Support Debugging Tool for Microsoft Dynamics GP

Training Workbook
Conditions and Terms of Use

Microsoft Confidential

This training package content is proprietary and confidential, and is intended only for users described in the training materials. This content and information is provided to you under a Non-Disclosure Agreement and cannot be distributed. Copying or disclosing all or any portion of the content and/or information included in this package is strictly prohibited.

The contents of this package are for informational and training purposes only and are provided "as is" without warranty of any kind, whether express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.

Training package content, including URL and other Internet Web site references, is subject to change without notice. Because Microsoft must respond to changing market conditions, the content 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. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Copyright and Trademarks

© 2012 Microsoft Corporation. All rights reserved.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in 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.

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.

For more information, see Use of Microsoft Copyrighted Content at http://www.microsoft.com/about/legal/permissions/.

Microsoft®, Internet Explorer, and Windows® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Microsoft products mentioned herein may be either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.
### About the Authors

<table>
<thead>
<tr>
<th>Author:</th>
<th>David Musgrave</th>
</tr>
</thead>
</table>

#### Bio:

David Musgrave has presented popular sessions at conferences and delivered training around the world. He is the creator of and primary contributor to the Developing for Dynamics GP MSDN blog. Based in Perth, Western Australia, David has been working as a Microsoft Dynamics GP (Great Plains) developer and consultant since v2.0 in 1994 (from 2001 with Microsoft). He is currently working with the Microsoft Dynamics GP Asia Pacific support team as their escalation engineer. Before joining Microsoft in 2001, David was the owner of Winthrop Dexterity Consultants and before that part owner of Sequel Technology. David is the original author of Named Printers, Omni Tools, Omni Price, Advanced Security, Field Level Security, Business Activity Statement (for Australian GST), Support Debugging Tool, and Menus for Visual Studio Tools, and he has made a number of contributions to the code in the core Microsoft Dynamics GP Dictionary.

### Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>30-Jan-2011</td>
<td>Draft 1.0 - Created Date - David Musgrave</td>
</tr>
<tr>
<td>27-Feb-2011</td>
<td>Draft 1.0 - Completed - David Musgrave</td>
</tr>
</tbody>
</table>
Table of Contents

Module 1: Introduction and Feature Summary ................................................................. 7
Lesson 1: Introduction ....................................................................................................... 8
  Introduction .................................................................................................................. 8
  Support Debugging Tool ............................................................................................. 8
  History ......................................................................................................................... 9
  Versions ....................................................................................................................... 9
  Support Debugging Tool Portal .................................................................................. 9
Lesson 2: Features .......................................................................................................... 10
  Features ....................................................................................................................... 10

Module 2: Installation and Configuration ........................................................................ 12
Lesson 3: Installation, Security and Navigation .............................................................. 13
  Installation .................................................................................................................. 13
  Security ....................................................................................................................... 13
  Navigation ................................................................................................................... 15
Lesson 4: Recommended Configuration .......................................................................... 17
  Recommended Configuration ....................................................................................... 17
Lesson 5: SQL Profile Tracing Configuration .................................................................. 22
  SQL Profile Tracing Configuration ............................................................................. 22
Lesson 6: Macro Recording Configuration ...................................................................... 29
  Macro Recording Configuration ................................................................................ 29
Lesson 7: Installation ...................................................................................................... 32
  About Support Debugging Tool for Microsoft Dynamics GP .................................... 32
  Un-installing ............................................................................................................... 32

Module 3: Solving Security Issues .................................................................................. 34
Lesson 8: Capturing Security Issues .............................................................................. 35
  Capturing the Security Profiler Log .......................................................................... 35
  Loading the Security Profiler Log ............................................................................. 36
Lesson 9: Resolving Security Issues .............................................................................. 37
  Using the Security Information window ................................................................... 37
Lesson 10: Viewing Security Access .............................................................................. 39
  Using the Security Information Resources window .................................................. 39
Lesson 11: Viewing SQL Roles ...................................................................................... 41
Displaying SQL Server Roles ..............................................................................................................41
Module 4: Identifying Dictionary Resources .........................................................................................43
Lesson 12: Finding Resources ............................................................................................................44
Finding Resources ..........................................................................................................................44
Lesson 13: Finding a Resource and Checking Access ............................................................................49
Checking Resources Security Access ..................................................................................................49
Module 5: Sending Screenshots and Emails .......................................................................................51
Lesson 14: Capturing Screenshots ........................................................................................................52
Sending Screenshots and the System Summary report .......................................................................52
Lesson 15: Sending Emails ..................................................................................................................56
Sending Emails ................................................................................................................................56
Module 6: Manually Capturing Application Logs ..............................................................................58
Lesson 16: Manual Logging Mode .........................................................................................................59
Manually Capturing Application Logs ..............................................................................................59
Capturing Individual Application Logs .............................................................................................60
Checking for Stranded SQL Profile Traces ..........................................................................................61
Module 7: Administrator Settings and Tools .....................................................................................62
Lesson 17: Administrator Settings .........................................................................................................63
Company Settings, Color Coding & Email Setup .................................................................................63
Lesson 18: Dex.ini Configuration .........................................................................................................66
Updating Dex.ini Settings Automatically ............................................................................................66
Module 8: Backing Up and Transferring Data ....................................................................................68
Lesson 19: Exporting Data to XML files ...............................................................................................69
Exporting Data ..................................................................................................................................69
Limiting Exported Data .....................................................................................................................70
Lesson 20: Importing Data from XML files ...........................................................................................71
Importing Data ....................................................................................................................................71
Module 9: Troubleshooting Customizations ......................................................................................72
Lesson 21: Dictionary Control window ..................................................................................................73
Disabling / Enabling products ............................................................................................................73
Viewing the Dynamics.set Launch File ..............................................................................................74
Changing the Dynamics.set Launch File ............................................................................................74
Module 10: Executing Scripts ............................................................................................................75
Lesson 22: Executing T-SQL Scripts ................................................................. 76
Executing SQL Scripts ....................................................................................... 76
Lesson 23: Executing Dexterity Scripts .............................................................. 78
Executing Dexterity Scripts ................................................................................ 78
Module 11: Automatically Capturing Application Logs ......................................... 80
Lesson 24: Automatic Debugger Mode ............................................................... 81
Automatically capturing Application Logs .......................................................... 81
Lesson 25: Automatic Debugger Mode Status ..................................................... 86
Viewing Active Automatic Debugger Mode Triggers .......................................... 86
Lesson 26: Starting and Stopping Automatic Debugger Mode Triggers ............... 87
Automatically Starting Triggers ......................................................................... 87
Manually Starting Triggers .................................................................................. 87
Manually Stopping Triggers ................................................................................ 87
Module 12: Exporting and Importing Configuration Settings ................................ 88
Lesson 27: Exporting and Importing Settings ...................................................... 89
Exporting and Importing Settings Files .............................................................. 89
Module 13: Advanced Debugging and Development Techniques .......................... 90
Lesson 28: Non-Logging Triggers ....................................................................... 91
Using Non-Logging Triggers to fix or prototype code ......................................... 91
Lesson 29: Capturing Parameters with Automatic Debugger Mode triggers .......... 94
Creating a wrapper trigger to store parameters for Support Debugging Tool to use where parameters are needed ......................................................... 94
Lesson 30: Working with Report Writer ............................................................. 95
Creating Custom Report Writer functions ......................................................... 95
Module 14: Other Considerations and Information ............................................... 96
Lesson 31: Terminal Server Environment .......................................................... 97
Consideration for setup in Terminal Server environment .................................... 97
Lesson 32: Limitations of the SDT ..................................................................... 98
Limitations of the Support Debugging Tool ......................................................... 98
Module 1: Introduction and Feature Summary

This module introduces the Support Debugging Tool for Microsoft Dynamics GP.

Before You Begin

Before starting this module, you should:

- Have access to a computer with an internet browser.

What You Will Learn

After completing this module, you will be able to:

- Understand what the Support Debugging Tool for Microsoft Dynamics GP is.
- Know for which versions of Microsoft Dynamics GP the tool is available.
- Locate the Support Debugging Tool Portal blog page.
- Understand the differences between Standard and Advanced modes.
Lesson 1: Introduction

This lesson provides an introduction to the Support Debugging Tool for Microsoft Dynamics GP.

What You Will Learn

After completing this lesson, you will be able to:

- Understand why the Support Debugging Tool was created.
- Know where to find the latest builds and information.

Introduction

In the world of the Microsoft Dynamics GP administrator or consultant there are many tools available to assist with the day to day administrative tasks. Traditionally, the tools we have used include creating a DEXSQL.LOG or a SQL Profile Trace, or using other tools available from the SQL Server Management Studio.

Another tool now available to the administrator or consultant is the Support Debugging Tool for Microsoft Dynamics GP (SDT). This free tool is available to all Microsoft Dynamics GP sites. Currently, the tool can be downloaded by Microsoft Dynamics Partners from PartnerSource, then they can provide it to their customers.

Support Debugging Tool

The Support Debugging Tool for Microsoft Dynamics GP is a Dexterity based suite of utilities designed to assist System Administrators, Partner Consultants and Support Engineers to keep Dynamics GP running smoothly. The tool already has many features and functions; however it is a living application that will continue to be enhanced based on feedback.
History

The Support Debugging Tool was initially created in August 2006 to resolve a critical escalation where data damage could be seen after the fact, but the cause could not be reproduced. The tool was configured at the customer’s site and within a few days, it identified the exact source code script that was at fault, allowing for a fast fix.

Since then the features of the tool have been expanded to provide many functions needed by people who support and administer Microsoft Dynamics GP. The first publically released build was build 9, since then there have been a number of subsequent builds:

- Build 9 in September 2008
- Build 10 in December 2008
- Build 11 in June 2009
- Build 12 in February/March 2010 – Last build for v8.0
- Build 13 in May 2010
- Build 14 in December 2010 – Last build for v9.0
- Build 15 in June/July 2011
- Build 16 in January 2012

Versions

The Support Debugging Tool was first developed for version 8.0 and there have been versions for all Microsoft Dynamics GP versions since. Development for a particular version is discontinued when that version is no longer supported by Microsoft.

- The final release for version 8.0 was Build 12 and is no longer supported.
- The final release for version 9.0 was Build 14 and is no longer supported.
- The current release for version 10.0 is Build 16.
- The current release for version 11.0 (GP 2010) is Build 16.

Support Debugging Tool Portal

Please see the Support Debugging Tool Portal blog page on the Developing for Dynamics GP blog for release details of the tool, Frequently Asked Questions (FAQ) and the changes in the various builds:

[http://aka.ms/SDT](http://aka.ms/SDT)
Lesson 2: Features

This lesson provides a summary of the features of the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Understand the differences between Standard and Advanced Modes.
- List the features included in each mode.

Features

The features of the Support Debugging Tool are divided into Standard Mode and Advanced Mode features. Standard Mode features are available to all users (unless prevented by application level security). Advanced Mode features require the Microsoft Dynamics GP System Password for access as well as the current user having system administrator or database owner (dbo) privileges on SQL Server.

Standard Mode

The standard mode features cannot access or alter data and so are safe for all users to use. They include:

- Manual Logging Mode
- Individual Logging Control
- Having Automatic Debugger Mode active
- Dex.ini Settings
- Resource Information
- Security Profiler
- Security Information
- Configuration Export/Import
- Screenshot
- Send Email
Advanced Mode

The advanced mode features have the ability to view or alter data in the system as well as execute Dexterity sanScript and Transact-SQL scripts. This is why they are protected by the System Password and require the user to have a high level of SQL Server privileges. They include:

- Automatic Debugger Mode
- Dictionary Control
- XML Table Export
- XML Table Import
- Runtime Execute
- SQL Execute
- Configuration Maintenance
- Administrator Settings
- Dex.ini Configuration

To access an advanced mode feature a user must have security access at the application level AND system administrator or database owner privileges at the SQL Server level AND know the System Password if one is in use.

Even if Advanced Mode is enabled, users without sufficient privileges will still only see the Standard Mode features on the menus.
Module 2: Installation and Configuration

This module guides you through the installation and configuration of the Support Debugging Tool.

Before You Begin

Before starting this module, you should:

- Have a working knowledge of how to navigate around Microsoft Dynamics GP.

What You Will Learn

After completing this module, you will be able to:

- Install the Support Debugging Tool.
- Set up the Support Debugging Tool in the recommended configuration.
- Configure the SQL Profile Tracing functionality.
- Configure the Macro Recording functionality.
- Know how to check the version and build details.
- Know how to un-install the Support Debugging Tool.
Lesson 3: Installation, Security and Navigation

This lesson provides an overview of installing the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Install the Support Debugging Tool files from the archive file.
- Understand what Security Roles are added to the system.
- Navigate to the Support Debugging Tool using the menus.

Installation

The Support Debugging Tool for Microsoft Dynamics GP is installed by copying the Debugger.cnk (self-installing dictionary), Debugger.txt (readme file) and Debugger.pdf (this user guide manual) to Microsoft Dynamics GP application folder. When Microsoft Dynamics GP is next launched, select “Yes” to include new code.

**Important:**

If installing on a Windows Vista, Windows 7 or Windows Server 2008 system and User Access Control (UAC) is active, please launch Microsoft Dynamics GP with the Run as Administrator option to complete the installation.

The Support Debugging Tool for Microsoft Dynamics GP does not use any SQL tables to store its data, instead it uses an XML file called Debugger.xml. By default this file is stored in the data subfolder beneath the application folder and will be created if required when logging into Microsoft Dynamics GP. The Debugger.xml file may be supplied with the initial installation files when the Support Debugging Tool for Microsoft Dynamics GP has been pre-configured.

Security

Security access must be granted to the forms of the Support Debugging Tool for Microsoft Dynamics GP before it can be used by users other than those belonging to the POWERUSER security role.

The Support Debugging Tool for Microsoft Dynamics GP will automatically create the Security Tasks and Security Roles required to use the tool. The following two Security Roles are created.

- MBS DEBUGGER USER (Debugging Tool User)
- MBS DEBUGGER ADMIN (Debugging Tool Administrator)
The administrator security role grants access to all areas of the tool, while the user security role only grants access to the Standard Mode features. Advanced Mode features are only available to Microsoft Dynamics GP User IDs that also have the SQL Server sysadmin fixed server role or membership of the db_owner role on the system database (DYNAMICS) and the current company database, even if security is granted.

After installing the Support Debugging Tool for Microsoft Dynamics GP: If logging into Microsoft Dynamics GP as a user belonging to the POWERUSER security role, and no users have been granted access to the MBS DEBUGGER USER security role, the system will offer to add this security role to all users for you.

If you respond Yes, the system remind to you to add the MBS DEBUGGER ADMIN security role to another other users who need access to the Advanced Mode features and do not already have access to the POWERUSER Security Role.

Also, if Advanced Mode is not enabled, it will offer to change that setting for you.
Important:

If the System Password is enabled, it will need to be entered before Advanced Mode can be enabled. Enabling Advanced Mode just displays the additional features of the Support Debugging Tool for Microsoft Dynamics GP, a user without sufficient privileges will not be able to access the Advanced Mode features.

To manually grant security to the forms of the Support Debugging Tool for Microsoft Dynamics GP use the User Security Setup window (Microsoft Dynamics GP >> Tools >> Setup >> System >> User Security). After selecting the user and company, select one of the two security roles below:

MBS DEBUGGER USER (Debugging Tool User)
MBS DEBUGGER ADMIN (Debugging Tool Administrator)

Navigation

Once logged into Microsoft Dynamics GP, a user with security access granted can find the Support Debugging Tool for Microsoft Dynamics GP under the Tools menu underneath the Microsoft Dynamics GP menu (highlighted below). It also has the keyboard shortcut Ctrl+D assigned to it.
Tip:
The Support Debugging Tool for Microsoft Dynamics GP also adds the Raise All Windows option to the main application menu, to allow for an easy method to send the main application window to the background. It also has the keyboard shortcut Ctrl+Shift+R assigned to it.

In addition, the Support Debugging Tool for Microsoft Dynamics GP is also found under the Tools menu on each individual window of Microsoft Dynamics GP (highlighted below).

You may need to press and release the Alt key on the keyboard to allow the window menu bar to activate before the shortcut keys work.

Note:
If a user is not going to be using any of the windows of the Support Debugging Tool for Microsoft Dynamics GP, they do not need to be assigned to a security role. Automatic Debugger Mode will work regardless of security settings.
Lesson 4: Recommended Configuration

This lesson describes how to setup the recommended configuration for the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Setup the Support Debugging Tool in the recommended configuration.
- Enable Advanced Mode.

Recommended Configuration

The Support Debugging Tool for Microsoft Dynamics GP does not store its settings and data in SQL tables. Instead it uses an XML file called Debugger.xml. The recommended configuration is for the Support Debugging Tool for Microsoft Dynamics GP to be installed on all workstations in the system and to point each workstation to use a single Debugger.xml file stored in a shared location.

Below are step by step instructions to install and setup this recommended configuration:

1. Initially install on a single instance of Microsoft Dynamics GP. Just copy the files from the archive to the application folder, usually under C:\Program Files\Microsoft Dynamics\GP.

2. Extract the chunk file by launching Microsoft Dynamics GP using Run as Administrator and click Yes if asked “Do you wish to include new code now?”

3. Log into Microsoft Dynamics as ‘sa’ or a user with similar permissions.

4. If asked to add the Standard Mode Security Settings to all users, click Yes.
5. You will then be reminded that Advanced Mode settings will need to be set up manually.

6. If asked about enabling Advanced Mode, click Yes.

7. **Optional:** To manually change security settings, go to the User Security Setup window (Microsoft Dynamics GP >> Tools >> Setup >> System >> User Security), select the appropriate user and company and grant access to one or both of the following roles:

   For Standard Mode features:
   MBS DEBUGGER USER (Debugging Tool User)

   For Advanced Mode features:
   MBS DEBUGGER ADMIN (Debugging Tool Administrator)

   **Tip:** It is recommended to grant all users in the system access to MBS DEBUGGER USER. Only System Administrators need access to MBS DEBUGGER ADMIN, unless they already have access to the POWERUSER Security Role.
8. Open the Support Debugging Tool main window (Microsoft Dynamics GP >> Tools >> Support Debugging Tool).

You can also open the main window from the Tools menu on each individual window.

Screenshots of menus shown in previous section.

9. From the Support Debugging Tool main window select Options >> Dex.ini Settings.

Tip: Unless Advanced Mode is already enabled, the Options menu will only contain the Standard Mode features. Enabling Advanced Mode features will make them available on the menus only if the user has access.
10. From the Dex.ini Settings window, on the Debug tab, make sure the Debugger Advanced Mode Features option is enabled and select a Specified Pathname location for Debugger Setup files. This pathname should be a network share available to all workstations, possibly a folder in the same location that the OLE note attachments are stored.

![Dex.ini Settings window](image)

**Note:**
The pathname can be specified using a UNC path in the format `\Server\Share\Folder`.

11. Click OK to save the changes. This will update the current workstation’s DEX.INI file only. The next step will allow these two settings to be automatically rolled out to all workstations that have the Support Debugging Tool installed.

**Critical:**
Do not enable Debugger Setup Mode on a live system as this prevent Triggers from starting automatically. This mode is designed for use by consultants and support engineers.
12. From the main window select Options >> Dex.ini Configuration to open the Dex.ini Configuration window. Change the Path Default Setting and the Advanced Mode Setting to be controlled by the Administrator during login process.

13. Click OK to save the changes. Now all workstations with the Support Debugging Tool installed will have these two settings automatically applied.

14. Install the Support Debugging Tool on all other workstations in the system.

That is all that is required for the initial setup. To install on other workstations just requires the copying of the files and the including of new code.
Lesson 5: SQL Profile Tracing Configuration

This lesson explains how to set up the SQL Profile Trace functionality of the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Configure SQL Profile Tracing in the Support Debugging Tool.

SQL Profile Tracing Configuration

For more information on setting up and enabling SQL Profile Tracing please see the section under the Administrator Settings window in the User Guide manual.

Below are step by step instructions to configure the recommended settings for SQL Profile Tracing:

1. On the SQL Server machine create a folder on a local drive for where the SQL Profile Trace files will be stored while they are being created. Note this local path for later.

2. Share this local folder on the network, so that all Microsoft Dynamics GP users will have Full Control to the folder. Note this network UNC path for later.

3. Create a user (for example: SQLTraceUser) to be used by SQL Profile Tracing system. The user can be a local user on the SQL Server or a domain user, but needs local Administrator rights on the SQL Server machine. It is recommended to set the password to not expire. Note the User ID and password for later.

4. Log into Microsoft Dynamics as ‘sa’ or a user with similar permissions.

5. Open the Support Debugging Tool main window (Microsoft Dynamics GP >> Tools >> Support Debugging Tool).
6. From the Support Debugging Tool main window select Options >> Dex.ini Settings.

[Image of Support Debugging Tool window]

7. From the main window select Options >> Administrator Settings to open the Administrator Settings window.

Tip: If the Administrator Settings choice is not available on the menu, then Advanced Mode is not yet enabled. Please see the previous Recommended Configuration section for steps to enable Advanced Mode.
8. From the Administrator Settings window, on the General tab, click Edit SQL Profile Trace Settings to open the SQL Profile Trace Settings window.

9. On the SQL Profile Trace Settings window, make sure Single User Authentication Mode is selected. In this mode only the single user created earlier will need permissions to create SQL Traces and the permissions for individual users do not need to be changed or elevated.
10. Enter the user created previously and press tab. The system will then ask if you want to process the SQL Server Actions to enable the Authentication Mode, click Yes.

![Image](image1.png)

11. As each step of the SQL Server actions needed to enable the Authentication Mode are completed a desktop alert will be displayed. You will also be asked for the password for the user for the Enable xp_cmdshell proxy account step. The password is not validated at this time, so please ensure it is entered correctly.

![Image](image2.png)

**Tip:**
To see the list of individual steps for enabling or disabling the Authentication Mode, click the Process Single User Mode SQL Server Action or Process Multi User Mode SQL Server Action button. You can select to manually run all of the steps or select individual steps from the list.
12. You can change the Maximum Trace file size and Maximum number of Trace files if desired, or just leave the default values.

![SQL Profile Trace Settings](image1)

13. Enter in the Local Path set up previously (as created in step 1) and press tab. The system will then ask if you want to create the SQL Profile Trace SQL Components, click Yes to create the stored Procedures in the DYNAMICS system database.

![Do you want to create the SQL Profile Trace SQL Components?](image2)
14. Enter the UNC Network Path set up previously (as created in step 2) and press tab.

15. Make sure the Copy SQL Profile Trace files to Debugger Settings location option is enabled. This will copy the completed trace files from the SQL Server to the folder used for the Debugger Settings and logs.

16. Click OK to save the settings and close the SQL Profile Trace window.
17. On the Administrator Settings window, on the General tab, enable the Capture SQL Profile Trace option and set the desired Trace Mode (use Small, if unsure). This will enable SQL Profile Tracing for Manual Logging Mode and as the default value for Automatic Debugger Mode.

18. Click OK to save the settings and close the Administrator Settings window.
Lesson 6: Macro Recording Configuration

This lesson explains how to set up the Macro Recording functionality of the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Configure Macro Recording in the Support Debugging Tool.

Macro Recording Configuration

For more information on enabling Macro Recording please see the section under the Administrator Settings window in the User Guide manual.

Below are step by step instructions to enable Macro Recording:

1. Log into Microsoft Dynamics as ‘sa’ or a user with similar permissions.
2. Open the Support Debugging Tool main window (Microsoft Dynamics GP >> Tools >> Support Debugging Tool).
3. From the Support Debugging Tool main window select Options >> Dex.ini Settings.
4. From the main window select Options >> Administrator Settings to open the Administrator Settings window.

Tip: If the Administrator Settings choice is not available on the menu, then Advanced Mode is not yet enabled. Please see the previous Recommended Configuration section for steps to enable Advanced Mode.

5. If running Microsoft Dynamics GP 2010 or later, skip to step 9.

6. For v10.00 only: From the Administrator Settings window, on the General tab, click Edit Macro Recording Settings to open the Macro Recording Settings window.

7. For v10.00 only: Check the Disable Mouse Wheel option. Click OK.
8. For v10.00 only: If you changed this option, click OK to close the Administrator Settings window and close and restart the Microsoft Dynamics GP application. Log back in and re-open the Administrator Settings window.

9. On the Administrator Settings window, on the General tab, enable the Capture Macro Recording option. This will enable Macro Recording for Manual Logging Mode and as the default value for Automatic Debugger Mode.

10. Click OK to save the settings and close the Administrator Settings window.
Lesson 7: Installation

This lesson explains how to check the version and build information of the Support Debugging Tool and how to un-install the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Locate the version and build information for the Support Debugging Tool.
- Un-install the Support Debugging Tool.

About Support Debugging Tool for Microsoft Dynamics GP

You can open this window by selecting About Support Debugging Tool for Microsoft Dynamics GP from the Options button drop list on the main window.

The About Support Debugging Tool for Microsoft Dynamics GP window shows the current version, build and last modified date information.

Un-installing

You can un-install the Support Debugging Tool for Microsoft Dynamics GP from this window. Clicking Un-install will remove the Support Debugging Tool for Microsoft Dynamics GP from the menus and security tables and remove any Dex.ini settings added.

If SQL Profile Tracing is enabled, you will be asked if you want to remove the SQL Server permissions and components created by the Support Debugging Tool.
You will also be asked if you want the Dynamics.set launch file updated to remove the Support Debugging Tool for Microsoft Dynamics GP, so that it does not re-install itself next time Microsoft Dynamics GP is launched.

**Important:**

If User Account Control (UAC) is preventing writer access to the application folder, you will see the following dialog displayed. You will need to use Run as Administrator to allow access and complete the un-install.
Module 3: Solving Security Issues

This module guides you through using the Support Debugging Tool to resolve security issues.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.
- Set up the SQL Profile Tracing configuration.
- Set up the Macro Recording configuration.

What You Will Learn

After completing this module, you will be able to:

- Identify windows or reports causing security errors.
- Identify Security Tasks and Security Roles required for access to a resource.
- Display all resources accessible by a specific User, Security Role or Security Task.
- Display SQL Roles for a specific User.
Lesson 8: Capturing Security Issues

This lesson provides an overview of capturing Security issues with Security Profiler window.

What You Will Learn

After completing this lesson, you will be able to:

- Use the Security Profiler window to capture Security issues.
- Send Security Profiler Logs to the System Administrator

Capturing the Security Profiler Log

When a user receives a Security Privileges error (see below) they can use the Security Profiler to capture the details of the error.

![Security Privileges Error Message]

Use the Tools menu or Ctrl-D keyboard shortcut to open the main window and select Options >> Security Profiler. Once the Security Profiler window is open; perform the steps to recreate the security error so that the details can be captured.

![Security Profiler Window]

Clicking Export allows the Security Profiler Log to be sent to the System Administrator.
Loading the Security Profiler Log

The System Administrator can open the Security Profiler window on their system and import the Security Profiler log sent to them by the user. Now because the System Administrator has privileges to access the security settings, the security button will be active.

Using the right click context menu or selecting the resource and clicking Security, opens the Security Information window for the selected resource and user/company combination.
Lesson 9: Resolving Security Issues

This lesson provides an overview of resolving Security issues with Security Information window.

What You Will Learn

After completing this lesson, you will be able to:

- Use the Security Information window to identify the cause of Security issues.
- Use the Security Information window as a launching point to fix Security issues.

Using the Security Information window

The Security Information window can be opened directly from the Support Debugging Tool’s Options menu, or from the Security Profiler and Resource Information windows. If opened from another window the selected resource and user / company will be selected already.

If opened directly, the current user and company will be selected but no resource will be selected. You will need to select a resource using the drop down list at the top of the left hand pane.
The left hand pane of the Security Information window screenshot tells us the following information:

- LESSONUSER1 for company Fabrikam does not have access to the window.
- If they did have access the DEFAULT USER Alternate Modified ID would access the original resource.
- To give them access they need to be POWERUSER or have access to the AAA_CUSTOMER_EXTRA Security Task which belongs to the selected AAA CUSTOMER EXTRA Security Role.

The right hand pane of the Security Information window screenshot tells us the following information:

- DYNSA and sa users have access for Fabrikam company when other users do not have access.

Double clicking on the User/Company node (highlighted) on the left hand pane will open the User Security window to allow the necessary changes to be made.

Single clicking on node on the right hand pane will change the user and company selection for the left hand pane so you can see how other users do or don't have access and why.

Expanding the nodes will show which Security Roles and Tasks a user has access too.

Changing the right hand pane view allows many views into the security data to see what the relationships between Users and Security Roles and Security Tasks are. For example: Seeing which users have access to particular Security Tasks and then which Security Roles those tasks belong to.
Lesson 10: Viewing Security Access

This lesson explains how to show Security Access for specific Users, Security Roles and Security Tasks with the Security Information Resources window.

What You Will Learn

After completing this lesson, you will be able to:

- Use the Security Information Resources window to view Security Access information.
- Export Security Access information to a comma or tab delimited text file.

Using the Security Information Resources window

The Security Information Resources window is opened from the Security Information window using the Show Resources button. The window shows the resources that the entity selected on the right hand pane of the Security Information window has access to. You can select a User, Security Role, Security Task or Alternate/Modified Forms & Report ID.

Select the entity in the right hand pane and click Show Resources

The Security Information Window will open. The window populates in the background that allows the user to select a different entity or change the settings, such as which resources to include and whether to show the Resource Series.
The Security Information Resources window can be left open and will update automatically when the selection changes on the right hand pane of the Security Information window.

Selecting a resource in the Security Information Resource window will make it the current resource selected in the parent Security Information window.

To export the data into a form that can be viewed in a spreadsheet, use the Export Button to export as tab delimited. The resulting file can be provided to auditors if desired.
Lesson 11: Viewing SQL Roles

This lesson explains how to view the SQL Server Role information for the system, both for Dynamics GP users and databases and for the entire system.

What You Will Learn

After completing this lesson, you will be able to:

- Use the Security Information window to display SQL Server roles for users and companies.
- Display the SQL Server role information in multiple views.

Displaying SQL Server Roles

The Security Information window can also be used to display the SQL Server roles assigned to users for each database.

Use the Button Drop Down list above the right hand pane to change the view to one of the three SQL Roles views.

Once a SQL Roles view is displayed, the additional option to Show All SQL Users & Databases is displayed. Selecting this option will display all users and databases in the system rather than just those specifically related to Microsoft Dynamics GP.
Changing between the three views allows the data to be displayed in different ways depending on the information you wish to see.

For example: in view SQL Roles by User, we can see what access each user has at the Server level and against each database.
Module 4: Identifying Dictionary Resources

This module guides you through using the Support Debugging Tool to identify Dictionary Resources such as Fields, Tables, Windows, Forms and Reports.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.
- Set up the SQL Profile Tracing configuration.
- Set up the Macro Recording configuration.

What You Will Learn

After completing this module, you will be able to:

- Find Dictionary Resources using the Resource Information window.
- Identify Security Tasks and Security Roles required for access to the selected resource.
Lesson 12: Finding Resources

This lesson provides an overview of searching for dictionary resources with the Resource Information window.

What You Will Learn

After completing this lesson, you will be able to:

- Use the Resource Information window to find information about dictionary resources.
- Use the Resource Explorer series of windows.

Finding Resources

If you need to find any information about a form, window, field, table or report resources in any dictionary, the Resource Information window is the answer. The window is opened from the Support Debugging Tool’s Options menu.

Once open, just select the mode with the Resource Type drop down list and then type in what information you know and let it find the resource you are looking for. It can perform exact, begins with or contains searches with or without case sensitivity.

If you type in a Display Name or Physical Name, the Resource Information will scan every dictionary (product) loaded to find a match.
If the match found is not what you are looking for, click Search Again to continue searching. Click Clear to start a new search.

You can also use the powerful multi-dictionary Resource Explorer windows to look through form, window, field, table, and report resources. Use the Menu Explorer to find forms and windows using the menu navigation structure.

You can also find tables that contain a particular table or tables associated to a form with an optional field filter.

Example Scenario

An example of finding field resources in a table could be locating the Hold field in the Customer Maintenance window. Using the Resource Explorer or Menu Explorer lookup windows, we can find the Customer Maintenance form (Debtor Maintenance on International English system) and the Hold field.

Clicking OK returns this information to the Resource Information window.
Clicking on the Associated Tables button opens the Associated Tables Lookup. Because a field was shown on the Resource Information window, the lookup is filtered to show tables containing that field.

Unchecking the filtering checkbox will display all tables associated with the form.
Using this filter can narrow down the possible locations for where a particular field is stored in the data model. If only one table is listed when the filter is active, then the table has been identified for you.

**Note:** You can also use a similar filter to show tables which contain a specified field across the entire database. This lookup accessed by using the Tables Containing Field button is not limited to fields attached to a particular form.
Selecting the table immediately returns you to the Resource Information window with that table displayed.

Opening the Table Explorer will then display details of the table and the fields contained within it, including the Display, Technical and Physical Names.
Lesson 13: Finding a Resource and Checking Access

This lesson explains how to find a resource and check the security access for that resource using the Resource Information and Security Information windows.

What You Will Learn

After completing this lesson, you will be able to:

- Use the Resource Information to identify dictionary forms.
- Launch the Security Information to analyze the Security Tasks and Roles associated with the form.

Checking Resources Security Access

Another example of using Resource Information to help identify how to give security access for a resource to a user could be the Create Return window. This window is accessed from the Additional menu from the Sales Transaction Entry window.

We don't know what the actual window is, so we change the Search Mode to Begins With and turn off Case Sensitivity, then type "create ret" into the Display Name field and press tab.

Resource Information locates the window as form SVC_RMA_Create_Return in the FieldService dictionary (Resource ID 22,190).
We then use the Right click context menu to select Security Information or click the Security button to open the Security Information window.

Once the Security Information window is open we can select the user we are interested in (for example LESSONUSER1) to see if they have access. In this example, they don’t as it says Access Denied.

Expand the System Level nodes to see that to gain access to the window a user needs access to the TRX_FSS_RT01* Security Task which can be granted by adding either the RT_AGENT* or the RT_MANAGER* Security Roles to the user.
Module 5: Sending Screenshots and Emails

This module guides you through using the Support Debugging Tool to send screenshots, system status and emails to the System Administrator.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.
- Microsoft Outlook installed (default), or alternate email configuration set up in Administrator Settings window.
- Default email address, Email Subject and Email Body are set up in Administrator Settings window.

What You Will Learn

After completing this module, you will be able to:

- Send Screenshots, System Status report and Dynamics.set & Dex.ini files to the System Administrator.
- Send Emails to the System Administrator.
Lesson 14: Capturing Screenshots

This lesson provides an overview of capturing and sending screenshots using the ScreenShot window in the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Capture screenshots and save or email them.
- View and email the System Summary report

Sending Screenshots and the System Summary report

The ScreenShot utility can be opened by clicking the Camera button on the Standard Toolbar, pressing the Ctrl-S keyboard shortcut, selecting Capture Screenshots from the Application or Window Level Tools menu, or selecting Capture Screenshots from the Options menu.

It allows screenshots of all open windows in a Dynamics GP to be saved or emailed along with the current DYNAMICS.SET launch file and the DEX.INI settings file. It will also include a comprehensive System Summary with details of the workstation environment, SQL Server, Runtime engine and installed Dictionaries.

Once the ScreenShot window is opened, you can select for which of the open windows screenshots will be created.
Click in the i information button will display the System Summary to the screen. Here is an example of a System Summary:

Support Debugging Tool System Summary (23/02/2012 17:37:21)

Version: 11.00.0016, Last Modified: 05-Jan-2012.
Support Debugging Tool Folder Path: C:\T\

Registration Information
Product Name: Microsoft Dynamics GP
Site Name: Two, Inc.
Number of Users: 4095
Language-Country: English-Australia

Login Information
User ID: sa
Company Name: Fabrikam, Ltd.
Database Name: TWO

Environment Information
User Name: dmusgrav
Computer Name: PTYDMUSGRAVELT1
Operating System: Windows 7 Service Pack 1 Build (7601)
Physical Memory: 1407 MB available of 8122 MB total (82% used)
User Account Control (UAC): No
Temporary Folder Path: C:\Users\dmusgrav\AppData\Local\Temp\
Environment Variable: HOMEDRIVE=C:
Environment Variable: HOMEPATH=\Users\dmusgrav
Database Type: SQL Server

SQL Version: Microsoft SQL Server 2008 R2 (SP1) - 10.50.2500.0 (X64)
SQL Server: PTYDMUSGRAVELT1\SQL2008R2
SQL Server NetBIOS Name: PTYDMUSGRAVELT1
SQL Session ID: 2
SQL Session SPID: 54
ODBC Data Source Name: GP110
ODBC Manager Version: 03.80.0000
ODBC Driver Version: 10.50.2500

Database Information (System and 1 Company Databases)
Fixed Server Roles for User: sysadmin
System Database: DYNAMICS
Fixed Database Roles for User:

Company Database: TWO, ID: -1, Name: Fabrikam, Ltd.
Fixed Database Roles for User:
Data Size - Maximum: 372.01 MB, Used: 330.59 MB, Free: 41.41 MB
Log Size - Maximum: 76.42 MB, Used: 2.01 MB, Free: 74.41 MB
Application Security for Company: Enabled

Product Information
Dexterity Runtime: 11.00.0349.000
Runtime Folder: C:\Dyn1100\
Launch File Folder: C:\Dyn1100\
Data Folder: C:\Dyn1100\Data\
Dex.ini File Path: C:\Dyn1100\Data\dex.ini
Launch File Path: C:\Dyn1100\Dynamics.set
Launch File Workstation: Windows

Dictionary Information (21 Products)
54

0: 11.00.1752: Microsoft Dynamics GP
Application Dictionary Path: C:\Dyn1100\Dynamics.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\FORMS.DIC (Found)
Custom Reports Dictionary Path: C:\Dyn1100\Data\REPORTS.DIC (Found)

105: 6.00.0061: IntegrationManager
Application Dictionary Path: C:\Dyn1100\IM.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\INFORMS.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\IMRPTS.DIC (Missing)

949: 11.00.1740: FieldService
Application Dictionary Path: C:\Dyn1100\SrvcAdv.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\FRMS949.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\RPTS949.DIC (Missing)

1042: 11.00.1734: Interfund Management
Application Dictionary Path: C:\Dyn1100\IFund.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\1042FORM.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\1042RPTS.DIC (Missing)

1493: 11.00.1734: SmartList
Application Dictionary Path: C:\Dyn1100\EXPL1493.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\EXP1493F.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\EXP1493R.DIC (Missing)

1632: 11.00.1734: Cash Flow Management
Application Dictionary Path: C:\Dyn1100\CFM.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\CFMFORM.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\CFMRPTS.DIC (Missing)

1878: 11.00.1734: Excel-Based Budgeting
Application Dictionary Path: C:\Dyn1100\XLBudget.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\XL1878F.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\XL1878R.DIC (Missing)

2199: 11.00.1752: Business Activity Statement
Application Dictionary Path: C:\Dyn1100\BAS_Rpt.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\BAS2199F.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\BAS2199R.DIC (Missing)

2277: 11.00.1734: Purchase Order Enhancements
Application Dictionary Path: C:\Dyn1100\POE2277.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\POE2277F.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\POE2277R.DIC (Missing)

2416: 11.00.1738: Control Account Management
Application Dictionary Path: C:\Dyn1100\CAM2416.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\CAMFORM.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\CAMRPTS.DIC (Missing)

2547: 11.00.1734: Enhanced Commitment Management
Application Dictionary Path: C:\Dyn1100\ECM2547.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\ECM2547F.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\ECM2547R.DIC (Missing)

2992: 11.00.1734: CopierSeries
Application Dictionary Path: C:\Dyn1100\QK2992.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\QDF2992.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\QDDR2992.DIC (Missing)

3104: 11.00.1734: Advanced Security
Application Dictionary Path: C:\Dyn1100\AdvSecur.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\ADVS_FRM.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\ADVS_RPT.DIC (Missing)

3107: 11.00.0042: Extender
Application Dictionary Path: C:\Dyn1100\extud.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\UDFORM.DIC (Found)
Custom Reports Dictionary Path: C:\Dyn1100\Data\UDREPT.DIC (Missing)

3278: 11.00.1734: Report Scheduler
Application Dictionary Path: C:\Dyn1100\RPTSCHED.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\P3278.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\R3278.DIC (Missing)

3830: 11.00.0050: SmartList Builder
Application Dictionary Path: C:\Dyn1100\SLBuild.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\F3830.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\R3830.DIC (Missing)

4612: 11.00.0003: Advanced Go Tos
Application Dictionary Path: C:\Dyn1100\advgoto.dic (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\F4612.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\R4612.DIC (Missing)

5261: 11.00.0016: Support Debugging Tool
Application Dictionary Path: C:\Dyn1100\DEBUGGER.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\DEBU_FRM.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\DEBU_RPT.DIC (Missing)

5597: 11.00.1734: HITB Report
Application Dictionary Path: C:\Dyn1100\HTB5597.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\HTBF5597.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\HTBR5597.DIC (Missing)

6499: 11.00.1734: Dynamics Online Services
Application Dictionary Path: C:\Dyn1100\DO6499.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\DO6499F.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\DO6499R.DIC (Missing)

32001: 11.00.0001: Customer User Defined Fields
Application Dictionary Path: C:\Dyn1100\CUSTUDF.DIC (Found)
Custom Forms Dictionary Path: C:\Dyn1100\Data\CUDF_FRM.DIC (Missing)
Custom Reports Dictionary Path: C:\Dyn1100\Data\CUDF_RPT.DIC (Missing)

Attachment Information

32001: Customer User Defined Fields: MBS_RM_Customer_Maintenance:
MBS_RM_Customer_Maintenance: Customer Maintenance Extras:
MBS_RM_Customer_Maintenance001.bmp: Original window

32001: Customer User Defined Fields: RM_Customer_Maintenance:
RM_Customer_Maintenance: Debtor Maintenance: RM_Customer_Maintenance001.bmp: Alternate window

32001: Customer User Defined Fields: RM_Customer_Maintenance:
RM_Customer_Options: Debtor Maintenance Options: RM_Customer_Maintenance003.bmp: Alternate window


** End of System Summary **
Lesson 15: Sending Emails

This lesson explains how to send emails with the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Send emails from inside the Microsoft Dynamics GP application.
- Send emails on workstations even when there is no email client installed.

Sending Emails

The Support Debugging Tool includes an email engine that can be used to send emails to the System Administrator or any other recipient. It can have a default email address, subject line and body text set up (using the Administrator Settings window).

For example: it can be used to automate and standardize reporting of software issues by users. Open the Send Email window from the Options menu of the Support Debugging Tool’s option menu.

The email engine supports sending emails via Outlook, a SMTP Server and (for Dynamics GP 2010 onwards) any MAPI compliant client.
**Note:** Use the Administrator Settings window to set up default email body and subject text as well the method of sending emails.

**Tip:** The Support Debugging Tool’s Send Email window can be used even when there is no email client application installed by setting up the Email Mode to use SMTP in the Administrator Settings window. This can be useful on Terminal Server environments.
Module 6: Manually Capturing Application Logs

This module guides you through using the Support Debugging Tool to manually capture application logs.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.
- Set up the SQL Profile Tracing configuration.
- Set up the Macro Recording configuration.

What You Will Learn

After completing this module, you will be able to:

- Start and Stop Manual Logging Mode.
- Start and Stop Individual Logs.
Lesson 16: Manual Logging Mode

This lesson provides an overview of manually capturing application logs.

What You Will Learn

After completing this lesson, you will be able to:

- Use Manual Logging Mode to capture logs.
- Use Logging Options to capture logs individually.
- Check for Stranded SQL Profile Traces.

Manually Capturing Application Logs

Manual Logging Mode can be used to capture any of the DEXSQL.LOG, Dexterity Script Log, Dexterity Script Profile, SQL Profile Trace and Macro Recording without needing to exit Dynamics GP to enable the Debug Menu or change DEX.INI settings.

Many intermittent issues in Dynamics GP can be “fixed” by exiting Dynamics GP and logging in again. If you are forced to exit Dynamics GP to change DEX.INI settings when you log in again, you might not be able to replicate the issue.
Manual Logging can be turned on and off from the main Support Debugging Tool window, from the application and window level Tools menu or using the keyboard shortcuts Ctrl-Shift-F9 and Ctrl-Shift-F10.

Just turn on the logging just prior to the event you are attempting to capture and turn it off immediately after. The logs will be stored and save with user/company/date/time stamped files. If Advanced Mode is enabled, clicking on the Manual Logging Mode hyperlink will open windows explorer to the correct data folder.

**Note:** Which of the logging modes are used for Manual Logging Mode are controlled by the selections on the General Tab of the Administrator Settings window.

### Capturing Individual Application Logs

Clicking on the Logging Options button on the main Support Debugging Tool window will open the Logging Option window. From this window you can manually turn on and off the individual logging modes.

**Tip:** Using Manual Logging Mode to turn on multiple logging modes together is the recommended method. Use of Logging Options to control Individual Logging Modes is only provided for testing and legacy support.
Checking for Stranded SQL Profile Traces

Clicking the Show SQL Profile Traces button from the main Support Debugging Tool window opens the Active SQL Profile Traces window. From this window it is possible to view the active SQL Profile Traces in the system.

A stranded SQL Profile Trace is a Trace that was started by a session, but was not stopped when the application closed. This can occur when Microsoft Dynamics GP has an abnormal termination while a trace is running.

A normal user can only see Support Debugging Tool Traces for their own user. A user with sysadmin SQL rights can see traces for all users and for all traces on the SQL Server, even those ones not started by the Support Debugging Tool.

Important:

Only traces started by the Support Debugging Tool can be stopped using the Active SQL Profile Traces window.
Module 7: Administrator Settings and Tools

This module guides you through using the Support Debugging Tool’s Administrator Settings and Dex.ini Configuration windows.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.
- Make sure Advanced Mode is enabled.

What You Will Learn

After completing this module, you will be able to:

- Select which Logging Modes should be used for Manual Logging Mode.
- Control the Automatic behavior of the Security Profiler.
- Differentiate between companies with visual cues.
- Configure default address, subject and body for emails.
- Setup the email transport to be used for the system.
- Update Dex.ini settings on individual workstations without having to visit them.
Lesson 17: Administrator Settings

This lesson provides an overview of the Administrator Settings window.

What You Will Learn

After completing this lesson, you will be able to:

- Select Logging Modes for Manual Logging mode.
- Change Security Profiler Automatic Open Mode setting.
- Use visual cues to differentiate between companies.
- Setup email defaults and transport mode and credentials.

Company Settings, Color Coding & Email Setup

The Administrator Settings window can be opened from the Options menu on the main Support Debugging Tool window. If the option is not available, enable Advanced Mode from the Dex.ini Settings window.

The General Tab of Administrator Settings window can be used to change which logging modes are used Manual Logging Mode (and also as the defaults for Advanced Debugging Mode). The settings for SQL Profile Traces and Macro Recording can also be accessed from this window (covered previously).

From the General Tab, you can also select default settings and behavior for Security Profiler and ScreenShot. Setting the Security Profiler to automatically “Open on Errors & Warnings” will mean that security errors are captured when they first occur and don’t need to be reproduced.

You can also disable the inclusion of form information on the Security Privilege warning.
Using the Company Tab you can differentiate companies using visual cues such as color coding and also by changing the window title bar include the Company ID and/or User ID. You can also make a dialog show when first logging into Test or Historical companies.

The Email 1 tab is used to set up default values for the email engine, setting the administrator's email as well as default Subject and Body Text for the email. You can also define the signature added to all emails send from the Support Debugging Tool.
The Email 2 tab is used to define the transport method for sending Emails. The choices are Microsoft Outlook, SMTP Server or MAPI compliant client (which includes Microsoft Outlook).
Lesson 18: Dex.ini Configuration

This lesson provides an overview of the Dex.ini Configuration window.

What You Will Learn

After completing this lesson, you will be able to:

- Roll out Dex.ini setting changes to all workstations on a system.
- Cleanup unwanted Dex.ini settings.
- Restore Dex.ini setting values to specific values.
- Clear login credentials on a Terminal Server environment.

Updating Dex.ini Settings Automatically

The Dex.ini Configuration window can be opened from the Options menu on the main Support Debugging Tool window. If the option is not available, enable Advanced Mode from the Dex.ini Settings window.

We have already seen the Dex.ini Configuration window used to control the Administrator default settings for a system, but it can also be used to automatic add, change or remove Dex.ini settings from all workstations (without needing to physically visit the machine).

Examples of use include:

- Turning off the System Print Dialog (setting NoPrintDialog=TRUE).
- Changing the OLEPath setting to a new location after copying the files to a new share.
- Cleaning up printer spooler entries added on a Terminal Server or Citrix system.
- Clearing the SQLLastUser setting to forget the last User on a Terminal Server of Citrix system.
- Reset the default size and position for the Notes windows with the NoteWindow setting.
### Administrator controlled D excell settings for Support Debugging Tool

The settings below are used to automatically update D excell settings during the login process to values set by the Administrator.

<table>
<thead>
<tr>
<th>Setting Path Setting</th>
<th>Admin Setting</th>
<th>Path Not Set by</th>
<th>Advanced Mode Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNTX</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Path Set by</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Administrator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>set by Adminis</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>set by Adminis</td>
<td></td>
</tr>
</tbody>
</table>

**Do not update any D excell settings automatically**

(This checkbox applies to the current workstation only)
Module 8: Backing Up and Transferring Data

This module guides you through using the Support Debugging Tool to backup and restore data or transfer data between systems.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.
- Make sure Advanced Mode is enabled.

What You Will Learn

After completing this module, you will be able to:

- Export data to XML files.
- Import previously exported data.
Lesson 19: Exporting Data to XML files

This lesson explains how to export data to XML files using the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Export data from multiple tables to a single XML file.
- Apply SQL Where clauses to limit the exported data to a subset of the entire table.

Exporting Data

The XML Table Export window is from the Options menu on the main Support Debugging Tool window. If the option is not available, enable Advanced Mode from the Dex.ini Settings window.

The XML Table Export window is used to create a profile ID with a list of tables in the system. Then the contents of the tables are exported to a single xml file.

Multiple Profile IDs can be created to allow for different sets of files to be exported. Usually a Profile ID will include all related tables and all tables in a logical table group.
An XML file created by XML Table Export can then be entered in the XML Table Import window and a selection of the tables can be imported back.

This functionality can be used:

- To backup data, so it can be restored back to the same state
- To transfer data or settings between systems, such as security settings.
- To obtain data from a customer's system without needing the entire database
- Provide data in form that can be read by any xml aware software.
- Transferring data between systems using different account frameworks.

**Limiting Exported Data**

The XML Table Export window allows an Option Where Clause to be added for each table. The SQL where clause is they applied to table to limit the records exported by the tool.

The lookup button for the Optional Where Clause field can be used to insert field names into the where clause line.

Limiting the data can be used:

- To backup only the data that you are likely to change.
- Export transactions for a date range or specific customers.
Lesson 20: Importing Data from XML files

This lesson explains how to import data from previously exported XML files using the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Use the XML Table Import window to import data previously exported data.

Importing Data

The XML Table Import window is from the Options menu on the main Support Debugging Tool window. If the option is not available, enable Advanced Mode from the Dex.ini Settings window.

If the XML Table Export window is open, the Import Path will be set to the Export Path of the XML Table Export window.

When importing data, you can select to replace the entire contents of a table or if merging data into a table you can specify if overwriting duplicate records is allowed.

Tip:

XML Table Import can also be used to import tables or records exported by Automatic Debugger Mode triggers.
Module 9: Troubleshooting Customizations

This module guides you through using the Support Debugging Tool to troubleshoot issues that are possibly caused by third party products or customizations.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.
- Make sure Advanced Mode is enabled.

What You Will Learn

After completing this module, you will be able to:

- Enable and disable triggers for third party products.
- Enable and disable alternate windows for third party products.
- Enable and disable modified windows for customizations.
- View and validate the Dynamics.set Launch File.
- Remove products from the Dynamics.set Launch File.
- Change the order of products in the Dynamics.set Launch File.
Lesson 21: Dictionary Control window

This lesson provides an overview of using the Dictionary Control window.

What You Will Learn

After completing this lesson, you will be able to:

- Disable and enable Dexterity triggers for third party Products.
- Disable and enable alternate and modified windows.

Disabling / Enabling products

When troubleshooting an issue, there are times where we believe that the problem could be caused by a third party Dexterity customization or a modified window.

In these situations, the usual troubleshooting process is to manually edit the DYNAMICS.SET launch file to remove products one by one until the behavior changes. Then you know that the last removed product is likely to be the cause. This process is time consuming and has a risk of corrupting the DYNAMICS.SET file. In addition, security settings need to be altered if an alternate window or report exists for a product that has been removed from the launch file.

It is possible to disable triggers with the Customization Status window; however this would need to be manually changed each time Dynamics GP is re-launched as there is no memory of which products are disabled.

The Dictionary Control window makes disabling products simple. It can disable triggers for the current session only, or keep them disabled after exiting and logging in again. It can also disable alternate and/or modified windows for a product for the current session or after exiting and logging in again.

The Dictionary Control window is accessed from the Options button on the main Support Debugging Tool window. If the option is not available, then enable Advanced Mode from the Dex.ini settings window.
Viewing the Dynamics.set Launch File

To view the Dynamics.set Launch file, press the i button on the right hand edge of the window. This will open the Show Launch File window. This window shows each line of the Dynamics.set Launch File with a description of what that line should be used for. If you check the bottom of the data shown and it is missing any lines of the final set of three paths, it could mean that there is some corruption in the file.

Changing the Dynamics.set Launch File

You can also remove or re-order dictionaries in the launch file using buttons on the right hand edge of the window. Re-ordering the dictionaries can resolve issues when triggers from two products are clashing. Sometimes the actions of one product can cause another product to fail. By swapping the order of the products in the launch file, we can control the order that the triggers execute and so avoid the errors occurring.
Module 10: Executing Scripts

This module guides you through using the Support Debugging Tool’s ad-hoc scripting capabilities.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.
- Make sure Advanced Mode is enabled.
- Have knowledge of SQL Transact-SQL to use SQL Execute.
- Have knowledge of Dexterity sanScript to use Runtime Execute.

What You Will Learn

After completing this module, you will be able to:

- Create and execute SQL Transact-SQL scripts.
- Create and execute Dexterity sanScript scripts.
Lesson 22: Executing T-SQL Scripts

This lesson explains how to execute ad-hoc SQL Transact-SQL scripts.

What You Will Learn

After completing this lesson, you will be able to:

- Create and execute SQL Transact-SQL scripts.
- Use the Table & Field Lookup.

Executing SQL Scripts

The SQL Execute window can be used like a Query Analyzer window from inside Dynamics GP. It can be used to execute any Transact-SQL statements, such as select, insert, update and delete statements.

Open the SQL Execute window from the Options button of the Support Debugging Tool main window.
This window can be very useful if access to the SQL server or tools is restricted by a system administrator. Scripts created in SQL Execute can be loaded can called from Automatic Debugger Mode and Runtime Execute.

The SQL Execute window can either use the SQL physical names to refer to tables and columns or it can use Dexterity technical names surrounded by braces {}. For example:

```sql
select {'Customer Number' of table RM_Customer_MSTR} from {table RM_Customer_MSTR}
```

or

```sql
select {Customer Number} from {RM_Customer_MSTR}
```

when executed automatically converts to

```sql
select CUSTNMBR from RM00101.
```

**Tip:**

Use the Tables Button to lookup the names for Tables and Fields from any loaded Dexterity product.

**Critical:**

Knowledge of SQL Server Transact-SQL is required to use this functionality.
Lesson 23: Executing Dexterity Scripts

This lesson explains how to execute ad-hoc Dexterity sanScript scripts.

What You Will Learn

After completing this lesson, you will be able to:

- Create and execute Dexterity sanScript scripts.
- Use the Form, Window & Field and the Table & Field Lookups.
- Use Helper Functions and Insert common code constructs

Executing Dexterity Scripts

The Runtime Execute window can be used to execute a Dexterity sanScript script in the context of any installed dictionary on a system, without needing the Dexterity development environment installed.

Open the Runtime Execute window from the Options button of the main Support Debugging Tool window.

The scripts can be used to call existing scripts in an existing dictionary or when procedural code can be more effective is performing the desired actions.
The script can use Helper functions to re-use and call other scripts created by the Runtime Execute window and SQL scripts created by the SQL Execute window. This means that we can execute Dexterity code against any table in any dictionary as well as SQL statements against any table.

Calling other scripts can work like calling procedures in Dexterity. There are helper functions that can be used to pass data between the scripts so “parameters” can be passed.

Tip: Use the Names Button to lookup the names for Forms, Windows and Fields from any loaded Dexterity product.

Tip: Use the Tables Button to lookup the names for Tables and Fields from any loaded Dexterity product.

Critical: Knowledge of Dexterity sanScript is required to use this functionality.
Module 11: Automatically Capturing Application Logs

This module guides you through using the Support Debugging Tool to capture application logs based on events and conditions in the application.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.
- Set up the SQL Profile Tracing configuration.
- Set up the Macro Recording configuration.
- Make sure Advanced Mode is enabled.

What You Will Learn

After completing this module, you will be able to:

- Create Automatic Debugger Mode triggers.
- View the Automatic Debugger Mode Status.
- Enable and Disable Automatic Debugger Mode triggers.
Lesson 24: Automatic Debugger Mode

This lesson provides an overview of creating Automatic Debugging Mode Triggers.

What You Will Learn

After completing this lesson, you will be able to:

- Create an Automatic Debugger Mode trigger.
- Select actions for an Automatic Debugger Mode trigger.
- Edit the script for an Automatic Debugger Mode trigger.
- Adjust options for an Automatic Debugger Mode trigger.

Automatically capturing Application Logs

Automatic Debugger Mode works by using Dexterity triggers to watch for an “event” in the system. It then turns on the application logs (as defined on the trigger) and waits for the event to occur.

When the event occurs, the trigger fires and runs a Dexterity sanScript script. The script looks for the situation we are trying to capture. If it has occurred, the logs are stored and specified actions (such as displaying a message or sending an email) are performed. If it has not occurred, the logs are deleted. Then the logging is restarted and the tool waits for the next event.

The trigger events can be almost any user action, table level change or program script executed.

An example use could be monitoring a particular table for when a field value changes and then checking if the field value is correct by looking at other tables. The trigger would use the save table event and the script would be able to access any table in the system.

Open the Support Debugging Tool setup window used creating Automatic Debugger Mode triggers from the Option button on the Support Debugging Tool main window.
Resource Tab

The screenshot following shows the settings on the Resource tab to monitor the Customer Master (RM_Customer_MSTR) table for changes to the Hold field.
Actions Tab

The actions to be performed when the trigger fires are specified on the Actions tab. In this example we are displaying a message using a desktop alert.

We also have options to send emails, capture and email Screenshots, export table records or the entire table.
Script Tab

The script and the dictionary context to execute it in are specified on the Script tab. In our case the script is watching for when the Hold field is saved (add or update) as checked.

The script can use Helper functions to re-use and call Dexterity scripts created by the Runtime Execute window and SQL scripts created by the SQL Execute window. This means that we can execute Dexterity code against any table in any dictionary as well as SQL statements against any table. You can choose which method is best for your purpose.

Tip:
Use the Names Button to lookup the names for Forms, Windows and Fields from any loaded Dexterity product.

Tip:
Use the Tables Button to lookup the names for Tables and Fields from any loaded Dexterity product.

Critical:
Knowledge of Dexterity sanScript is required to use this functionality.
**Options Tab**

The Options tab is used to specify an optional date range for the trigger and also which application logs to capture.

In our case we will capture all of the available logs.

![Support Debugging Tool Setup](image.png)

**Note:** To use the SQL Profile Trace and Macro Recording modes they need to be configured using the Administrator Settings window.
Lesson 25: Automatic Debugger Mode Status

This lesson provides an overview of viewing active Automatic Debugging Mode Triggers.

What You Will Learn

After completing this lesson, you will be able to:

- View the active Automatic Debugger Mode triggers.

Viewing Active Automatic Debugger Mode Triggers

You can view the active Automatic Debugger Mode triggers using the Automatic Debugger Mode Status window. To open this window either click on the “Automatic Debugger Mode” hyperlink or use the Options button on the main Support Debugging Tool window.
Lesson 26: Starting and Stopping Automatic Debugger Mode Triggers

This lesson provides an overview of Starting and Stopping Automatic Debugging Mode Triggers.

What You Will Learn

After completing this lesson, you will be able to:

- Start and Stop Automatic Debugger Mode triggers.

Automatically Starting Triggers

Automatic Debugger Mode triggers will start automatically after login when the Start Trigger Automatically on Login checkbox is selected for the trigger.

**Important:**

Triggers marked to automatically start will not start if Debugger Setup Mode is enabled in the Dex.ini setting window.

**Tip:**

Triggers marked to automatically start can be defined to only start for specific user and company combinations using the User button from the Support Debugging Tool Setup window.

Manually Starting Triggers

To manually start Automatic Debugger Mode triggers, use the Turn On Automatic Debugger Mode button on the main Support Debugging Tool window. You can select to start the Default trigger only, only triggers marked as automatic start (and not disabled), or all triggers (except disabled triggers).

Manually Stopping Triggers

To manually stop Automatic Debugger Mode triggers, open the Automatic Debugger Mode Status window and use the Unregister button. You can select an individual trigger to be stopped, stop all logging triggers, or stop all triggers.
Module 12: Exporting and Importing Configuration Settings

This module guides you through Exporting and Importing configuration settings for the Support Debugging Tool.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.

What You Will Learn

After completing this module, you will be able to:

- Select the path and configuration settings to export.
- Export configuration settings.
- Import configuration settings.
Lesson 27: Exporting and Importing Settings

This lesson provides an overview of exporting and importing configuration settings.

What You Will Learn

After completing this lesson, you will be able to:

- Select the path and settings to export.
- Export settings to a XML file.
- Import previously exported settings.

Exporting and Importing Settings Files

All of the settings created in the Support Debugging Tool can be selectively exported and sent to a customer or partner to install on a live system.

From the Configuration Export/Import window, select the path and the settings to export and click the Export button.

To import, select the file and click the Import Button. You will be shown the contents of the settings file and warned if any settings will be overwritten (similar to how VBA packages work).

When you click OK, the Path and checkbox settings you have made will be saved. This makes it easy to keep exporting the same settings as you continuing working on them.
Module 13: Advanced Debugging and Development Techniques

This module guides you through using the Support Debugging Tool for a number of advanced debugging and development techniques.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.

What You Will Learn

After completing this module, you will be able to:

- Create Non-Logging Automatic Debugger Mode triggers.
- Use Dexterity triggers to capture and store parameters.
- Use the Runtime Execute window to create custom Report Writer functions.
Lesson 28: Non-Logging Triggers

This lesson provides an overview of creating Automatic Debugging Mode Triggers.

What You Will Learn

After completing this lesson, you will be able to:

- Create an Automatic Debugger Mode trigger.
- Select actions for an Automatic Debugger Mode trigger.
- Edit the script for an Automatic Debugger Mode trigger.
- Adjust options for an Automatic Debugger Mode trigger.

Using Non-Logging Triggers to fix or prototype code

The Support Debugging Tool's Advanced Debugging mode has the ability to create non-logging triggers. These are triggers which can be registered against any of the supported events which do not start any of the logging processes. They can be used to help capture information when debugging issues or they can be used to create on-the-fly triggers which can be used to fix issues or prototype code.

A non-logging trigger is specified with the “Do not activate Logging Mode” checkbox and can still be started automatically after logging in.

Tip:
The Minimize Debugger Log Entries option will avoid logging details of when the trigger is firing unless there is a problem. It is recommended that this option is only enabled once the trigger has been tested and is working fine.
Non-logging triggers can be used to alter behavior of the original code by manipulating data in tables or fields. You can use secondary non-logging triggers to restore the data back to its original values after the event. You can use the built-in helper functions to store and retrieve parameters to keep track of the original values.

Non-logging triggers on focus events can use additional actions when the OUT\_Condition parameter is set to true.

- If the Trigger Attach mode is set to run before the original script, you can issue a Reject Script command to abort further processing.
- If the Trigger Event mode is set for the Scrolling window Scroll fill event, you can issue a Reject Record command to prevent the current record from being displayed.
- If the Trigger Event mode is set for a field’s Field Change or Field Post events, you can issue commands to keep the focus or restore the previous value.
Non-logging triggers can be used to implement temporary fixes until the updated code with the fix is released by the original developers.

For examples of non-logging triggers in use, please have a look at these blog posts:


**Lesson 29: Capturing Parameters with Automatic Debugger Mode triggers**

This lesson provides a technique that can be used to capture function and procedure parameters for use with Automatic Debugging Mode Triggers.

**What You Will Learn**

After completing this lesson, you will be able to:

- Create Dexterity triggers to capture and store parameters.
- Retrieve previously stored parameters inside an Automatic Debugger Mode trigger.

**Creating a wrapper trigger to store parameters for Support Debugging Tool to use where parameters are needed**

The Support Debugging Tool can trigger on functions and procedures, however it is unable to capture the parameters for a function or procedure because you need to include the parameter list in the trigger handler’s code at development time.

If you need access to the parameters, you can use Dexterity to create a small chunk to trigger on the function or procedure and use the Helper function in the Support Debugging Tool to store the parameters.

```plaintext
call with name "MBS_Param_Set" in dictionary 5261, "Variable", MBS_Parameter;
```

Then the script in Automatic Debugger Mode trigger can use the Helper functions to read the stored parameter values and then delete the data when it is no longer needed.

```plaintext
call with name "MBS_Param_Get" in dictionary 5261, "Variable", MBS_Parameter;
call with name "MBS_Param_Del" in dictionary 5261, "Variable";
```

**Note:**

This method will work even if the custom chunk is later in the DYNAMICS.SET launch file than the Support Debugging Tool as the Support Debugging Tool registers its triggers on login and not with the Startup script.

For an example have a look at the following blog post:

Lesson 30: Working with Report Writer

This lesson provides an overview of creating custom Report Writer functions.

What You Will Learn

After completing this lesson, you will be able to:

- Create a custom Report Writer function with Runtime Execute.

Creating Custom Report Writer functions

The Support Debugging Tool has the facility to allow custom RW Functions to be created by calling Runtime Execute scripts.

When a modified report uses the \texttt{rw\_ReportStart}, \texttt{rw\_ReportEnd}, \texttt{rw\_TableHeaderString}, \texttt{rw\_TableHeaderCurrency}, \texttt{rw\_TableLineString}, or \texttt{rw\_TableLineCurrency} report writer user defined functions and passes the Dictionary ID of 5261 and the Script ID as the first two parameters, the script will be executed and the saved parameter returned as the result.

These functions can be used to capture information and write them into the Debugger log file. They can also be used to create custom RW functions.

The Helper Functions window can generate the code template with the parameter handling to be used with each RW Function.

See the following Knowledge Base (KB) article:

https://mbs.microsoft.com/knowledgebase/KBDisplay.aspx?scid=kb;en-us;888884

The following blog article has details instructions for an example:

Module 14: Other Considerations and Information

This module covers additional considerations and information relating to the Support Debugging Tool.

Before You Begin

Before starting this module, you should:

- Install the Support Debugging Tool.
- Set up the tool using the recommended configuration.

What You Will Learn

After completing this module, you will be able to:

- Understand how the Support Debugging Tool works in a Terminal Server environment.
- Understand the limitations of the Support Debugging Tool.
Lesson 31: Terminal Server Environment

This lesson provides information on how the Support Debugging Tool works in a Terminal Server or Citrix environment.

What You Will Learn

After completing this lesson, you will be able to:

- Understand what the considerations for running the Support Debugging Tool in a Terminal Server environment are.

Consideration for setup in Terminal Server environment

The DEXSQL.LOG does not allow a name to be specified, which means that it will capture the information for all users using the same application folder.

Manual Logging Mode allows logs to be captured for a single user without needing to make changed to the Dex.ini file. Avoiding changes to the Dex.ini file means that the logging does not affect other users.

Automatic Debugger Mode triggers can have users and companies specified, so that only the selected users will have the logging enabled.

In the Dex.ini settings window you can specify a different path for the DEXSQL.LOG to be stored. If you use a drive letter (for example: H:\) which is assigned to the user's home folder, you can limit the DEXSQL.LOG to only contain data from one user.

If using a farm of Terminal Servers, you might want to specify a single shared location for the Debugger files to be stored. You can use the Dex.ini Configuration window to set this path as the Administrator default setting.
Lesson 32: Limitations of the SDT

This lesson provides information on the limitations of the Support Debugging Tool.

What You Will Learn

After completing this lesson, you will be able to:

- Know the limitations of the Support Debugging Tool.

Limitations of the Support Debugging Tool

The Table based Dexterity Triggers cannot monitor changes made directly by SQL. This could be by pass-through SQL commands, SQL Stored Procedures, SQL Triggers, or direct table updates from eConnect, Integration Manager SQL optimized connectors or any other application.

Screenshot cannot capture Modal windows, System Dialogs or Dexterity Exception Dialogs.

* End of document - Support Debugging Tool Training Workbook.docx - DM - 26 March 2012 *