GP Power Tools for Microsoft Dynamics™ GP

User’s Guide

Build 27
Copyright
Copyright © 2014-2020 Winthrop Development Consultants. All rights reserved.

Limitation of Liability
This document is provided "as-is". Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

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 Winthrop Development Consultants.

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

Trademarks
Microsoft, Excel, Internet Explorer, Microsoft Dexterity, Microsoft Dynamics, Outlook, and SQL Server are trademarks of the Microsoft group of companies. FairCom and c-tree Plus are trademarks of FairCom Corporation and are registered in the United States and other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Warranty disclaimer
Winthrop Development Consultants disclaims any warranty regarding the sample code contained in this documentation, including the warranties of merchantability and fitness for a particular purpose.

License Agreement
Use of this product is covered by a license agreement provided with the software product.

Application & documentation designed, developed and supported by

David Musgrave of Winthrop Development Consultants
Contents

Chapter 1:  Introduction  1
   Examples of use  4
   Support  5
   GP Power Tools Portal  5

Chapter 2:  Installation and Configuration  6
   Installation  7
   Security  8
   Navigation  11
   Recommended Configuration  15
   SQL Profile Tracing Configuration  20
   Macro Recording Configuration  25
   About GP Power Tools  26
   GP Power Tools Registration  28
   GP Power Tools Update Check  30
   GP Power Tools Feedback Survey  31
   Advanced Mode Access  32
   GP Power Tools and the Web Client  33

Chapter 3:  System Module Features  34
   Manual Logging Mode  35
   ScreenShot  41
   Send Email  45
   Calculator  49
   Dex.ini Settings  51
   Configuration Export/Import  60
   Administrator Password Setup  62
   Logging Settings  64
   Email Settings  75
   Configuration Maintenance  79
   Setup Backup and Restore  80
   Additional System Features  81

Chapter 4:  Administrator Tools Features  83
   Resource Information  84
   Security Profiler  98
   Security Information  102
   Security Log  110
   Security Analyzer  114
   Deny Based Security – Introduction  117
   Deny Based Security – Enhanced Security  118
   Deny Based Security – Security Denied  122
   Deny Based Security – Security Hidden  124
   Administrator Settings  126
   Dex.ini Configuration  142
MBS_Runtime_Execute_Delayed 312
MBS_SQL_Check_Exists 313
MBS_Export_SQL_Query_To_File 314
MBS_SQL_Results 315
MBS_SQL_Results_Goto 316
MBS_SQL_Goto_Get_Data 317
MBS_Net_Execute 318
MBS_Script_Load_Dex 320
MBS_Script_Load_SQL 321
MBS_Script_Load_Net 322
MBS_Param_Set 323
MBS_Param_Get 324
MBS_Param_Del 325
MBS_Param_DelAll 326
MBS_Auto_Log 327
MBS_Trigger_Start 328
MBS_Trigger_Stop 329
MBS_Logging_Start 330
MBS_Logging_Stop 331
MBS_DUOS_Set 332
MBS_DUOS_Get 333
MBS_DUOS_Del 334
MBS_DUOS_DelAll 335
MBS_SQL_Lookup 336
MBS_SQL_Lookup_Parameter 337
MBS_Form_Lookup 338
MBS_Form_Lookup_Parameter 339
MBS_Project_Start 340
MBS_Project_Stop 341
MBS_Script_Substitute 342
MBS_Parameter_Placeholder 343
MBS_Parameter_String 344
MBS_Parameter_Number 345
MBS_Parameter_Currency 346
MBS_Parameter_Boolean 347
MBS_Parameter_Date 348
MBS_Parameter_Time 349
MBS_Parameter_Load 350
MBS_Parameter_Set_String 351
MBS_Parameter_Set_Number 352
MBS_Parameter_Set_Currency 353
MBS_Parameter_Set_Boolean 354
MBS_Parameter_Set_Date 355
MBS_Parameter_Set_Time 356
MBS_Parameter_Get_String 357
MBS_Parameter_Get_Number
MBS_Parameter_Get_Currency
MBS_Parameter_Get_Boolean
MBS_Parameter_Get_Date
MBS_Parameter_Get_Time
MBS_getmsg
MBS_Token
MBS_Field_ParseText
MBS_Security_Form_Check

Chapter 9: RW Functions
rw_ReportStart
rw_ReportEnd
rw_TableHeaderString
rw_TableHeaderCurrency
rw_TableLineString
rw_TableLineCurrency

Chapter 10: Service Procedures
ServiceCreateCustom
ServiceDeleteCustom
ServiceGetCustom
ServiceUpdateCustom
ServicePostCustom

Chapter 11: Developer APIs
MBS_Email_API
MBS_WindowPositionCheck
MBS_WindowPositionMemory
MBS_WindowPositionMemoryResize

GP Power Tools Index
Chapter 1: Introduction

GP Power Tools is a primarily Dexterity based with Visual C# and Visual Basic.Net components suite of utilities and tools created to assist with the task of supporting Microsoft Dynamics GP.

GP Power Tools is divided into three modules which can be purchased separately with some standard features available to all modules. The modules are:

- Administrator Tools
- Developer Tools
- Database Tools

The standard features are part of the System Module which is automatically registered when one or more of the three modules above is registered.

The features of GP Power Tools are also divided into User level (Standard Mode) and Administrator level (Advanced Mode) features.

Standard Mode features are read-only and can be safely used by all users in a system. Advanced Mode features include scripting and accessing system settings and should only be used by system Administrators. To access an Advanced Mode feature, a user will need elevated privileges at the SQL Server level in addition to application level security and an optional system or administrator password.

Below is a list of what features are contained in each module with Advanced Mode features highlighted with an asterisk (*).

The **System Module** contains the following features:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logging Control including Manual Logging Mode</td>
<td>Manually turn on SQL Logging and Dexterity Logging and Profiling</td>
</tr>
<tr>
<td>ScreenShot</td>
<td>Capture and either email or save Screenshots and System Status information</td>
</tr>
<tr>
<td>Send Email</td>
<td>Send Email messages from within the application</td>
</tr>
<tr>
<td>Calculator</td>
<td>Touch friendly standard calculator with clipboard integration</td>
</tr>
<tr>
<td>Dex.ini Settings</td>
<td>Change System and Debugger Dex.ini Settings for the current workstation</td>
</tr>
<tr>
<td>Configuration Export/Import</td>
<td>Export and Import settings</td>
</tr>
<tr>
<td>Administrator Password Setup*</td>
<td>Create optional separate password to be used when accessing Advanced mode features</td>
</tr>
<tr>
<td>Logging Settings*</td>
<td>Change system wide Logging Settings such as shared path location, default logs and SQL Profile Trace setup</td>
</tr>
<tr>
<td>Email Settings*</td>
<td>Change system wide Email Settings controlling the email engine used by the tool</td>
</tr>
<tr>
<td>Configuration Maintenance*</td>
<td>Clear GP Power Tools data tables</td>
</tr>
<tr>
<td>Setup Backup and Restore*</td>
<td>Backup all data in SQL Tables to Debugger.xml file and restore from Debugger.xml to SQL Tables</td>
</tr>
</tbody>
</table>
The **Administrator Tools** module contains the following features:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Information</td>
<td>Obtain Details of any Table, Form, Window, Field or Report resource</td>
</tr>
<tr>
<td>Security Profiler</td>
<td>Monitor all Security check activity</td>
</tr>
<tr>
<td>Security Information</td>
<td>Display Security settings for specific resources for a user and company</td>
</tr>
<tr>
<td>Security Log</td>
<td>Security activity tracking for users, companies and the entire system</td>
</tr>
<tr>
<td>Security Analyzer</td>
<td>Displays results of various queries against the Security data for the system to highlight possible security related issues</td>
</tr>
<tr>
<td>Enhanced Security</td>
<td>The primary interface to Deny Based Security additional security layer including denying security and hiding menu navigation options on a per user per company basis</td>
</tr>
<tr>
<td>Security Denied</td>
<td>Deny Based Security maintenance window for security denied</td>
</tr>
<tr>
<td>Security Hidden</td>
<td>Deny Based Security maintenance window for security hidden</td>
</tr>
<tr>
<td>Administrator Settings*</td>
<td>Change system wide Administrator Settings controlling the behavior of the tool, including Company Color Themes and Automatic Logout</td>
</tr>
<tr>
<td>Dex.ini Configuration*</td>
<td>Automatically update Dex.ini settings across multiple workstations</td>
</tr>
<tr>
<td>Dictionary Control*</td>
<td>Enable and Disable third party products and VBA and Visual Studio customizations</td>
</tr>
<tr>
<td>Company Login Filter*</td>
<td>Filter companies available when logging in based on the installation folder and/or launch file name used</td>
</tr>
<tr>
<td>Window Position Memory*</td>
<td>Automatically remember a user’s preferred window position, size and state for any window in Microsoft Dynamics GP</td>
</tr>
<tr>
<td>User Activity Log*</td>
<td>User Login Activity tracking to record logins and logouts and track daily maximum session count on a system, user and company basis</td>
</tr>
<tr>
<td>Login Limits*</td>
<td>Limit user logins on a system, user and company basis</td>
</tr>
<tr>
<td>Launch File Configuration*</td>
<td>Automatically update Dynamics.set launch files across multiple workstations</td>
</tr>
<tr>
<td>Dynamic Product Selection*</td>
<td>Allows selection between multiple versions of the same window or report as the window or report is opened</td>
</tr>
</tbody>
</table>
The **Developer Tools** module contains the following features:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigger Status</td>
<td>Display currently active triggers and</td>
</tr>
<tr>
<td>Runtime Executer</td>
<td>Run published Dexterity sanScript scripts</td>
</tr>
<tr>
<td>SQL Executer</td>
<td>Run published Transact SQL scripts</td>
</tr>
<tr>
<td>.Net Executer</td>
<td>Run published C# and VB.Net scripts</td>
</tr>
<tr>
<td>Project Setup*</td>
<td>Group together all the related Triggers, Scripts (Dex, SQL, .Net) and</td>
</tr>
<tr>
<td></td>
<td>Parameter Lists into a Project</td>
</tr>
<tr>
<td>Trigger Setup*</td>
<td>Automatically starts logging and watches for specified events</td>
</tr>
<tr>
<td>Runtime Execute Setup*</td>
<td>Create and run Dexterity sanScript scripts</td>
</tr>
<tr>
<td>SQL Execute Setup*</td>
<td>Create and run Transact SQL scripts</td>
</tr>
<tr>
<td>.Net Execute Setup*</td>
<td>Create and run C# and VB.Net scripts</td>
</tr>
<tr>
<td>Parameter Lists*</td>
<td>Create interactive parameter dialogs to be used with scripting features.</td>
</tr>
<tr>
<td>Dynamic Trigger Logging*</td>
<td>Track execution paths by dynamically registering triggers</td>
</tr>
<tr>
<td></td>
<td>against events in Dynamics GP and logging when they occur</td>
</tr>
</tbody>
</table>

The **Database Tools** module contains the following features:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Table Export*</td>
<td>Export any table(s) to an XML file</td>
</tr>
<tr>
<td>XML Table Import*</td>
<td>Import previously exported tables</td>
</tr>
<tr>
<td>Database Validation*</td>
<td>Validate SQL user and database information and table structures</td>
</tr>
<tr>
<td>SQL Login Maintenance*</td>
<td>Reset Users’ Passwords and view or change password policy settings</td>
</tr>
</tbody>
</table>

On a registered system, if you open a window from an unregistered module, you might be asked if you wish to open the window in Preview Mode. If a window is opened in Preview Mode, you may use the window to explore its features, however its functionality will be limited.
Examples of use

GP Power Tools has many uses. Here are some examples:

- When issues or bugs occur, GP Power Tools helps you identify the specific series of events that led up to them.

- Got performance problems? Use GP Power Tools to quickly and simply turn on all logging and profiling capabilities without restarting GP.

- Do you want to know the details about dictionary resources? GP Power Tools gives you a complete and in-depth look at all security objects, including Forms, Windows, Tables, Reports, Fields, and Scripts.

- Find out what's causing security access issues by using GP Power Tools to identify the responsible forms, reports, or tables.

- Deny security access to individual resources on a per user per company basis without needing to duplicate security tasks and roles.

- To help with troubleshooting issues, GP Power Tools can easily enable or disable third party products or change the order of the products in the launch file.

- When you need to import or export data to any GP table at all, think GP Power Tools.

- Do you need to run SQL, Dexterity, C# or VB.Net scripts? You can do it with GP Power Tools even if you don't have Dexterity, Visual Studio or SQL Administration Tools installed.

- Capture, save, and email screenshots of all open windows and send a system status report at the same time with GP Power Tools.

- Even if your local system doesn't have Outlook installed, GP Power Tools can be used to send email to the system administrator.

- Overcome those difficult Report Writer (RW) user-defined function issues with GP Power Tools.

- GP Power Tools makes it easy to roll out Dex.ini setting changes to all workstation in your system.

- Before doing a GP upgrade, GP Power Tools can validate your SQL users, databases and table structures to help ensure it goes smoothly.

- How about creating brand new Security Tasks and Roles by tracing user activity that you capture interactively in GP Power Tools or from security activity tracking logs?

- Need extra functionality for a web service integration, create custom web services with GP Power Tools.
Support

Support for GP Power Tools is provided by Winthrop Development Consultants.

Support cases can be logged using the link below:

https://www.winthropdc.com/support.htm

GP Power Tools Portal

You can also find release histories, FAQ documents and lots of articles as well as links to download and purchase at the GP Power Tools Portal:

http://WinthropDC.com/GPPT
Chapter 2: Installation and Configuration

This chapter includes the following sections:

- Installation
- Security
- Navigation
- Recommended Configuration
- SQL Profile Tracing Configuration
- Macro Recording Configuration
- About GP Power Tools
- GP Power Tools Registration
- GP Power Tools Update Check
- GP Power Tools Feedback Survey
- Advanced Mode Access
- GP Power Tools and the Web Client
Installation

GP Power Tools is installed by downloading the installer and executing it. Follow the onscreen instructions to install the product files into the Microsoft Dynamics GP application folder and the Addins subfolder.

The installation contains the following files:

- **GPPTools.cnk** (self-installing dictionary)
- **GPPTools.txt** (readme file)
- **GPPTools.pdf** (this user guide manual)
- **GPPTools_License.doc** (the license agreement which you accept by using the tool)
- **Dex.chm** (Dexterity Help file)
- **Application.GpPowerTools.dll** (signed Dictionary Assembly)
- **Application.GpPowerTools.xml** (IntelliSense data for Visual Studio)
- **Application.GpPowerTools.Metadata.xml** (IntelliSense data for Visual Studio, version 14.0 or later)
- **Addins/WinthropDC.GpPowerToolsVC.dll** (Visual C# support)
- **Addins/WinthropDC.GpPowerToolsVB.dll** (Visual Basic.Net support)

Check the properties of all the dll files and Unblock them if necessary.

When Microsoft Dynamics GP is next launched, if asked, select “Yes” to include new code.

*If installing on a Windows Vista, Windows 7, Windows 8 (or 8.1), Windows Server 2008, Windows Server 2008 R2 or Windows Server 2012 system and User Account Control (UAC) is active, please launch Microsoft Dynamics GP with the Run as Administrator option to complete the installation.*

Log into Microsoft Dynamics GP with a user the appropriate SQL privileges to create tables, such as ‘sa’ or ‘DYNSA’. GP Power Tools will automatically create its SQL tables in the system database.

If you had a previous installation, GP Power Tools will read the Debugger.xml settings file to populate the initial data in the SQL tables.
Security

Security access must be granted to the forms of GP Power Tools before it can be used by users other than those belonging to the POWERUSER security role.

GP Power Tools will automatically create the Security Tasks and Security Roles required to use the tool. The following Security Roles are created.

GP POWER TOOLS USER (GP Power Tools User)

GP POWER TOOLS ADMIN (GP Power Tools Administrator)

GP POWER TOOLS PASSWORD (GP Power Tools Administrator Password)

GP POWER TOOLS SERVICES (GP Power Tools Services) for GP 2015 or later.

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 GP Power Tools: If logging into Microsoft Dynamics GP as a user belonging to the POWERUSER security role, and no users have been granted access to the GP POWER TOOLS USER security role, the system will offer to add this security role to all users for you.

If you respond Yes, the system will remind you to add the GP POWER TOOLS ADMIN security role to other users who need access to the Advanced Mode features (and do not already have access to the POWERUSER Security Role). You have the option to open the User Security Setup window when setup is completed.
You will then have the option to select a shared location for logs and export files to be stored in. If you select No, the default location is the Data folder in application folder for Microsoft Dynamics GP. If you select Yes, you will be presented with a dialog to select the path you wish to use. This path should point to a folder which has full control permissions for all users and can be specified using either a UNC pathname or a shared drive letter available to all users.

If you selected a shared location, then you will be asked if you would like to make this Administrator Controlled. Making the setting Administrator Controlled, automatically rolls the setting out to all workstations in the system on their next login and is the Recommended Configuration.
To manually grant security to the forms of GP Power Tools use the User Security Setup window (Microsoft Dynamics GP >> Tools >> Setup >> System >> User Security). After selecting the user and company, select one of the security roles below:

- GP POWER TOOLS USER (GP Power Tools User)
- GP POWER TOOLS ADMIN (GP Power Tools Administrator)
- GP POWER TOOLS PASSWORD (GP Power Tools Administrator Password)
- GP POWER TOOLS SERVICES (GP Power Tools Services) for GP 2015 or later.

If a user is not going to be using any of the windows of GP Power Tools, they do not need to be assigned to a security role. Automatic Trigger Mode will work regardless of security settings.
Navigation

Once logged into Microsoft Dynamics GP, a user with security access granted can find GP Power Tools Logging Control main window under the Tools menu underneath the Microsoft Dynamics GP menu (highlighted below). It also has the keyboard shortcut Ctrl+D assigned to it.

GP Power Tools also adds the Raise All Windows option to the main application menu and Tools menus, to allow for an easy method to send the main application window to the background. The option has the keyboard shortcut Ctrl+Shift+R assigned to it.

Also, added to the main application menu is the Exit After Processes option, which will request the application to exit after it has completed all background processing. The option has the keyboard shortcut Ctrl+Shift+X assigned to it.
From the GP Power Tools Logging Control main window, the Options button drop list can be used to open other windows.

In addition, GP Power Tools 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.

If using Microsoft Dynamics GP 2013 R2 or later in desktop mode with ribbons enabled instead of the menus, you can access GP Power Tools under the Tools button on the ribbon.
GP Power Tools can also be opened from the Standard Toolbar and from Quick Links on the Home Page.

All GP Power Tools windows are also available via the standard application menus under the GP Power Tools submenus. The GP Power Tools Logging Control main window can be found under Transactions >> GP Power Tools >> GP Power Tools Logging Control.

Finally, you can use the GP Power Tools Area Page by clicking on the GP Power Tools Navigation Pane button.

If the GP Power Tools button or icon are not visible, you might need to use the Navigation Pane Options or Show More Buttons menus from the bottom of the Navigation Pane.

From Microsoft Dynamics GP 2013 (Version 12.0) onwards, GP Power Tools adds the Find a Window option to the main application menu and window level Tools menu. This opens a normally hidden Microsoft Dynamics GP core window that can search the navigation menus for matching items and open them for you. The option has the keyboard shortcut Ctrl+F assigned to it.
Once the GP Power Tools Area Page is displayed, all the various windows will be displayed, including the main GP Power Tools Logging Control window (under Transactions).

When running on the Web Client, use the GP Power Tools area page or the Quick Links on the Home Page to open GP Power Tools.
Recommended Configuration

GP Power Tools stores its settings in SQL tables. If the SQL tables are empty and you had a previous install, the XML setup file called Debugger.xml while be accessed once to load the data into the SQL tables.

While a shared location is no longer required for the data storage it is recommended for the storage of all logs and export files created. This avoids having to visit an individual workstation to have access to the files.

The recommended configuration is for GP Power Tools to be installed on all workstations in the system and to point each workstation to use a single shared location.

Below are step by step instructions to install and set up the Recommended Configuration:

1. Initially install on a single instance of Microsoft Dynamics GP.

2. Launch Microsoft Dynamics GP using Run as Administrator and click Yes if asked “Do you wish to include new code now?”

3. If upgrading from a previous install, you might be asked to relaunch Microsoft Dynamics GP after changes were made to the Dynamics.set launch file. If, so go back to step 2.

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

5. If the SQL tables need to be created or updated, you will see a progress dialog in the bottom right of your screen as the tables are created. Any existing data will be preserved.

6. If upgrading a previous install, the SQL tables are empty and a Debugger.xml file can be located, it will be read to populate the SQL tables. You can import a different Debugger.xml file later if desired.
7. If asked to add the base user level of security access to all users, click Yes.

8. You will then be reminded that administrator level security settings will need to be set up manually. You can ask to open the User Security Setup window when setup is completed.

9. If asked to select a shared location for the setup files and logs, click Yes and select the path you wish to use. This path should point to a folder which has full control permissions for all users and can be specified using either a UNC pathname or a shared drive letter available to all users.
10. If asked about making the path setting for the shared location Administrator Controlled, click Yes.

![Microsoft Dynamics GP]

Do you want to make this path setting for shared location Administrator Controlled? Note: Setting a shared location for all users is the Recommended Configuration.

Yes  No

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

For user features:
- GP POWER TOOLS USER (GP Power Tools User)
- GP POWER TOOLS SERVICES (GP Power Tools Services)
  for GP 2015 or later.

For administrator features:
- GP POWER TOOLS ADMIN (GP Power Tools Administrator)
- GP POWER TOOLS PASSWORD (GP Power Tools Administrator Password)

*It is recommended to grant all users in the system access to GP POWER TOOLS USER. Only System Administrators need access to GP POWER TOOLS ADMIN, unless they already have access to the POWERUSER Security Role.*

12. Install GP Power Tools on all other workstations in the system.

The Recommended Configuration is now configured. To install on other workstations just requires the copying of the files and the including of new code.
Below are the manual steps showing where the responses to the dialogs for Folder Location settings can be manually changed:

To update the current workstation only:

1. Open the Dex.ini Settings window by selecting Dex.ini Settings from the Cards section of the GP Power Tools Area Page or by selecting Dex.ini Settings from the Options button drop list on the main window.

2. From the Dex.ini Settings window, on the Debug tab, select a Specified Pathname location for logs and export files.

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

3. Click OK to save the changes.
To update the Administrator controlled setting:

1. Open the Logging Settings window by selecting Logging Settings from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Logging Settings from the Options button drop list on the main window.

2. From the Logging Settings window, select a shared folder where all logs and export files will be written. This path should point to a folder which has full control permissions for all users and can be specified using either a UNC pathname or a shared drive letter available to all users.

3. Click OK to save the changes.

That is all that is required for the Recommended Configuration.

The pathname can be specified using a UNC path in the format \Server\Share\Folder\.
SQL Profile Tracing Configuration

For more information on setting up and enabling SQL Profile Tracing please see the section under the Logging Settings window.

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. Open the Logging Settings window by selecting Logging Settings from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Logging Settings from the Options button drop list on the main window.

5. From the Logging Settings window, click Edit SQL Profile Trace Settings to open the SQL Profile Trace Settings window.
6. 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.

7. 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.
8. 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.

9. You can change the Maximum Trace file size and Maximum number of Trace files if desired, or just leave the default values.
10. 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.

11. Enter the UNC Network Path set up previously (as created in step 2) and press tab.

12. 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.
13. Click OK to save the settings and close the SQL Profile Trace window.

14. On the Logging Settings window, 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 Trigger Mode.

15. Click OK to save the settings and close the Logging Settings window.
Macro Recording Configuration

For more information on enabling Macro Recording please see the section under the Logging Settings window.

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 Logging Settings window by selecting Logging Settings from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Logging Settings from the Options button drop list on the main window.

3. On the Logging Settings window, enable the Capture Macro Recording option. This will enable Macro Recording for Manual Logging Mode and as the default value for Automatic Trigger Mode.

4. Click OK to save the settings and close the Logging Settings window.
About GP Power Tools

You can open the About GP Power Tools window by selecting About GP Power Tools from the Routines section of the GP Power Tools Area Page or by selecting About GP Power Tools from the Options button drop list on the main window.

The About GP Power Tools window shows the current version, build and last modified date information.

You can uninstall GP Power Tools from this window. Clicking Uninstall will remove GP Power Tools 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 GP Power Tools.

You will also be asked if you want the Dynamics.set launch file updated to remove GP Power Tools, so that it does not re-install itself next time Microsoft Dynamics GP is launched.
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.

You can also re-install GP Power Tools from this window. Clicking Re-install will, after a confirmation dialog, remove GP Power Tools from the menus and security tables, then re-run the installation as discussed in the Installation section above.

To check which modules are registered you can click on the Info button.
GP Power Tools Registration

You can open the GP Power Tools Registration window by selecting GP Power Tools Registration from the Routines section of the GP Power Tools Area Page or by selecting GP Power Tools Registration from the Options button drop list on the main window.

The GP Power Tools Registration window can also be opened by clicking the Registration Button on the About GP Power Tools window, or from the Additional menu on the Microsoft Dynamics GP Registration window.

The current Microsoft Dynamics GP registration details of the system along with the registration keys for each module will be displayed.

Use the Contact Details button to complete or update the Contact Details for the site, including consenting to the Privacy Policy. The Privacy Policy must be accepted for the OK Button to be enabled.
The contact details must be for the end user of the system and not for a partner.
The Email address must be the valid address for the end user as it must be unique.
The Email address links the site details with the registration keys.

Use the Update Keys button to retrieve current keys for the system from the registration server.

Use the individual Trial Key button to retrieve the key for that module and if no key is available, request a 30 day trial of that module.

If a Product Key has been manually provided, it entered directly into the Product Key field on the window. It is recommended to use copy and paste to minimize typing errors.

The Automatically check for updated keys option can be used to control the frequency that the system will automatically request updated keys from the registration server when the current keys are expired or optionally due to expire.

The Automatic check for updated keys only is executed when an Administrator user logs in.
GP Power Tools Update Check

GP Power Tools can automatically check online to see if an updated build or hotfix has been released. The GP Power Tools Update Check window can be opened by using the Options menu on the About GP Power Tools window and selecting Check for Updates.

The Automatically check for updates option can be used to control the frequency that the system will check for updates.

*If an update is available, you can select not to be notified again for this update. This will skip the one update and notify you when the next update is available.*

If the frequency has not been set, the following dialog will be displayed. Select Yes to open the GP Power Tools Update Check window.

*The Automatic check for update only is executed when an Administrator user logs in. This check is for information only, no updates to the system will be made.*
GP Power Tools Feedback Survey

GP Power Tools includes a dialog to prompt users to provide feedback via an online survey (http://WinthropDC.com/GPPT/Survey.htm).

The feedback is vital to keep improving GP Power Tools based on what the Microsoft Dynamics GP community want and need.

The dialog only automatically opens for users with POWERUSER application security or the SQL Server sysadmin fixed server role. It will open two days after a new installation of GP Power Tools or immediately with an upgrade of GP Power Tools.

Once the dialog is opened, a selection of an action from the drop-down list is required to close the window and continue. You can decide to complete the Survey which will open the default web browser to the page, or you can decide to postpone the survey to a later time (next login, tomorrow, 30 days, or after installing the next build).

Once completed, the dialog will display the date and user details.

The dialog can also be opened manually by all users by selecting GP Power Tools Feedback Survey from the Routines section of the GP Power Tools Area Page or by selecting GP Power Tools Feedback Survey from the Options button drop list on the main GP Power Tools Logging Control window.
Advanced Mode Access

To be able to access the Advanced Mode features of GP Power Tools, the current Microsoft Dynamics GP User ID will need to have either the SQL Server sysadmin fixed server role or membership of the db_owner role on the system database (DYNAMICS) and the current company database.

If the Microsoft Dynamics GP system password is configured, you will need to enter this password before the window will open.

Optionally, GP Power Tools can be configured to use its own Administrator password instead of the Microsoft Dynamics GP system password. If setup you will need to enter this Administrator password before the window will open.

Advanced Mode features are protected because they should be used only by Microsoft Dynamics GP system administrators, partner consultants or support engineers.

Some Advanced Mode features allow direct access to data stored on the SQL Server. Other features can be used to disable functionality of Microsoft Dynamics GP.
GP Power Tools and the Web Client

GP Power Tools works with the Web Client, however some features are disabled as the functionality is not supported in the Web Client environment.

Below is a summary of features which are disabled or modified when running on the Web Client:

- Accessing the tool is only via the Quick Links pane on the Home page.
- Macro Logging Mode is disabled.
- ScreenShot cannot capture bitmap images but can save or email System Status and other files.
- Changing Windows Titles to show User and/or Company is disabled.
- Preventing Windows opening outside the visible desktop is disabled.
- Changing background colors with Company Color Schemes is disabled.
- Microsoft Outlook Client email mode is not supported.
- Changing the launch file from Dictionary Control is disabled.
- Disabling VBA from Dictionary Control is disabled.
- Disabling Visual Studio Tools from Dictionary Control is disabled.
- Remembering position and size of windows is disabled.
- Using splitters on windows with two panes is disabled.
- Desktop Alerts show using a System Dialog.
- User Account Control (UAC) checks are disabled.
- Database Validation is not available when running on the web client.
- Use of Visual Studio dialogs such as MessageBox.Show() is not supported.
- Window Position Memory functionality is disabled.
Chapter 3: System Module Features

This chapter includes the following sections:

- Manual Logging Mode
- ScreenShot
- Send Email
- Calculator
- Dex.ini Settings
- Configuration Export/Import
- Administrator Password Setup*
- Logging Settings*
- Email Settings*
- Configuration Maintenance*
- Setup Backup and Restore*
- Additional System Features

* Advanced Mode Feature
Manual Logging Mode

The Microsoft Dynamics GP core application runs on the Dexterity runtime engine from which the following logging facilities are available:

SQL Logging
SQL Logging tracks all communication between the Microsoft Dynamics GP client and the SQL Server. The default file in which the SQL communication is stored is named DEXSQL.LOG.

The SQL Logging is tracked on a per workstation basis and will include information from more than one application session, if more than one session of Microsoft Dynamics GP is launched from the same application folder. This is normally the case for Terminal Server and Citrix installations.

Communication with the SQL Server using alternate methods of data access is not logged. For example; data access via Visual Basic for Applications (VBA) using ActiveX Data Objects (ADO) will not be captured by SQL Logging.

SQL Profile Tracing
SQL Profile Tracing can be used to log all activity at the SQL Server, including commands from inside Stored Procedures. The default file in which the SQL Profile Trace is stored is named Trace.trc.

SQL Profile Tracing is not enabled until it has been setup using the SQL Profile Trace Settings window under the Logging Settings.

SQL Profile Tracing will capture all activity at the SQL Server for the current user in the DYNAMICS database and the current company database, so communication with the SQL Server using alternate methods of data access (as described above) will be logged.

Dexterity Script Logging
Dexterity Script Logging tracks all Dexterity event script, procedure and function calls, including the script hierarchy. The default file in which the script log is stored is named Script.log.

Dexterity Script Profiling
Dexterity Script Profiling tracks the number of calls to each event script, procedure and function and how much time the calls have taken. It also tracks all table activity initiated by Dexterity and the time taken. The default file in which the script profile is stored is named Profile.txt.

Macro Recording
Macro Recording captures all activity performed by the user at the User Interface. The recorded Macro can be replayed to repeat the actions, or opened in Notepad.exe for analysis. The default file in which the macro is stored is named Macro.mac.
Macro Recording can be enabled using the Logging Settings window.

Macro Recording can only work when a single instance of Microsoft Dynamics GP is running on a workstation, or if multiple instances are running, Macro Recording will only work on the first instance launched.

Macro Recording is disabled when running on the Web Client.

Using any logging facility will create additional processing overhead for the application. Logging should only be used when actually looking to resolve an issue with the system.

Manual Logging Mode
By default, Manual Logging Mode will activate all of these logging options with a single mouse click without requiring the application to be restarted. The Dexterity runtime will continue to log activity in the application until stopped.

You can use the Logging Settings window to select which logging modes are enabled when using Manual Logging Mode.

To ensure that the log files are not overwritten, the User, Company and date and time information are appended to the default file name.

To start Manual Logging Mode, click on the Turn On button (highlighted below).

To stop Manual Logging Mode, click on the same button, now labeled as Turn Off.
Manual Logging can also be turned on using the Ctrl+Shift+F9 keyboard shortcut and turned off again using the Ctrl+Shift+F10 keyboard shortcut.

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.

When using Manual Logging Mode to look at a specific issue (possible programming bug or performance problem), it is important to minimize the information captured in the logs to just the events directly related to the issue. To achieve this, request the user to perform all the actions in Microsoft Dynamics GP up to just prior to where the issue occurs. At this point, activate the manual logging and perform the action that exhibits the issue. Once the issue has occurred, stop the logging as soon as possible.

The results of the logging can be found in the folder where GP Power Tools is storing its data files. The default location is the data subfolder beneath the Microsoft Dynamics GP application folder. The location can be changed from the default path using the Pathname location for Debugger Setup files, exports and logs option on the Dex.ini Settings windows (see section in this chapter).

The individual logs will be stored in the following files:

- **GPPTools_<User>_<Company>.log** or optionally **GPPTools_<User>_<Company>_<Date>.log**
  
  This file will contain all the details of the actions performed by GP Power Tools including the names of the files created during the logging process. Any error or warning messages from GP Power Tools will also be logged to this file. Use the Logging Settings window add the optional date to the file name.

- **DEXSQL_<Date>_<Time>.LOG**
  
  These files will contain the SQL Logging results.

- **Trace_<User>_<Company>_<Date>_<Time>_<Mode>.trc**
  
  These files will contain the SQL Profile Tracing results.

- **Script_<User>_<Company>_<Date>_<Time>.log**
  
  These files will contain the Dexterity Script Logging results.

- **Profile_<User>_<Company>_<Date>_<Time>.txt**
  
  These files will contain the Dexterity Script Profiling results.

- **Macro_<User>_<Company>_<Date>_<Time>.mac**
  
  These files will contain the Macro Recording results.

<User> will be substituted with the current User ID and <Company> will be substituted with the current Company ID code (InterCompany ID). <Date>, <Time> will contain the date and time at which the logging was started in the format YYYYMMDD_HHMMSS. <Mode> will be replaced with a letters A to E depending on the SQL Profile Trace mode used.
When using the Dex.ini Setting to Start Logging on next startup, the file names used will not have a User ID or Company ID code as these will not be known until after login has completed.

There is an optional password which can be used to control access to Manual Logging Mode. It is set up using the Logging Password field on the General Tab of the Logging Settings window.

**Individual Logging Control**

Individual Logging Control allows each of the logging options to be controlled independently. To access the Individual Logging Control features it must be enabled via Logging Settings. Then you can click on the Logging Options button on the GP Power Tools main window.

The pathnames of the resulting files can be left as default, created automatically based on User, Company and date and time information or they can be manually specified.

Turning all logging methods on using the Based on date and time and Based on User, Company, date and time is the same as using Manual Logging Mode.

Access to Individual Logging Modes can be enabled using the Enable Individual Logging Modes option on the General Tab of the Logging Settings window.
CHAPTER 3

SYSTEM MODULE FEATURES

SQL Profile Traces
Active SQL Profile Traces can be viewed by pressing the Show SQL Profile Traces button on the GP Power Tools main window. This will open the Active SQL Profile Traces window.

The window defaults to showing GP Power Tools Traces only for the SQL Profile Trace Application, and the Current User only for SQL Profile Trace User. A user with the sysadmin rights at the SQL Server level will be allowed to select All Traces on SQL Server or All Users modes.

Stranded SQL Profile Traces are traces created by GP Power Tools where the Microsoft Dynamics GP has unexpectedly terminated and left the trace running at the SQL Server. They can be stopped from this window by selecting the traces (use control and shift keys to multi-select) and then click Stop SQL Profile Trace.

The Show SQL Profile Traces button is enabled once SQL Profile Tracing has been enabled. For more information on setting up and enabling SQL Profile Tracing please see the section under the Logging Settings window or the section in the previous chapter.
When logging into Microsoft Dynamics GP, if there are stranded traces for the current user and company, the following dialog will be displayed.

![Microsoft Dynamics GP dialog](image)

The user can select whether to stop the stranded traces, leave them running or open the Active SQL Profile Traces window.

Also when logging into Microsoft Dynamics GP, if there are stranded traces for the current user in other companies which the user is currently not logged into, the following dialog will be displayed.

![Microsoft Dynamics GP dialog](image)

Again the user can select whether to stop the stranded traces, leave them running or open the Active SQL Profile Traces window.
ScreenShot

ScreenShot is a tool which can capture screenshots of all the open windows in the application as well as a System Status report and then either email or save the files.

ScreenShot creates reduced color bitmaps (4 bits per pixel, 16 colors) to ensure that the size of the email is kept to a minimum. It can capture all open windows regardless of whether they are overlaid by other windows.

The System Status report contains information about the system including registration information, current login information, environmental information (such as operating system, database and ODBC versions), product information (including all version and build numbers) and a list of the attached screenshots.

You can open the GP Power Tools ScreenShot window by selecting Capture Screenshots from the Transactions section of the GP Power Tools Area Page or by selecting Capture Screenshots from the Options button drop-down list on the main window.

You can open it directly from the Tools menu underneath the Microsoft Dynamics GP menu (highlighted below). It also has the keyboard shortcut Ctrl+S assigned to it.

If the Standard Toolbar is displayed, you can launch ScreenShot from the Capture Screenshots button (highlighted below).
You can also use the Capture Screenshots option on Quick Links on the Home Page. When running on the Web Client, use the Quick Links on the Home Page to open Capture Screenshots as the other navigation options are not available.

In addition, Capture Screenshots 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.

Once ScreenShot is activated, the following window will be displayed.
Below is a description of the individual fields on the window:

**Save Path**
This is the root path that will be used when saving screenshots. The actual path used will be a subfolder based on the user ID and company ID code.

**Include Dex.ini Settings File**
This checkbox tells ScreenShot whether to include the Global level Dex.ini settings file as an attachment for the email. The default setting for this checkbox can be set up in the Administrator Settings window.

**Include User Dex.ini Settings File**
This checkbox tells ScreenShot whether to include the User level Dex.ini settings file as an attachment for the email. The default setting for this checkbox can be set up in the Administrator Settings window.

**Include Current Launch File**
This checkbox tells ScreenShot whether to include the launch file, usually Dynamics.set, as an attachment for the email. The default setting for this checkbox can be set up in the Administrator Settings window.

**Include info for all databases**
This checkbox tells ScreenShot whether to include information for all databases or just the system database and current company database in the System Status report. Not including information for all databases gives better performance on systems with many companies. The default setting for this checkbox can be set up in the Administrator Settings window.

**Open Windows**
This is a list of windows currently open on the system. It is automatically updated when a form is open or closed. If you open a secondary window on a form, you can refresh the list with the Refresh Button. You can use the checkboxes to select which screenshots should be included.

**Info Button**
This button can be used to preview the System Status report. You can use Ctrl-A to select the contents of the report and then Ctrl-C to copy it to the clipboard.
Refresh Button
This button will refresh the window list with the currently open windows.

Mark All Button
This button will select all windows to be emailed or saved. This button will be disabled when running on the Web Client.

Unmark All Button
This button will de-select all windows so that individual windows can be selected. This button will be disabled when running on the Web Client.

Email Button
This button will create an email with the selected screenshots and System Status report attached. The System Status will also be included as the body of the email. All that the user needs to do is add a recipient and click Send. The default email settings can be set up in the Email Settings window.

Save Button
This button will save the selected screenshots and System Status report to a folder based on the Save Path and the current user ID and company ID code.

Cancel Button
This button will close ScreenShot.

When running on the Web Client, ScreenShot is unable to create the bitmap images and so this functionality is disabled.
Send Email

You can open the Send Email window by selecting Send Email from the Transactions section of the GP Power Tools Area Page or by selecting Send Email from the Options button drop list on the main window.

You can open it directly from the Tools menu underneath the Microsoft Dynamics GP menu (highlighted below). It also has the keyboard shortcut Ctrl+E assigned to it.

If the Standard Toolbar is displayed, you can launch Send Email from the Send Email button (highlighted below).

You can also use the Send Email option on Quick Links on the Home Page. When running on the Web Client, use the Quick Links on the Home Page to open Send Email as the other navigation options are not available.

In addition, Send Email 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.

The Send Email window can be used to create and send email messages from inside the Microsoft Dynamics GP application. This window will also appear to the user when other features in GP Power Tools are configured to send emails and the option to Preview emails is enabled in the Email Settings window.

The default email settings can be set up in the Email Settings window. This includes the Email address to use in the To address and the Default Subject and Default Body Text.
Email addresses can be in the following formats and multiple addresses should be separated by a semi-colon (;):

- name@domain.com
- Full Name<name@domain.com>
- Full Name (when in Microsoft Outlook mode only)

Below is a description of the individual fields on the window:

**From Field**
This is a single email address used as the sender’s email when sending via SMTP mode. The default value is set up in the Email Settings window as the Sender’s Email.

**To Field**
This is the list of email addresses to be used as the To value when sending the email. The To Button is available when a MAPI compliant email client is installed and allows the selection of addresses from an address book. The default value is set up in the Email Settings window as the Administrator Email.

**Cc Field**
This is the list of email addresses to be used as the Cc (Carbon Copy) value when sending the email. The Cc Button is available when a MAPI compliant email client is installed and allows the selection of addresses from an address book.

**Bcc Field**
This is the list of email addresses to be used as the Bcc (Blind Carbon Copy) value when sending the email. The Bcc Button is available when a MAPI compliant email client is installed and allows the selection of addresses from an address book.

**Subject**
This is the Subject line to be used when sending the email. The default value is set up in the Email Settings window as the Default Subject.

**Attachments**
This is a drop-down list containing the paths to the files to be attached when sending the email.

**Add Button**
This button opens a dialog to select a file to be added to the list of attachments.

**Remove Button**
This button removes the currently selected attachment from the list.

**Body**
This is the Body text to be used when sending the email. The default template can be set up in the Email Settings window as the Default Body Text.

**Send Button**
This button will process the email and send it. The transport protocols and other email settings can be set up in the Email Settings window.
Cancel Button
This button will abort the email and close the window.

When the Send Email window is manually opened, it behaves as though Preview and Auto Send options are enabled in the Email Settings window. This is to ensure that the Send Email window is the only user interface seen when manually sending emails.
Chapter 3  System Module Features

Calculator

You can open the Calculator window by selecting Calculator from the Transactions section of the GP Power Tools Area Page or by selecting Calculator from the Options button drop list on the main window.

You can open it directly from the Tools menu underneath the Microsoft Dynamics GP menu (highlighted below). It also has the keyboard shortcut Ctrl+Shift+C assigned to it.

If the Standard Toolbar is displayed, you can launch Calculator from the Calculator button (highlighted below).

You can also use the Calculator option on Quick Links on the Home Page. When running on the Web Client, use the Quick Links on the Home Page to open the Calculator as the other navigation options are not available.

In addition, Calculator 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.

The Calculator is a touch friendly standard calculator built directly into Microsoft Dynamics GP. It supports copying and pasting of values using the clipboard as well as memory functions. It is especially useful on the Web Client where access to a calculator app might not be possible.

Use the highlighted letters for the Memory Store (S), Memory Add (M), Memory Recall (R), Delete (D), Clear (C), All Clear (A) and Negate (N) functions. It was not possible to get the Delete or Backspace keys on the keyboard working.
Dex.ini Settings

You can open the Dex.ini Settings window by selecting Dex.ini Settings from the Transactions section of the GP Power Tools Area Page or by selecting Dex.ini Settings from the Options button drop list on the main window.

The Dex.ini Settings window allows control of some system and GP Power Tools options which are stored in the Dex.ini file. It is divided into four tabbed sections.

For Microsoft Dynamics GP 2013 onwards, all settings in this window are stored in the Global level Dex.ini with the exception of the Enable Debugger Setup Mode and Automatically open GP Power Tools main window after login options which are stored in the User level Dex.ini.

Debug Tab

The Debug tab contains settings related to the use of the logging and debugging features of Microsoft Dynamics GP as well as settings for GP Power Tools itself.
The following settings are available:

**Enable SQL Logging on next login**
This option will update the SQLLogSQLStmt and SQLLogODBCMessages Dex.ini settings to enable logging to the DEXSQL.LOG file on next login.

**Pathname location for SQL Log file**
This option will update the SQLLogPath Dex.ini setting to control the location of the DEXSQL.LOG file. This option controls where the log file is initially created. If using Automatic Trigger Mode or Manual Logging Mode, the log file will be renamed and possibly moved to a different folder.

**Rename DEXSQL.LOG at the beginning of each day**
This option is added by GP Power Tools to stop the DEXSQL.LOG file growing too large. It renames the log each day. It stores the date when it last renamed the file in the SQLLogRename Dex.ini setting in the format YYYYMMDD. This option should not be used with Automatic Trigger Mode.

**Enable Dexterity Debug Menu on next login**
This option will update the ScriptDebugger Dex.ini setting to control whether the Debug Menu will be available on next login.

**Dexterity Debug Menu Product**
This option updates the ScriptDebuggerProduct Dex.ini setting to control the default dictionary Product ID context for the Debug Menu.

**Show Debug Messages on next login**
This option updates the ShowDebugMessages Dex.ini setting to control whether internal debug message dialogs are displayed when the Debug Menu is enabled.

*If the Debug Menu is enabled, it is recommended that the Show Debug Messages option is not enabled for a production system. Having it enabled can cause additional dialogs to be displayed that could be confusing to users.*

**Enable Enhanced Script Log on next login**
This option updates the ScriptLogEnhanced Dex.ini setting to control whether the enhanced Dexterity Script Log features are enabled. Enabling this option adds time stamps and flagging of background processes to the script log. This option is enabled by default by GP Power Tools.

**Enable GP Power Tools Setup Mode**
Enabling this GP Power Tools option will prevent Triggers marked to Start Trigger Automatically on Login from starting. Setup Mode is designed to be used by consultants when setting up GP Power Tools for use at a customer’s site. It uses the MBS_Debug_SetupMode Dex.ini setting.

*GP Power Tools Setup Mode should not be enabled for a production system. It is designed to only be used on test systems or support engineer or partner consultant’s workstations.*
Automatically open Logging Control window after login
This option will make the GP Power Tools Logging Control main window open after a user logs in. It uses the MBS_Debug_AutoOpen Dex.ini setting.

Folder location for logs and export files
This option allows the location for any table exports and captured log files to be specified. It uses the MBS_Debug_Path Dex.ini setting.

Start Logging on next startup only
Enabling this GP Power Tools option will automatically start Manual Logging Mode on application startup. This enables the capture of the logs during the login process. This option will turn itself off after it has been used once. It uses the MBS_Debug_LogOnStartup Dex.ini setting.

SQL Logging
When using logging on next startup, you can specify which logging options to enable. This option enables SQL Logging. It uses the MBS_Debug_LogOnStartup Dex.ini setting.

Dexterity Script
When using logging on next startup, you can specify which logging options to enable. This option enables Dexterity Script Logging. It uses the MBS_Debug_LogOnStartup Dex.ini setting.

Dexterity Profile
When using logging on next startup, you can specify which logging options to enable. This option enables Dexterity Profile Logging. It uses the MBS_Debug_LogOnStartup Dex.ini setting.

You can use the Reset Window Positions Button to clear the Dex.ini settings used for remembering the last window size, position and state for the GP Power Tools windows. Be sure all other GP Power Tools windows are closed when using this option.
STARTUP TAB

The Startup tab contains settings related to the startup of Microsoft Dynamics GP.

The following settings are available:

**Name shown on Application title bar during initial loading**
This option will update the ApplicationName Dex.ini setting to control the name shown by the Dexterity Runtime title bar during application startup. Entering a value into this field will override the default application name of “Dexterity Runtime” while the application is launching. Once the application has launched, the title is updated with the product name as shown in the Dynamics.set launch file.

**Automatically Install Chunk Files without displaying dialog**
This option will update the AutoInstallChunks Dex.ini settings to allow chunk files to install without the user being prompted.

**Suppress Sample Company Date Warning**
This option will update the SAMPLEDATEMSG Dex.ini setting to allow Microsoft Dynamics GP to login to the Fabrikam sample company without displaying the date warning dialog.

**Use SQL Login Compatibility Mode**
This option will update the SQLLoginCompatibilityMode Dex.ini setting to allow Microsoft Dynamics GP to continue attempting to login using backwards compatible password encryption methods.
If you continue to use SQL Login Compatibility Mode, a failed login attempt will register as four attempts at the SQL Server. This can prematurely lock out a user when an incorrect password is entered (when enforce password policy is enabled for the SQL Login and the SQL Native Client is used for the ODBC DSN configuration).

Add Application Details to GPPTools_<User>_<Company> Log
This option will add an entry into the GPPTools_<User>_<Company>.log file each time a user logs into a company. It uses the MBS_Debug_LogAppDetails Dex.ini setting.

Enable selection of Data Server on Login
This option controls whether the Server drop-down list on the Login window is enabled. It uses the EnableServerDropDown Dex.ini setting.

Default last User ID used on login
This option controls whether the last User ID used is defaulted in on the Login window. It uses the DefaultLastUser Dex.ini setting.

Default last Company used on login
This option controls whether the last Company used is defaulted in on the Company Selection window. It uses the DefaultLastCompany Dex.ini setting.

Update last User ID and Company on exit
This option controls whether the last User ID and Company used are stored in the Dex.ini when exiting. This is useful when running multiple instances of Microsoft Dynamics GP, the last closed instance will record its settings rather than the last logged in settings. It uses the MBS_Debug_UpdateLastUserOnExit Dex.ini setting.

Disable Ribbons for workstation on next login
This option can disable Ribbons on the desktop client for the current workstation. It uses the EnableWCRibbons Dex.ini setting.

Open Application Maximized on next login
This option controls whether the application opens full screen for the current workstation. It uses the WindowMax Dex.ini setting.

Application Window Position
These options control the default application window position when not maximized for the current workstation. It uses the WindowPosX and WindowPosY Dex.ini settings.

Application Window Size
These options control the default application window size when not maximized for the current workstation. It uses the WindowWidth and WindowHeight Dex.ini settings.
Reports Tab
The Reports tab contains settings related to the behavior and debugging of the Microsoft Dynamics GP Report Writer.

The following settings are available:

Export Body Section as One Line
This option will update the ExportOneLineBody Dex.ini setting to control how the body section on a text report is printed. This option can be used when creating reports to be exported as tab-delimited or comma-delimited text files.

Number of Lines Per Page when Exporting Reports (inc. PDF)
This option will update the ExportLinesPerPage and ExportPDFLinesPerPage Dex.ini settings to control the number of lines on a report page when the report is exported rather than printed to a file (including PDF files).

Suggested values for this setting are 72 for A4 paper in portrait, 51 for A4 paper in landscape, 68 for US Letter paper in portrait and 52 for US Letter paper in landscape. Some trial and error testing might be required to find the best value.
Activate Debug Logging for the Report Writer
These options will update the DebugRW Dex.ini setting to ask the Report Writer to output a debugging log to the file DebugRW.txt. The actual value written to the Dex.ini is shown in the DebugRW Value field.

Mark All
Use this button to activate all the Report Writer debug logging.

Unmark All
Use this button to turn off Report Writer debug logging.

Activate Debug Font Logging for the Report Writer
This option will update the DebugFonts Dex.ini setting enable logging of font selections made by the Report Writer. The results will be written to a DebugLog.txt file. For more information see Knowledge Base (KB) article 870341:

http://support.microsoft.com/kb/870341

Activate Word Template Processing Engine Logging
This option will update the TPELogging and the KeepTemplateTempFiles Dex.ini settings to log the workings of the Template Processing Engine (TPE). The following files will be created in the %TEMP% folder: the TemplateProcessing*.txt file, the document file and the template file.

Disable Screen Output window position memory
This option can be used to disable the window position memory for the Report Writer Screen Output window. It will update the MBS_Debug_DisableScreenOutputMemory and the MBS_Debug_WinScreenOutput Dex.ini settings.
Other Tab

The Other tab contains other miscellaneous settings for use with Microsoft Dynamics GP.

The following settings are available:

**Suppress Date Change Dialog**
This option will update the SuppressChangeDateDialog Dex.ini setting to prevent the dialog to change the User Date from being displayed at midnight. Using this option will also stop the date from being changed in Microsoft Dynamics GP.

**Show Advanced Macro Menu**
This option will update the ShowAdvancedMacroMenu Dex.ini setting to enable the Advanced Macro Menu from the Tools >> Macro menu.

**Show All Menu Items**
This option will update the ShowAllMenuItems Dex.ini setting to show all menu items, even when the module is not installed, not registered or access has been denied.

**Suppress Sound from Application**
This option will update the SuppressSound Dex.ini setting to suppress all sound from Microsoft Dynamics GP.
Display More Info button on Process Monitor
This option will update the QueueMoreInfo Dex.ini setting to display the More Info button on the Process Monitor window (Microsoft Dynamics GP >> Process Monitor).

Enable Scrollbar width override
This option will update the MaxSWScrollbarSize Dex.ini setting to override the width of scrollbars in Microsoft Dynamics GP. This can be helpful when display DPI settings make the scrollbars too wide, so that they cover up the contents of fields.

Disable closing of the OLE Contain.exe on exit
This setting stops the application from attempting to close the OLE Contain.exe program on exit. It can improve performance when exiting the application. It updates the OLEClose Dex.ini setting.

Windows Bitmap Font Registry Settings
This option will attempt to change the registry to update the font files used for bitmap fonts under Windows 7 and later. These settings are initially created when the operating system is first installed and are not changed when changing the DPI setting for the system. If the fonts in the Microsoft Dynamics GP windows are not being displayed at the correct size, use this option to change the sizes.

Windows Bitmap Scaling Settings
These options will attempt to update the registry to enable the Bitmap Scaling functionality of Windows 8 or later and create a Manifest file to enable Bitmap Scaling for the current instance of the Microsoft Dynamics GP application.

Bitmap Scaling can be used to tell the operating system that an application does not automatically handle high DPI settings (anything greater than 100%). The result is that the application is rendered off screen at 100% and then scaled to the correct percentage on the display. Using a DPI setting on the monitor which is a multiple of 100% (such as 200% or 300%) will produce the clearest images with no blurring of fonts, otherwise expect some fuzziness.

A restart of the operating system is required for these setting to take effect.

On an operating system with User Account Control (UAC) enabled, Registry changes are only allowed if the application has been launched using Run as Administrator. If access to the registry is denied the following warning will be displayed:
Configuration Export/Import

You can open the Configuration Export/Import window by selecting Configuration Export/Import from the Routines section of the GP Power Tools Area Page or by selecting Maintenance >> Configuration Export/Import from the Options button drop list on the main window.

The Configuration Export/Import window can be used to export and import selected GP Power Tools settings.

Below is a description of the individual fields on the window:

Export Button
This button will export the selected settings to the file name selected.

Import Button
This button will import the contents of the selected file name. It will open the Import Settings File window to display the contents of the settings file. You can then select the objects that will be imported from the settings file.
Clear Button
This button will clear any selections and reset the File Name and Transfer User and Company Details with Triggers checkbox.

File Name
This is the file name used for exporting and importing. The file should use the extension .dbg.xml.

Transfer User and Company details
This checkbox selects whether the user and company selection for triggers and products is exported when the trigger or Dictionary Control product is exported.

*If you select a Development Project from the tree, all triggers, scripts and parameter lists assigned to that project will automatically be selected. If you select a trigger, script or parameter list which belongs to a project, that project will be selected, but no other components will be selected. If you do not want to export the project, you can unselect it.*

*If you want to export all components of a Project, use the Project Setup window.*
Administrator Password Setup

You can open the Administrator Password Setup window by selecting Administration Password Setup from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Administrator Password Setup from the Options button drop list on the main window. You will also need to be added the GP POWER TOOLS PASSWORD Security Role to access this window. This is an Advanced Mode feature.

The Administrator Password Setup window can be used to specify a separate password to the System Password to be used to before an Advanced Mode feature window can be opened. This would allow users to have access to GP Power Tools administrator level windows without having to provide them with the System Password. Note that the user will still need the appropriate application level and SQL Server level security access.

The following is a description of the individual fields on the window:

*Use separate password instead of System Password*
This checkbox tells GP Power Tools to use a separate GP Power Tools Administrator Password instead of the System Password when opening Advanced Mode feature windows.

*Don’t ask for users who have access to this window*
This checkbox tells GP Power Tools to not ask for the separate GP Power Tools Administrator Password for users who have application security access to the Administrator Password Setup window, except window opening this window. Add the GP POWER TOOLS PASSWORD Security Role to a user for access to this window.

*Password Fields*
These fields allow the separate GP Power Tools Administrator Password to be changed.
Chapter 3

System Module Features

Challenge 'sa' user with Administrator password on login

This checkbox will force the 'sa' user to have to correctly enter the Administrator password before they can select a company during login. If they fail to enter the password, they will be unable to complete logging in. This feature is designed to prevent IT staff from easily accessing the Microsoft Dynamics GP application.

If you enable the separate GP Power Tools Administrator Password, but don’t actually set a new password, you can disable GP Power Tools asking for a password, without having to remove the System Password.
Logging Settings

You can open the Logging Settings window by selecting Logging Settings from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Logging Settings from the Options button drop list on the main window. This is an Advanced Mode feature.

The Logging Settings window can change the settings used with the GP Power Tools logging features.

The following is a description of the individual fields on the window:

**Administrator Controlled Shared Folder Location for logs and export files.**
You can select a folder in a shared location for all logs and export files to be written to. This setting is automatically rolled out to all workstations. Therefore avoiding the need to visit individual workstations to change the Pathname location for Debugger Setup files, exports and logs setting in the Dex.ini Settings window manually. It will update the MBS_Debug_Path Dex.ini setting on login.

The Administrator Controlled Shared Folder Location Setting is stored in the sysUserDefaults (SY01402) table in the DYNAMICS SQL Database. On login, the setting is checked and the Dex.ini setting on the current workstation are updated if necessary. The pathname can be specified using a UNC path in the format \Server\Share\Folder\. 

**Logging Password**
You can specify an optional password to be requested before Manual Logging Mode can be enabled.
CHAPTER 3 SYSTEM MODULE FEATURES

Enable Individual Logging Modes
You use this option to enable Individual Logging Control. By default this option is disabled which hides the Logging Options button on the GP Power Tools main window.

Capture SQL Log
You can select which of the logging modes to enable, this option enables the SQL Logging when Manual Logging Mode is used.

Capture SQL Profile Trace
You can select which of the logging modes to enable, this option enables the SQL Profile Tracing when Manual Logging Mode is used.

SQL Profile Trace Mode
When using SQL Profile Tracing, you can use this option to select the type of SQL Profile Trace created. You can select between Small, Medium, Large and Performance. The Other mode can be used in conjunction with a customized MBS_SQL_Tracing_API_5 stored procedure in the DYNAMICS database.

Capture SQL Log
You can select which of the logging modes to enable, this option enables the SQL Logging when Manual Logging Mode is used.

Capture Dexterity Script Log
You can select which of the logging modes to enable, this option enables the Dexterity Script Logging when Manual Logging Mode is used.

Capture Dexterity Script Profile
You can select which of the logging modes to enable, this option enables the Dexterity Script Profiling when Manual Logging Mode is used.

Capture Macro Recording
You can select which of the logging modes to enable, this option enables the Macro Recording when Manual Logging Mode is used.

Macro Recording can only work when a single instance of Microsoft Dynamics GP is running on a workstation, or if multiple instances are running, Macro Recording will only work on the first instance launched.

When Manual Logging is stopped
You can select whether you want to email a zipped archive file of the logs captured by Manual Logging Mode when the logging is stopped. You can select to email logs automatically, or to ask before emailing logs.
Maximum email attachment file size for zipped log files
Use this option to select the maximum size allowed when emailing the zipped archive file containing the log files captured by Manual Logging Mode.

The maximum email attachment file size would be limited by the maximum attachment size allowed by the email services being used. Please contact the administrator of the email system to check what the maximum size allowed is.

Rename log each day
Select this setting to create a new GP Power Tools Log file for each user and company each day. This avoids the issue where the single un-dated file can get too large over time.

Number of days to keep logs
Use this option to control how many days’ worth of GP Power Tools logs are kept before they are automatically removed. This avoids the logging folder from getting filled up with too many files.

Edit SQL Profile Trace Settings
This button will open the SQL Profile Trace Settings window (see section below).
SQL Profile Trace Settings
The SQL Profile Trace Settings window contains all the options to enable SQL Profile Tracing and create the database objects needed.

The SQL Profile Tracing functionality of GP Power Tools creates a series of SQL Stored Procedures in the DYNAMICS system database:

- MBS_SQL_Tracing_API
- MBS_SQL_Tracing_API_1 (Small)
- MBS_SQL_Tracing_API_2 (Medium)
- MBS_SQL_Tracing_API_3 (Large)
- MBS_SQL_Tracing_API_4 (Performance)
- MBS_SQL_Tracing_API_5 (Other)
- MBS_SQL_Tracing_Read
- MBS_SQL_Tracing_Version
The following is a description of the individual fields on the window:

**Single User Authentication Mode**

Single User Authentication Mode uses a single Windows user to create the SQL Profile Traces. This is the preferred Authentication Mode as it does not require individual users to have their privileges elevated.

If the Authentication Mode is already enabled and you change the setting, you will receive a dialog to process the necessary changes at the SQL Server. It is recommended that you allow the system to make the changes.

If the SQL Profile Trace SQL Components are already created and you change the setting, you will receive a dialog to recreate them. It is recommended that you allow the system to make the changes.

**Multi User Authentication Mode**

Multi User Authentication Mode uses the individual Dynamics GP users to create the SQL Profile Traces and only uses the Windows user as a proxy for the xp_cmdshell command. Using this mode will elevate individual users’ rights to allow them to create traces.

If the Authentication Mode is already enabled and you change the setting, you will receive a dialog to process the necessary changes at the SQL Server. It is recommended that you allow the system to make the changes.
If the SQL Profile Trace SQL Components are already created and you change the setting, you will receive a dialog to recreate them. It is recommended that you allow the system to make the changes.

**Windows Administrator User ID**
Depending on Authentication Mode, this Windows User ID is used to create traces and/or as a proxy for the `xp_cmdshell` command.

The user can be either a local user on the SQL Server machine or a domain user. The user must be added to the local administrator group on the SQL Server machine. It is recommended that the password for the user is set to not expire.

The user does not need to be manually added to SQL Server, GP Power Tools will perform that step.

When you enter the User ID, you will receive a dialog asking to process the steps to enable the Authentication mode. It is recommended that you allow the system to make the changes.
If the Authentication Mode is already enabled and you change the User ID, you will receive a dialog to process the necessary changes at the SQL Server. It is recommended that you allow the system to make the changes.

If you remove the User ID, you will receive a dialog asking to process the steps to disable the Authentication mode. It is recommended that you allow the system to make the changes.

If the SQL Profile Trace SQL Components are already created and you change the User ID, you will receive a dialog to recreate them. It is recommended that you allow the system to make the changes.
**Process Single User Mode SQL Server Action**

There are seven setting changes required on SQL Server to allow Single User Authentication Mode to work. This button allows the steps to enable and disable the settings to be run individually or as one action.

As the system already prompts for these actions to be executed automatically, you would not normally need to manually run the actions using this button.

*If a new user is added to Microsoft Dynamics GP, you will need to run the Grant IMPERSONATE permission to all users option again to allow the new user to be able to create traces.*

The Enable xp_cmdshell proxy account with User ID option will ask for the password for the Windows Administrator User ID.

![Screenshot](image)

The password is not validated at this time. If it is not entered correctly, it will prevent the SQL Profile Trace File being copied to the Debugger Settings folder when the trace is stopped. The error will show in the GPPTools_<User>_<Company>.log file.

As each step is processed a Desktop Alert is displayed to show that the action completed.

**Process Multi User Mode SQL Server Action**

There are four setting changes required on SQL Server to allow Multi User Authentication Mode to work. This button allows the steps to enable and disable the settings to be run individually or as one action.

As the system already prompts for these actions to be executed automatically, you would not normally need to manually run the actions using this button.

*If a new user is added to Microsoft Dynamics GP, you will need to run the Grant IMPERSONATE permission to all users option again to allow the new user to be able to create traces.*

The Enable xp_cmdshell proxy account with User ID option will ask for the password for the Windows Administrator User ID.
The password is not validated at this time. If it is not entered correctly, it will prevent the SQL Profile Trace File being copied to the Debugger Settings folder when the trace is stopped. The error will show in the GPPTools_<User>_<Company>.log file.

As each step is processed a Desktop Alert is displayed to show that the action completed.

**Maximum Trace file size**
Use this setting to control the maximum size a SQL Profile Trace file can get to before a new file is created. The default value for this field is 25 MB.

If you set the field back to zero, it will restore the default values for Maximum Trace file size and Maximum number of Trace files.

If the SQL Profile Trace SQL Components are already created and you change this setting, you will receive a dialog to recreate them. It is recommended that you allow the system to make the changes.

**Maximum number of Trace files**
Use this setting to control the number of trace files created by the SQL Profile Trace. As the trace file reaches the Maximum Trace file size a new trace file will be created with a numbered suffix added to the filename. This setting controls how many of the individual trace files are kept and will delete the oldest trace files as new ones are created. The default value for this field is 10.

If you set the field back to zero, the trace will only create a single file which will grow in size until the trace is stopped.
If the SQL Profile Trace SQL Components are already created and you change this setting, you will receive a dialog to recreate them. It is recommended that you allow the system to make the changes.

**Folder on local drive on SQL Server**

This is the path to a folder that is local to SQL Server that is to be used as a temporary location for SQL Profile Trace files while they are being created.

The folder must use a path that is valid as seen from the SQL Server machine. All Microsoft Dynamics GP Users as well as the Windows Administrator User ID must have Full Control rights to this folder.

When you enter the path, you will receive a dialog asking to create the SQL Profile Trace SQL Components (stored procedures). It is recommended that you allow the system to make the changes.

If the SQL Profile Trace SQL Components are already created and you change the path, you will receive a dialog to recreate them. It is recommended that you allow the system to make the changes.
UNC Network shared path to above Folder
This is the path to the local folder on the SQL Server from the previous field once it has been shared on the network.

⚠️ The folder must be shared so that all Microsoft Dynamics GP Users as well as the Windows Administrator User ID have Full Control rights to this folder.

This path is used after the SQL Profile Trace is created to copy the trace files from the temporary location on the SQL Server to the Debugger Settings location.

Copy SQL Profile Trace files to Logs and Export files location
This checkbox can be used to control where the SQL Profile Trace files are copied from the temporary location on the SQL Server to the Logs and Export files location.

It is recommended that this setting is enabled.

Create SQL Profile Trace SQL Components
This button can be used to manually create the SQL Profile Trace SQL Components (stored procedures) on the SQL Server.

Remove SQL Profile Trace SQL Components
This button can be used to manually remove the SQL Profile Trace SQL Components (stored procedures) on the SQL Server.
Email Settings

You can open the Email Settings window by selecting Email Settings from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Email Settings from the Options button drop list on the main window. This is an Advanced Mode feature.

The Email Settings window allows you define default values and server transport protocols and security settings to be used when sending emails from GP Power Tools.

The following is a description of the individual fields on the window:

Administrator Email
This field can be used to specify the default To email address(es) when sending emails.

Email addresses can be in the following formats and multiple addresses should be separated by a semi-colon (;):

• name@domain.com
• Full Name<name@domain.com>
• Full Name (when in Microsoft Outlook mode only)

Default Subject
This field can be used to specify the default Subject line for the Send Email window.
Default Body Text for Send Email window
This button can be used to specify the default Body Text line for the Send Email window. This can be used to create a standard template for reporting issues to the system administrator.

Click on the Edit Body Text Button to open the Edit Body Text window.

Standard Signature to add to all emails
This button can be used to create a standard signature to add to the bottom of all emails sent from GP Power Tools. If no signature is defined, the text in the screenshot below will be used.

Click on the Edit Signature Button to open the Edit Signature window.

Email Mode
This field can be used to select whether the email engine is using a Microsoft Outlook Client (default setting) or a SMTP Server via CDO (Collaboration Data Objects) to send emails. Using SMTP instead of Outlook is useful for a Terminal Server environment where it is unlikely that an Outlook client is installed and set up on the Terminal Server.
For version 11.0 (for Microsoft Dynamics GP 2010) and later, you can also select to use any MAPI Compliant Client for sending emails. This will work for email clients other than Outlook if they are MAPI compliant. As Outlook is MAPI compliant, this mode also works for Outlook.

For version 12.0 (for Microsoft Dynamics GP 2013) and later, you can also select to use Exchange Web Services for sending emails. This will work directly with the Exchange Server and so does not require an email client to be installed.

When running on the Web Client, the Microsoft Outlook Client email mode is not supported. It is recommended to use the SMTP or Exchange modes which do not require an email client.

**Preview**
This option controls if the Send Email window is displayed whenever an email is sent.

**Auto Send**
This option controls if the email is automatically sent when an email is submitted. If Preview is unchecked, the email is submitted immediately, or if Preview is selected the email is submitted when the Send Button is clicked.

When using SMTP mode, Auto Send is always enabled. When using Outlook mode, this option controls whether the email is shown in the Outlook client before it is sent, without Auto Send the user will need to click the Send button in Outlook.

**Send HTML**
This option controls whether emails generated in GP Power Tools are sent as plain text or as HTML.

**Sender’s Email**
This field must contain a single valid email address for use as the sender’s email address when in SMTP mode. It is recommended to create a new email address for emails sent from Microsoft Dynamics GP.

The email address can be in the following formats:

- name@domain.com
- Full Name<name@domain.com>

**SMTP Server**
This field defines the SMTP Server’s address. It can be specified as a name or as an IP address.

**SMTP Server Port**
This field defines the SMTP Server Port to use, the default value is 25.

**Authentication**
This drop-down list specifies what level of authentication is required to send emails via the SMTP Server. The options are:
You can specify whether Basic or NTLM (Windows NT LAN Manager) Authentication is to be used and whether SSL (Secure Sockets Layer) should be used.

**User ID**
This field contains the user ID to login into the SMTP Server with. This would normally be the user ID associated with the Sender’s Email defined above.

**Password**
This field contains the password to login into the SMTP Server with. This would normally be the password associated with the Sender’s Email defined above.
Configuration Maintenance

You can open the Configuration Maintenance window by selecting Configuration Maintenance from the Routines section of the GP Power Tools Area Page or by selecting Maintenance >> Configuration Maintenance from the Options button drop list on the main window. This is an Advanced Mode feature.

The Configuration Maintenance window can be used to clear the contents of the GP Power Tools settings tables.

The following is a description of the individual fields on the window:

**Clear Button**

This button will clear the contents of the selected tables. You can use the shift and control keys to select multiple tables.

**Redisplay Button**

This button will refresh window and update the record count.

The system will always have a trigger ID named DEFAULT. This trigger will be automatically added when the MBS_TriggerSetup table is cleared.

To reset the Security Activity Tracking data used by the Security Log window, clear the contents of the MBS_SecurityLog table.
Setup Backup and Restore

You can open the Setup Backup and Restore window by selecting Setup Backup and Restore from the Routines section of the GP Power Tools Area Page or by selecting Maintenance >> Setup Backup and Restore from the Options button drop list on the main window. This is an Advanced Mode feature.

The Setup Backup and Restore window can be used to re-import a Debugger.xml file. It can also be used to backup and restore all settings from GP Power Tools.

The following is a description of the individual fields on the window:

Pathname for Debugger.xml file
This list contains the Dex.ini settings to be checked on login. The setting can be specified with an exact value (this is needed to add a new setting).

Backup Button
This Button will back up all settings to a file called Debugger.xml in the folder specified.

Restore Button
This will read the Debugger.xml file from the specified folder and replace all the settings from the imported file.

You can use this window to reimport a Debugger.xml file if the file imported during the upgrade from a previous install was not the correct file. It can also be used to keep a backup of all settings. If you want to export and import individual settings, use the Configuration Export/Import window.

This window exports and imports all settings for GP Power Tools. It completely overwrites the target contents in the Debugger.xml file (for Backup) and the GP Power Tools SQL Tables (for Restore). Use with caution.
Additional System Features

GP Power Tools adds some extra features to help users. Below is a summary of the features:

**Login Remember User**
GP Power Tools fixes an issue where the Remember User feature on the login window does not work when user level Dex.ini files are being used. It makes sure that the RememberUser Dex.ini Setting is stored in the correct location.

**Remember Last Company**
GP Power Tools remembers the last company logged into and selects that company when the Company Selection window is opened. The information is stored in the SQLLastCompany Dex.ini Setting.

**User Preferences Apply**
GP Power Tools fixes the User Preferences window to that is the Apply button is used more than once without closing the window, it now works.

**Find a Window**
GP Power Tools makes the Find a Microsoft Dynamics GP window feature available (just press Ctrl-F). It will search all menu navigation options for the specified text.

**Raise All Windows**
GP Power Tools adds the Raise All Windows menu option available from the application level menu and the Tools menu on all windows.

**Exit After Processes**
GP Power Tools adds the Exit After Processes menu option available from the application level menu.
Transaction being Edited

GP Power Tools adds the User Name to the “Transaction is being edited by another user” dialog. For Sales Order Processing: Sales Transaction Entry window and Purchase Order Processing: Purchase Order Entry window, Purchasing Invoice Entry window and Receiving Transaction Entry window.

![Microsoft Dynamics GP]

This transaction is being edited by another user. [David Murpoye]

OK
Chapter 4: Administrator Tools Features

This chapter includes the following sections:

- Resource Information
- Security Profiler
- Security Information
- Security Log
- Security Analyzer
- Enhanced Security
- Deny Based Security - Security Denied
- Deny Based Security - Security Hidden
- Administrator Settings*
- Dex.ini Configuration*
- Dictionary Control*
- Company Login Filter*
- Window Position Memory*
- User Activity Log*
- Login Limits*
- Launch File Configuration*
- Dynamic Product Selection*
- Additional Administrator Features

* Advanced Mode Feature
Resource Information

You can open the Resource Information window by selecting Resource Information from the Reports section of the GP Power Tools Area Page or by selecting Resources and Security >> Resource Information from the Options button drop list on the main window.

The Resource Information window will display technical, display, and physical names and resource IDs for any form, window, field, table, table group, report or script (procedure or function, global or form level) resource in the any dictionary currently installed in the Microsoft Dynamics GP application.

It can also provide information about non-dictionary resource Security Objects, such as Customization Tools, Document Access, Letters, Microsoft Dynamics GP Import, Navigation Lists, Series Posting Permissions, and SmartList Objects. If the products are installed, the following objects are also supported, SmartList Builder Permissions and Extender Resources. Security objects from other 3rd party products will show as Unknown Objects.

To use this window, enter the information you know into the appropriate field and the rest of the fields will be populated with the details for that resource.
For example, entering a window’s display name will identify the window’s technical name and resource ID, or entering a table’s physical name as it appears in SQL Server will identify the table’s dictionary, technical and display names as well as the resource ID.

This window can be useful when working with table and column names in SQL Server, because it will quickly convert the physical names used in SQL back to the technical names used in Modifier, Report Writer and Dexterity.

For a field on a window on a form, if the form is open, the value of the field will be displayed in the Field Information section.

Below is a description of the individual fields on the window:

**OK Button**
This button will close the Resource Information window.

**Back Button**
This button work backwards through the history of searched resources since the window was opened.

**Search Again Button**
This button will search for the next resource to match the search criteria. Searching again works for Technical, Display and Physical Names for all resource types. The mode of the search can be controlled by the Search Mode drop-down list and the Case Sensitive checkbox.

You can also select from the Search Results list on the right-hand side of the window instead of using the Search Again button to scroll through the list individually.
Clear Button
This button will clear the current search in preparation for a new search.

Open Button
This button will open the selected form or report resource.

Reports opened in this way will not have any options or restrictions applied and might contain unpredictable results. If the report uses a temporary table, this table will contain no data. Opening forms and reports from this window is only for testing purposes.

Security Button
This button will open the Security Information window for the selected resource. See sections below for more information. The Security Button will only be available if the current user has security access to the security windows under Tools >> Setup >> System.

Resource Type
This drop-down list controls whether Resource Information window is searching for Forms, Windows & Fields; Tables & Fields; Reports; Security Objects; or Procedures & Functions.

Search Mode
This drop-down list controls how text searches will be handled by the Resource Information window. The options are Exact Match, Begins with and Contains. The default setting is Exact Match.

Case Sensitive
This checkbox controls if the text searches on the Resource Information window will be case sensitive or not. The default setting is to be case sensitive.

Show currently selected Window and Field information
When this checkbox is selected for the Forms, Windows & Fields Resource Type, the Resource Information window will automatically display the details for the currently selected Form, Window and Field.

The Show currently selected Window and Field information feature only works for windows opened while the Resource Information window is open. So open the Resource Information window before opening the windows you want information about.

Associated Tables Button
This button is available when the Resource Information window is in Forms, Windows & Fields mode. It will display a list of tables associated with the currently selected form.

If a field is selected on the Resource Information window, you will have the option to filter the list of tables to only tables having the specified field. If the field is not available in any tables, this option will be disabled.
This linked table for the window is highlighted with different icon and a flag in the Linked column in the display.

Selecting a table from this window will change the Resource Information window into Table & Field mode and display the details of the selected table. If filtering on a field, the field will also be selected.
Display Keys Button

This button is available when the Resource Information window is in Tables & Fields modes. It will display a list of keys (indexes) for the currently selected table. The fields for the key and the key options are displayed.

Selecting a field from this window will display the details of the selected field.
Tables Containing Field Button
This button is available when the Resource Information window is in Forms, Windows & Fields and Tables & Fields modes. It will display a list of tables which contain the currently selected field. You have the option to select to only include tables which contain data.

Selecting a table from this window will change the Resource Information window into Table & Field mode and display the details of the selected table and field.
Display Parameters Button

This button is available when the Resource Information window is in Procedures & Functions mode. It will attempt to display a list of parameters for the currently selected procedure or function.

This functionality used Visual Studio Tools to read the Dexterity parameters from the Dictionary Assembly DLL files created for each dictionary. Not all procedures and functions are exposed to Visual Studio Tools, so scripts with anonymous or complex datatype parameters might not be found. If the Dictionary Assembly for a product dictionary is not available, the DAG.EXE tool can be used to generate it.

Next to the Control Type field is the Static Values expansion button which displays the Static Values associated with the data type.

Note: Some fields use Static Values populated by code.
When in Form, Window & Field mode, you can use the lookup button to select a form, window or field resource. Once clicked the Resource Explorer window will open.

To insert a form name or window name, select the resource in the right-hand pane and click OK. If no resources are selected on the right-hand pane, the currently selected resource in the left-hand pane will be used when OK is clicked. Clicking on the resource name in the status field at the bottom of the window toggles Dexterity and .Net view.

You can also use the menu lookup button to select a form, window or field resource based on the menu navigation model. Once clicked the Menu Explorer window will open.
The Menu Explorer window has the option to navigate via application menus (top of left pane) or by the Area Pages (bottom of left pane).

To insert a form name or window name, select the resource in the right-hand pane and click OK. If no resources are selected on the right-hand pane, the currently selected resource in the left-hand pane will be used when OK is clicked.

When in Table & Field mode, you can use the lookup button to select a table or field resource. Once clicked the Table Explorer window will open.

To insert a table name, select the resource in the right-hand pane and click OK. If no resources are selected on the right-hand pane, the currently selected resource in the left-hand pane will be used when OK is clicked. Clicking on the resource name in the status field at the bottom of the window toggles Dexterity and .Net view.
The Table Explorer window can also display the key (index) information for a table. Expand the table node in the tree to display the keys; selecting an individual key will display the key fields and the key options.

The Table Explorer window has the option to navigate to tables with or without table groups. Use the Table Groups checkbox to change views.

When in Report mode, you can use the lookup button to select a report resource. Once clicked the Report Explorer window will open.

To insert a report name, select the resource in the right-hand pane and click OK. Custom Reports are shown with a different icon in the right-hand pane.

When in Security Object mode, you can use the lookup button to select a security object. Once clicked the Security Object Explorer window will open.
To insert a security object, select the desired security object in the right-hand pane and click OK. Security objects from other 3rd party products will show as Unknown Objects.

When in Procedure & Function mode, you can use the lookup button to select a script resource. Once clicked the Script Explorer window will open.

To insert a script name, select the resource in the right-hand pane and click OK. Procedures and Functions are shown with different icons in the right-hand pane. Clicking on the script name in the status field at the bottom of the window toggles Dexterity and .Net view.

For version 14.0 onwards: You can select to only show Service Enabled Procedures, which shows a simplified tree structure in the left-hand pane.
Below is a description of the individual fields on the Explorer windows:

**OK Button**
This button will return the selected resource and close the window.

**Cancel Button**
This button will close the window without making a selection.

**Back Up Button**
This button will change the current selection to its parent on the tree.

**Export Button**
This button will allow the resources displayed in the list view to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window.

**Export Mode**
Use this drop-down list to select the format for the exported file. The file can be exported as Tab Delimited, Comma Delimited or as a HTML Table.

**Hidden Forms**
Use this check box to show forms which are normally hidden from the security system.

**Missing Resources**
Use this check box to show menu items which point to external or missing resources.

**Expanded Fields**
Use this check box to expand composite and array fields into the component parts.

**Only show Service Enabled Procedures**
Use this check box to limit the Script Explorer to only show Service Enabled Procedures (for version 14.0 or later).

The Resource Explorer windows which have two panes are Splitter enabled which allows the ratio between the left and right-hand panes to be adjusted. When running on the Web Client, the splitter functionality is disabled.
On the Resource Information window, you can click the Open Button or the Technical Name hyperlink to open the current resource. If the resource is a form or report, it will open. If the resource is a table, the standard Table Descriptions window will open.

If you click Window Technical Name hyperlink, the standard Window Descriptions window will open.
If you click Field Technical Name hyperlink, the standard Field Information window will open.

The Resource Information window is Right click enabled. If you right mouse click on any of the fields you can select Open Resource (same as Open Button), Security Info (same as Security Button) or Cancel from the context sensitive menu. The Security Info option will only be available if the current user has security access to the security windows under Tools >> Setup >> System.
Security Profiler

You can open the Security Profiler window by selecting Security Profiler from the Reports section of the GP Power Tools Area Page or by selecting Resources and Security >> Security Profiler from the Options button drop list on the main window.

After it has been opened, the Security Profiler window will monitor all application-level security requests and display the results.

Whenever a form or report is opened, the application-level security is checked to confirm that the current user has access. Security is also checked to find out whether a customized version (modified, alternate or modified alternate) of the form or report is to be used.

When a report is opened, access is checked for all of the tables linked to the report. To be able to print the report, access must be permitted for the report itself and all the tables linked to the report.

The Security Profiler will also track access to non-dictionary resource Security Objects, such as Customization Tools, Document Access, Letters, Microsoft Dynamics GP Import, Navigation Lists, Series Posting Permissions, and SmartList Objects. If the products are installed, the following objects are also supported, SmartList Builder Permissions and Extender Resources. Security objects from other 3rd party products will show as Unknown Objects.

The Security Profiler window displays each of the queries to the application-level security system and displays the results with all the relevant details of the resources involved.
The Security Profiler window can be used to identify which form or report is causing unexpected security privileges or access denied errors. Just open the Security Profiler and then perform the action in Microsoft Dynamics GP that causes the error to appear. The details of the resource causing the error will be displayed.

By default, GP Power Tools will append additional details to the dialog to identify the resource. This functionality can be disabled from the Administrator Settings window, if desired.

The Security Profiler window is monitoring only application-level security. It will not display security issues caused by Windows security or SQL Server security.

Below is a description of the individual fields on the window:

**OK Button**
This button will close the Security Profiler window.

**Export Button**
This button will allow a Security Profiler log to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window. This allows a user to provide all of the details of a security issue to the administrator for their analysis.

**Import Button**
This button can be used to import a previously exported Security Profiler log. This allows an administrator to view a log of security issues provided by a user.
Clear Button
This button can be used to clear the current contents of the Security Profiler window.

Open Button
This button will open the selected form or report resource.

Reports opened in this way will not have any options or restrictions applied and might contain unpredictable results. If the report uses a temporary table, this table will contain no data. Opening forms and reports from this window is only for testing purposes.

Security Button Drop List
This button Drop List has the option to open the Security Information window for the selected resource. See sections below for more information. The Security Button will only be available if the current user has security access to the Security Information window.

If the current user has access to the Security Task Setup window, the option to Start Capture of Resources and Security Objects will be available. This option will offer to clear the Security Profiler if it is not empty:

If the current user has started the capture of Resources and Security Objects, the option to Stop Capture and create/update Security Task will be available. When this option is selected it will open the Create/Update Security Task window.
This window can be used to create a new Security Task or update an existing Security Task with the items listed in the Security Profiler. If the user has access to the Security Role Setup window, the option to create a new Security Role or update an existing Security Role with the Security Task ID will be available.

Use the options to capture Resources and Security Objects and then create or update a Security Task based on the captured items to quickly build Security Tasks for specific activities within Microsoft Dynamics GP.

Print Button
This button will allow a report of the contents of the Security Profiler window to be printed.

The Security Profiler window is Right click enabled. If you right mouse click on an item in the list you can select Resource Info (same as double click), Open Resource (same as Open Button), Security Info (same as Security Button) or Cancel from the context sensitive menu. The Security Info option will only be available if the current user has security access to the security windows under Tools >> Setup >> System.

The Security Profiler window can be configured to open automatically when there is a security issue. This option is controlled from the Administrator Settings window.

The Security Profiler window has an Options Menu which can be used to Refresh Application Navigation. This option can be used by a user to update the application’s navigation menus to reflect changes made to security without having to exit and re-launch the application.
Security Information

You can open the Security Information window by selecting Security Information from the Reports section of the GP Power Tools Area Page or by selecting Resources and Security >> Security Information from the Options button drop list on the main window. Once opened, you can use the drop-down menu on top of the left pane to select a resource. You may select a Form (by Dictionary or by Menu) as well as a Table or Report resource, a Security Object, or a Service Enabled Procedure.

You can also open the Security Information window from the Resource Information window or the Security Profiler window. From these windows use the Security Button or the Security Info option from the local context (right click) menu to show security information for the selected resource.

The Security Information window is designed to display the security settings for the selected resource for a particular user and company combination. Once the information is displayed the administrator can use the Go To Button or double click to open the appropriate security administration window to make changes if necessary.

Below is an example of the Security Information window. It shows the security settings for the user including the security tasks that belong to security roles assigned to that user. Also shown is the alternate/modified form and report ID to show which version of a resource the user has access to. Under the System Level node, all security tasks, security roles and alternate/modified form and report IDs which reference the selected resource are displayed.
The tree in the left-hand pane is used to display the security status for the currently selected user and company for the selected security resource. The first 3 nodes of the tree describe the product dictionary, resource type (and series) and resource by Display and Technical Name.

If a resource is not available on the Web Client or not available to Limited Users, this will be displayed on an information node on the tree. Also a Limited User will be highlighted with a yellow dot on the icon and Limited User in the description.

The next section is the User node which shows if the user has access to the current resource and which Security Tasks and Security Roles provided that access. If the resource is a Form or Report, the Alternate Modified Form and Report ID will be shown to define which version of the resources access is granted to.

The third section is the System node which shows all Security Tasks and Security Roles which reference the current resource and all Alternate Modified Forms and Report IDs that reference the current resource. Security Tasks, Security Roles and Alternate Modified Forms and Report IDs in this view will have a green or red indicator to show whether the current user and company has access.

Expand the System Level node on the left-hand pane to see what tasks are assigned to the currently select resource or operation. Expand the tasks to see what roles can be used to give a user access to the currently selected resource or operation. If the only task and role available is POWERUSER, then the current resource or operation has not been added to any tasks.

Below is a description of the individual fields on the window:

User ID
This is the User ID for which security is being checked.

Company
This is the company for which security is being checked.

Show only Selected
When this checkbox is selected, only users with access will be shown.

Inactive
When this checkbox is selected, inactive users will be shown.

OK Button
This button will close the Security Information window.

Redisplay Button
This button will re-populate the security information tree. Use this button after making security changes to see the new updated security.
Legend Button
This button will open the Security Information Legend window to show the meanings of the different icons used.

Show Resources Button
This button will open the Security Information Resources window.

Resource Info Button
This button will open the Resource Information window.

Security Button
Use this button to access Deny Based Security and select from the Enhanced Security window, the Security Denied window and the Security Hidden window.

Go To Button
This button allows the user to open a system security window.

You can double click on the User ID/Company node to open the User Security Setup window; a Security Task ID to open the Security Task Setup window; a Security Role ID to open the Security Role Setup window; and an Alternate/Modified Forms and Reports ID to open the Alternate/Modified Forms and Reports window.

The right-hand pane on the Security Information window displays a number of different views into the company access and security information. Use the View Access button drop-down list to change view. When changing views, the currently selected object will remain selected if possible. This pane can be used even when no resource is selected before opening the Security Information window.

Below are the views available.
These views will provide a visual representation of the relationships between Security Tasks, Security Roles, Alternate Modified Forms and Report IDs, Users and Companies.

The Security Information window will highlight when security is not activated for the selected company. This can be enabled from Company Setup window (Microsoft Dynamics GP >> Tools >> Setup >> Company >> Company).

The Security Information window is Splitter enabled which allows the ratio between the left and right-hand panes to be adjusted. When running on the Web Client, the splitter functionality is disabled.

Security Information SQL Role Views
The Security Information window can also be used to show the SQL Server Roles assigned to users at the SQL Server level as well as for each database. There are three views available to view the data by Users, by Database and by Role.

The SQL Role information is read from the SQL Server the first time one of the three SQL Role views is selected. On a large system, there might be a small delay while the data is read from the SQL Server. To force the data to be read again, close and re-open the Security Information window.

Below is an example screenshot.
The data shown in the three views is restricted to only include Dynamics GP users and databases by default. To show all users and database, select the Show All SQL Users & Databases checkbox.

Once the option has been selected, the view will be refreshed to include the additional data for non-Dynamics GP users and databases.

**Security Information Resources**

When the Show Resources Button is clicked, the Security Information Resources window will open.

This window will display the resources associated with the currently selected User ID/Company combination, Security Role ID, Security Task ID or Alternate/Modified Forms and Report ID in the right-hand pane of the Security Information window. Changing the selection will cause the window to refresh.

You can use the check boxes to decide which resource types (Forms, Reports, Tables and Other) to include in the displayed resources. These selections can be changed while the window is populating.

The resources displayed are those for which the selected User ID/Company combination, Security Role ID, Security Task ID or Alternate/Modified Forms and Report ID has access to.
If the selected node in the right-hand pane of the Security Information window has a User ID and/or Company ID parent node, the system will be able to identify which Alternate/Modified Forms and Report ID to apply and so will display when an alternate and/or modified version of the resources has been selected.

Below is a description of the individual fields on the window:

**OK Button**
This button will close the Security Information Resources window.

**Redisplay Button**
This button will re-populate the window. Use this button after making security changes to see the new updated security.

**Legend Button**
This button will open the Security Information Legend window.

**Export Button**
This button will allow the resources displayed in the list view to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window.

**Export Mode**
Use this drop-down list to select the format for the exported file. The file can be exported as Tab Delimited, Comma Delimited or as a HTML Table.

**Show Series**
Use this checkbox if you want the series information included in the resource list.
**Display Security Tasks and Roles**

Use this checkbox if you want the Security Tasks and Security Roles displayed in the resource list. When this option is selected, multiple lines will be displayed for resources if there are more than one Security Task or Security Role which provides access to the resource.

**Print Button**

A report of the contents of the resource list can be printed using this button.

When opening the Security Information window, a background process is launched to check if all the dictionary resources and security objects have been added to the syCurrentResources (SY09400) table. If information is found to be missing, GP Power Tools will generate the additional data. GP Power Tools will also add the additional data when the table is cleared using the Clear Data window.

Once all the dictionary resources and security objects have been added to the syCurrentResources (SY09400) table, GP Power Tools will create a SUPERUSER Security Task with access to everything and a matching SUPERUSER Security Role. Using the SUPERUSER Security Role is similar to the POWERUSER Security Role but uses the security model rather than bypassing it. GP Power Tools will keep the SUPERUSER Security Task updated automatically.

The Security Information window has an Options Menu which can be used to Refresh Resource Information Table. This option can be used by a user to clear and then update the syCurrentResources (SY09400) table without having to use the Clear Data window. The SUPERUSER Security Task and SUPERUSER Security Role will also be updated using this option.
The following section covers additional functionality available in the Menu Explorer window.

When the Menu Explorer opened from the Security Information window, you have the option to filter the menus for the current user and company based on their security access.

The Menu Explorer also has the option to display the details for the menu command, by click the expansion button (shown above). This will open the Menu Command Details expansion area at the bottom of the window.
Security Log

You can open the Security Log window by selecting Security Log from the Reports section of the GP Power Tools Area Page or by selecting Resources and Security >> Security Log from the Options button drop list on the main window.

The Security Log window displays the data captured by the Security Activity Tracking option which can be enabled from the Administrator Settings window using the Enable Security Activity Tracking option.

Once the Security Activity Tracking is enabled, all security events (both granted and denied) are tracked. The logging does not track individual events, but instead totals up the number of events so you can see which resources are accessed the most. It also tracks the last three security events for a resource.

Each event is tracked for the user and company, user, company and system wide, and you select how you want to view the data.

You can use the check boxes to decide which resource types (Forms, Reports, Tables and Other) to include in the displayed resources. These selections can be changed while the window is populating.

Below is a description of the individual fields on the window:

**Display Mode**
This drop-down list allows you to select whether you wish to view data for the selected user and company, for a specific user or company or for all users and companies.
**User ID**
Use this field to select the User ID to display.

**Company**
Use this field to select the Company to display.

**Sort Mode**
This drop-down list can be used to select the order that the Security Log entries are initially displayed in. You can also adjust the sort after the data is displayed by clicking on the column headers.

**Excluded from Security**
This checkbox can be selected if you wish to see the resources which have been accessed which are excluded from the application security system.

**OK Button**
This button will close the Security Log window.

**Redisplay Button**
This button can be used to redisplay the current contents of the Security Log data to the window.

**Open Button**
This button will open the selected form or report resource.

*Reports opened in this way will not have any options or restrictions applied and might contain unpredictable results. If the report uses a temporary table, this table will contain no data. Opening forms and reports from this window is only for testing purposes.*

**Security Button Drop List**
This button Drop List has the option to open the Security Information window for the selected resource. See sections below for more information. The Security Button will only be available if the current user has security access to the Security Information window.

The option to Create/update Security Task from selected rows will open the Create/Update Security Task from Log window.
This window can be used to create a new Security Task or update an existing Security Task with the selected rows listed in the Security Log window. If the user has access to the Security Role Setup window, the option to create a new Security Role or update an existing Security Role with the Security Task ID will be available.

*Use the options to capture Resources and Security Objects and then create or update a Security Task based on the captured items to quickly build Security Tasks for specific activities within Microsoft Dynamics GP.*

**Export Button**
This button will allow the result set displayed in the list view to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window.

**Export Mode**
Use this drop-down list to select the format for the exported file. The file can be exported as Tab Delimited, Comma Delimited or as a HTML Table.

**Details Button**
This button will open the Security Log Details window to display other log entries for the currently select resource. This allows you to easily check which other users and/or companies are using a resource.

**Mark All Button**
This button will mark all lines in the current Security Log view as selected.

**Unmark All Button**
This button will mark all lines in the current Security Log view as unselected.
The Security Log window is Right click enabled. If you right mouse click on an item in the list you can select Resource Info (same as double click), Open Resource (same as Open Button), Security Info (same as Security Button) or Cancel from the context sensitive menu. The Security Info option will only be available if the current user has security access to the security windows under Tools >> Setup >> System.

To clear the data in the Security Log table to start capturing data again, use the Configuration Maintenance window to clear the data in the MBS_SecurityLog table.
Security Analyzer

The Security Analyzer is a tool for administrators to analyze the security settings of their Microsoft Dynamics GP system. It is designed to highlight potential security risks, provide information on unused settings as well as provide a comparison between security access and security resources actually used.

The Security Analyzer has over twenty queries which can be displayed in both Summary and Detail formats. The queries are divided into System Level queries and User & Companies queries. For the Users & Companies queries, you can select the User and/or Company to limit the query data to.

Some of the queries in the Security Analyzer window use the Security Log window’s activity data captured by the Security Activity Tracking option which can be enabled from the Administrator Settings window using the Enable Security Activity Tracking option.

Once the data is displayed, it can be exported if desired, or used to drill down to the relevant system windows to be able to make changes to the system and security settings.

You can open the Security Analyzer window by selecting Security Analyzer from the Reports section of the GP Power Tools Area Page or by selecting Resources and Security >> Security Analyzer from the Options button drop list on the main window.

Before the window opens you will be reminded that the Security Analyzer should be used on a workstation that that has the dictionaries for all products installed. This is to ensure that the security data in the system will be valid for all installed products.

When opening the Security Analyzer window, it will check if all the dictionary resources and security objects have been added to the syCurrentResources (SY09400) table. If information is found to be missing, GP Power Tools will generate the additional data. GP Power Tools will also add the additional data when the table is cleared using the Clear Data window.

After confirming all the products are installed and the updating the security resources table has completed, the Security Analyzer window will open.
Below is a description of the individual fields on the window:

**OK Button**
This button will close the Security Analyzer window.

**Redisplay Button**
This button will re-populate the window. Use this button after making changes to users and companies. To refresh the current query, use the Refresh button in the top right corner of the right-hand pane.

**Open Button**
This button will open the selected form or report resource.

**Security Button**
This button will open the Security Information window for the selected resource.

**Export Button**
This button will allow the resources displayed in the list view to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window.

**Export Mode**
Use this drop-down list to select the format for the exported file. The file can be exported as Tab Delimited, Comma Delimited or as a HTML Table.

**Go To Button**
This button allows the user to open a system security window.
Select the query you wish to view using the left-hand tree pane and it will be displayed in the right-hand list pane. You can change the order that Users and Companies are displayed in using the view button above the left-hand pane. You can also swap between Summary and Detail view using the view button above the right-hand pane. Double clicking on the data in a Summary view in the right pane will jump to the Detail view of the same query.

You can double click on the User ID/Company node to open the User Security Setup window; a Security Task ID to open the Security Task Setup window; a Security Role ID to open the Security Role Setup window; and an Alternate/Modified Forms and Reports ID to open the Alternate/Modified Forms and Reports window.

The Security Analyzer window is Right click enabled. If you right mouse click on an item in the list you can select Resource Info (same as double click), Open Resource (same as Open Button), Security Info (same as Security Button) or Cancel from the context sensitive menu. The Security Info option will only be available if the current user has security access to the security windows under Tools >> Setup >> System.

The Security Analyzer window is Splitter enabled which allows the ratio between the left and right-hand panes to be adjusted. When running on the Web Client, the splitter functionality is disabled.

The Security Analyzer window has an Options Menu which can be used to Refresh Resource Information Table. This option can be used by a user to clear and then update the syCurrentResources (SY09400) table without having to use the Clear Data window.
Deny Based Security – Introduction

The security system in Microsoft Dynamics GP controls access to all resources within the application. This includes dictionary resources such as Forms, Reports and Tables as well as other security objects such as Document Access and Posting Permissions.

For Microsoft Dynamics GP prior to Version 10.0, the security model was an optimistic user & company and class-based design. This design meant that a user had access to every resource within the application unless it was specifically denied. The settings for a user could be set at the user & company level or set at a class level and rolled down to users belonging to that class.

From Microsoft Dynamics GP Version 10.0, the security model was changed to a pessimistic task and role-based design. This design meant that a user had no access to any resources unless they were specifically granted to them. Granting access worked by grouping multiple resources or operations needed to perform a function into tasks. Multiple tasks could then be assigned to roles. Finally, a user could be assigned to multiple roles depending on the work they do within each company.

The task and role-based model works really well to grant access to users based on the work they do but does not easily allow for minor differences between users.

For example: If you have two users with the same roles but wish to deny access to a single window from one user, the process to remove access to one resource is cumbersome and difficult to maintain. You would need to duplicate any task that gave access to the window and remove that one window from the now duplicated tasks, then you would need to duplicate any role that linked to any of the original tasks now updated and change them to use the duplicated tasks. Finally, you would need to assign the now duplicated roles to the user.

Based on the above example, you can see that over time, your security data would be filled with duplicated tasks and roles without only minor differences between them and no easy method to compare the differences.

There has to be a better way.... Introducing to Deny Based Security.

Deny Based Security adds an optional additional layer to the Microsoft Dynamics GP security model, which allows individual resources or operations to be denied on a per user & company basis regardless of what has been granted by the task and role model. Once a resource is marked as denied for a user & company combination, access will never be available for that user & company.

Deny Based Security also adds the ability to hide items from the menu navigation when those items cannot be controlled by security. This works for both menu items linked to forms excluded from security and menu items which run scripts rather than opening forms.

The Security Denied functionality, once applied, works whether GP Power Tools is installed or not. The Security Hidden functionality does require the GP Power Tools to remain installed (which is the recommended configuration anyway).
Deny Based Security – Enhanced Security

You can open the Enhanced Security window by selecting Enhanced Security from the Reports section of the GP Power Tools Area Page or by selecting Resources and Security >> Enhanced Security from the Options button drop list on the main window.

The Enhanced Security window allows you view the security resources or operations via the navigation model (Menus or Area Pages) or via the dictionary model.

The status icons on the left-hand pane show the security status for the user & company selected in the right-hand pane. The status icons on the right-hand pane show the security status for the resource or operation selected in the left-hand pane.

*Checkboxes are used for security resources or operations which can be controlled by the Security Denied functionality. Radio Buttons are used for menu commands which can be controlled by the Security Hidden functionality. A checkbox or radio button will show as disabled if the resource is excluded from security or the user belongs to the POWERUSER role.*

Clicking on an enabled status icon will toggle the item as Security Denied or Security Hidden accordingly. Changes are made immediately and do not need the OK Button to be clicked.

Below is a description of the individual fields on the window:

**OK Button**

This button will close the Enhanced Security window.
CHAPTER 4  ADMINISTRATOR TOOLS

Redisplay Button
This button will re-populate the left-hand pane of the window. To refresh the Users and Companies, use the Refresh button in the top right corner of the right-hand pane.

Legend Button
This button will open the Enhanced Security Legend window to show the meanings of the different icons used.

Copy Button
Use this button to copy security settings from the current to user & company to other users in the current company:
Or copy security settings from the current to user & company the current user in other companies:

Use these windows to copy Security Denied and Security Hidden data to other users and companies in the system. For your convenience, these windows can also copy Security Roles and Alternate Modified Forms and Report IDs as well as Field Level Security settings. You have the option to make an exact copy (Reset target before copying) or combine the settings (Add settings to target).

Resource Info Button
This button will open the Resource Information window.

Security Button
Use this button to access the Security Information window or the other Deny Based Security windows and select from the Security Denied window and the Security Hidden window.

The following is a description of the fields on the window:

User ID
This is the User ID for which security is being displayed.

Company
This is the company for which security is being displayed.

Display Excluded and Missing Resources
Selecting this checkbox will show all resources in the left-hand pane, even if the resource is excluded from security or is missing.

Show Table Groups
Unselecting this checkbox will show the table resources under Dictionary Tables without using the table group logical tables.
Display only Selected Users
Selecting this checkbox will only show users in the right-hand pane if they have access to the resource selected in the left-hand pane.

The Enhanced Security window has an Options Menu which can be used to Refresh Application Navigation. This option can be used to update the application’s navigation menus to reflect changes made to security without having to exit and re-launch the application.

The Menus navigation view uses the data in the Menu Master syMenuMstr (SY07110) table to create the view. Some products do not correctly add their menu items to the table and so their menus will not show on the view.

The Options Menu on Enhanced Security window can be used to Scan for missing Menu Entries. This option will open the following window can scan the menu navigation in memory and look for entries which exist in memory but not in the table.

You can then select the entries you would like to add to the table and these will show on the Enhanced Security Menus view next time it is used.
Deny Based Security – Security Denied

You can open the Security Denied window by selecting Security Denied from the Reports section of the GP Power Tools Area Page or by selecting Resources and Security >> Security Denied from the Options button drop list on the main window.

The Security Denied window is used for maintenance, exporting and reporting of the Deny Based Security – Security Denied data. Using this window allows all the Security Denied for users to be easily viewed without having to explore the tree views on the Enhanced Security window.

Changes in the window are made immediately and do not need the OK Button to be clicked.

Below is a description of the individual fields on the window:

**OK Button**
This button will close the Security Denied window.

**Redisplay Button**
This button will re-populate the window.

**Delete Button**
This button will permanently remove the Security Denied records with a marked checkbox.

**Open Button**
Use this button to open the Window or Report currently selected.
Security Button
Use this button to access the Security Information window or the Enhanced Security window.

Export Button
This button will allow the resources displayed in the list view to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window.

Export Mode
Use this drop-down list to select the format for the exported file. The file can be exported as Tab Delimited, Comma Delimited or as a HTML Table.

Print Button
This button will allow a report of the contents of the Security Denied window to be printed.

The following is a description of the fields on the window:

Display Mode
Use this drop-down list in conjunction with the User ID and Company ID fields to control which records are displayed in the Security Denied view.

User ID
This is the User ID for which security is being displayed.

Company
This is the company for which security is being displayed.

Sort Mode
Use this drop-down list to control the order that the records are displayed in the Security Denied view.

Include
Use these resource type checkboxes to filter the records are displayed in the Security Denied view.

Mark All Button
This button will mark all lines in the current Security Denied view as selected.

Unmark All Button
This button will mark all lines in the current Security Denied view as unselected.

The Security Denied window has an Options Menu which can be used to Refresh Application Navigation. This option can be used to update the application’s navigation menus to reflect changes made to security without having to exit and re-launch the application.
Deny Based Security – Security Hidden

You can open the Security Hidden window by selecting Security Hidden from the Reports section of the GP Power Tools Area Page or by selecting Resources and Security >> Security Hidden from the Options button drop list on the main window.

The Security Hidden window is used for maintenance, exporting and reporting of the Deny Based Security – Security Hidden data. Using this window allows all the Security Hidden for users to be easily viewed without having to explore the Menus tree view on the Enhanced Security window.

Changes in the window are made immediately and do not need the OK Button to be clicked.

Below is a description of the individual fields on the window:

**OK Button**
This button will close the Security Hidden window.

**Redisplay Button**
This button will re-populate the window.

**Delete Button**
This button will permanently remove the Security Hidden records with a marked checkbox.

**Open Button**
Use this button to open the command currently selected.
Security Button
Use this button to access the Enhanced Security window.

Export Button
This button will allow the resources displayed in the list view to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window.

Export Mode
Use this drop-down list to select the format for the exported file. The file can be exported as Tab Delimited, Comma Delimited or as a HTML Table.

Print Button
This button will allow a report of the contents of the Security Hidden window to be printed.

The following is a description of the fields on the window:

Display Mode
Use this drop-down list in conjunction with the User ID and Company ID fields to control which records are displayed in the Security Hidden view.

User ID
This is the User ID for which security is being displayed.

Company
This is the company for which security is being displayed.

Sort Mode
Use this drop-down list to control the order that the records are displayed in the Security Hidden view.

Mark All Button
This button will mark all lines in the current Security Hidden view as selected.

Unmark All Button
This button will mark all lines in the current Security Hidden view as unselected.

The Security Hidden window has an Options Menu which can be used to Refresh Application Navigation. This option can be used to update the application’s navigation menus to reflect changes made to security without having to exit and re-launch the application.
Chapter 4

Administrator Tools

Administrator Settings

You can open the Administrator Settings window by selecting Administrator Settings tab from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Administrator Settings from the Options button drop list on the main window. This is an Advanced Mode feature.

The Administrator Settings window can change settings used within GP Power Tools. It is divided into three tabbed sections.

Colors Tab

The Colors tab contains settings for controlling the Company Color Scheme settings. These color themes are designed to prevent the accidental entry of data into an incorrect company.

The following is a description of the individual fields on the window:

Activate Company based Color Schemes

This option can be used to change the background colors for each company to allow an effective visual cue as to the company currently being used.

Once the option is activated, the colors for the Window Toolbar Color, Window Background Color, Window Heading Color, Field Background Color and Scrolling Window Line Color can be selected using the Select Theme button drop-down list which offers 110 preset themes. Any Custom Color Themes loaded or created will also be shown in the Select Theme button drop-down list. If a preset or custom theme is selected, the Theme Group and Theme Name will be displayed.
The colors can also be manually selected with the Select Buttons. The Reset Buttons can be used to restore the default colors. You can also use the checkboxes against each color to select which colors will be affected by the Spinner Controls. These up and down buttons can be used to adjust the individual Red, Green and Blue components of the selected colors or to adjust all three values together to lighten or darken the selected colors.

The current color scheme and the new color scheme are displayed. When the Apply Button or OK Button is clicked, the new color scheme will be applied.

The Color Scheme data is stored at both the Company and System level. This design allows for live company databases to be copied into a test company database while maintaining the correct color schemes.

When running on the Web Client, the Activate Company based Color Schemes option is disabled as it is not supported.

Company Colors Lookup

This lookup button can be used to change the company for which the Color Scheme is being edited. When the button is clicked, the Company Colors Lookup window is opened.

The colors for a company are displayed on the window when the company is highlighted. This allows you to see the colors for each company without having to switch companies and open the Administrator Settings window in each company.

Changing the Color Schemes for other companies updates the settings in the system database only. This system setting is then transferred into the company database the next time the company is used.
Company Colors Users

This button can be used to optionally control for which users the company based Color Schemes are enabled. When the button is clicked, the Company Colors User window is opened.

When the Allow selection of users for Company based Schemes checkbox is selected, you will be able to use the User Selection checkbox and the Mark All and Unmark All buttons to control for which users the Company based Color Schemes are enabled. By default, all users (including new users) are enabled.

For each user, the number of companies they have access to is shown, this information can be used when deciding if a user should have the Company based Color Schemes enabled.

Changes made to the User Selection checkbox are saved immediately and so will still take effect even if the Administrator Settings window is closed without saving.
CHAPTER 4

ADMINISTRATOR TOOLS

Custom Color Themes

Selecting Custom Color Themes from the Select Theme button dropdown list will open the Custom Color Theme Setup window.

This window can be used to maintain Custom Color Themes. You can create new themes based on the colors Administrator Settings window or create new themes.

The Theme Group and Theme Name fields are required and dictate how the themes are displayed. Themes with the same Theme Group will be grouped together.

Theme colors can be edited with same controls as the Administrator Setting window and can be saved and deleted. Exit the Custom Color Theme Setup window using the Cancel button.
Usability Tab
The Usability tab contains settings for improving the usability of Microsoft Dynamics GP.

![Usability Tab](image)

The following is a description of the individual fields on the window:

**Display dialog on login for this company**
This setting can be used to display a dialog after a user has logged into a company. The settings are **No Message on Login** or one of the two choices below:

- **Test Company Message**: “This company is set up for testing only. Do not use this company when processing live data.”
- **Historical Company Message**: “This company is used for storing historical information only. Do not use this company when processing current-year data.”

This feature is providing a user interface to the existing dialog functionality as described in Knowledge Base (KB) article 885542:

http://support.microsoft.com/kb/885542

This feature takes effect on next login or after switching company.
### Change Window Titles in Windows Start Bar

This setting can be used to prefix the window titles as seen in the Windows Start Bar with the User ID or User Name and/or Company ID or Company Name.

This feature is useful when running multiple instances of Microsoft Dynamics GP on a single workstation. It allows users to easily identify which window belongs to each instance of the application by displaying the User ID or User Name and/or Company ID or Company Name in the Windows Start Bar.

This feature takes effect on next login or after switching company.

*When running on the Web Client, the Change Window Titles in Windows Start Bar option is disabled as it is not supported.*

### Prevent application windows from opening outside of the visible desktop area

This setting checks the location of all windows as they open and if they will not be in the visible desktop, their position will be adjusted to make sure they are fully visible.

When this checkbox is selected, you can use the Users Button to open a window to allow selection of the users and/or companies to apply this feature to.

This feature takes effect on next login or after switching company.

*The Window Position Memory window can be used to disable specific windows from this feature if they are hidden windows which are now being displayed when they should remain hidden.*
Add extra width to company name drop-down list on Company Login window

This setting expands the fields on the Company Login window to use the full width of the window to make it easier to read long company names. This feature uses the MBS_Debug_CompanySwitchWidth Dex.ini setting.

This feature takes effect on next login or after switching company.

After Login warn user when password is due to expire

When this setting is enabled, on the first login of the day GP Power Tools will check if the current user’s password will expire and if the number of days is less than the specified warning period, a dialog will be displayed offering the user to change their password.

This feature was added to avoid the situation where a user’s password could expire at the SQL Server level while they are currently logged into Microsoft Dynamics GP.

Number of days prior to password expiry to start warnings

This setting controls the number of days warning for the password expiry warning.

After logging in, if the user’s password is going to expire within the selected number of warning days, the following dialog will be displayed.
Selecting Yes will automatically open the User Preferences Password window to allow the user to change their password.

This feature only works for SQL Server 2008 systems or later.

Only require System or Administrator Password to be entered once per session

When this setting is enabled and you are asked for the System Password or GP Power Tools Administrator Password, a correct answer will be remembered for the rest of the session and you will not be required to enter the password until you log off or change company.
Features Tab
The Features tab contains settings for adjusting the behavior of various features of GP Power Tools.

The following is a description of the individual fields on the window:

Settings Applied Message
This drop-down list allows the selection of how the “Settings Applied” message should be displayed when click the Apply button on the various settings windows in GP Power Tools.

Include Dex.ini Settings File
This checkbox specifies the default setting for ScreenShot.

Include User Dex.ini Settings File
This checkbox specifies the default setting for ScreenShot.

Include Current Launch File
This checkbox specifies the default setting for ScreenShot.

Include info for all databases
This checkbox specifies the default setting for ScreenShot.

Automatic Open Mode
This option controls whether the Security Profiler window should automatically open when a security error occurs. The options are Do not open Automatically, Open on Errors only; and Open on Error & Warnings.
An Error is a situation that will cause the application to generate a dictionary not loaded or permission denied error dialog. A Warning is a situation where no error dialog will be generated, but the resources defined in the settings will not be opened as expected.

**Disable updating Security Privilege warning to include form name**

This option controls whether the Security Privilege warning dialog (screenshot below) includes additional information about the resource for which security access has been denied.

By default, once GP Power Tools is installed, additional information will be included on the dialog (screenshot below). This will help administrators identify the issues even if the Security Profiler window is not in use.

Enabling this option will disable the additional information and revert back to the standard dialog.

**Disable logging of Security Errors and Warnings**

This option disabled the logging of Security Errors and Warnings to the GPPTools_<User>_Company_.log log files.

**Enable Security Activity Tracking**

This option enables the recording of all Security events to track the number of times resources are accessed and whether the access for granted or denied. Use the Security Log window to review the data captured.
Enable Security Activity Tracking when opening Smartlist

This option re-enables the recording of Security events when opening the Smartlist window. By default, capturing this information has been disabled as it can cause performance issues especially when your system has a large number of Smartlist favorites.
Options Tab
The Options tab contains settings for enabling and disabling various features of GP Power Tools.

The following is a description of the individual fields on the window:

**Disable Window Position Memory feature**
When this checkbox is selected, the Window Position Memory feature will be disabled. Takes effect immediately for current workstation and on next login for other workstations.

When this checkbox is selected, you can use the Users Button to open a window to allow selection of the users and/or companies to apply this feature to.

**Disable User Setup Additional Information window automatically opening**
When this checkbox is selected, the User Setup Additional Information window will not open automatically when the User Setup window is opened.

This window is used store additional information against each Microsoft Dynamics GP user. The User Email Address is used by the Database Validation feature to send emails when resetting users’ passwords.

If a default company is selected and the system is currently logged into that company, the lookups can be used to select Employee ID and related information.
The Default Site ID field is used to auto populate the Default Site on the following windows:

- Item Transaction Entry
- Item Transfer Entry
- Item Enquiry
- Sales Transaction Entry
- Invoice Entry

There are also four User Defined String fields and two User Defined Date fields. The prompts for these six fields can be defined using the following option on the Administrator Settings window.

The User Setup Additional Information window is synchronized with the User Setup window so the information is displayed, saved and deleted using the controls on the User Setup window. Clicking the OK Button on the window, just closes the window, but does not save anything until the user is saved.

Even if the User Setup Additional Information window does not open automatically, it can be opened using the Additional Menu on the User Setup window or pressing the Ctrl-I keyboard shortcut.
Change User Setup Additional Information User Defined Field Prompts

Click the Edit Button to open the window to change the prompts for the User Defined String fields and User Defined Date fields used on the User Setup Additional Information window.

<table>
<thead>
<tr>
<th>User Defined String 1 Prompt</th>
<th>User Defined String 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Defined String 2 Prompt</td>
<td>User Defined String 2</td>
</tr>
<tr>
<td>User Defined String 3 Prompt</td>
<td>User Defined String 3</td>
</tr>
<tr>
<td>User Defined String 4 Prompt</td>
<td>User Defined String 4</td>
</tr>
<tr>
<td>User Defined Date 1 Prompt</td>
<td>User Defined Date 1</td>
</tr>
<tr>
<td>User Defined Date 2 Prompt</td>
<td>User Defined Date 2</td>
</tr>
</tbody>
</table>

Enable User Activity Tracking

When this checkbox is selected, the User Activity Log feature will be enabled. Takes effect immediately for current workstation and on next login for other workstations.

The User Activity Log tracks when users’ login and logout and tracks the daily maximum session count on a system, user and company basis.

Days to keep daily Max User Count data for

Use this setting to control for how many days the daily maximum session count data are kept. This data includes a list of the sessions logged into Microsoft Dynamics GP when the maximum count was reached.

Enable Automatic Logout of inactive users

When this checkbox is selected, the Automatic Logout feature will be enabled. This feature takes effect immediately for current workstation and on next login for other workstations.

Automatic Logout mimics the user selecting File >> Exit from the menus and so will perform a well-behaved exit of Microsoft Dynamics GP. It does NOT do anything that will force terminate the application, as this is dangerous and can cause orphaned data and data corruption.

Automatic Logout uses timed background process which executes every minute and will attempt to logout once the user has been inactive for more than the number of minutes specified. You can look at the Process Monitor to see the process and its current status.
Use the After X Minutes field to define how long a user must be inactive before Automatic Logout attempts to log out of Microsoft Dynamics GP.

When this checkbox is selected, you can use the Users Button to open a window to allow selection of the users and/or companies to apply this feature to.

**Enable a second level of Automatic Logout**

When this checkbox is selected, the Automatic Logout feature will check the number of available licenses remaining and once it reaches the specified threshold, Automatic Logout will use a second shorter time before attempting to log out. This allows the feature to get more aggressive with logging out inactive users when the available licenses are low.

Use the When only X% of licenses available field to define at what percentage of available licenses remaining does the second level of Automatic Logout activate.

Use the After X Minutes field to define how long a user must be inactive before the second level of Automatic Logout attempts to log out of Microsoft Dynamics GP.

**Check for SQL activity before logging out inactive users**

When this checkbox is selected, the Automatic Logout feature will check for the last activity at the SQL Server level before logging out. Enabling this option is not recommended as other background timed processes can update the last SQL activity and thus prevent Automatic Logout from working. There is a warning displayed when selecting the option.
CHAPTER 4
ADMINISTRATOR TOOLS

Disable Automatic Logout warning dialog when logging out
When this checkbox is selected, the Automatic Logout feature will disable the Automatic Logout Warning Dialog window which counts down the final minute and gives the user the choice to Exit Now or Cancel.

![Automatic Logout window]

When this checkbox is selected, you can use the Users Button to open a window to allow selection of the users and/or companies to apply this feature to.

If the User Activity Log feature is enabled, statistics are tracked on how many times a user cancels this dialog as well as how many times the Automatic Logout feature triggers and how many of those times resulted in a successful attempt to logout.

Disable Automatic Logout warning dialog taking focus
When this checkbox is selected, the Automatic Logout feature will disable the Automatic Logout Warning Dialog window attempting to take focus and jump to the foreground when it refreshes each second of the countdown. You can use the Test Button to open the dialog on request to test the behavior of the dialog.

Please note that if you click Exit Now on the dialog, even when opened with the Test Button, Automatic Logout will attempt to exit Microsoft Dynamics GP.

Attempt to save changes on open windows when logging out
When this checkbox is selected, the Automatic Logout feature will attempt to close open windows in reverse order to when they were opened. If a window needs saving and has all required fields entered, Automatic Logout will simulate the user selecting the Save Button or OK Button.

This additional “smart” option will increase the chances of a successful log out even if a user leaves their screen with unsaved data while ensuring that their changes are not lost.

Please note that if there are windows which cannot be closed or if a dialog opens, the Automatic Logout attempt will be suspended until the dialogs have been handled by manual intervention.
Dex.ini Configuration

You can open the Dex.ini Configuration window by selecting Dex.ini Configuration from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Dex.ini Configuration from the Options button drop list on the main window. This is an Advanced Mode feature.

The Dex.ini Configuration window can be used to automate changes to Dex.ini settings for all workstations in the system.

The following is a description of the individual fields on the window:

**Settings List**

This list contains the Dex.ini settings to be checked on login. The setting can be specified with an exact value (this is needed to add a new setting), or can be specified using a “contains” or “begins with” search. The search can be applied against the Dex.ini settings listed in the Dex.ini file (i.e. left of the = sign) or against the entire line in the file. Using a search can find and update multiple lines in the Dex.ini file if more than one setting or line meets the Search Mode and Search String criteria.

The Silent checkbox should be checked (default) if the Dex.ini setting should be updated without asking the user.

The Log checkbox should be checked (default) if the Dex.ini setting changes should be recorded in the Debugger log files.

The Value field contains the value to change the Dex.ini setting to. Leaving this field blank will remove the Dex.ini setting from the Dex.ini file.
The Target Dex.ini field allows the selection of whether this setting should be applied to the Global Level Dex.ini file (default), to the User level Dex.ini file, or to Both Dex.ini files.

Before the line in the Setting List is saved, it is checked for possible damaging settings and if they exist an additional confirmation is required. Possible changes to the following Dex.ini settings are detected: Pathname, Initial, Synchronize, Workstation, Workstation2, FileHandler, DatabaseType.

**Do not update any Dex.ini settings automatically**

This field can be used on individual workstations to prevent GP Power Tools from automatically changing any Dex.ini Settings. It will update the MBS_Debug_ConfigurationOverride Dex.ini setting. This can be useful on test or administration workstations which might not want their Dex.ini settings changed.

The Apply Button can be used to save the changes to the setup files without closing the window.

Click the Info button to display a list of Dex.ini settings that can be changed by the current Search Mode and Search String criteria. The Display Dex.ini Settings window will open.
Setting changes specified in the Dex.ini Configuration window are checked against the Dex.ini file when a user logs in. The system looks for settings which differ from the specified values. If the change is marked as Silent, the setting will be automatically updated.

If any changes need to be made where the Silent checkbox is not selected, then the user will be presented with a dialog asking them to confirm which changes should be applied.

If a user opts to deselect a Dex.ini change that setting will be displayed again when the user logs in again or changes company.
You can use the Edit Dex.ini button to open the Dex.ini Settings Inspector window. This window allows viewing and editing of both the Global and User level Dex.ini settings.
Dictionary Control

You can open the Dictionary Control window by selecting Dictionary Control from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Dictionary Control from the Options button drop list on the main window. This is an Advanced Mode feature.

Dictionary Control can be used to troubleshoot issues with third party dictionaries. You can effectively remove dictionaries from the system one-by-one until the issue stops. Then the last dictionary to be removed can be investigated further.

You can use Trigger Status to disable Dexterity triggers for a specific Product in a similar fashion to the Customization Status window in Microsoft Dynamics GP. The added benefit of Dictionary Control is that it can remember the settings and automatically disable the product on the next login.

Dictionary Control can also disable alternate and/or modified windows for third party dictionaries using the Alternate Status option. This does not change any security settings.

The drop-down lists at the bottom of the window can be used to change settings for all dictionaries.
Using Dictionary Control to disable the triggers and alternate windows for a third party dictionary can produce the same effect as removing the dictionary from the Dynamics.set launch file without requiring any backups or manual editing.

If you set a product to one of the “after login” options, you can use the User button in the top right corner of the window to specify which users and companies should have this dictionary control setting in effect after logging in. Once clicked the Disabled After Login for Users window will open.

You can view this window by users, by companies or by user classes and navigate the tree to select the user and company combinations as required.

If all users are selected on the tree, the tree selections will be cleared and the mode will change from Selected Users and Companies to All Users and Companies. If no users are selected on the tree, the mode will change to All Users and Companies.

The Exclude Selected Users and Companies rather than include them option allows you to invert the behavior of the window. This is handy when it is easier to specify the users and companies for whom the disabling should not take effect.

Sometimes issues can occur because of the order of the dictionaries in the system. Different dictionaries adding triggers for the same event in the application can sometimes clash causing unexpected or undesirable results. The order that triggers from different products will execute is affected by the order of the products in the Dynamics.set Launch File. By changing the order of the products, you can change the order of the triggers and avoid the issues.
Under most circumstances having two or more dictionaries triggering from the same event would not cause any problems regardless of the order the triggers are executed in. Sometimes, a trigger from one dictionary can make changes to data which affect the behaviour of a trigger from a second dictionary thus causing the code to fail. It is this type of situation which can often be fixed by re-ordering the dictionaries.

Dictionary Control allows the order of the products to be changed using the Top, Up, Down and Bottom buttons. You can also remove a product with the Delete Button. Any changes to the Dynamics.set launch file will be saved when OK is clicked. You will be requested to restart Microsoft Dynamics GP after the changes have been saved.

A backup of the original file will be saved as Backup X of Dynamics.set, where X will be a number starting at 1.

If using the Dictionary Control window to disable access to modified windows be aware that by displaying the original window, users might have access to fields previously hidden or disabled on the modified version of the window.

If User Account Control (UAC) is preventing write 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 changes.
Dictionary Control now has the ability to disable Visual Basic for Applications (VBA) and Visual Studio Tools (VSTools) on next login.

The following is a description of the additional checkboxes on the window.

**Disable Visual Basic for Applications (VBA) on next login**
This checkbox disables Visual Basic for Applications (VBA) when the application is restarted. This option uses the VBADisable Dex.ini setting.

**Enable Visual Basic for Applications after one login**
This checkbox automatically re-enables Visual Basic for Applications (VBA) for the application after the first restart. This option uses the MBS_Debug_VBADisableReset Dex.ini setting.

**Disable Visual Studio Tools (VST) Addins on next login**
This checkbox disables Visual Studio Tools (VST) Addins when the application is restarted. This option uses the MBS_Debug_VSTDdisable Dex.ini setting.

**Enable Visual Studio Tools Addins after one login**
This checkbox automatically re-enables Visual Studio Tools (VST) Addins for the application after the first restart. This option uses the MBS_Debug_VSTDdisableReset Dex.ini setting.

The Visual Basic for Applications and Visual Studio Tools options are not available if running on the Web Client. The Visual Studio Tools options will be disabled if User Account Control (UAC) is preventing write access to the application folder. This is because the Dynamics.exe.config file must be renamed as part of the process of disabling Visual Studio Tools Addins.

Disabling Visual Studio Tools will disable the ability to execute .Net scripts (Visual C# or Visual Basic.Net) as well as the ability to execute Dexterity sanScript in the context of a Modified dictionary.
If you want to check exactly what is contained in the Dynamics.set launch file and confirm that each line is in the correct position you can click the Info button to open the Show Launch File window.

The Description column in this window describes what information should be on the current line of the file for the file to be valid.

To use Dictionary Control, a user must have security access to the Customization Status window. This window may have access disabled automatically on each login when using Field Level Security and Field Security IDs are active for the current user and company.

Dictionary Control cannot be used to disable alternate windows and forms or triggers in GP Power Tools. As the core Microsoft Dynamics GP dictionary cannot have alternate windows, Dictionary Control cannot be used to disable alternate windows.

When running on the Web Client, Dictionary Control cannot be used to modify the launch file and so the movement and delete buttons are disabled.

Dictionary Control handles checks by third party products of the security tables to ensure they correctly identify which version of a window is currently open. This prevents triggers running on the incorrect version of a window and generating errors.
Company Login Filter

You can open the Company Login Filter window by selecting Company Login Filter from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Company Login Filter from the Options button drop list on the main window. This is an Advanced Mode feature.

The Company Login Filter window is used to set up filtering and re-ordering of the companies available in the Company Login window for a user depending on the instance of the application environment currently being used.

The companies available for a specific instance of the application environment can be controlled by a Dext.ini Setting which selects the active Company Login Filter profile and optionally by path of the Launch File used to start Microsoft Dynamics GP (usually Dynamics.set).

Examples of Use:

- On a multinational system, where different countries or regions have different localization dictionaries (such as VAT or GST), you can ensure that a company database is only used with the matching application client for each company.

- On a system with different customization dictionaries or different custom forms and reports for different companies, you can ensure that the correct companies can only be used with the appropriate application client.
The following is a description of the individual fields on the window:

**Profile ID**
This field contains a unique identifier for each Company Login Filter profile in the system. The lookup button can be clicked to select from existing profile IDs.

Note that the Profile IDs starting with the prefix character of tilde (~) are reserved for use by Microsoft Support.

**Profile Name**
This field contains a description for the Company Login Filter profile.

**Enable current Profile on this workstation**
Selecting this field will set the MBS_Debug_CompanyFilter Dex.ini Setting for the current workstation to the current Profile ID.

If enabling a Profile ID without any companies selected for the Default Path, the following warning will be displayed:

If you get locked out of Microsoft Dynamics GP because Company Login Filter removes access to all companies even when logging in as the ‘sa’ user, edit the Dex.ini file and remove the MBS_Debug_CompanyFilter setting.
Show Disabled Companies
Selecting this field will display disabled companies at the bottom of the drop-down list instead of removing them entirely with the prefix entered into the next field.

Using the Show Disabled Companies option provides a visual indication of the companies a user has security access to but cannot use from the current application instance.

Prefix for Disabled Companies
Use this field to define a prefix to be used when showing disabled companies instead of removing them from the drop-down list.

Auto select if only one Company
Selecting this field will automatically select the company if there is only one company available after the filter has been applied.

Display Company Database
This field is used to select if the Company Database is shown on the drop-down list and whether it is shown as a prefix or suffix to the Company Name.

Company Display Sort Order
This field is used to select the order that the companies are listed in the drop-down list. This includes a Custom Defined Order, which can be set using the right hand Top, Up, Down and Bottom buttons.

The following is a description of the additional buttons on the window:

Duplicate Button
Use this button to duplicate the current profile ID to a new profile ID. This is useful when an existing profile ID is very similar to the new one you want to create.

A new profile ID must be specified in the dialog which opens.

When setting up a Company Login Filter you can create a new profile ID for each application folder than is using a separate Dex.ini file. If you are using a single application folder with a single Dex.ini file with different Launch File names or paths, you can use a single profile ID with multiple paths specified.

When a profile is created, it will always have a default path created named “Default Filter when no Path matched”. This default path will always be at the top of the list of paths.
You may add additional paths which are checked with a case insensitive "contains" comparison against the current Launch File path. The comparisons for the paths are executed in the order the paths are listed on the window. If no paths are matched, the default path will be used.

For each path in the left-hand pane, you can select which companies will be can be accessed in the right-hand pane.

You can add a Path using the Add Button or using the button drop-down list at the top of the left-hand pane.

Company Login Filter allows the order of the paths to be changed using the left hand Top, Up, Down and Bottom buttons.

You can also edit an existing path with the Edit Button or double clicking on it. You can also remove a path with the Delete Button.

Once the setup has been completed, upon login if the MBS_Debug_CompanyFilter Dex.ini Setting has a value and the profile can be found, then the paths are compared and a set of company access settings will be used to restrict access on the Company Login window. If there are no valid companies available, a message to say that will be displayed.

If you create a Company Login Filter configuration that denies you access to all companies and you can no longer log into Microsoft Dynamics GP, delete the MBS_Debug_CompanyFilter Dex.ini Setting to regain access and then adjust your settings as required.

To attempt to avoid this situation the following warning is displayed when saving a profile which does not have any companies enabled for the Default Path:
Window Position Memory

You can open the Window Position Memory window by selecting Window Position Memory from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Window Position Memory from the Options button drop list on the main window. This is an Advanced Mode feature.

The Window Position Memory window is used to specify which windows in the Microsoft Dynamics GP application should remember their position, size and state on a per user basis. Any window (form) in any dictionary can be added to the list and all sub-windows on selected window (form) will be remembered.

You can also disable the Window Position Check functionality enabled in Administrator Settings (Prevent application windows from opening outside of the visible desktop area) on a per window basis for hidden windows that need to remain hidden.

Below is a description of the individual fields on the window:

**OK Button**
This button will save the settings and close the Window Position Memory window.

**Cancel Button**
This button will close the Window Position Memory window without saving any changes made.

**Mark All Button**
This button will select all windows currently showing in the right-hand pane.
Unmark All Button
This button will de-select all windows currently showing in the right-hand pane.

If de-selecting a window that has user data associated with it, you will be warned that this data will be removed when the list of active windows is saved. Click “Don’t ask again” if you wish the warning to be hidden for the rest of the time the window is open.

Default Button
This button will add the default windows back to the list. The default windows include the main transaction and cards windows from all core modules and the SmartList window.
**Reset Button**

This button will open the Reset Window Position Memory Settings window.

This window allows the administrator to reset the already stored window position, size and state data for the selected users. Once reset, the windows will open in the default position, size and will store settings again when it is closed.

Users can reset their own settings without needing the help of an administrator using the Reset Window Memory Settings option from the Additional menu on the User Preferences window.
Users can also reset their own settings from the Dex.ini Settings window using the Reset Window Positions button which resets both GP Power Tools windows and windows controlled by Window Position Memory.

Hidden Forms
Use this checkbox to enable the display of windows normally hidden by the system. This might be required if the window you want to add cannot be found.

Warnings
Use this checkbox to disable or re-enable the warnings when removing windows from Window Position Memory.

When the Microsoft Dynamics GP node or All dictionary node on the left-hand tree is selected, the right-hand list will contain all of the currently active windows. When specific dictionary node is selected, the right-hand list will contain all of the currently active windows in that dictionary. When any other nodes are selected, the right-hand list will contain all windows in the application for the selected dictionary and series with the currently active windows showing as checked.
When Window Position Memory has been enabled for the SmartList window, GP Power Tools takes over control of handling the window from SmartList and makes it behave correctly. There is an issue on the latest versions of Microsoft Dynamics GP when the ribbon is enabled on the desktop client where the window size shrinks each time it is opened. There is also an issue when the SmartList window is closed while maximized. Both of these issues are fixed when GP Power Tools has control.

The Window Position Memory feature can be disabled using the Options Tab of the Administrator Settings window. Changing this option will take effect immediately on the current workstation and on next login for other workstations.

The Window Position Memory window can be used to disable specific windows from this feature if they are hidden windows which are now being displayed when they should remain hidden.
**User Activity Log**

You can open the User Activity Log window by selecting User Activity Log from the Setup section of the GP Power Tools Area Page or by selecting Administration >> User Activity Log from the Options button drop list on the main window. This is an Advanced Mode feature.

The User Activity Log window displays the data captured by the User Activity Tracking option which can be enabled from the Administrator Settings window using the Enable User Activity Tracking option.

Once the User Activity Tracking is enabled, all user login and logout events are tracked. The logging does not track individual events, but instead totals up the number of events so you can see which users are logging in the most. It also tracks the last three login events for a user. Data from the Automatic Logout feature are also tracked, Automatic Logout can be enabled from the Administrator Settings window.

Each event is tracked for the user and company, user, company and system wide, and you select how you want to view the data.

Below is a description of the individual fields on the window:

**Display Mode**

This drop-down list allows you to select whether your wish to view data for the selected user and company, for a specific user or company or for all users and companies.

**User ID**

Use this field to select the User ID to display.
Company
Use this field to select the Company to display.

Sort Mode
This drop-down list can be used to select the order that the Security Log entries are initially displayed in. You can also adjust the sort after the data is displayed by clicking on the column headers.

OK Button
This button will close the User Activity Log window.

Redisplay Button
This button can be used to redisplay the current contents of the User Activity Log data to the window.

Open Button
This button will open the User Setup window for the selected User.

Max. Users Button
This button will open the User Activity Log Maximum Users window.

This window displays a graph of the daily maximum sessions on a system, per user or per company basis. The graph can be viewed for a date range or showing the maximum values first.

The graph is auto scaling and hovering over any bar will show a tooltip with the date and maximum value on that date. Clicking on a bar will display a list of the logged in sessions when the maximum occurred.
Export Button
This button will allow the result set displayed in the list view to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window.

Export Mode
Use this drop-down list to select the format for the exported file. The file can be exported as Tab Delimited, Comma Delimited or as a HTML Table.
Login Limits

You can open the Login Limits window by selecting Login Limits from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Login Limits from the Options button drop list on the main window. This is an Advanced Mode feature.

The Login Limits window is used to set up limits for how many sessions are allowed for a user logging into Microsoft Dynamics GP. While you can create more than one profile, only one profile can be marked as the Active Profile and be used at any one time.

The login limits options can be set on a system wide, per user, per user and company and per company basis.

Examples of Use:

- You can limit users to one session system wide, but then mark the system administrator and company account accounts as exempt from the limits.

- You can mark a company as always accessible for a user, thus allowing them to always access that company as well as one other company. (as per the one session limit specified above).

- You can also set a limit for the maximum number of sessions that can access specific companies, thus preventing too many sessions be used for one company meaning none are available for another company.
The following is a description of the individual fields on the window:

**Profile ID**
This field contains a unique identifier for each Login Limits profile in the system. The lookup button can be clicked to select from existing profile IDs.

Note that the Profile IDs starting with the prefix character of tilde (~) are reserved for use by Microsoft Support.

**Profile Name**
This field contains a description for the Login Limits profile.

**Active Profile**
Selecting this checkbox marks this profile as active so it will be applied on next login.

The following options become enabled depending on the node selected in the tree in left-hand pane of the window. You can select a System Node, User Node, User and Company Node or Company Node.

**Include sessions for all user types instead of just Full user**
Selecting this field will adjust how the currently used sessions will be counted. Normally only Full Users are counted, but you can opt to include Limited or Service Users as well.

**Default maximum sessions per User**
Use this field to define the default system wide maximum number of allowed sessions per User. Leave as 0 for no limit.

**Exempt user from system maximum sessions limit**
Selecting this field will exempt the selected user

**Override maximum sessions per User**
Use this field to define the override user level maximum number of sessions for the selected user. Leave as 0 for no limit.
Always allow access to this Company
Use this checkbox to allow the selected user to always have access to the selected company. This session is not included in the maximum session limits specified at the system or user levels.

Maximum number of sessions for this Company
Use this field to define a maximum sessions limit for the selected company. Leave as 0 for no limit.

To be able to set the Maximum number of sessions for this Company you will need to change the tree to the “by Company” view using the view button drop down list above the tree and then select the desired Company node.

The following is a description of the additional buttons on the window:

Duplicate Button
Use this button to duplicate the current profile ID to a new profile ID. This is useful when an existing profile ID is very similar to the new one you want to create.

A new profile ID must be specified in the dialog which opens.

Once configured, the active profile will be checked at login and the user can be presented with a warning dialog when they click OK if they have exceeded the number of sessions they are allowed:
If the maximum number of sessions for a company has been reached, the following dialog will be displayed:

![Microsoft Dynamics GP Dialog]

When the dialogs are displayed a log entry will also be written to the current GP Power Tools log file.
Launch File Configuration

You can open the Launch File Configuration window by selecting Launch File Configuration from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Launch File Configuration from the Options button drop list on the main window. This is an Advanced Mode feature.

The Launch File Configuration window can be used to automate changes to Dynamics.set launch file for all workstations in the system. It works by defining rules for changes desired. These rules are checked on login and applied (if necessary) after creating a backup.

The following is a description of the individual fields on the window:

**Rule List**
This list contains the rules to be checked on login. The columns show the details for each rule.

To edit a rule, just select the row in the list and make the desired changes. The rule will be automatically updated if all the rule field settings are valid.

To add a rule use the Add Button and select the rule to use from Launch File Rule drop down list. Then enter the desired changes. The rule will be automatically updated if all the rule field settings are valid.
Launch File Configuration allows the order of the rules to be changed using the Top, Up, Down and Bottom buttons. This controls the order in which the rules are applied.

You can also remove a rule with the Delete Button.

**Launch File Rule**

This drop down list is used to select the rule to apply to the launch file. Depending on the selection the Rule Information will be updated, and the Rule Fields will be enabled or disabled. Rules available are:

- Reorder by Product ID
- Reorder by Product Name
- Add New Product
- Remove Existing Product
- Rename Product Name
- Move Above Product
- Move Below Product
- Update Dictionary Paths
- Update Location ID Folders
- Update Dictionary Files
- Add Location ID
- Remove Location ID

**Rule Fields**

These fields will be enabled and disabled depending on the Launch File Rule selected. The rule will be automatically updated if all the rule field settings are valid.

When adding or removing a product, an expansion button will be available which opens the Launch File Configuration Additional Files window. Use this window to list additional files such as DLLs which need to be renamed with a .bak extension when removing a product or have the .bak extension removed when adding a product.
Do not update the Launch File automatically

This field can be used on individual workstations to prevent GP Power Tools from automatically updating the Launch File. It will update the MBS_Debug_LaunchConfigurationOverride Dex.ini setting. This can be useful on test or administration workstations which might not want their Launch File changed.

The Apply Button can be used to save the changes to the setup files without closing the window.

Click the Preview button to view a preview of what changes would be made to the current workstation based on the rules defined. The Launch File Configuration Preview window will open. Changes to the rules are immediately reflected in the preview window.

Launch File Rules specified in the Launch File Configuration window are checked against the current launch file (usually Dynamics.set) when a user logs in. If any changes are required, a backup of the existing launch file will be created, and a new updated launch file will be written. The user will then be notified and asked to restart Microsoft Dynamics GP.
It is recommended to ensure that you do not have any rules which contradict each other as this could create an infinite loop where GP Power Tools keeps updating the Launch File each time Microsoft Dynamics GP is started and thus prevents access.

If you create rules which will leave the Advanced Security product above Smartlist in the Launch File, when saving you will be shown the dialog below which will offer to add a rule to fix this situation. If left unchanged, Smartlist can open a login dialog on startup before the actual Microsoft Dynamics GP login window opens.
Dynamic Product Selection

You can open the Dynamic Product Selection window by selecting Dynamic Product Selection from the Setup section of the GP Power Tools Area Page or by selecting Administration >> Dynamic Product Selection from the Options button drop list on the main window. This is an Advanced Mode feature.

Dynamic Product Selection allows the dynamic selection of different versions of a window or report to be selected as the window or report is opened. This can be used to allow access to more than one version of a window (original, modified, alternate or modified alternate) without requiring security settings to be changed or logging in as another user.

The Dynamic Product Selection window is used to configure which windows or reports have Dynamic Product Selection enabled and which versions of the window or report to offer for selection by the user. The settings are linked to the Microsoft Dynamics GP Alternate/Modified Forms and Report ID as used on the User Security window and can also have additional user and/or company selection.

The following is a description of the individual fields on the window:

Modified/Alternate ID
This field contains a unique identifier for each Modified/Alternate ID in the system. The lookup button can be clicked to select from existing Modified/Alternate IDs.
While it is possible to have as many Modified/Alternate IDs as you wish, only those which match an existing Microsoft Dynamics GP Alternate/Modified Forms and Report ID will be used. To select from the list of existing Microsoft Dynamics GP Alternate/Modified Forms and Report IDs, use the lookup button on the right-hand side of the window.

Note that the Modified/Alternate IDs starting with the prefix character of tilde (~) are reserved for use by Microsoft Support.

Description
This field contains the description from the matched Microsoft Dynamics GP Alternate/Modified Forms and Report ID, if it exists.
Resource Type
This drop down list selects whether to setup dynamic product selection for windows or reports resources.

Resource Tree
Use the left-hand tree pane to select which resource should have Dynamic Product Selection enabled. The Users Button can be used to fine tune the users and/or companies the resource is enabled for beyond that they are using the specified Modified/Alternate ID.

Selection List
Use the right-hand list pane to select which versions of the resource are to be made available by Dynamic Product Selection.

The order of the choices to be changed using the left hand Top, Up, Down and Bottom buttons.

You can also fine tune the selections based on user and/or company using the User Button. Once clicked the Enabled for Users window will open, see the section below for more details.

Dialog mode when selecting product
Controls whether keyboard entry dialog should be used even when there are only two or three choices which can use a button dialog. Using the keyboard entry dialog uses one dialog to display a list of available options and a second dialog to enter the selection desired.

Description of Modified/Alternate Resource
Use this field to enter a description to display to the user rather than the name of the dictionary.

Short Description used for dialog buttons
Use this option to change the labels used on the dialog buttons rather than a simple letter of the alphabet. On the keyboard entry dialog, you can use the number of the selection or type in the short description.

The following is a description of the additional buttons on the window:

Duplicate Button
Use this button to duplicate the current Modified/Alternate ID to a new Modified/Alternate ID. This is useful when an existing Modified/Alternate ID is very similar to the new one you want to create.

A new Modified/Alternate ID must be specified in the dialog which opens.
**Users Button**

Use this button to specify which users and companies should have the current resource enabled. Once clicked the Enabled for Users window will open.

You can view this window by users, by companies or by user classes and navigate the tree to select the user and company combinations as required.

If all users are selected on the tree, the tree selections will be cleared and the mode will change from Selected Users and Companies to All Users and Companies. If no users are selected on the tree, the mode will change to All Users and Companies.

The Exclude Selected Users and Companies rather than include them option allows you to invert the behavior of the window. This is handy when it is easier to specify the users and companies for whom the dynamic product selection should not be enabled.

Once configured, when a user opens a window or report the Alternate/Modified Forms and Reports ID being used for the current user and company matches a Modified/Alternate ID and the resource has Dynamic Product Selection enabled and has more than one selection available, a dialog will be displayed and the user can make a selection of which version of the window or report they wish to open.
If there are three or less options and the button dialog mode is selected, then a button dialog will be displayed.

If there are more than three options or the keyboard dialog mode is selected, then two dialogs will be used, the first dialog with a list of options will be displayed:

Then a second dialog is displayed to allow the keyboard entry of the desired choice.

Note a valid selection must be entered to proceed past this dialog, or you can select cancel to open the default version as controlled by security settings (displayed with the asterisk in the first dialog).

Dynamic Product Selection handles checks by third party products of the security tables to ensure they correctly identify which version of a window is currently open. This prevents triggers running on the incorrect version of a window and generating errors.
Additional Administrator Features

GP Power Tools adds some extra features to help administrators. Below is a summary of the features:

Security Resource Descriptions
When opening GP Power Tools security related windows, the Security Resource Descriptions (SY09400) syCurrentResources table is updated to include resources from any missing dictionaries and for resources types not updated by core code.

SUPERUSER Security Task and Role
When GP Power Tools updates the Security Resource Descriptions table, it also creates and maintains a SUPERUSER Security Role and SUPERUSER Security Task. The SUPERUSER Security Role is designed to be used instead of the POWERUSER Security Role. Its advantage is that it uses the security system but grants access to everything instead of bypassing the security system like POWERUSER.

SUPERUSER Workflow Setup
The Workflow Setup looks for POWERUSER for some features, GP Power Tools allows the same features to be used by a SUPERUSER. This allows Workflow Setup to be used without needing to go back to a POWERUSER.

User Company Access Fix
GP Power Tools fixes the issue which can cause an error when changing User Company access when the settings in SQL Server and Microsoft Dynamics GP do not match

User Setup Additional Information
GP Power Tools adds a window to the User Setup window to allow for the storage of additional data against a user. For more information see the setup options in the Administrator Settings window.
Chapter 5: Developer Tools Features

This chapter includes the following sections:

- Runtime Executor
- SQL Executor
- .Net Executor
- Project Setup*
- Automatic Trigger Mode*
- Runtime Execute Setup*
- SQL Execute Setup*
- .Net Execute Setup*
- Parameter Lists*
- Dynamic Trigger Logging*
- Additional Developer Features

* Advanced Mode Feature
Runtime Executer

You can open the Runtime Executer window by selecting Runtime Executer from the Inquiry section of the GP Power Tools Area Page or by selecting Scripting >> Runtime Executer from the Options button drop list on the main window.

The Runtime Executer window can be used to run any Dexterity sanScript script created with the Runtime Execute Setup window which has been marked as Published to Executer Window.

Scripts can only be executed from this window and cannot be viewed, edited or deleted. This window is designed to be used to expose specific scripts to be used by standard users, without needing to give them access to the Runtime Execute Setup window.

The following is a description of the individual fields on the window:

**Script ID**
This field contains a unique identifier for each Runtime Execute Setup script in the system. The lookup button can be clicked to select from existing published script IDs.

**Execute Button**
Use this button to execute the script.
SQL Executer

You can open the SQL Executer window by selecting SQL Executer from the Inquiry section of the GP Power Tools Area Page or by selecting Scripting >> SQL Executer from the Options button drop list on the main window.

The SQL Executer window can be used to run any Transact SQL statements created with the SQL Execute Setup window which has been marked as Published to Executer Window.

Scripts can only be executed from this window and cannot be viewed, edited or deleted. This window is designed to be used to expose specific scripts to be used by standard users, without needing to give them access to the SQL Execute Setup window.

The following is a description of the individual fields on the window:

**Script ID**

This field contains a unique identifier for each SQL Execute Setup script in the system. The lookup button can be clicked to select from existing published script IDs.

**Execute Button**

Use this button to execute the script.
Export Button
This button will allow the result set displayed in the list view to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window.

Export Mode
Use this drop-down list to select the format for the exported file. The file can be exported as Tab Delimited, Comma Delimited or as a HTML Table.

Gotos Button
Use this button drop-down menu to execute a SQL Goto on the selected rows in the returned data.
.Net Executer

You can open the .Net Executer window by selecting .Net Executer from the Inquiry section of the GP Power Tools Area Page or by selecting Scripting >> .Net Executer from the Options button drop list on the main window.

The .Net Executer window can be used to run any Visual C# or Visual Basic.Net script created with the .Net Execute Setup window which has been marked as Published to Executer Window.

Scripts can only be executed from this window and cannot be viewed, edited or deleted. This window is designed to be used to expose specific scripts to be used by standard users, without needing to give them access to the .Net Execute Setup window.

To be able to execute .Net scripts, the WinthropDC.GpPowerToolsVC.dll and WinthropDC.GpPowerToolsVB.dll Addins must be installed.

The following is a description of the individual fields on the window:

Script ID
This field contains a unique identifier for each .Net Execute Setup script in the system. The lookup button can be clicked to select from existing published script IDs.

Execute Button
Use this button to execute the script.
Project Setup

You can open the Project Setup window by selecting Project Setup from the Cards section of the GP Power Tools Area Page or by selecting Scripting >> Project Setup from the Options button drop list on the main window. This is an Advanced Mode feature.

The Project Setup window can be used to group together multiple triggers, scripts and parameter lists into a single development or customization project which can be easily exported without needing to manually select the resources on the Configuration Export/Import window.

The supported resources are listed below.

- Automatic Trigger Mode triggers
- Runtime Execute Setup scripts
- SQL Execute Setup scripts
- .Net Execute Setup scripts
- Parameter Lists

The following is a description of the individual fields on the window:

**Project ID**
This field contains a unique identifier for each Project Setup project in the system. The lookup button can be clicked to select from existing project IDs.
The Notes Button can be clicked to enter Release Notes. Use the Timestamp Button to add a timestamp to the bottom of the release notes.

Note that the Project IDs starting with the prefix character of tilde (~) are reserved for use by Microsoft Support.

**Project Description**
This field contains a description of the project.

**Current Project**
This checkbox indicates that this project is the current project and will be automatically loaded into the window when it is first opened.
Chapter 5

Developer Tools

Configuration File Path
This is the file name used for exporting. The file should use the extension .dbg.xml. The path is automatically generated based on the Project ID, but can be manually changed, if desired.

Transfer User and Company details
This checkbox selects whether the user and company selection for triggers is exported when the trigger is exported.

Project Component List
This list shows all the components that make up the project. Double click on any item to open that item in the appropriate window.

The following is a description of the additional buttons on the window:

Duplicate Button
Use this button to duplicate the current project ID and create a new project ID. This is useful when an existing project ID is very similar to the new one you want to create.

A new project ID must be specified in the dialog which opens.

Duplicating a project just creates a new Project Setup record and does not duplicate any of the resources linked to the source project.

Export Button
This button will export all the settings linked to the current project to the configuration file path specified.

Import Button
This button will import from the configuration file path specified. You will have an option to Remove Project objects not being imported so you can remove objects that are no longer required when importing a project.
Start Button
This button will start all triggers linked to the current project or all automatic start triggers.

Stop Button
This button will stop all triggers linked to the current project or stop all triggers active in the system.

Add Button
This button will allow you to quickly add new Triggers, Scripts and Parameter Lists to the current project.

Information Button
This button will open a window to list all the triggers, scripts and parameter lists linked to the current project.

Reset Path Button
This button will reset the Configuration File Path field back to its default value for the system. You can then adjust the path as desired.
The following is a description of the Options menu available:

**Save and Continue**
Use this menu option to save the current trigger without clearing the window. Control-S can be used as a shortcut.

*A project that is currently linked to triggers, scripts or parameter lists cannot be deleted. If you attempt to delete a project while it is still in use, the information dialog shown above will also open to show you where the project is used.*
Automatic Trigger Mode

Automatic Trigger Mode uses the logging options and Dexterity triggers to log application and SQL activity up to a specific event and exception condition. GP Power Tools can look for multiple issues.

Introduction

The Automatic Trigger Mode of GP Power Tools came about as a result of a specific support incident. The Dynamics support team was assisting a customer with a situation that produced invalid data in a table, but no cause could be replicated. Looking at the customer’s data it was verified that there was an incorrect value in the table. No one was able to identify when the previously correct value in the table was being changed to the incorrect value. GP Power Tools was used to monitor the table field in question and log the steps which led up to the field changing to the incorrect value. It was able to identify the situation and provide the exact scripts being executed up to the point the exception occurred. This information allowed the code issue to be identified and fixed.

How to Setup

To use Automatic Trigger Mode, you must create a trigger ID for each issue or exception condition being monitored. For each trigger ID, an event must be identified which can be used to look for the exception condition. For example, if the exception condition involves data in a table, the trigger event used could be when the table in question is saved. If the exception condition involves a field on a window, the trigger event could be when the field in question is changed.

After the trigger event is selected, a conditional script is written using Dexterity sanScript to check whether the exception condition has actually occurred. Scripts written for this purpose will require the assistance of an experienced Dexterity developer.

Finally, the actions to perform when the exception condition occurs are defined. The trigger ID can be marked to start automatically. When the Start Trigger Automatically on Login checkbox is selected, it is possible to limit the Trigger ID to only automatically start for specified users and/or companies as well as a specified date range.
Registration
When Automatic Trigger Mode is started either manually from the GP Power Tools main window or automatically on login, GP Power Tools registers Dexterity triggers based on the trigger IDs being activated. Once the triggers are registered all logging options are activated. GP Power Tools then waits for one of the triggers to fire.

When manually activating the Automatic Trigger Mode, you can select whether to activate:

- just the DEFAULT trigger ID only,
- the logging trigger IDs marked to start automatically only,
- all logging trigger IDs in the system (except those marked as disabled),
- the non-logging trigger IDs marked to start automatically only,
- all non-logging trigger IDs in the system (except those marked as disabled), or
- all triggers for the selected project. Once selected the Project Lookup window opens to select a project to start.

Non-logging triggers are triggers that can be registered to perform actions independently of the normal Automatic Trigger Mode triggers. They will not activate Automatic Trigger Mode and will not start the system logging. Non-logging triggers can be used to store system values prior to other triggers or used to prototype possible changes to fix an issue without the creation of a Dexterity chunk-based trigger.
Clicking on the Automatic Trigger Mode hyperlink will open the Trigger Status window which displays the Dexterity triggers are currently registered by GP Power Tools. If the trigger needs to store a previous value for a field, it will also be shown on this window.

You can also open the Trigger Status window by selecting Trigger Status from the Inquiry section of the GP Power Tools Area Page or by selecting Scripting >> Trigger Status from the Options button drop list on the main window.

From the Trigger Status window you can use the Unregister button to unregister single or multiple triggers of either the logging or non-logging type. If all logging triggers are disabled, you will be presented with a dialog providing the options to swap to logging only mode stop logging entirely.

You can also use the Register button to start logging or non-logging trigger. This button functions the same as the Turn On Automatic Debugger Mode button on the previous window.
**Triggering**
When an event being monitored occurs and the Dexterity trigger is initiated or “fired” GP Power Tools looks up the trigger ID and runs the associated script to check if the exception condition has actually happened.

If the issue or exception condition is identified to have occurred by the associated script, GP Power Tools will log the results and save the log files as described in the Manual Logging Mode section. GP Power Tools then restarts the logging and continues to wait for the next trigger to fire.

If the actions to export the table record or the entire table were selected, the following files will be created:

- **Record_<User>_<Company>_<Date>_<Time>.xml**
  
  *This file will contain the exported table record.*

- **Table_<User>_<Company>_<Date>_<Time>.xml**
  
  *This file will contain the exported records for the entire table.*

These export files can be found in the folder where GP Power Tools is storing its data files. The default location is the data subfolder beneath the Microsoft Dynamics GP application folder. The location can be changed from the default path using the Pathname location for Debugger Setup files, exports and logs option on the Dex.ini Settings windows (see section the previous chapter).

If the action to display a message or desktop alert was selected, a dialog and/or alert with the display message specified will be shown.

If the exception condition has not occurred, then GP Power Tools resets and continues to wait for the next trigger to fire.
Log File
All actions by GP Power Tools are logged in the GPPTools_<User>_<Company>.log file. Below is an example log when the conditions failed.

Below is an example log when the conditions were met.
**Setup**

You can open the Trigger Setup window by selecting Trigger Setup from the Cards section of the GP Power Tools Area Page or by selecting Scripting >> Setup Automatic Trigger Mode from the Options button drop list on the main window. This is an Advanced Mode feature.

The Trigger Setup window is used to define the Dexterity triggers that will be used to look for the exception conditions.

The window is divided into a header section and four tabs; the Resource Tab, the Actions Tab, the Script Tab and the Options Tab.

The system will always have a trigger ID named DEFAULT. If this trigger ID is deleted, it will be added back automatically as a blank trigger ID. The use of this trigger ID is optional.
Below is a description of the individual header fields on the window:

**Trigger ID**
This field contains a unique identifier for each trigger in the system. The lookup button can be clicked to select from existing trigger IDs.

The Notes Button can be clicked to enter Release Notes. Use the Timestamp Button to add a timestamp to the bottom of the release notes.

Note that the Trigger IDs starting with the prefix character of tilde (˜) are reserved for use by Microsoft Support.

**Trigger Description**
This field contains a description for the trigger.
**Trigger Type**

This drop-down list specifies the type of trigger being defined. The following objects can be selected:

- Table
- Table restricted to Form
- Procedure
- Function
- Focus Event
- Focus Event with Table
- Warning Dialog
- Timed Event
- Form Level Menu
- Field Context Menu
- Login/Logout Event
- Scheduled Event
- Application Level Menu

**Trigger Event**

This drop-down list specifies the event for the selected object. The following events can be selected depending on the trigger type selected:

- Table
  - Save Record
  - Delete Record
  - Read Record
- Table restricted to Form
  - Save Record
  - Delete Record
  - Read Record
- Procedure
  - Global Level
  - Form Level
- Function
  - Global Level
  - Form Level
- Focus Event
  - Form Pre
  - Form Post
  - Window Pre
  - Window Post
  - Window Activate
  - Scroll Fill
  - Scroll Pre
  - Scroll Change
  - Scroll Post
  - Scroll Insert
  - Scroll Delete
  - Field Pre
  - Field Change
  - Field Post
  - Modal Dialog
  - Context Menu
While GP Power Tools can trigger against global and form level procedures and functions, it is unable to obtain the parameter lists for those procedures and functions.

To use the Application Level Menu Trigger Type, the Visual Studio Integration Toolkit product must be installed.
Trigger Attach

This drop-down list specifies when the code for the Dexterity trigger is run when the selected event for the selected object occurs. The following attach modes can be selected depending on the trigger type selected:

- Table
  - After Table Event
- Table restricted to Form
  - After Table Event
- Procedure
  - Before Original
  - After Original
- Function
  - Before Original
  - After Original
- Focus Event
  - Before Original
  - After Original
- Focus Event with Table
  - Before Original
  - After Original
- Warning Dialog
  - Before Original
  - After Original
- Timed Event
  - After Timed Event
- Form Level Menu
  - After Menu Selected
- Field Context Menu
  - After Menu Selected
- Login/Logout Event
  - After Login Event
  - After Logout Event
- Scheduled Event
  - After Logging In
  - After Time XX:XX
  - After Login on DOW
  - After Login on Day X
- Application Level Menu
  - After Menu Selected

When using table trigger type, it is possible to trigger only after a successful table event. This means this option cannot be used to capture a failed save, delete or read event.
**Disabled**
When this checkbox is marked, the current trigger is disabled and will never be activated.

**Start Trigger Automatically on Login**
When this checkbox is marked, the current trigger will be activated automatically after logging into Microsoft Dynamics GP. Use the Users button to specify the individual user and companies to limit for whom the trigger is used.

**Do not activate Logging Mode**
When this checkbox is marked, the trigger will not start logging and will not activate the Automatic Trigger Mode. It allows a trigger to be registered and used without the overhead of maintaining the log files.

Non-logging triggers can be started automatically on login or started manually from the Automatic Trigger Mode Turn On button. To stop a non-logging trigger, use the Unregister button on the Trigger Status window.

**Minimize Log Entries**
When using a Non-logging trigger, this option can be enabled to prevent the trigger generating entries in the GPPTools_<User>_<Company>.log file unless an error occurs.

**Duplicate Button**
Use this button to duplicate the current trigger ID and create a new trigger ID. This is useful when an existing trigger ID is very similar to the new one you want to create.

A new trigger ID must be specified in the dialog which opens.
**Users Button**

Use this button to specify which users and companies should have the current trigger start automatically. Once clicked the Start Trigger Automatically on Login for Users window will open.

You can view this window by users, by companies or by user classes and navigate the tree to select the user and company combinations as required.

*If all users are selected on the tree, the tree selections will be cleared and the mode will change from Selected Users and Companies to All Users and Companies. If no users are selected on the tree, the mode will change to All Users and Companies.*

The Exclude Selected Users and Companies rather than include them option allows you to invert the behavior of the window. This is handy when it is easier to specify the users and companies for whom the trigger should not be activate.
CHAPTER 5

DEVELOPER TOOLS

Administration Button

Use this button to administer multiple Automatic Trigger Mode Trigger IDs at the same time. Once clicked the Trigger Administration window will open.

When the Trigger Administration window is opened, the current Trigger ID is saved automatically. The Trigger Administration window is modal and must be closed before continuing to use other windows.

The window shows the current status of the Trigger IDs in the system. Triggers can be Enabled or Disabled, have their start mode changed between Manual and Automatic, or be deleted in bulk from this window.

To make changes, select the Trigger IDs (use control and shift keys to multi-select) and use the Change State, Change Start Mode, and Mark To Delete Buttons.

The selected changes will be made when OK is clicked. Clicking Cancel will close the window without applying any pending changes.
Resource Tab

The Resource tab contains the definition of the resource to apply the trigger against.

The following is a description of the individual resource selection fields on the tab. The actual fields available depend on the settings for Trigger Type and Trigger Event fields. The lookup button can be used to open the Resource Explorer or the Table Explorer to select the required resource:

**Product Name**
This drop-down list contains a list of products currently installed on the Microsoft Dynamics GP workstation.

**Modified**
This checkbox can be used to force the trigger register in the context of the modified dictionary. This allows the trigger to reference Modifier added local fields.

![Warning]
To be able to register triggers against modified dictionaries, the WinthropDC.GpPowerToolsVB.dll Addins must be installed.

**Form Name**
This field contains the technical name for the form selected.

**Table Name**
This field contains the technical name for the table selected.
Window Name
This field contains the technical name for the window selected.

Field Name
This field contains the technical name for the field selected.

Procedure Name
This field contains the technical name for the procedure selected.

Function Name
This field contains the technical name for the function selected.

Menu Entry
This field contains the description to be displayed on the Form Menu created by this trigger.

Accelerator Key
This field contains an optional accelerator shortcut key (used with Control) for the menu entry.

The Resource tab changes when the Trigger Type of Application Level Menu is selected so you can enter the one or two sets of information need to specify the menu commands.
Actions Tab
The Actions tab contains the actions to perform when the trigger has fired.

The following is a description of the individual action fields on the tab. These actions will be processed when the trigger fires and the conditional script returns true or if the Perform actions when fired regardless of condition checkbox is selected.

**Perform actions when fired regardless of condition**
Check this checkbox when you want the actions to be processed when the trigger is fired regardless of whether the conditional script returns true.

**Perform actions when fired and condition not met**
Check this checkbox when you want the actions to be processed when the trigger is fired and the conditional script returns false.

**Display Message to screen using system dialog**
Select this checkbox if you want the message displayed to the screen in a system dialog box.

**Display Message to screen using desktop alert**
Select this checkbox if you want the message displayed to the screen in a desktop alert.
**Dialog Message**

This field contains the message which will be logged and displayed if the Display Message checkbox is selected. When the Field Name is specified, the message can contain the %1 placeholder which will be substituted with the field value when the message is displayed.

**Display Message to screen using simple system dialog instead of text box dialog**

Select this checkbox if you want the message displayed to the screen in a simple system dialog instead of a text box dialog.

**Send Email using Administrator Email or Email Address below**

When this checkbox is selected, an email with the log details of the trigger will be sent to the Administrator Email address as setup in the Administrator Settings window, or to the specified Email Address.

**Include zipped log files**

Check this option to include the captured log files in a zipped archive file in the email sent.

**If less than X MB**

Specify the maximum allowed size for the zipped archive file.

**Email Address**

This field can be used to specify an email address to use instead of the default Administrator Email.

**Export Current Table Record to XML**

When this checkbox is selected, the current table buffer contents will be exported as an XML file. This action is only valid for Table triggers.

**Export Entire Table to XML restricted by Where Clause**

When this checkbox is selected, the entire table contents will be exported as an XML file. This action is only valid for Table triggers. A SQL Where Clause can be specified to restrict the records exported.

**Optional Where Clause**

This field can be used with the Export Entire Table option to define a SQL Where Clause to restrict the records exported to XML. This field is only valid for Table triggers.

**Issue Reject Record**

When this checkbox is selected, a reject record command will be issued to prevent the current record being shown in a scrolling window. This action is only valid for Non-logging Focus Event triggers attached to the Scroll Fill Event.

**Open Window Hidden**

When this checkbox is selected, the window will open hidden (off display) and remain so until it is closed. This action is only valid for Non-logging Focus Event triggers attached to the Window Pre Event.

**Issue Reject Script**

When this checkbox is selected, a reject script command will be issued to abort the original code from executing. This action is only valid for Non-logging Focus Event triggers running before the original code.
Keep Focus on Field
When this checkbox is selected, the focus will be kept on the current field. This action is only valid for Non-logging Focus Event triggers running before the original code when the Reject Script option is used.

Restore Field Value
When this checkbox is selected, the original value of the current field will be restored. This action is only valid for Non-logging Focus Event triggers running before the original code when the Reject Script option is used.

Capture Screenshots to default logging folder or email
When this checkbox is selected, the ScreenShot utility will be used to capture screenshots of all open windows and either save them to the logging folder or email them.

Email Screenshots using Administrator Email or Email Address below
When this checkbox is selected, an email with the captured screenshots will be sent to the Administrator Email address as setup in the Administrator Settings window, or to the specified Email Address.

Include Dex.ini Settings File
This checkbox tells the ScreenShot utility whether to include the Global level Dex.ini settings file as an attachment for the email. The default setting for this checkbox can be set up in the Administrator Settings window.

Include User Dex.ini Settings File
This checkbox tells the ScreenShot utility whether to include the User level Dex.ini settings file as an attachment for the email. The default setting for this checkbox can be set up in the Administrator Settings window.

Include Current Launch File
This checkbox tells the ScreenShot utility whether to include the launch file, usually Dynamics.set, as an attachment for the email. The default setting for this checkbox can be set up in the Administrator Settings window.

Include info for all databases
This checkbox tells the ScreenShot utility whether to include information for all databases or just the system database and current company database in the System Status report. Not including information for all databases gives better performance on systems with many companies. The default setting for this checkbox can be set up in the Administrator Settings window.
Script Tab

The Script tab contains the Conditional script to be executed when the trigger fires.

The following is a description of the individual script fields on the tab.

**Script Context**

This drop-down list contains a list of products currently installed on the Microsoft Dynamics GP workstation. It is used to select the dictionary context that the conditional script will be executed in. The script context is usually the same as the dictionary ID, but can be changed if the script needs to be executed in a different dictionary to where the trigger is registered.

**Modified**

This checkbox can be used to force the script to execute in the context of the modified dictionary. This allows the script to reference Modifier added local fields.

To be able to execute scripts against modified dictionaries, the WinthropDC.GpPowerToolsVB.dll Addins must be installed.

**Check Security**

This checkbox can be control when the script is executed. If selected, the script will only run if version of the form opened matches the Script Context product and Modified checkbox.
**Conditional Script**

This text field contains the script to be executed when the trigger fires. The script will be populated with a default script when the trigger type, trigger event and resource information are selected. The script will have the required parameters, including a boolean `OUT_Condition`. The script can be used to check for the exception condition being targeted and then set `OUT_Condition` to true if the condition has occurred. The script is checked for syntax errors when saved.

*Using the Helper Functions (see below), a script created in the Runtime Execute Setup window, the SQL Execute Setup window or the .Net Execute Setup window can be loaded and executed from within a conditional script of a trigger.*

The following is a description of the additional buttons on the tab:

**Help Button**

Use this button (highlighted on screenshot) to open the full Dexterity Help file.

**Default Button**

Use this button to reset the Message and Conditional Script fields to the default settings based on the trigger and resource settings.

**Parameters Button**

Use this button to insert a Parameter Placeholder into the script for the Parameter List selected with the Parameter ID on the Options tab.
**Insert Button**

Use this button to insert a Dexterity sanScript code construct. The available constructs are shown below:

![Insert Button](image)

**Helper Button**

Use this button to open the Insert Helper Function window. The appropriate code for the selected helper function will be inserted into the script.

![Insert Helper Function](image)

Helper functions can be used to read or write a window or table field in any window or table from any loaded dictionary. When setting a window field you can select whether to execute the field’s change script. When setting a table field you can select whether adding a new record is allowed.

The table-based help functions currently support up to four key fields. The individual helper functions are covered in more detail in a later chapter.
Names Button

Use this button to insert a dictionary resource into the script.

Once Forms, Windows & Fields is selected the Resource Explorer window will open.

To insert a form name or window name, select the resource in the right-hand pane and click OK. If no resources are selected on the right-hand pane, the currently selected resource in the left-hand pane will be used when OK is clicked.

Once Tables & Fields is selected the Table Explorer window will open.

To insert a table name, select the resource in the right-hand pane and click OK. If no resources are selected on the right-hand pane, the currently selected resource in the left-hand pane will be used when OK is clicked.
Once Reports is selected the Report Explorer window will open.

To insert a report name, select the resource in the right-hand pane and click OK.

Once Procedures & Functions is selected the Script Explorer window will open.

To insert a procedure or function name, select the resource in the right-hand pane and click OK.

Selecting Procedures & Functions with Parameters will allow for the full call syntax to be inserted into the script with the parameter list (if available).
The following is a description of the Script menu available for the tab:

**Find …**
Use this menu option to open the script editor Find window to search for text. Control-F can be used as a shortcut.

**Find Next**
Use this menu option to find the next occurrence. Control-G can be used as a shortcut.

**Replace …**
Use this menu option to open the script editor Replace window to search and replace text. Control-R can be used as a shortcut.

**Replace and Find Next**
Use this menu option to replace and find the next occurrence. Control-B can be used as a shortcut.

**Goto Line …**
Use this menu option to open the script editor Goto Line window to jump to a specified line. Control-N can be used as a shortcut.

**Save and Continue**
Use this menu option to save the current trigger without clearing the window. Control-S can be used as a shortcut.

**Check Syntax**
Use this menu option to check the syntax of the current script. Any errors will be displayed in a dialog window. Control-K can be used as a shortcut.
Options

Use this menu option to open the Options window to allow the syntax highlighting colors, font style, and size to be changed. Control-O can be used as a shortcut.

Generate Dexterity Pass Through

Use this menu option to generate Dexterity pass through sanScript code from a trigger script including the trigger registration that can be copied and pasted into a Dexterity development dictionary. Control-D can be used as a shortcut.
Names Button Uses Clipboard
Use this menu option to control whether the Names Button returns directly to the script (default) or to the clipboard.

To be able to use the clipboard, the WinthropDC.GpPowerToolsVB.dll Addins must be installed.
Options Tab
The Options tab contains optional settings which change the behavior of the trigger.

The following is a description of the individual script fields on the tab.

**Project ID**
Use this field to add the current trigger to a development project.

**Parameter ID**
For Non Logging Triggers using a Focus Event, Add Form Menu or Add Field Context Menu Type you can specify a Parameter List to be used with the script.

**Start Date**
You can specify a Start Date to restrict the dates that a trigger will automatically start.

**End Date**
You can specify an End Date to restrict the dates that a trigger will automatically start.
If the Start Date and the End Date are the same, the trigger will only be active for a single day. If the End Date is before the Start Date, then the trigger will be inactive during the date range. The status field will show the behavior based on the selected dates. If a Start Date is not specified, the trigger will be active up to the End Date. If an End Date is not specified, the trigger will be active from the Start Date.

**Execution Mode**
For Scheduled Event Trigger Type only: Use this drop-down list to select how often the scheduled event should execute. Select from every time, once per system, once per user, once per company or once per user/company combination.

**Do not run missed event on next login**
For Scheduled Event Trigger Type only: By default, if a scheduled event is missed because Microsoft Dynamics GP was not logged in at the time it was scheduled, it will execute on the next login. Select this checkbox to skip the missed event and just wait for the next scheduled time.

**Error Handling**
For Scheduled Event Trigger Type only: Use this drop-down list to select how error are handled. You can select not to retry, to retry once or up to 5 times when the trigger is incomplete (usually caused by a script error), or retry once or up to 5 times when the trigger completed but the conditional script returned false.

**Number of execution logs to keep**
For Scheduled Event Trigger Type only: The system keeps logs each time a scheduled event is triggered, use this field to specify how much history should be kept. Drilling down on the field will open the Trigger Setup Scheduled Log window:
Capture SQL Log
You can select which of the logging modes to enable, this option enables the SQL Logging when this trigger is active. This option is not valid for Non-logging triggers.

Capture SQL Profile Trace
You can select which of the logging modes to enable, this option enables the SQL Profile Tracing when this trigger is active. This option is not valid for Non-logging triggers.

SQL Profile Tracing is not enabled until it has been setup using the SQL Profile Trace Settings window under the Administrator Settings.

SQL Profile Trace Mode
When using SQL Profile Tracing, you can use this option to select the type of SQL Profile Trace created. You can select between Small, Medium, Large and Performance. The Other mode can be used in conjunction with a customized MBS_SQL_Tracing_API_5 stored procedure in the DYNAMICS database. This option is not valid for Non-logging triggers.

Capture Dexterity Script Log
You can select which of the logging modes to enable, this option enables the Dexterity Script Logging when this trigger is active. This option is not valid for Non-logging triggers.

Capture Dexterity Script Profile
You can select which of the logging modes to enable, this option enables the Dexterity Script Profiling when this trigger is active. This option is not valid for Non-logging triggers.

Capture Macro Recording
You can select which of the logging modes to enable, this option enables the Macro Recording when this trigger is active. This option is not valid for Non-logging triggers.

Macro Recording can only work when a single instance of Microsoft Dynamics GP is running on a workstation, or if multiple instances are running, Macro Recording will only work on the first instance launched.

Only restart selected logs when trigger fires
Using this checkbox, you can control which logging modes are restarted when the trigger fires. By default, all active logging modes are restarted each time a trigger fires. If this checkbox is enabled, only the logging modes selected for this trigger will be restarted when this trigger fires. This option is not valid for Non-logging triggers.

Stop Trigger after Condition met
Using this checkbox, you can specify that a trigger should only be used once per session. When the Trigger fires and the condition is met, the trigger will be stopped until next login or manual restart.

Disable trigger after Condition met
Using this checkbox, you can specify that a trigger should only be used once. When the trigger fires and the condition is met, the trigger will be disabled preventing it from starting until it is re-enabled.
Restriction of Scope

GP Power Tools has a restriction which must be taken into account when using the Automatic Trigger Mode.

When using a table trigger type, GP Power Tools uses a Dexterity database trigger. A Dexterity database trigger is only capable of tracking changes made to the tables using Dexterity commands.

Changes made to tables using tools or applications other than Dexterity will not be picked up by GP Power Tools. This can include table changes made directly by SQL Query Analyzer, pass-through SQL commands, SQL stored procedures, SQL triggers, or updates from eConnect, Integration Manager’s SQL Optimized or Microsoft Dynamics GP eConnect adapters, ADO (ActiveX Data Objects) from VBA (Visual Basic for Applications) or any other external application.
Runtime Execute Setup

You can open the Runtime Execute Setup window by selecting Runtime Execute Setup from the Cards section of the GP Power Tools Area Page or by selecting Scripting >> Runtime Execute Setup from the Options button drop list on the main window. This is an Advanced Mode feature.

The Runtime Execute Setup window can be used to run any Dexterity sanScript code without requiring the Dexterity development environment. Scripts written in this window can be used to manipulate tables using Dexterity commands or to call existing functions and procedures in any dictionary.

Script IDs created in this window can be loaded and executed from an Automatic Trigger Mode trigger, another Runtime Execute Setup script or a .Net Execute Setup script. This allows code re-use in a similar fashion to having multiple procedure calls as well as mixing of languages.
The following is a description of the individual fields on the window:

**Script ID**
This field contains a unique identifier for each Runtime Execute Setup script in the system. The lookup button can be clicked to select from existing script IDs.

Note that the Script IDs starting with the prefix character of tilde (~) are reserved for use by Microsoft Support.

**Script Name**
This field contains a description of the script.
Published to Executer Window
This checkbox indicates if the current script can be accessed from the read only Runtime Executer window.

Select Custom Script Purpose
This drop-down list can be used to specify a custom script purpose for the script. Changing the script purpose will replace the script with the template code needed. Purposes include using the script for Report Writer functions (as described in chapter 7), using the script for Service Enabled Procedures, using the script to register custom SmartList Builder Gotos, using the script for handling SQL Gotos, and using the script for handling URL Drill Backs.

URL Drill Backs allow external applications to all custom scripts in Microsoft Dynamics GP. They are called with a URL in the format below:


Project ID
Use this field to add the current script to a development project.

Parameter ID
Use this field to specify a Parameter List to be used with the script.

Execute Dexterity SanScript code in the context of Product
This drop-down list contains a list of products currently installed on the Microsoft Dynamics GP workstation.

Modified
This checkbox can be used to force the script to execute in the context of the modified dictionary. This allows the script to reference Modifier added local fields.

To be able to execute scripts against modified dictionaries, the WinthropDC.GpPowerToolsVB.dll Addins must be installed.

Script
This text field contains the script to be executed. It cannot have any parameters. The script runs as though it is a global procedure in the context of the dictionary specified in the drop-down list. The script is checked for syntax errors when saved.

Runtime Execute Setup can be used to manipulate data in tables when complex business logic is required. In this situation writing the equivalent code in Transact SQL can be difficult. You could loop through a range of records in table and conditionally make different changes depending on the data in the records. For example, re-formatting phone numbers in the Customer Master table to different formats depending on whether they are domestic, international or mobile/cell numbers.
The following is a description of the additional buttons on the window:

**Help Button**
Use this button (highlighted on screenshot) to open the full Dexterity Help file.

**Parameters Button**
Use this button to insert a Parameter Placeholder into the script for the Parameter List selected with the Parameter ID. See the section under Automatic Trigger Mode for more information.

**Insert Button**
Use this button to insert a Dexterity sanScript code construct. See the section under Automatic Trigger Mode for more information.

**Helper Button**
Use this button to open the Insert Helper Function window and insert a helper function into the script. See the section under Automatic Trigger Mode for more information.

**Names Button**
Use this button to insert a dictionary resource into the script. See the section under Automatic Trigger Mode for more information.

**Execute Button**
Use this button to execute the script in the context of the dictionary specified. Any compile errors will be shown in the status pane below the script. Execution errors will cause an Exception Error Dialog to open.

If a selection of the script is currently highlighted, you can decide to execute the highlighted section or the entire script.

**Duplicate Button**
Use this button to duplicate the current script ID to a new script ID. This is useful when an existing script ID is very similar to the new one you want to create.
A new script ID must be specified in the dialog which opens.

**Users Button**

Use this button to specify which users and companies the script should be published to. Once clicked Publish Script for Users window will open.

You can view this window by users, by companies or by user classes and navigate the tree to select the user and company combinations as required.

*If all users are selected on the tree, the tree selections will be cleared and the mode will change from Selected Users and Companies to All Users and Companies. If no users are selected on the tree, the mode will change to All Users and Companies.*

The Exclude Selected Users and Companies rather than include them option allows you to invert the behavior of the window. This is handy when it is easier to specify the users and companies for whom the script should not be published to.
The following is a description of the Script menu available for the window:

**Find …**
Use this menu option to open the script editor Find window to search for text. Control-F can be used as a shortcut.

![Find Window](image)

**Find Next**
Use this menu option to find the next occurrence. Control-G can be used as a shortcut.

**Replace …**
Use this menu option to open the script editor Replace window to search and replace text. Control-R can be used as a shortcut.

![Replace Window](image)

**Replace and Find Next**
Use this menu option to replace and find the next occurrence. Control-B can be used as a shortcut.

**Goto Line …**
Use this menu option to open the script editor Goto Line window to jump to a specified line. Control-N can be used as a shortcut.

![Goto Window](image)

**Save and Continue**
Use this menu option to save the current script without clearing the window. Control-S can be used as a shortcut.

**Check Syntax**
Use this menu option to check the syntax of the current script. Any errors will be displayed in a dialog window. Control-K can be used as a shortcut.
Options
Use this menu option to open the Options window to allow the syntax highlighting colors, font style, and size to be changed. Control-O can be used as a shortcut.

![Options window]

Execute
Use this menu option to execute the script. Control-E can be used as a shortcut.

Generate Dexterity Pass Through
Use this menu option to generate Dexterity pass through sanScript code from a prototype script that can be copied and pasted into a Dexterity development dictionary. Control-D can be used as a shortcut.

![Generate Dexterity Pass Through window]
Changing the Script Language to Visual C# provides the C# code that can be pasted into a Visual Studio project.

Changing the Script Language to Visual Basic .Net provides the VB code that can be pasted into a Visual Studio project.

Names Button Uses Clipboard
Use this menu option to control whether the Names Button returns directly to the script (default) or to the clipboard.

To be able to use the clipboard, the WinthropDC.GpPowerToolsVB.dll Addins must be installed.
SQL Execute Setup

You can open the SQL Execute Setup window by selecting SQL Execute Setup from the Cards section of the GP Power Tools Area Page or by selecting Scripting >> SQL Execute Setup from the Options button drop list on the main window. This is an Advanced Mode feature.

The SQL Execute Setup window can be used to run any Transact SQL statements without requiring the SQL Administration Tools or MS Query. Commands written in this window can be used to view or manipulate data in any table. This window is similar to the Query Analyzer window that is installed with the SQL Server client tools.

Script IDs created in this window can be loaded and executed from an Automatic Trigger Mode trigger, a Runtime Execute Setup script or a .Net Execute Setup script. This allows a Transact SQL query to be used within Dexterity or .Net code.
The following is a description of the individual fields on the window:

**Script ID**
This field contains a unique identifier for each SQL Execute Setup script in the system. The lookup button can be clicked to select from existing script IDs.

![Sql Execute Lookup](image1)

The Notes Button can be clicked to enter Release Notes. Use the Timestamp Button to add a timestamp to the bottom of the release notes.

![Sql Execute Setup Release Notes](image2)

*Note that the Script IDs starting with the prefix character of tilde (~) are reserved for use by Microsoft Support.*

**Script Name**
This field contains a description of the script.
Published to Executer Window
This checkbox indicates if the current script can be accessed from the read only SQL Executer window.

Execute Script for all Companies
A non-published script can be executed against multiple companies using this option. Use the Expansion Button to select companies.

To the left of the company selection list are Mark All and Mark None buttons which can be used to quickly change the selection of the companies. You can also select the System Database if desired. If the selection of the databases is changed from the default (system database not selected and all company databases selected), the selection will be saved along with the script.

Project ID
Use this field to add the current script to a development project.

Parameter ID
Use this field to specify a Parameter List to be used with the script.

Execute Query in which SQL Database
This drop-down list contains a list of SQL databases. The System database and each of the company databases appear in this list.

Limit results set to fixed number of lines
You can use this field to limit the amount of data returned in the results set. Set its value to zero (0) for no limit.

Setting the value of this field to zero (0) can cause SQL Execute Setup to take a long time to display the results if the returned results set is very large.

Database
This non-editable field shows the name of the selected SQL Database.

Script
This text field contains the Transact SQL statements to be executed.
SQL Execute Setup can be used to manipulate data in tables when large set-based changes are required. In this situation writing the equivalent Dexterity sanScript code may not be the most efficient method.

The following is a description of the additional buttons on the window:

Show Dexterity Technical Name Syntax Button
Use this button to display examples of how Dexterity Technical Names can be used in the script.

Divider Adjustment Buttons
Use these buttons to adjust the position of the horizontal window divider between script and results data.

Parameters Button
Use this button to insert a Parameter Placeholder into the script for the Parameter List selected with the Parameter ID. See the section under Automatic Trigger Mode for more information.

Names Button
Use this button to insert a table or field resource into the script. Once clicked the Table Explorer window will open. See the section under Automatic Trigger Mode for more information.

Execute Button
Use this button to execute the script in the context of the SQL database specified. Any execution errors will cause an Exception Error Dialog to open. Results can be shown as unformatted text or as a list.

If a selection of the script is currently highlighted, you can decide to execute the highlighted section or the entire script.
Before the SQL script is executed, it is checked for possible damaging commands and if they exist an additional confirmation is required.

If a GO Statement is used in the script, make sure it is used at the beginning of the line. When GO Statements exist, the script will be executed in sections with one section for each GO statement. Only the final section will display results, so place any select statements to be displayed after the final GO Statement.

Dexterity table and field names can be used in the Transact SQL when surrounded by braces { }. They will be converted to the equivalent physical names prior to the code being executed.

The alias keyword can be used to specify an alias other than the table’s physical name. The show keyword can be used to display the field’s Dexterity display name as the column name. The field keyword is used to limit the generated physical equivalents to be only the column name without the table name or alias prefix.

**Duplicate Button**

Use this button to duplicate the current script ID to a new script ID. This is useful when an existing script ID is very similar to the new one you want to create.

A new script ID must be specified in the dialog which opens.
**Users Button**

Use this button to specify which users and companies the script should be published to. Once clicked Publish Script for Users window will open.

---

You can view this window by users, by companies or by user classes and navigate the tree to select the user and company combinations as required.

*If all users are selected on the tree, the tree selections will be cleared and the mode will change from Selected Users and Companies to All Users and Companies. If no users are selected on the tree, the mode will change to All Users and Companies.*

The Exclude Selected Users and Companies rather than include them option allows you to invert the behavior of the window. This is handy when it is easier to specify the users and companies for whom the script should not be published to.

**Export Button**

This button will allow the result set displayed in the list view to be exported to a file or directly to an email. The default email settings can be set up in the Email Settings window.

**Export Mode**

Use this drop-down list to select the format for the exported file. The file can be exported as Tab Delimited, Comma Delimited or as a HTML Table.
Gotos Button

Use this button drop-down menu to setup SQL Gotos or execute an existing SQL Goto on the selected rows in the returned data.

SQL Gotos allow further actions to be taken on the selected rows of the data returned from SQL Execute scripts. This feature uses Runtime Executer Setup scripts with the Script Purpose set to SQLExecuteGotoHandler to define the required actions and the SQL Execute Setup Gotos window to configure the label to display on the Goto Button and the order of the SQL Gotos. The Runtime Executer Setup Script is executed for each selected line in the result set.

Use the Add Button to add a new SQL Goto and then select the Script ID and define the label to display on the Goto Button. Use “&” if you wish to add a keyboard shortcut and “&&” if you want to add an ampersand character. The order of the SQL Gotos to be changed using the Top, Up, Down and Bottom buttons.

The following is a description of the Script menu available for the window:

Find …

Use this menu option to open the script editor Find window to search for text. Control-F can be used as a shortcut.
Find Next
Use this menu option to find the next occurrence. Control-G can be used as a shortcut.

Replace ...
Use this menu option to open the script editor Replace window to search and replace text. Control-R can be used as a shortcut.

Replace and Find Next
Use this menu option to replace and find the next occurrence. Control-B can be used as a shortcut.

Goto Line ...
Use this menu option to open the script editor Goto Line window to jump to a specified line. Control-N can be used as a shortcut.

Save and Continue
Use this menu option to save the current script without clearing the window. Control-S can be used as a shortcut.

Check Syntax
Use this menu option to check the syntax of Dexterity resource names contained in curly braces in the current script. Control-K can be used as a shortcut.

Convert References
Use this menu option to convert the Dexterity resource names contained in curly braces in the current script to their SQL equivalents. Control-0 can be used as a shortcut.
Options

Use this menu option to open the Options window to allow the font style and size to be changed. Control-O can be used as a shortcut.

The Color options are disabled because the syntax highlighting is not available for SQL scripts.

Execute

Use this menu option to execute the script. Control-E can be used as a shortcut.

Generate Dexterity Pass Through

Use this menu option to generate Dexterity pass through sanScript code from a prototype script that can be copied and pasted into a Dexterity development dictionary. Control-D can be used as a shortcut.
Names Button Uses Clipboard

Use this menu option to control whether the Names Button returns directly to the script (default) or to the clipboard.

![Warning]

To be able to use the clipboard, the WinthropDC.GpPowerToolsVB.dll Addins must be installed.
Following are some examples of using the resource name conversions and keywords:

```sql
select * from {table RM_Customer_MSTR}
```
is converted to

```sql
select * from RM00101
```

```sql
select ('Customer Number' of table RM_Customer_MSTR)
from {table RM_Customer_MSTR}
```
is converted to

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

```sql
select ('Customer Number' of table RM_Customer_MSTR show)
from {table RM_Customer_MSTR}
```
is converted to

```sql
select RM00101.CUSTNMBR as [Customer Number]
from RM00101
```

```sql
select ('Customer Number' of table RM_Customer_MSTR field)
from {table RM_Customer_MSTR}
```
is converted to

```sql
select CUSTNMBR
from RM00101
```

```sql
select ('Customer Number' of table RM_Customer_MSTR show field)
from {table RM_Customer_MSTR}
```
is converted to

```sql
select CUSTNMBR as [Customer Number]
from RM00101
```

```sql
select ('Customer Number' of table RM_Customer_MSTR show alias a)
from {table RM_Customer_MSTR alias a}
```
is converted to

```sql
select a.CUSTNMBR as [Customer Number]
from RM00101 a
```

The “table” keyword, specifying the table for a field, and surrounding field names containing spaces with single quotes are optional, so

```sql
select (Customer Number)
from {RM_Customer_MSTR}
```
is converted to

```sql
select CUSTNMBR
from RM00101
```
.Net Execute Setup

You can open the .Net Execute Setup window by selecting .Net Execute Setup from the Cards section of the GP Power Tools Area Page or by selecting Scripting >> .Net Execute Setup from the Options button drop list on the main window. This is an Advanced Mode feature.

The .Net Execute Setup window can be used to run any Visual C# or Visual Basic.Net code without requiring the Visual Studio development environment. Scripts written in this window can use the form, window, table and field resources from any dictionary or to call existing functions and procedures in any dictionary.

Script IDs created in this window can be loaded and executed from an Automatic Trigger Mode trigger, a Runtime Execute Setup script or another .Net Execute script. This allows code re-use in a similar fashion to having multiple procedure calls as well as mixing of languages.

To be able to execute scripts, the WinthropDC.GpPowerToolsVC.dll and WinthropDC.GpPowerToolsVB.dll Addins must be installed.
Chapter 5
Developer Tools

The following is a description of the individual fields on the window:

**Script ID**
This field contains a unique identifier for each .Net Execute Setup script in the system. The lookup button can be clicked to select from existing script IDs.

![Image of Script ID window]

The Notes Button can be clicked to enter Release Notes. Use the Timestamp Button to add a timestamp to the bottom of the release notes.

![Image of Notes window]

**Script Name**
This field contains a description of the script.
Published to Executer Window
This checkbox indicates if the current script can be accessed from the read only .Net Executer window.

Project ID
Use this field to add the current script to a development project.

Parameter ID
Use this field to specify a Parameter List to be used with the script.

Script Language
This field is used to select the .Net language to be used for this script. You can select from Visual C# or Visual Basic.Net#. When changing the Script Language, the existing script (if any) will be replaced with the base template for the selected language.

Script
This text field contains the script to be executed. It cannot have any parameters. The script is checked for syntax errors when saved.

The following is a description of the additional buttons on the window:

Divider Adjustment Buttons
Use these buttons to adjust the position of the horizontal window divider between script and output data.

Parameters Button
Use this button to insert a Parameter Placeholder into the script for the Parameter List selected with the Parameter ID. See the section under Automatic Trigger Mode for more information.

Insert Button
Use this button to insert a Visual C# or Visual Basic.Net code construct.

Helper Button
Use this button to open the Insert Helper Function window and insert a helper function into the script. See the section under Automatic Trigger Mode for more information.

Names Button
Use this button to insert a dictionary resource into the script. See the section under Automatic Trigger Mode for more information.

Execute Button
Use this button to execute the script in the context of the dictionary specified. Any compile errors will be shown in the status pane below the script.

Duplicate Button
Use this button to duplicate the current script ID to a new script ID. This is useful when an existing script ID is very similar to the new one you want to create.
A new script ID must be specified in the dialog which opens.

**Users Button**

Use this button to specify which users and companies the script should be published to. Once clicked Publish Script for Users window will open.

You can view this window by users, by companies or by user classes and navigate the tree to select the user and company combinations as required.

*If all users are selected on the tree, the tree selections will be cleared and the mode will change from Selected Users and Companies to All Users and Companies. If no users are selected on the tree, the mode will change to All Users and Companies.*

The Exclude Selected Users and Companies rather than include them option allows you to invert the behavior of the window. This is handy when it is easier to specify the users and companies for whom the script should not be published to.
References Button
Use this open the .Net Execute References window. This window can be used to add additional References to dictionary assembly or system dlls.

The following is a description of the Script menu available for the window:

Find …
Use this menu option to open the script editor Find window to search for text. Control-F can be used as a shortcut.

Find Next
Use this menu option to find the next occurrence. Control-G can be used as a shortcut.

Replace …
Use this menu option to open the script editor Replace window to search and replace text. Control-R can be used as a shortcut.

Replace and Find Next
Use this menu option to replace and find the next occurrence. Control-B can be used as a shortcut.
**Goto Line ...**

Use this menu option to open the script editor Goto Line window to jump to a specified line. Control-N can be used as a shortcut.

![Goto Line window](image)

**Save and Continue**

Use this menu option to save the current script without clearing the window. Control-S can be used as a shortcut.

**Check Syntax**

Use this menu option to check the syntax of the current script. Any errors will be displayed in a dialog window. Control-K can be used as a shortcut.

**References**

Use this menu option to open the .Net Execute References window. Control-0 can be used as a shortcut.

**Options**

Use this menu option to open the Options window to allow the font style, and size to be changed. Control-O can be used as a shortcut.

![Options window](image)

The Color options are disabled because the syntax highlighting is not available for .Net scripts.

**Execute**

Use this menu option to execute the script. Control-E can be used as a shortcut.
Names Button Uses Clipboard

Use this menu option to control whether the Names Button returns directly to the script (default) or to the clipboard.

To be able to use the clipboard, the WinthropDC.GpPowerToolsVB.dll Addins must be installed.
Parameter Lists

You can open the Parameter List Maintenance window by selecting Parameter Lists from the Cards section of the GP Power Tools Area Page or by selecting Scripting >> Parameter Lists from the Options button drop list on the main window. This is an Advanced Mode feature.

Parameter Lists provide a method to create a custom user interface to request information from the user prior to a script being executed. The selections made by the user can then be used in the scripts to change the behavior of the script or select the data range the script runs against.

Parameter Lists can be used Automatic Trigger Mode Non-Logging Trigger scripts (for Focus Events, Form Menu and Field Context Menu Types), Runtime Execute Setup scripts, SQL Execute Setup scripts and .Net Execute Setup scripts.

Once a Parameter List dialog has been used, the parameters are available in the script which opened the dialog and any script called by that script as long as the called scripts are linked to the same Parameter ID or have a blank Parameter ID.

To use the data returned from the Parameter List Dialog, just insert a Parameter Placeholder into the script. This will be replaced with the data from the dialog prior to the script being executed. Parameter Placeholders are special language dependent character combinations which return the correct data type so that the script will compile and will also be recognized by the script pre-processor so they can be substituted.
The following is a description of the individual fields on the window:

**Parameter ID**
This field contains a unique identifier for each Parameter List in the system. The lookup button can be clicked to select from existing parameter IDs.

Parameter Description
This field contains a description of the parameter list.
Chapter 5

Developer Tools

Project ID
Use this field to add the current parameter list to a development project.

Parameter Title
This field contains the title for the parameter list. This value will be used at the top of the Parameter List dialog as well as for the title of the dialog’s window.

Parameter Instructions
This field contains the instructions for the user on how they should use the parameter list. This field will be displayed at the top of the Parameter List dialog.

The following is a description of the parameter definition fields for the ten parameters on the window:

Parameter Active
This checkbox specifies whether the current parameter is enabled. This allows a parameter to be temporarily disabled without having to remove the rest of the settings for the parameter.

Parameter Hidden
This checkbox specifies whether the current parameter is hidden. This allows a parameter to be available for use with coding without being shown to the end user.

Parameter Prompt
This field contains a name of the parameter and will be used for the prompt of the parameter.

Parameter Type
Use this drop-down list to select the data type for the parameter. You can select from:

- Checkbox
- Number
- Currency
- Quantity
- String
- Lookup
- Long String
- Date
- Token Date
- Time
- List
- List (SQL)

Parameter Mode
This drop-down list sets whether the parameter is a single value or a range of values. You can select from:
**Parameter Options**

This drop-down list sets various options depending on the Parameter Type selected.

For Lookup Type, you can select from:

- Account Lookup
- Customer Lookup
- Customer Class Lookup
- Vendor Lookup
- Vendor Class Lookup
- Item Lookup
- Item Class Lookup
- Item Site Lookup
- User Lookup
- Custom Lookup (SQL)
- Custom Lookup (Form)
- Custom Lookup (SQL) Uppercase
- Custom Lookup (Form) Uppercase

For List Type, you can select from:

- Cleared Value(s)
- First List Entry
- Default Value(s)

For other types, you can select from:

- Cleared Value(s)
- Default Value(s)
- Min & Max Values

For String and Long String Types, you can also select from:

- Uppercase Cleared Value(s)
- Uppercase Default Value(s)
- Uppercase Min & Max Values

**Parameter Length/Decimal**

This drop-down list controls the length of the field or the number of decimal places for the parameter fields.

For Number Type, you can select the size of the parameter 16 or 32 bit and whether the number should be signed or unsigned.

For Currency and Quantity Types, you can select the number of decimal places from 0 to 5.

For String and Long String Types, you can select the length of the string in 5 character increments.
For List Type, you can select what will be returned by the parameter, you can select from:

- Position
- Data
- String

For all other Types, the length value is fixed.

**Parameter Expansion Button**

Click this button to open the appropriate Parameter List Maintenance Additional Information window.

For Drop-down lists, it will open the Parameter List Drop-down List Maintenance window. This window is used to set up the drop-down list values for a List Type parameter. The drop-down list will be populated with the string on each line in the order it is listed in the window. You can specify an integer (32 bit) value for each entry using a comma.

For SQL Drop-down lists, it will open the Parameter List Drop-down List SQL Script window. This window is used to select a SQL Execute Setup script which returns an ordered query with two columns; a string and an integer (32 bit) value.
For SQL Custom Lookups, it will open the Parameter List Lookup SQL Script window. This window is used to select a SQL Execute Setup script which returns a query with three string columns; an ID string, a Description string and a string to be returned (usually the same as the ID value).

For Form based Custom Lookups, it will open the Parameter List Lookup Form Definition window. This window is used to define the form, window and field information required to drive an existing lookup form in any dictionary installed in Microsoft Dynamics GP.
Parameter Single/Minimum/From Value
Depending on the Parameter Option selected, this field can be used to specify a Minimum value for the parameter, or a default value for the Single or From field of the parameter.

Parameter Maximum/To Value
Depending on the Parameter Option selected, this field can be used to specify a Maximum value for the parameter, or a default value for the To field of the parameter.

The order of the ten parameter in the Parameter List can be adjusted using the small up buttons and down buttons on the right-hand side of the window.

The following is a description of the additional buttons on the window:

Duplicate Button
Use this button to duplicate the current parameter ID to a new parameter ID. This is useful when an existing parameter list is very similar to the new one you want to create.

A new parameter ID must be specified in the dialog which opens.
Preview Button

Use this button to show a preview of what the Parameter List dialog will look like when it executed. If the OK button is used to close the Parameter List dialog, an informational dialog will open showing the results returned for the various languages.

The Parameter List dialog window will automatically resize to the size needed to display only the used and active parameters. If a parameter is unused or not active it will leave a gap in the dialog, except at the bottom where the window will be resized smaller.

The following is a description of the Options menu available:

Save and Continue

Use this menu option to save the current parameter list without clearing the window. Control-S can be used as a shortcut.
Dynamic Trigger Logging

You can open the Dynamic Trigger Logging window by selecting Dynamic Trigger Logging from the Cards section of the GP Power Tools Area Page or by selecting Scripting >> Dynamic Trigger Logging from the Options button drop list on the main window. This is an Advanced Mode feature.

There are times when you are unable to use convention Dexterity Script Logging to follow the flow of scripts in Microsoft Dynamics GP. Some examples are:

- Dexterity Script Logging is unavailable when using Service Based Architecture (SBA) and cannot be enabled.
- Dexterity Script Logging sometimes causes instability which can cause Microsoft Dynamics GP to crash.

Dynamic Trigger Logging can be used as an alternative method to track the flow of scripts. By registering triggers before and after any event (Focus, Table or Script) in the system and logging when that trigger fires, you can track when code is executed.

You will need to know the focus events, table names and script names in advance to be able to register dynamic triggers against them. When working with a Service Procedure for Service Based Architecture, you could capture logs of the code running in the desktop client to get all the procedure and function names. Once you have the names, they can be used to set up the dynamic triggers.
When you first open the Dynamic Trigger Logging window, if the file path has not been written into the Dex.ini file, you will be asked if you want to use the default setup file name.

The following is a description of the individual fields on the window:

**File Path**
This field contains the path to the Dynamic Trigger Logging setup file.

**Trigger Type**
Select the type of trigger from Focus Trigger, Table Trigger or Script Trigger.

**Trigger Mode**
Select the mode of the trigger depending on the type. Focus Triggers can use Pre, Change, Post, Print, Activate, Fill, Insert and Delete. Table Triggers can use Read, Read Lock, Read Both, Add, Update, Save and Delete. Script Triggers use a single Script mode for both Procedures and Functions.

**Product Dictionary**
Select a product dictionary from the list of installed products. The Dictionary ID field will be updated automatically.

**Dictionary ID**
Select a product using its dictionary ID. The Product Dictionary field will be updated automatically.

**Form Name**
Enter the form name or use the lookup to select. This field is required for Focus Triggers and is optional for Table Triggers and Script Triggers.

**Window/Table/Procedure/Function Name**
Enter the window, table, procedure or function name as appropriate or use the lookup to select. Functions are denoted by ending with “()”.

**Field Name**
Enter the field name or use the window lookup to select. Adding a field name to a Focus Trigger or Table Trigger will get the value of the field displayed in the log entries created.
The following is a description of the additional buttons on the window:

**OK Button**
This button will save the triggers to the Dynamic Trigger Logging setup file as defined by the File Path field and writes the File Path into the MBS_Debug_LogListPath Dex.ini Setting.

**Cancel Button**
This button will close the window without making any changes.

**Clear Button**
Use this clear the window. Note it will only clear the File Path field and remove the MBS_Debug_LogListPath Dex.ini Setting.

**Delete Button**
Use this clear the window. Note it will clear the File Path field and remove the MBS_Debug_LogListPath Dex.ini Setting and delete the Dynamic Trigger Logging setup file (if it exists).

**Redisplay Button**
Use this button redisplay the list of triggers and scroll to the bottom of the list ready to add a new Trigger.
When you launch Microsoft Dynamics GP, the MBS_Debug_LogListPath Dex.ini Setting is checked and if it contains a path valid setup file, Dynamic Trigger Logging will parse the setup file and register the triggers specified. When triggers fire, they will write a record in the GP Power Tools log files.

The Dynamic Trigger Logging setup file is a text file and can be edited manually outside of Microsoft Dynamics GP using your favorite text editor or Notepad.exe. The format is explained in the section at the end of the GPPTools.txt file installed with GP Power Tools.

While the triggers registered by Dynamics Trigger Logging do not perform any function other than writing a log entry, it is recommended that this feature is only used by Dexterity developers.
Additional Developer Features

GP Power Tools adds some extra features to help developers. Below is a summary of the features:

**Macro Play Fast**
Added to the Macro menu is the option to Play Fast. This option is the same as the normal play macro option but runs about three times faster.

**Script Debugger Context**
When the Dexterity Script Debugger is opened, the Script Debugger Context window is opened automatically. This window can be used to change the Script Debugger Dictionary Context easily without needing to change Dex.ini settings or restart the application. Use the checkbox at the bottom of the window to control if single or double click is needed to change context.
Resource Information Context
When the Dexterity Script Debugger is enabled, the Resource Information window (when in Form, Window & Fields mode) has a Link to Dexterity Script Debugger option which will link the Dictionary drop down list on the Resource Information window to Script Debugger Context.

<table>
<thead>
<tr>
<th>Resource Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Edit Tools Help Debug</td>
</tr>
<tr>
<td>OK Back Search Again</td>
</tr>
<tr>
<td>Resource Type Form, Window &amp; Fields</td>
</tr>
<tr>
<td>Link to Dexterity Script Debugger</td>
</tr>
<tr>
<td>Form, Report or Table Information</td>
</tr>
<tr>
<td>Product Name 0: Microsoft Dynamics GP</td>
</tr>
</tbody>
</table>

Runtime Execute Context
When the Dexterity Script Debugger is enabled, the Runtime Execute Setup window will default the Dictionary Context to match the Script Debugger Context.

Open Script Debugger on Startup
When the Dexterity Script Debugger is enabled, the MBS_Debug_Break Dex.ini Setting can be used to force the Script Debugger to open upon starting Microsoft Dynamics GP. You can then use the Script Debugger Context window to change Dictionary context and the Script Debugger window to set breakpoints.
Chapter 6: Database Tools Features

This chapter includes the following sections:

- XML Table Export*
- XML Table Import*
- Database Validation*
- SQL Login Maintenance*

* Advanced Mode Feature
XML Table Export

You can open the XML Table Export window by selecting XML Table Import from the Utilities section of the GP Power Tools Area Page or by selecting Database >> XML Table Export from the Options button drop list on the main window. This is an Advanced Mode feature.

The XML Table Export window can be used to copy the contents of one or more tables residing in any product into an XML file. All tables selected will be exported into the single XML file listed on the Export Path.

Using separate Profile IDs allows multiple sets of tables to be stored for particular related groups.

The following is a description of the individual fields on the window:

Profile ID
This field contains a unique identifier for each XML Table Export profile in the system. The lookup button can be clicked to select from existing profile IDs.
Note that the Profile IDs starting with the prefix character of tilde (~) are reserved for use by Microsoft Support.

**Profile Name**
This field contains a description for the XML Table Export profile.

**Table List**
Select the tables you want to export and add them to the list. You can use the lookup or manually enter the Table Technical Name or Table Physical Name fields.

**Export Path**
This field contains the path of the file name to which the tables will be exported as XML.

The following is a description of the additional buttons on the window:

**Duplicate Button**
Use this button to duplicate the current profile ID to a new profile ID. This is useful when an existing profile ID is very similar to the new one you want to create.

A new profile ID must be specified in the dialog which opens.

**Export Button**
Use this button to export the data to the file named in the Export Path field.
For each table specified in the scrolling window section of this window, you can specify an Optional SQL Where Clause to restrict the records export for that table.

XML Table Export can be used to obtain a customer's data for specific tables without requiring a full SQL database backup. Just select the tables for which you need the data and click OK to save the selection. Then use the Configuration Export/Import window to export the setting file to send to the customer. The customer can then import the settings and use the XML Export window to export the desired tables.

During the export or import process, the following progress window will be displayed.

XML Table Export can be used to backup data before running test scenarios so the data can be restored afterwards to allow the scenarios to be run again with the same initial data.

There is no data validation or business logic checking when data is imported using XML Table Import. This is similar to the Dexterity Table Import Utility. It is best to ensure that all related tables are exported by XML Table Export.

If both the XML Table Export window and XML Table Import window are open, the import path will default to the export path from the XML Table Export window.
XML Table Import

You can open the XML Table Import window by selecting XML Table Import from the Utilities section of the GP Power Tools Area Page or by selecting Database >> XML Table Import from the Options button drop list on the main window. This is an Advanced Mode feature.

The XML Table Import window can be used to import the contents of a number of tables from an XML file previously exported by the XML Table Export window.

Select the XML file as the Import Path. The tables contained in the file will be listed.

Select the tables you want to import and then click Import to start importing.

When importing data into tables it is possible that the tables already contain data and that duplicate records may occur. XML Table Import has overwrite options to handle this situation.
The following Overwrite options are available:

**Overwrite Table Contents**
Checking this option will cause the original contents of the table to be deleted prior to importing the XML file. None of the original data will be kept.

**Overwrite Duplicate Records**
Checking this option will allow XML Table Import to overwrite a duplicate record with the data from the XML file. If this option is not checked and a duplicate occurs, the data from the XML file will be ignored and a duplicate record error logged.

During the export or import process, the progress window will be displayed.

XML Table Import can be used to restore data from backups you made before running test scenarios. This allows the scenarios to be run again with the same initial data.

There is no data validation or business logic checking when data is imported using XML Table Import. This is similar to the Dexterity Table Import Utility. It is best to ensure that all related tables are exported by XML Table Export.

If both the XML Table Export window and XML Table Import window are open, the import path will default to the export path from the XML Table Export window.
Database Validation

Please make sure you review the Using Database Validation section for the steps to use this window.

You can open the Database Validation window by selecting Database Validation from the Utilities section of the GP Power Tools Area Page or by selecting Database >> Database Validation from the Options button drop list on the main window. This is an Advanced Mode feature.

The Database Validation window is designed to perform a number of system checks to ensure that your SQL Server settings and databases correctly match what is expected by Microsoft Dynamics GP. If any issues are found, the Database Validation will provide options to resolve them.

Running Database Validation before upgrading or after copying databases between SQL Servers can resolve any potential issues before they occur.

Before the window opens you will be reminded to ensure that all users are logged out and that your company and system databases have been backed up before executing any of the fixing functionality.

Running the Database Validation checks to identify issues is read-only and does not require backups or exclusive use of the system.

You will also be reminded that Database Validation should be executed on a workstation that has the dictionaries for all products installed. This is to ensure that all dictionaries are present when comparing SQL Server tables to tables in the dictionaries.
When the window first opens, Database Validation performs its first series of checks.

**Users and Databases:**
- Confirm that the ‘DYNSA’ SQL Server Login exists
- Confirm that ‘DYNSA’ is assigned as dbo for the SQL Databases
- Confirm that the ‘DYNSA’ GP User ID exists
- Confirm that ‘DYNSA’ is assigned access to all GP Companies
- Confirm that ‘sa’ is assigned access to all GP Companies
- Identify GP Companies for which there is no SQL Database
- Identify GP User IDs for which there is no SQL Login
- Identify GP Users for not assigned to the DYNGRP SQL Role
- Identify Company Access records for missing Users or Companies
- Identify missing Database Users as per Company Access records

**Dynamics GP Utilities:**
Using the records in DB_Upgrade and DU00020 tables in the System database.
- Identify records for companies that are not installed
- Identify records for product dictionaries that are not installed

**Account Framework:**
- Identify Account Framework from Application Dictionary
- Identify Account Framework from setup tables in System Database
- Identify Account Framework from GL_Account_MSTR (GL00100) table in each Company Database
The results of the checks are then displayed when the window opens.

*Please note that this screenshot is intentionally showing errors. A system with no errors will have green ticks next to all of the users, companies and databases shown in the top left and top right panes.*

The following is a description of the individual fields on the window:

**OK Button**
This button closes the window, saving the list of Exempted tabled.

**Redisplay Button**
This button restarts Database Validation window and re-runs the initial checks listed above.

**Process Button**
When this button is pressed, Database Validation reads all the table definitions from the installed Dexterity product dictionaries. It then reads the tables and views from the selected SQL Server databases and identifies which tables and views exist in both SQL Server and the product dictionaries and those only found in one location.
To avoid Database Validation incorrectly classifying tables or views as missing, please ensure that the workstation used to run Database Validation has all installed product dictionaries installed.

**Validate Button**

When this button is pressed, Database Validation will compare the data structures for tables in the selected database tables against the matching table definitions in the product dictionaries and identify any differences.

Running the Validation process can take a while to run as it has to compare the data structures for all the selected tables in multiple databases. This process runs in the foreground and you will not be able to perform other tasks while it is running.

**Add Exemption Button**

This button is enabled when one or more missing tables or views are selected. It can be used to quickly add tables and/or views to the Exemptions list.
CHAPTER 6  DATABASE TOOLS

Remove Exemption Button
This button is enabled when one or more exempt tables or views are selected. It can be used to quickly remove tables and/or views from the Exemptions list.

Exemptions Button
This button opens the Database Validation Exemptions window so that manual changes can be made to the Exemptions list.

The Exemptions list is populated with some known tables and views automatically, but you can manually add additional tables and views which will then be excluded from the validation process.

Exemption Mode can be from the System Database, All Company Databases or for Specified Company Database. Object Mode can be Dexterity Objects or SQL Objects and the View Mode can be Table or View. When specifying a Dexterity Object, you can select the object by entering its physical name or by selecting the Product Name and entering the Technical Name.

To remove entries from the Exemptions list, you can remove individual lines, or select a number of lines and Remove Selected or Remove All.
**Legend Button**

This button opens the Database Validation Legend window which explains the various icons and indicators used by Database Validation.

![Legend Button](image)

**Print Button**

This button opens the Print Report window where you can select what information to include on the printed report.

![Print Button](image)
Fix Users Button

This button opens the Fix Users and Databases window. The various sections of this window will only be enabled if there are errors for that section to be resolved. To the left of the top two panes on the window are Mark All and Mark None buttons which can be used to quickly change the checkbox selections in that pane.

The Edit button can be used to open the Database Validation Email Settings window to edit the email sent when resetting passwords.
Password Reset Emails can be sent automatically when Database Validation knows the User's email address. Use the User Setup Additional Information window to enter this and other user related data.

The Process Button will be enabled if any fixing options have been selected. Click the Process Button to fix the selected issues.

Fix Utilities Button
This button opens the Fix Utilities window.

The Process Button will be enabled if any fixing options have been selected. Click the Process Button to fix the issues found.
**Fix Framework Button**

This button opens the Fix Account Framework window.

**Fix Account Framework**

- **Cancel** button
- **Process** button

**Account Framework**

- **Update settings in System Database to match Application Account Framework**
- **Update settings in System Database to match Company selected below**

<table>
<thead>
<tr>
<th>Location</th>
<th>No. Segments</th>
<th>Segment 1</th>
<th>Segment 2</th>
<th>Segment 3</th>
<th>Segment 4</th>
<th>Segment 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>TST10</td>
<td>5 (3)</td>
<td>9 (3)</td>
<td>9 (4)</td>
<td>5 (2)</td>
<td>5 (0)</td>
<td>5 (0)</td>
</tr>
<tr>
<td>TST10</td>
<td>5 (3)</td>
<td>9 (3)</td>
<td>9 (4)</td>
<td>5 (2)</td>
<td>5 (0)</td>
<td>5 (0)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>No. Segments</th>
<th>Segment 1</th>
<th>Segment 2</th>
<th>Segment 3</th>
<th>Segment 4</th>
<th>Segment 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>DYNAMICS</td>
<td>5 (5)</td>
<td>21 (20)</td>
<td>21 (20)</td>
<td>21 (20)</td>
<td>21 (20)</td>
<td>21 (20)</td>
</tr>
</tbody>
</table>

Numbers in parenthesis show what is actually used instead of database values.

Only one of the two above actions can be processed at any one time.

The Override to Convert Table Structures without using Dynamics Utilities option should be used with caution as it will bypass any additional conversion steps that might have been performed by Dynamics GP Utilities.

The Process Button will be enabled if any fixing options have been selected. Click the Process Button to fix the issues found.

**Fix Tables Button**

This button opens the Fix Tables window. The various sections of this window will only be enabled if there are errors for that section to be resolved.

The table errors are divided into four sections:

- **Empty tables with incorrect Structure Errors**. These tables can be dropped and recreated without needing to consider any existing data.

- **Empty SQL Tables missing from Dexterity dictionaries**. These tables can possibly be removed as they have no data and appear not to be used. They could be from a product that was installed and never used and has since been removed.

- **Tables containing data with incorrect Structure Errors which can be upgraded with Dynamics GP Utilities**. These are tables with date that have structure errors, however there is a conversion available via Dynamics GP Utilities. You should attempt to upgrade with Dynamics GP Utilities first as this should perform the proper conversion steps and might update data as part of the upgrade.

The Override to Convert Table Structures without using Dynamics Utilities option should be used with caution as it will bypass any additional conversion steps that might have been performed by Dynamics GP Utilities.
- Tables containing data with incorrect Structure Errors which cannot be upgraded with Dynamics GP Utilities. These are tables with data that have structure errors, but there is no Dynamics GP Utilities conversion available. You can use Database Validation to automatically backup the data, drop and recreate the tables and restore the data.

To the left of each pane on the window is a Mark All and Mark None button which can be used to quickly change the checkbox selections of the tables in that pane.

Ensure you have a backup of all Microsoft Dynamics GP databases before running the Fix Tables process. If you are unsure about running any Fix Tables process, please contact your support consultants to discuss.
The Process Button will be enabled if any fixing options have been selected. Click the Process Button to fix the selected issues.

After running the Fix Tables process, it is recommended to run the Database Maintenance Utility (DBMaintenance.exe) to update or create any additional SQL Server resources.

**Only include SQL Table & Views which have a DEX_ROW_ID column**

This option limits the SQL Tables and Views reviewed to only include ones that include a DEX_ROW_ID column. Keeping this option selected prevents Database Validation looking at additional SQL objects that are not used with Dexterity product dictionaries.

**Only Show Tables with Account Fields**

This option filters the scrolling window to only include tables which include an Account Number field. This is useful when looking for tables with Account Framework issues.
Show Structure Errors Button

This button opens the Table Structure Errors window. This window details the database structure errors for the selected table.

Click OK to close the window.
The Database Validation window has an Options Menu which can be used to Reset User SQL Logins and Passwords. This option can be used for force a reset of selected users’ or all users’ passwords by removing their SQL Logins and allowing Database Validation to recreate them. It will not remove the SQL Login for any users currently logged into the system.

If you wish to reset user passwords, view or change password policy settings or change a user’s status, you might find the SQL Login Maintenance window a better approach as it does not remove the SQL Login and require it to be recreated. See the next section for more information.
Using Database Validation
The following section explains the process of using Database Validation on your system.

1. Backup all system and company SQL Server databases.

2. Ensure that no other users are logged in.

3. Ensure that all Dexterity product dictionaries are installed on the current workstation.

4. After Database Validation has performed its initial checks, the window will open and display what issues it has found.

5. Use the Fix Users and Databases window to resolve any issues with Users and Databases. The window will refresh after the process.

6. Use the Fix Utilities window to resolve any issues with the data in the Dynamics GP Utilities version tables. The window will refresh after the process.

7. Use the Fix Account Framework window to resolve any issues with the data Account Framework in the system. The window will refresh after the process.
8. Once all these issues are fixed, the Database Validation window show now display with no errors.

9. Click Process to read the tables and views from all Dexterity product dictionaries installed and match them to the tables and views in the selected SQL Server system database and company databases in the top right-hand pane.

10. Once the processing has completed. You can explore the tree in the top right pane and see what tables and views have been found to exist in both the Dexterity product dictionaries and the SQL Server databases. The scrolling window in the bottom half of the window will display the tables depending on the node selected in the top right pane.

11. You can select Missing tables and views and add them to the Exemptions using the Add Exemptions button, so they don’t show as missing next time.
12. You can select all or some of the Found Tables in the databases. The selection can be made by clicking on the node checkboxes or by selecting a node and then clicking on the tables in the scrolling window. Clicking on the node checkboxes in the tree can be used to mark all or mark none.

13. Click Validate to compare the table structures for the selected tables found in both Dexterity product dictionaries and SQL Server databases. This process can take some time depending on how many tables are being checked. It runs multiple passes to complete the process.
14. After the Validation process is complete, an optional report can be printed to show the results. You can use the tree to explore the results and show the tables which have been identified as Structure Incorrect.

15. If you wish to see the detail of the Structure Errors, select the individual table in the scrolling window and click the Show Structure Errors button.
16. Use the Fix Tables window to resolve any issues with Tables. The window will refresh after the process.

17. When closing the Database Validation window, print or save the Database Validation Log report generated with all the actions processed by Database Validation.

18. Exit Dynamics GP and run the Database Maintenance utility against all system and company databases.

19. Make a second backup of all system and company SQL Server databases of your validated system.
SQL Login Maintenance

You can open the SQL Login Maintenance window by selecting Database Validation from the Utilities section of the GP Power Tools Area Page or by selecting Database >> SQL Login Maintenance from the Options button drop list on the main window. This is an Advanced Mode feature.

The SQL Login Maintenance window is designed to provide a simple method to reset user passwords, view or change password policy settings or make users inactive or active for multiple users at one time.

The Database Validation has an option to Reset User SQL Logins and Passwords. Using the SQL Login Maintenance window instead resets the password without removing and recreating SQL Logins. If there is a problem with the SQL Login you can use Database Validation to fix it before using SQL Login Maintenance.

The following is a description of the individual fields on the window:

User List
This list shows all the users in the system with their current status and password policy settings. Double clicking on a user will open User Setup for the selected user. From the User Setup window, you can check the User Setup Additional Information window and ensure the User Email Address is populated. Use the checkboxes to select the users you wish to apply any changes to.

Send Password changed emails
When this checkbox is selected, SQL Login Maintenance will send emails to users when resetting their password as long as the User Email Address is populated for the user in the User Setup Additional Information window.
The Edit button can be used to open the Database Validation Email Settings window to edit the email sent when resetting passwords.

![Database Validation Email Settings window](image)

**Reset User Passwords**
Select this checkbox if you want to reset the passwords for the selected users.

**Automatically Generate Passwords**
Select this checkbox if you want GP Power Tools to generate individual passwords for each user.

**User Password**
If not generating passwords, use this field to enter a single password to user for all users.

**Apply Advanced SQL Server options**
Select this checkbox if you want to change Advanced SQL Server options for the selected users.

**Enforce Password Policy**
Select this checkbox to update users to enable the system’s password policies.

**Enforce Password Expiration**
Select this checkbox to update users to enable the password expiration policy.

**Change Password Next Login**
Select this checkbox to force users to change their password on next login. This option can only be used when resetting passwords.

**Apply User Status**
Select this checkbox if you want to change User Status settings for the selected users.

**User Status**
Use this drop down list to select if users should be marked Active, Inactive or as Lesson Users.
The following is a description of the additional buttons on the window:

**Cancel Button**
This button closes the window without taking any further actions.

**Apply Button**
This button will apply the selected setting changes to the selected users.

**Redisplay Button**
This button will refresh the window with the current users and their settings.

**Mark All Button**
Use this button to mark all the users which can be updated.

**Unmark All Button**
Use this button to unmark all the users which can be updated.
Chapter 7: Dex.ini Settings

GP Power Tools Settings

GP Power Tools uses the Dex.ini file to store a number of settings. The default location for the Dex.ini file is in the data subfolder beneath the Microsoft Dynamics GP application folder. These settings are explained below:

**MBS_Debug_Path**
This setting can point to a location for the Debugger.xml setup file. The default for this setting is missing, which means that the Debugger.xml file will be stored in the data subfolder beneath the Microsoft Dynamics GP application folder.

**MBS_Debug_SetupMode**
This setting can be TRUE or missing and denotes whether Setup Mode is enabled. The default for this setting is missing, which means that Setup Mode is not enabled.

**MBS_Debug_AutoOpen**
This setting can be TRUE or missing and denotes whether GP Power Tools window should open automatically after logging into a company.

**MBS_Debug_Version**
This setting tracks the last used version of GP Power Tools on the current workstation.

**MBS_Debug_LogOnStartup**
This setting can be TRUE or missing and denotes whether to automatically start logging when Microsoft Dynamics GP is next started. The default for this setting is missing, which means that the feature is disabled.

**MBS_Debug_RuntimeCheck**
This setting can be FALSE or missing and denotes whether the Runtime Engine version and build information is checked for compatibility. The default for this setting is missing, which means that the version and build will be checked.

**MBS_Debug_ShowRuntime**
This setting can be TRUE or missing, and denotes whether the Runtime Engine is shown when creating Dexterity sanScript scripts in either the Trigger Setup window or the Runtime Execute Setup window. If this setting is enabled, the Resource Explorer window and Table Explorer window will also display resources from the Runtime Engine dictionary DEX.DIC. The default for this setting is missing, which means the runtime engine is not displayed.
**MBS_Debug_ConfigurationOverride**
This setting can be TRUE or missing, and denotes whether GP Power Tools is allowed to automatically update Dex.ini Settings for this workstation as defined in the Dex.ini Configuration window. Set to TRUE to prevent any updates.

**MBS_Debug_LaunchConfigurationOverride**
This setting can be TRUE or missing, and denotes whether GP Power Tools is allowed to automatically change the Launch File for this workstation as defined in the Launch File Configuration window. Set to TRUE to prevent any updates.

**MBS_Debug_LogAppDetails**
This setting can be TRUE or missing and denotes whether GP Power Tools should log an entry into the GPPTools_<User>_Company.log file each time a user logs into a company.

**SQLLogRename**
This setting can be used to automatically rename the DEXSQL.LOG file each day. The value will be the date of the last rename in the form YYYYMMDD.

**SQLLastCompany**
This setting is used to automatically store the last Company ID selected for the current workstation. This allows the company selection drop-down list to be defaulted to the last company used.

**DefaultLastCompany**
This setting can be FALSE or missing and is used to disable the automatic defaulting of the last company used when logging into Microsoft Dynamics GP or switching companies.

**MBS_Debug_UpdateLastUserOnExit**
This setting can be FALSE or missing and is used to disable writing the last user and company details when exiting from Microsoft Dynamics GP.

**MBS_Debug_CompanySwitchWidth**
This setting can be TRUE or missing and is used to expand the fields on the Company Login window to use the full width of the window.

**MBS_Debug_WinDebugger**
This setting is used to store the last window size, position, and state for the GP Power Tools main window.

**MBS_Debug_WinDebuggerSetup**
This setting is used to store the last window size, position, and state for the GP Power Tools Setup window.

**MBS_Debug_WinDebuggerStatus**
This setting is used to store the last window size, position, and state for the Trigger Status window.
**MBS_Debug_WinResourceInformation**
This setting is used to store the last window size, position, and state for the Resource Information window.

**MBS_Debug_WinSecurityProfiler**
This setting is used to store the last window size, position, and state for the Security Profiler window.

**MBS_Debug_WinSecurityInfo**
This setting is used to store the last window size, position, and state for the Security Information window.

**MBS_Debug_WinSecurityInfoResource**
This setting is used to store the last window size, position, and state for the Security Information Resources window.

**MBS_Debug_WinSecurityLog**
This setting is used to store the last window size, position, and state for the Security Log window.

**MBS_Debug_WinSecurityLogDetail**
This setting is used to store the last window size, position, and state for the Security Log Details window.

**MBS_Debug_WinSecurityAnalyzer**
This setting is used to store the last window size, position, and state for the Security Analyzer window.

**MBS_Debug_WinSecurityEnhanced**
This setting is used to store the last window size, position, and state for the Enhanced Security window.

**MBS_Debug_WinSecurityDeny**
This setting is used to store the last window size, position, and state for the Security Denied window.

**MBS_Debug_WinSecurityHide**
This setting is used to store the last window size, position, and state for the Security Hide window.

**MBS_Debug_WinDictionaryControl**
This setting is used to store the last window size, position, and state for the Dictionary Control window.

**MBS_Debug_WinCompanyFilter**
This setting is used to store the last window size, position, and state for the Company Login Filter window.

**MBS_Debug_WinWindowMemory**
This setting is used to store the last window size, position, and state for the Window Position Memory window.
MBS_Debug_WinActivityLog
This setting is used to store the last window size, position, and state for the User Activity Log window.

MBS_Debug_WinActivityLogMaxUser
This setting is used to store the last window size, position, and state for the User Activity Log Maximum Users window.

MBS_Debug_WinLoginLimits
This setting is used to store the last window size, position, and state for the Login Limits window.

MBS_Debug_WinLaunchFileConfig
This setting is used to store the last window size, position, and state for the Launch File Configuration window.

MBS_Debug_WinProductSelection
This setting is used to store the last window size, position, and state for the Dynamic Product Selection window.

MBS_Debug_WinXMLTableExport
This setting is used to store the last window size, position, and state for the XML Table Export window.

MBS_Debug_WinXMLTableImport
This setting is used to store the last window size, position, and state for the XML Table Import window.

MBS_Debug_WinDatabaseValidation
This setting is used to store the last window size, position, and state for the Database Validation window.

MBS_Debug_WinLoginMaintenance
This setting is used to store the last window size, position, and state for the SQL Login Maintenance window.

MBS_Debug_WinProjectSetup
This setting is used to store the last window size, position, and state for the Project Setup window.

MBS_Debug_WinRuntimeExecute
This setting is used to store the last window size, position, and state for the Runtime Execute Setup window.

MBS_Debug_WinRuntimeExecuter
This setting is used to store the last window size, position, and state for the Runtime Executer window.

MBS_Debug_WinSQLExecute
This setting is used to store the last window size, position, and state for the SQL Execute Setup window.
MBS_Debug_WinSQLExecute
This setting is used to store the last window size, position, and state for the SQL Executer window.

MBS_Debug_WinSQLResults
This setting is used to store the last window size, position, and state for the SQL Results window.

MBS_Debug_WinNetExecute
This setting is used to store the last window size, position, and state for the Net Executive window.

MBS_Debug_WinNetExecute
This setting is used to store the last window size, position, and state for the Net Executer window.

MBS_Debug_WinParameterMaintenance
This setting is used to store the last window size, position, and state for the Parameter List Maintenance window.

MBS_Debug_WinTriggerListMaintenance
This setting is used to store the last window size, position, and state for the Dynamic Trigger Logging window.

MBS_Debug_WinConfigurationExportImport
This setting is used to store the last window size, position, and state for the Configuration Export/Import window.

MBS_Debug_WinConfigurationMaintenance
This setting is used to store the last window size, position, and state for the Configuration Maintenance window.

MBS_Debug_WinScreenShot
This setting is used to store the last window size, position, and state for the ScreenShot window.

MBS_Debug_WinLoggingSettings
This setting is used to store the last window size, position, and state for the Logging Settings window.

MBS_Debug_WinEmailSettings
This setting is used to store the last window size, position, and state for the Email Settings window.

MBS_Debug_WinAdminSettings
This setting is used to store the last window size, position, and state for the Administrator Settings window.

MBS_Debug_WinConfigSettings
This setting is used to store the last window size, position, and state for the Dex.ini Configuration window.
MBS_Debug_WinSendEmail
This setting is used to store the last window size, position and state for the Send Email window.

MBS_Debug_WinResourceExplorer
This setting is used to store the last window size, position and state for the Resource Explorer window.

MBS_Debug_WinMenuExplorer
This setting is used to store the last window size, position and state for the Menu Explorer window.

MBS_Debug_WinTableExplorer
This setting is used to store the last window size, position and state for the Table Explorer window.

MBS_Debug_WinReportExplorer
This setting is used to store the last window size, position and state for the Report Explorer window.

MBS_Debug_WinObjectExplorer
This setting is used to store the last window size, position and state for the Security Object Explorer window.

MBS_Debug_WinScriptExplorer
This setting is used to store the last window size, position and state for the Script Explorer window.

MBS_Debug_WinTableLookup
This setting is used to store the last window size, position and state for the Table Lookup window.

MBS_Debug_WinFieldLookup
This setting is used to store the last window size, position and state for the Field Lookup window.

MBS_Debug_WinCalculator
This setting is used to store the last window size, position and state for the Calculator window.

MBS_Debug_WinKeyLookup
This setting is used to store the last window size, position and state for the Table Keys Lookup window.

MBS_Debug_WinScreenOutput
This setting is used to store the last window size, position and state for the Report Writer Screen Output window.

MBS_Debug_DisableScreenOutputMemory
This setting can be used to disable the window position memory feature for the Report Writer Screen Output window.
MBS_Debug_Automate_File
This setting is used by Microsoft Support to provide the full path or filename to a Diagnostics configuration settings file to be loaded after logging into Microsoft Dynamics GP. If the full path is not provided, the file can be located in the Debugger logs folder, the application’s Data folder, or the folders where the DYNAMICS.EXE or DYNAMICS.SET are located. Trigger IDs, Script IDs and Profile IDs loaded with this option should be prefixed with a tilde (~) character. By default, this setting is removed after use.

MBS_Debug_Automate_Script
This setting is used by Microsoft Support to provide the Script ID for a Runtime Execute Setup Diagnostics script to be executed after logging into Microsoft Dynamics GP. The Script ID executed with this option should be prefixed with a tilde (~) character. By default, this setting is removed after use.

MBS_Debug_Automate_Status
This setting is used by Microsoft Support to control the behavior of the Diagnostics automation features of GP Power Tools. By default, this setting is removed after use. The valid flags (which can be added together) are as follows:

1 - Do not delete settings loaded from configuration settings file.
2 - Do not delete Diagnostics Automation Dex.ini settings.
4 - Do not delete configuration settings XML file.
8 - Do not display “Please Wait” dialogs while loading settings file.

MBS_Debug_DisableSplitters
This setting can be used to disable the splitter functionality on the Security Information and Resource Explorer windows. Set it to TRUE to disable the splitters.

MBS_Debug_VBADisableReset
This setting is used by GP Power Tools to signify that Visual Basic for Applications (VBA) should be re-enabled after one login.

MBS_Debug_VSTDDisable
This setting is used by GP Power Tools to disable Visual Studio Tools Addins on login.

MBS_Debug_VSTDDisableReset
This setting is used by GP Power Tools to signify that Visual Studio Tools Addins should be re-enabled after one login.

MBS_Debug_SkipVersionChecks
This setting is used to allow GP Power Tools to run on a different version of Dexterity than the one it was built for. It is to be used when testing GP Power Tools on upcoming versions of Microsoft Dynamics GP.

MBS_Debug_LastRunSystem
This setting is used to track when GP Power Tools was last run on a particular workstation.
**MBS_Debug_LastRunUser**
This setting is used to track when GP Power Tools was last run by a particular user.

**MBS_Debug_LogWinData**
This setting is used to enable logging for the automatic window positioning code for troubleshooting purposes.

**MBS_Debug_CompanyFilter**
This setting is used to specify the Company Login Filter Profile ID to use for the current workstation.

**MBS_Debug_LogListPath**
This setting is used to specify the text file containing the settings for Dynamic Trigger Logging.

**MBS_Debug_Break**
This setting can be TRUE or missing and is used to force the Script Debugger (if enabled) to open automatically when starting Microsoft Dynamics GP.

**MBS_Debug_LookupPosition**
This setting can be FALSE or missing and is used to disable the Lookup Window Positioning which ensures Lookup windows open next to the calling window.

**MBS_Debug_NamesUseClipboard**
This setting can be TRUE or missing and is used to enable Names Button Uses Clipboard option on the script menu.
System Settings

GP Power Tools can also manipulate the values of certain system settings stored in the Dex.ini settings file:

**SQLLogSQLStmt**
This setting can be TRUE or FALSE and controls whether statements Microsoft Dynamics GP sends to the SQL Server are logged to the DEXSQL.LOG file by default.

**SQLLogODBCMessages**
This setting can be TRUE or FALSE and controls whether ODBC messages returned from the SQL Server back to the Microsoft Dynamics GP client are logged to the DEXSQL.LOG file by default.

**SQLLogAllODBCMessages**
This setting can be TRUE or FALSE and controls whether all ODBC messages returned from the SQL Server back to the Microsoft Dynamics GP client are logged to the DEXSQL.LOG file by default.

**SQLLogPath**
This setting can be used to change the default location of the DEXSQL.LOG file.

**ScriptDebugger**
This setting can be TRUE or FALSE and controls whether the Dexterity Debug menu is available in runtime mode.

**ScriptDebuggerProduct**
This setting contains the Dexterity Product ID that will be used to set the initial context of the Debug menu. The default value is 0 for Dynamics.

**ShowDebugMessages**
This setting can be TRUE or FALSE and controls whether internal debug message dialogs are displayed when the Debug Menu is enabled. It is recommended that this should be set to FALSE for production systems.

**ScriptLogEnhanced**
This setting can be TRUE or FALSE and controls whether the enhanced Dexterity Script Log features are enabled. Enabling this option adds time stamps and flagging of background processes to the script log. The default value is set to TRUE by GP Power Tools.

**ApplicationName**
This setting contains the name to be shown on the title bar when the application first launches. If this value is not defined, the name in the title bar will default to “Dexterity Runtime”.

**AutoInstallChunks**
This setting allows chunks to be included without prompting when Microsoft Dynamics GP is launched.
CHAPTER 7

DEX.INI SETTINGS

AllowWrongDex
This setting allows a mismatched Dex.dic and Dexterity Runtime version to be used. It is not recommended to use this option.

SkipVersionChecks
This setting allows Microsoft Dynamics GP to launch without errors even when the dictionary version numbers do not match the version information in the database. It is not recommended to use this option.

SAMPLEDATEMSG
This setting prevents the Fabrikam sample company date warning dialog from opening when logging in.

SQLLoginCompatibilityMode
This setting controls if Microsoft Dynamics GP continues to use SQL Login Compatibility Mode.

ExportOneLineBody
This setting controls whether text report body sections are exported as a single line in the export file.

ExportLinesPerPage
This setting controls the number of lines to include on a report page when it is exported to a file.

ExportPDFLinesPerPage
This setting controls the number of lines to include on a report page when it is exported to a PDF file.

DebugRW
This setting is used to configure the Report Writer to create a debugging log file named DebugRW.txt that will appear in the data subfolder beneath the Microsoft Dynamics GP application folder.

SuppressChangeDateDialog
This setting prevents the Change Date dialog from being displayed at midnight. If used, the User Date will not change at midnight.

ShowAdvancedMacroMenu
This setting will enable the Advanced Macro Menu from the Tools >> Macro menu.

ShowAllMenuItems
This setting will leave all menu items showing even if the module is not installed or if access is denied.

SuppressSound
This setting disables all sound from the Microsoft Dynamics GP application.
QueueMoreInfo
This setting can be used to enable the More Info button on the Process Monitor window.

MouseWheel
This setting can be used to disable Mouse Wheel scrolling in the application.

MaxSWScrollbarSize
This setting can be used to override the width of scrollbars in the application. The default value is 17 pixels.

DebugFonts
This setting can be used to enable logging of Report Writer selections to the DebugLog.txt file.

TPELogging
This setting can be used to enable logging of the internals of the Template Processing Engine (TPE) for word templates.

KeepTemplateTempFiles
This setting can be used to disable the automatic removal of the temporary files used when the Template Processing Engine (TPE) runs.

VBADisable
This setting can be used to disable Visual Basic for Applications when restarting Microsoft Dynamics GP.

EnableServerDropDown
This setting can be used to disable the Data Source Server selection when logging into Microsoft Dynamics GP.

DefaultLastUser
This setting can be used to disable the defaulting of the last user used when logging into Microsoft Dynamics GP.

EnableWCRibbons
This setting can be used to disable the GP 2013 R2 or later Web Client style ribbons in the desktop client for the current workstation.

WindowMax
This setting can be used to control whether the application opens full screen for the current workstation.

WindowPosX & WindowPosY
These settings can be used to control the default application position when not maximized for the current workstation.

WindowWidth & WindowHeight
These settings can be used to control the default application size when not maximized for the current workstation.
OLEClose
This setting can be used to control whether the application attempts to close the OLE Contain.exe program on exit for the current workstation.
Script Editor Settings

GP Power Tools uses some of the Dexterity Script Editor Dex.ini settings:

**ScriptEditorSyntaxColoring**
This setting stores whether Syntax Highlighting is enabled.

**ScriptKeywordColor**
This setting stores the color selection for keywords.

**ScriptIdentifierColor**
This setting stores the color selection for identifiers.

**ScriptNumberColor**
This setting stores the color selection for numbers.

**ScriptStringColor**
This setting stores the color selection for strings.

**ScriptCommentColor**
This setting stores the color selection for comments.

**ScriptOperatorColor**
This setting stores the color selection for operators.

**ScriptErrorColor**
This setting stores the color selection to display Scripting Highlighting errors.

**ScriptEditorFontName**
This setting stores the font style section.

**ScriptEditorFontSize**
This setting stores the font size section.
Chapter 8: Helper Functions

GP Power Tools has a number of helper functions which can be used to make cross-dictionary Dexterity sanScript simpler to write. The Helper Function Assistant window will automatically insert the code required to use these functions.

Below are the details of the helpers available:

- `MBS_Get_Window_Value`
- `MBS_Set_Window_Value`
- `MBS_Get_Table_Value1`
- `MBS_Set_Table_Value1`
- `MBS_Get_Table_Value2`
- `MBS_Set_Table_Value2`
- `MBS_Get_Table_Value3`
- `MBS_Set_Table_Value3`
- `MBS_Get_Table_Value4`
- `MBS_Set_Table_Value4`
- `MBS_Runtime_Execute`
- `MBS_Runtime_Execute_Modified`
- `MBS_Runtime_Execute_Background`
- `MBS_Runtime_Execute_Delayed`
- `MBS_SQL_Check_Exists`
- `MBS_Export_SQL_Query_To_File`
- `MBS_SQL_Results`
- `MBS_SQL_Results_Goto`
- `MBS_SQL_Goto_Get_Data`
- `MBS_Net_Execute`
- `MBS_Script_Load_Dex`
- `MBS_Script_Load_SQL`
- `MBS_Script_Load_Net`
- `MBS_Param_Set`
- `MBS_Param_Get`
- `MBS_Param_Del`
- `MBS_Param_DelAll`
- `MBS_Auto_Log`
- `MBS_Trigger_Start`
- `MBS_Trigger_Stop`
- `MBS_DUOS_Set`
- `MBS_DUOS_Get`
- `MBS_DUOS_Del`
- `MBS_DUOS_DelAll`
- `MBS_SQL_Lookup`
- `MBS_SQL_Lookup_Parameter`
- `MBS_Form_Lookup`
- `MBS_Form_Lookup_Parameter`
- `MBS_Project_Start`
- `MBS_Project_Stop`
• MBS_Script_Substitute
• MBS_Parameter_Placeholder
• MBS_Parameter_String
• MBS_Parameter_Number
• MBS_Parameter_Currency
• MBS_Parameter_Boolean
• MBS_Parameter_Date
• MBS_Parameter_Time
• MBS_Parameter_Load
• MBS_Parameter_Set_String
• MBS_Parameter_Set_Number
• MBS_Parameter_Set_Currency
• MBS_Parameter_Set_Boolean
• MBS_Parameter_Set_Date
• MBS_Parameter_Set_Time
• MBS_Parameter_Get_String
• MBS_Parameter_Get_Number
• MBS_Parameter_Get_Currency
• MBS_Parameter_Get_Boolean
• MBS_Parameter_Get_Date
• MBS_Parameter_Get_Time
• MBS_getmsg
• MBS_Token
• MBS_Field_ParseText
• MBS_Security_Form_Check
MBS_Get_Window_Value

This call is used to obtain the value of a window field from any open form in any dictionary.

The parameter list for this call is:

- in integer IN_Prod_ID;
- in string IN_Form_Name;
- in string IN_Window_Name;
- in string IN_Field_Name;
- out anonymous field OUT_Field_Value;
- out integer OUT_Status;

An example script is:

```plaintext
local integer l_status;
local string l_field;
call with name "MBS_Get_Window_Value" in dictionary 5261, Dictionary, "Form", "Window", "Field", l_field, l_status;
if l_status = OKAY then
    warning str(l_field);
end if;
```
MBS_Set_Window_Value

This call is used to set the value of a window field from any open form in any dictionary. You have the option to also run the target field’s change script.

The parameter list for this call is:

in integer IN_Prod_ID;
in string IN_Form_Name;
in string IN_Window_Name;
in string IN_Field_Name;
in anonymous field IN_Field_Value;
in boolean IN_Run_Flag;
out integer OUT_Status;

An example script is:

local integer l_status;
local string l_field;
l_field = "Value";
call with name "MBS_Set_Window_Value" in dictionary 5261, Dictionary, "Form", "Window", "Field", l_field, true {run script}, l_status;
if l_status <> OKAY then
  warning str(l_status);
end if;
MBS_Get_Table_Value1

This call is used to obtain the value of a field located in any table in any
dictionary using an index containing one field.

All table and field names need to be the technical names and surrounded
by single quotes if they contain a space. The status returned will contain
the number of errors that occurred, a value of OKAY (zero) means the call
was successful. The Key Name fields need to contain the technical names
of the segment fields of the index being used.

The parameter list for this call is:

\[
\begin{align*}
in & \text{ integer } \text{IN\_Prod\_ID}; \\
in & \text{ string } \text{IN\_Table\_Name}; \\
in & \text{ string } \text{IN\_Field\_Name}; \\
\text{out} & \text{ anonymous field } \text{OUT\_Field\_Value}; \\
\text{out} & \text{ integer } \text{OUT\_Status}; \\
in & \text{ integer } \text{IN\_Index}; \\
in & \text{ string } \text{IN\_Key\_Name1}; \\
\text{in} & \text{ anonymous field } \text{IN\_Key\_Value1}; \\
\end{align*}
\]

An example script is:

\[
\begin{align*}
&\text{local integer l\_status;} \\
&\text{local string l\_field;} \\
&\text{local string l\_key1;} \\
&\text{l\_key1 = "Value1";} \\
&\text{call with name "MBS\_Get\_Table\_Value1" in dictionary 5261,} \\
&\text{Dictionary, "Table", "Field", l\_field, l\_status, l \{Index\},} \\
&\text{"Key1", l\_key1;} \\
&\text{if l\_status = OKAY then} \\
&\quad \text{warning str(l\_field);} \\
&\text{end if;} \\
\end{align*}
\]
MBS_Set_Table_Value1

This call is used to update the value of a field located in any table in any dictionary using an index containing one field. You can specify whether the creation of a new table record is allowed.

All table and field names need to be the technical names and surrounded by single quotes if they contain a space. The status returned will contain the number of errors that occurred, a value of OKAY (zero) means the call was successful. The Key Name fields need to contain the technical names of the segment fields of the index being used.

The parameter list for this call is:

- in integer IN_Prod_ID;
- in string IN_Table_Name;
- in string IN_Field_Name;
- in anonymous field IN_Field_Value;
- out integer OUT_Status;
- in integer IN_Index;
- in boolean IN_Allow_Add;
- in string IN_Key_Name1;
- in anonymous field IN_Key_Value1;

An example script is:

```plaintext
local integer l_status;
local string l_field;
local string l_key1;
l_key1 = "Value1";
l_field = "Value"
call with name "MBS_Set_Table_Value1" in dictionary 5261,
Dictionary, "Table", "Field", l_field, l_status, 1 {Index}, true
{allow add},
   "Key1", l_key1;
if l_status <> OKAY then
   warning str(l_status);
end if;
```
MBS_Get_Table_Value2

This call is used to obtain the value of a field located in any table in any dictionary using an index containing two fields.

All table and field names need to be the technical names and surrounded by single quotes if they contain a space. The status returned will contain the number of errors that occurred, a value of OKAY (zero) means the call was successful. The Key Name fields need to contain the technical names of the segment fields of the index being used.

The parameter list for this call is:

in integer IN_Prod_ID;
in string IN_Table_Name;
in string IN_Field_Name;
out anonymous field OUT_Field_Value;
out integer OUT_Status;
in integer IN_Index;
in string IN_Key_Name1;
in anonymous field IN_Key_Value1;
in string IN_Key_Name2;
in anonymous field IN_Key_Value2;

An example script is:

local integer l_status;
local string l_field;
local string l_key1, l_key2;
l_key1 = "Value1";
l_key2 = "Value2";
call with name "MBS_Get_Table_Value2" in dictionary 5261, Dictionary, "Table", "Field", l_field, l_status, 1 (Index),
"Key1", l_key1,
"Key2", l_key2;
if l_status = OKAY then
  warning str(l_field);
end if;
MBS_Set_Table_Value2

This call is used to update the value of a field located in any table in any dictionary using an index containing two fields. You can specify whether the creation of a new table record is allowed.

All table and field names need to be the technical names and surrounded by single quotes if they contain a space. The status returned will contain the number of errors that occurred, a value of OKAY (zero) means the call was successful. The Key Name fields need to contain the technical names of the segment fields of the index being used.

The parameter list for this call is:

- in integer IN_Prod_ID;
- in string IN_Table_Name;
- in string IN_Field_Name;
- in anonymous field IN_Field_Value;
- out integer OUT_Status;
- in integer IN_Index;
- in boolean IN_Allow_Add;
- in string IN_Key_Name1;
- in anonymous field IN_Key_Value1;
- in string IN_Key_Name2;
- in anonymous field IN_Key_Value2;

An example script is:

```gp
local integer l_status;
local string l_field;
local string l_key1, l_key2;
l_key1 = "Value1";
l_key2 = "Value2";
l_field = "Value"
call with name "MBS_Set_Table_Value2" in dictionary 5261, Dictionary, "Table", "Field", l_field, l_status, 1 {Index}, true {allow add},
         "Key1", l_key1,
         "Key2", l_key2;
if l_status <> OKAY then
  warning str(l_status);
end if;
```
MBS_Get_Table_Value3

This call is used to obtain the value of a field located in any table in any dictionary using an index containing three fields.

All table and field names need to be the technical names and surrounded by single quotes if they contain a space. The status returned will contain the number of errors that occurred, a value of OKAY (zero) means the call was successful. The Key Name fields need to contain the technical names of the segment fields of the index being used.

The parameter list for this call is:

```
in integer IN_Prod_ID;
in string IN_Table_Name;
in string IN_Field_Name;
out anonymous field OUT_Field_Value;
out integer OUT_Status;
in integer IN_Index;
in string IN_Key_Name1;
in anonymous field IN_Key_Value1;
in string IN_Key_Name2;
in anonymous field IN_Key_Value2;
in string IN_Key_Name3;
in anonymous field IN_Key_Value3;
```

An example script is:

```
local integer l_status;
local string l_field;
local string l_key1, l_key2, l_key3;
l_key1 = "Value1";
l_key2 = "Value2";
l_key3 = "Value3";
call with name "MBS_Get_Table_Value3" in dictionary 5261, Dictionary, "Table", "Field", l_field, l_status, 1 {Index},
    "Key1", l_key1,
    "Key2", l_key2,
    "Key3", l_key3;
if l_status = OKAY then
    warning str(l_field);
end if;
```
This call is used to update the value of a field located in any table in any dictionary using an index containing three fields. You can specify whether the creation of a new table record is allowed.

All table and field names need to be the technical names and surrounded by single quotes if they contain a space. The status returned will contain the number of errors that occurred, a value of OKAY (zero) means the call was successful. The Key Name fields need to contain the technical names of the segment fields of the index being used.

The parameter list for this call is:

- `IN_Prod_ID`: integer
- `IN_Table_Name`: string
- `IN_Field_Name`: string
- `IN_Field_Value`: anonymous field
- `OUT_Status`: integer
- `IN_Index`: integer
- `IN_Allow_Add`: boolean
- `IN_Key_Name1`: string
- `IN_Key_Value1`: anonymous field
- `IN_Key_Name2`: string
- `IN_Key_Value2`: anonymous field
- `IN_Key_Name3`: string
- `IN_Key_Value3`: anonymous field

An example script is:

```
local integer l_status;
local string l_field;
local string l_key1, l_key2, l_key3;
l_key1 = "Value1";
l_key2 = "Value2";
l_key3 = "Value3";
l_field = "Value"
call with name "MBS_Set_Table_Value3" in dictionary 5261, Dictionary, "Table", "Field", l_field, l_status, 1 {Index}, true {allow add},
   "Key1", l_key1,
   "Key2", l_key2,
   "Key3", l_key3;
if l_status <> OKAY then
   warning str(l_status);
end if;
```
MBS_Get_Table_Value4

This call is used to obtain the value of a field located in any table in any dictionary using an index containing four fields.

All table and field names need to be the technical names and surrounded by single quotes if they contain a space. The status returned will contain the number of errors that occurred, a value of OKAY (zero) means the call was successful. The Key Name fields need to contain the technical names of the segment fields of the index being used.

The parameter list for this call is:

- in integer IN_Prod_ID;
- in string IN_Table_Name;
- in string IN_Field_Name;
- out anonymous field OUT_Field_Value;
- out integer OUT_Status;
- in integer IN_Index;
- in string IN_Key_Name1;
- in anonymous field IN_Key_Value1;
- in string IN_Key_Name2;
- in anonymous field IN_Key_Value2;
- in string IN_Key_Name3;
- in anonymous field IN_Key_Value3;
- in string IN_Key_Name4;
- in anonymous field IN_Key_Value4;

An example script is:

```gp
local integer l_status;
l_local string l_field;
l_local string l_key1, l_key2, l_key3, l_key4;
l_key1 = "Value1";
l_key2 = "Value2";
l_key3 = "Value3";
l_key4 = "Value4";
call with name "MBS_Get_Table_Value4" in dictionary 5261,
Dictionary, "Table", "Field", l_field, l_status, l (Index),
"Key1", l_key1,
"Key2", l_key2,
"Key3", l_key3,
"Key4", l_key4;
if l_status = OKAY then
    warning str(l_field);
end if;
```
MBS_Set_Table_Value4

This call is used to update the value of a field located in any table in any dictionary using an index containing four fields. You can specify whether the creation of a new table record is allowed.

All table and field names need to be the technical names and surrounded by single quotes if they contain a space. The status returned will contain the number of errors that occurred, a value of OKAY (zero) means the call was successful. The Key Name fields need to contain the technical names of the segment fields of the index being used.

The parameter list for this call is:

- in integer IN_Prod_ID;
- in string IN_Table_Name;
- in string IN_Field_Name;
- in anonymous field IN_Field_Value;
- out integer OUT_Status;
- in integer IN_Index;
- in boolean IN_Allow_Add;
- in string IN_Key_Name1;
- in anonymous field IN_Key_Value1;
- in string IN_KEY_Name2;
- in anonymous field IN_KEY_Value2;
- in string IN_KEY_Name3;
- in anonymous field IN_KEY_Value3;
- in string IN_KEY_Name4;
- in anonymous field IN_KEY_Value4;

An example script is:

```
local integer l_status;
local string l_field;
local string l_key1, l_key2, l_key3, l_key4;
l_key1 = "Value1";
l_key2 = "Value2";
l_key3 = "Value3";
l_key4 = "Value4";
l_field = "Value"
call with name "MBS_Set_Table_Value4" in dictionary 5261,
Dictionary, "Table", "Field", l_field, l_status, 1 {Index}, true
{allow add},
    "Key1", l_key1,
    "Key2", l_key2,
    "Key3", l_key3,
    "Key4", l_key4;
if l_status <> OKAY then
    warning str(l_status);
end if;
```
MBS_Runtime_Execute

This call is used to execute Dexterity sanScript in the context of the specified dictionary.

The parameter list for this call is:

\begin{verbatim}
inout text INOUT_Text;
in integer IN_Prod_ID;
out integer OUT_Status;
\end{verbatim}

An example script is:

\begin{verbatim}
local integer l_status;
local text l_text;
local integer l_dict;

clear l_text;
l_text = l_text + "warning ""Hello World"";" + char(13);
l_dict = 0; (Dictionary)
call with name "MBS_Runtime_Execute" in dictionary 5261, l_text,
l_dict, l_status;
if l_status <> OKAY then
  warning l_text;
end if;
\end{verbatim}
MBS_Runtime_Execute_Modified

This call is used to execute Dexterity sanScript in the context of the specified modified dictionary.

This allows Dexterity to reference Modifier added local fields.

The parameter list for this call is:

- inout text INOUT_Text;
- in integer IN_Prod_ID;
- out integer OUT_Status;

An example script is:

```plaintext
local integer l_status;
local text l_text;
local integer l_dict;

clear l_text;
l_text = l_text + "warning ""Hello World"";" + char(13);
l_dict = 0; {Dictionary}
call with name "MBS_Runtime_Execute_Modified" in dictionary 5261,
l_text, l_dict, l_status;
if l_status <> OKAY then
    warning l_text;
end if;
```
**MBS_Runtime_Execute_Background**

This call is used to execute Dexterity sanScript in the context of the specified dictionary after any background processes by adding it to the background queue.

The parameter list for this call is:

- `inout text INOUT_Text;`
- `in integer IN_Prod_ID;`
- `out integer OUT_Status;`

An example script is:

```plaintext
local integer l_status;
local text l_text;
local integer l_dict;

clear l_text;
l_text = l_text + "warning ""Hello World"";" + char(13);
l_dict = 0; (Dictionary)
call with name "MBS_Runtime_Execute_Background" in dictionary 5261,
l_text, l_dict, l_status;
if l_status <> OKAY then
    warning l_text;
end if;
```
**MBS_Runtime_Execute_Delayed**

This call is used to execute Dexterity sanScript in the context of the specified dictionary after all foreground scripts have completed by running it delayed.

The parameter list for this call is:

```plaintext
inout text INOUT_Text;
in integer IN_Prod_ID;
out integer OUT_Status;
```

An example script is:

```plaintext
local integer l_status;
local text l_text;
local integer l_dict;

clear l_text;
l_text = l_text + "warning ""Hello World"";" + char(13);
l_dict = 0; {Dictionary}
call with name "MBS_Runtime_Execute_Delayed" in dictionary 5261,
l_text, l_dict, l_status;
if l_status <> OKAY then
  warning l_text;
end if;
```
MBS_SQL_Check_Exists

This call is used to execute a SQL Select statement in the context of the current company database and indicate whether any data records were returned.

The text field returned will contain the error message, or the number of records returned with or without data depending on the options passed in.

The parameter list for this call is:

\[
\text{inout text INOUT_TSQL;} \\
\text{in boolean IN_Return_Data;} \\
\text{in boolean IN_Return_Columns;} \\
\text{out integer OUT_Status;} \\
\]

An example script is:

\[
\text{local integer l_status;} \\
\text{local text l_text;} \\
\text{clear l_text;} \\
\text{l_text = l_text + "select * from table" + char(13);} \\
\text{call with name "MBS_SQL_Check_Exists" in dictionary 5261, l_text, true, true, l_status;} \\
\text{case l_status} \\
\hspace{1em} \text{in [OKAY]} \\
\hspace{2em} \text{warning l_text;} \\
\hspace{1em} \text{in [MISSING]} \\
\hspace{2em} \text{warning l_text;} \\
\hspace{1em} \text{else} \\
\hspace{2em} \text{warning l_text;} \\
\text{end case;} \\
\]
MBS_Export_SQL_Query_To_File

This call is used to execute a SQL Select statement in the context of the current company database and export the result set as a text file.

The parameter list for this call is:

- inout text INOUT_Code;
- inout string INOUT_Pathname;
- in boolean IN_Header;
- in boolean IN_Quotes;
- in integer IN_Mode; { 0 - CSV, 1 - Tab, 2 - User Defined }
- in string IN_Delimiter;
- in boolean IN_Append;
- out long OUT_Rows;
- out integer OUT_Status;

An example script is:

```plaintext
local integer l_status;
local text l_text;
local string l_path;
local long l_rows;

clear l_text;
    l_text = l_text + "select * from table" + char(13);
call with name "MBS_Export_SQL_Query_To_File" in dictionary 5261,
    l_text, l_path, true {Header}, true {Quotes}, 0 {CSV}, ""
    {Delimiter}, false, {Append}, l_rows, l_status;
if l_status = OKAY then
    l_text = str(l_rows) + " rows exported to " + l_path + ".";
    warning l_text;
end if;
```
MBS_SQL_Results

This call is used to execute a SQL Select statement in the context of the current company database and display the results returned in a SQL Results window. The results can be exported from this window if desired.

The parameter list for this call is:

    inout text INOUT_TSQL;

An example script is:

    local text l_text;
    clear l_text;
    l_text = l_text + "select * from table" + char(13);
    call with name "MBS_SQL_Results" in dictionary 5261, l_text;
MBS_SQL_Results_Goto

This call is used to execute a SQL Select statement in the context of the current company database and display the results returned in a SQL Results window. The results can be exported from this window if desired or further actions can be started using the SQL Gotos.

The parameter list for this call is:

- inout text INOUT_TSQL;
- in string IN_ScriptID;

An example script is:

local text l_text;

clear l_text;
l_text = l_text + "select * from table" + char(13);
call with name "MBS_SQL_Results" in dictionary 5261, l_text, "ScriptID";
MBS_SQL_Goto_Get_Data

This call is used in a Runtime Execute Setup script to retrieve data from a SQL result set for use with SQL Gotos.

The parameter list for this call is:

- in integer IN_Window;
- in long IN_Position;
- in string IN_Column;
- in integer IN_Type; { 1 = string, 2 = long, 3 = currency, 4 = date, 5 = time }
- out anonymous field OUT_Field;

An example script is:

```plaintext
in integer MBS_SQLGotoWindow;
in long MBS_SQLGotoCount;
local long MBS_SQLGotoPos;
local string MBS_SQLGotoValue1;
local string MBS_Message;

if MBS_SQLGotoCount > 0 then
    for MBS_SQLGotoPos = 1 to MBS_SQLGotoCount do
        call with name "MBS_SQL_Goto_Get_Data" in dictionary 5261,
            MBS_SQLGotoWindow, MBS_SQLGotoPos, "Customer ID" { Column Label },
            1 { Column Datatype: 1 = string, 2 = long, 3 = currency, 4 = date, 5 = time },
            MBS_SQLGotoValue1;
    end for;
end if;
```
MBS_Net_Execute

This call is used to execute .Net scripts (Visual C# or Visual Basic.Net).

The parameter list for this call is:

in integer IN_Mode;
inout text INOUT_References;
inout text INOUT_Script;
inout text INOUT_Results;
out boolean OUT_Success;

An example Visual C# script is:

```csharp
local integer MBS_Mode;
local text MBS_References;
local text MBS_Script;
local text MBS_Results;
local boolean MBS_Success;

MBS_Success = false;
MBS_Mode = 1; {1 for C#, 2 for VB}

clear MBS_Script;
MBS_Script = MBS_Script + "using System.Windows.Forms;" + char(13);
MBS_Script = MBS_Script + "using Microsoft.Dexterity.Bridge;" + char(13);
MBS_Script = MBS_Script + "using Microsoft.Dexterity.Applications;" + char(13);
MBS_Script = MBS_Script + "using Microsoft.Dexterity.Applications.DynamicsDictionary;" + char(13);
MBS_Script = MBS_Script + "using Microsoft.Dexterity.Applications.GpPowerToolsDictionary;" + char(13);
MBS_Script = MBS_Script + "namespace NetExecute" + char(13);
MBS_Script = MBS_Script + "{" + char(13);
MBS_Script = MBS_Script + "    public class Program" + char(13);
MBS_Script = MBS_Script + "    {" + char(13);
MBS_Script = MBS_Script + "        public void Run()" + char(13);
MBS_Script = MBS_Script + "        {" + char(13);
MBS_Script = MBS_Script + "            MessageBox.Show("Hello from C");" + char(13);
MBS_Script = MBS_Script + "        }" + char(13);
MBS_Script = MBS_Script + "    }" + char(13);
MBS_Script = MBS_Script + "}" + char(13);
MBS_Script = MBS_Script + "}" + char(13);

clear MBS_References;
MBS_References = MBS_References + "System.Windows.Forms.dll" + char(13);
MBS_References = MBS_References + "..\Application.Dynamics.dll" + char(13);
MBS_References = MBS_References + "..\Microsoft.Dexterity.Bridge.dll" + char(13);
MBS_References = MBS_References + "..\Microsoft.Dexterity.Shell.dll" + char(13);
MBS_References = MBS_References + "..\Application.GpPowerTools.dll" + char(13);

call with name "MBS_Net_Execute" in dictionary 5261,
    MBS_Mode, MBS_References, MBS_Script, MBS_Results, MBS_Success;
if not MBS_Success then
    warning MBS_Results;
end if;
```
An example Visual Basic.Net script is:

```vbnet
local integer MBS_Mode;
local text MBS_References;
local text MBS_Script;
local text MBS_Results;
local boolean MBS_Success;

MBS_Success = false;
MBS_Mode = 2; {1 for C#, 2 for VB}

clear MBS_Script;
MBS_Script = MBS_Script + "Imports System.Windows.Forms" + char(13);
MBS_Script = MBS_Script + "Imports Microsoft.VisualBasic" + char(13);
MBS_Script = MBS_Script + "Imports Microsoft.Dexterity.Bridge" + char(13);
MBS_Script = MBS_Script + "Imports Microsoft.Dexterity.Applications" + char(13);
MBS_Script = MBS_Script + "Imports Microsoft.Dexterity.Applications.DynamicsDictionary" + char(13);
MBS_Script = MBS_Script + "Imports Microsoft.Dexterity.Applications.GpPowerToolsDictionary" + char(13);

MBS_Script = MBS_Script + "Namespace NetExecute" + char(13);
MBS_Script = MBS_Script + "    Class Program" + char(13);
MBS_Script = MBS_Script + "        Public Function Run() As Object" + char(13);
MBS_Script = MBS_Script + "            MessageBox.Show("Hello from VB.Net")" + char(13);
MBS_Script = MBS_Script + "        End Function" + char(13);
MBS_Script = MBS_Script + "    End Class" + char(13);
MBS_Script = MBS_Script + "End Namespace" + char(13);

clear MBS_References;
MBS_References = MBS_References + "System.Windows.Forms.dll" + char(13);
MBS_References = MBS_References + "..\Application.Dynamics.dll" + char(13);
MBS_References = MBS_References + "..\Microsoft.Dexterity.Bridge.dll" + char(13);
MBS_References = MBS_References + "..\Microsoft.Dexterity.Shell.dll" + char(13);
MBS_References = MBS_References + "..\Application.GpPowerTools.dll" + char(13);

call with name "MBS_Net_Execute" in dictionary 5261,
    MBS_Mode, MBS_References, MBS_Script, MBS_Results, MBS_Success;
if not MBS_Success then
    warning MBS_Results;
end if;
```
MBS_Script_Load_Dex

This call is used to load a Dexterity sanScript script from a Runtime Execute Setup Script ID. It is designed to be used with the MBS_Runtime_Execute Helper Function.

The parameter list for this call is:

in string IN_ScriptID;
inout text INOUT_Text;
inout integer INOUT_Dict;

An example script is:

local text l_text;
local integer l_dict;

call with name "MBS_Script_Load_Dex" in dictionary 5261, "XXXX", l_text, l_dict;
MBS_Script_Load_SQL

This call is used to load a SQL script from a SQL Execute Setup Script ID. It is designed to be used with the MBS_SQL_Check_Exists Helper Function.

The parameter list for this call is:

- in string IN_ScriptID;
- inout text INOUT_Text;

An example script is:

```plaintext
local text l_text;

call with name "MBS_Script_Load_SQL" in dictionary 5261, "XXXX", l_text;
```
MBS_Script_Load_Net

This call is used to load a Visual C# or Visual Basic.Net script from a .Net Execute Setup Script ID. It is designed to be used with the MBS.Net_Execute Helper Function.

The parameter list for this call is:

in string IN_ScriptID;
out integer IN_Mode;
inout text INOUT_References;
inout text INOUT_Script;

An example script is:

local integer MBS_Mode;
local text MBS_References;
local text MBS_Script;

call with name "MBS_Script_Load_Net" in dictionary 5261, "XXXX", MBS_Mode, MBS_References, MBS_Script;
MBS_Param_Set

This call is used to store a value in the DUOS SY_User_Object_Store (SY90000) table which can then be read by another script. It is designed to be used with the MBS_Runtime_Execute and MBS_Param_Get Helper Functions as a method of passing parameters.

The parameter list for this call is:

```plaintext
in string IN_Parameter;
in string IN_Value;
```

An example script is:

```plaintext
local string l_string;

l_string = "Value";
call with name "MBS_Param_Set" in dictionary 5261, "Variable", l_string;
```
MBS_Param_Get

This call is used to read a previously set value from the DUOS SY_User_Object_Store (SY90000) table. It is designed to be used with the MBS_Runtime_Execute and MBS_Param_Set Helper Functions as a method of passing parameters.

The parameter list for this call is:

```plaintext
in string IN_Parameter;
out string OUT_Value;
```

An example script is:

```plaintext
local string l_string;

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

This call is used to remove a previously set value from the DUOS SY_User_Object_Store (SY90000) table. It is designed to be used with the MBS_Runtime_Execute and MBS_Param_Set Helper Functions as a method of passing parameters.

The parameter list for this call is:

in string IN_Parameter;

An example script is:

call with name "MBS_Param_Del" in dictionary 5261, "Variable";
MBS_Param_DelAll

This call is used to remove all previously stored parameter values for the current user from the DUOS SY_User_Object_Store (SY90000) table. It is designed to be used with the MBS_Runtime_Execute and MBS_Param_Set Helper Functions.

There is no parameter list for this call.

An example script is:

call with name "MBS_Param_DelAll" in dictionary 5261;
MBS_Auto_Log

This call is used to add a message into the GP Power Tools log file. It is designed to be used with the Automatic Trigger Mode to record additional information when a trigger fires.

*Using this Helper Function within a Trigger Script will write to the GP Power Tools log file and also be recorded in the email body if the trigger is set to send an email.*

The parameter list for this call is:

in string IN_Message;

An example script is:

call with name "MBS_Auto_Log" in dictionary 5261, "Message";
MBS_Trigger_Start

This call is used to activate an Automatic Trigger Mode Trigger and is designed to be used with Non-logging triggers in the Automatic Trigger Mode.

The parameter list for this call is:

in string IN_TriggerID;

An example script is:

call with name "MBS_Trigger_Start" in dictionary 5261, "XXXX";
MBS_Trigger_Stop

This call is used to deactivate an Automatic Trigger Mode Trigger and is designed to be used with Non-logging triggers in the Automatic Trigger Mode.

The parameter list for this call is:

\[
\text{in string } \text{IN}_{-}\text{TriggerID};
\]

An example script is:

\[
call \text{ with name } \text{"MBS}_{-}\text{Trigger}_{-}\text{Stop" in dictionary 5261, } \text{"XXXX"};
\]
MBS_Logging_Start

This call is used to programmatically start Manual Logging Mode and is designed to be used with Non-logging triggers in the Automatic Trigger Mode.

There are no parameters for this call.

An example script is:

```
call with name "MBS_Logging_Start" in dictionary 5261;
```
MBS_Logging_Stop

This call is used to programmatically stop Manual Logging Mode and is designed to be used with Non-logging triggers in the Automatic Trigger Mode.

There are no parameters for this call.

An example script is:

call with name "MBS_Logging_Stop" in dictionary 5261;
**MBS_DUOS_Set**

This call is used to store a value in the DUOS SY_User_Object_Store (SY90000). It is designed to be used with the MBS_DUOS_Get Helper Function.

The parameter list for this call is:

```plaintext
in string IN_Object;
in string IN_ID;
in string IN_Property;
in string IN_Value;
```

An example script is:

```plaintext
local string l_string;
local string l_object;

l_object = "ID";
l_string = "Value";
call with name "MBS_DUOS_Set" in dictionary 5261, "Object", l_object, "Property", l_string;
```
MBS_DUOS_Get

This call is used to read a previously set value from the DUOS SY_User_Object_Store (SY90000) table. It is designed to be used with the MBS_DUOS_Set Helper Function.

The parameter list for this call is:

   in string IN_Object;
   in string IN_ID;
   in string IN_Property;
   out string OUT_Value;

An example script is:

   local string l_string;
   local string l_object;

   l_object = "ID";
   call with name "MBS_DUOS_Get" in dictionary 5261, "Object",
   l_object, "Property", l_string;
**MBS_DUOS_Del**

This call is used to remove a previously set value from the DUOS SY_User_Object_Store (SY90000) table. It is designed to be used with the MBS_DUOS_Set Helper Function.

The parameter list for this call is:

```plaintext
in string IN_Object;
in string IN_ID;
in string IN_Property;
```

An example script is:

```plaintext
local string l_object;

l_object = "ID";
call with name "MBS_DUOS_Del" in dictionary 5261, "Object",
l_object, "Property";
```
MBS_DUOS_DelAll

This call is used to remove all previously stored values for an object from the DUOS SY_User_Object_Store (SY90000) table. It is designed to be used with the MBS_DUOS_Set Helper Function.

The parameter list for this call is:

- in string IN_Object;
- in string IN_ID;

An example script is:

```plaintext
l_object = "ID";
call with name "MBS_DUOS_DelAll" in dictionary 5261, "Object",
l_object;
```
MBS_SQL_Lookup

This call is used to open a lookup and return the selected value to a field. It uses the Custom SQL Lookup from Parameter Lists. You need to provide a SQL Execute Setup script which returns a query with three string columns; an ID string, a Description string and a string to be returned (usually the same as the ID value). The other parameters are the seed value and return field (usually the same window field).

The parameter list for this call is:

    in string IN_Script_ID;
    in string IN_Seed_Value;
    inout anonymous INOUT_Return_Field;

An example script is:

    local string l_string;

    call with name "MBS_SQL_Lookup" in dictionary 5261, "XXXX",
    l_string, Return_Field;
**MBS_SQL_Lookup_Parameter**

This call is used to open a lookup and return the selected value to a field. It uses the Custom SQL Lookup from Parameter Lists. You need to provide a Parameter List Parameter ID and the Position for parameter that is set up for a Custom Lookup (SQL) mode. The other parameters are the seed value and return field (usually the same window field).

The parameter list for this call is:

- in string IN_Parameter_ID;
- in integer IN_Position;
- in string IN_Seed_Value;
- inout anonymous INOUT_Return_Field;

An example script is:

```plaintext
local string l_string;

call with name "MBS_SQL_Lookup_Parameter" in dictionary 5261, "XXXX", Y, l_string, Return_Field;
```
MBS_Form_Lookup

This call is used to open a lookup and return the selected value to a field. It uses the Custom Form Lookup from Parameter Lists. You need to define the form, window and field information required to drive an existing lookup form in any dictionary installed in Microsoft Dynamics GP. The other parameters are the seed value and return field (usually the same window field).

The parameter list for this call is:

```plaintext
in integer IN_Dict;
in string IN_Form;
in string IN_Window;
in string IN_Field;
in string IN_FieldSortBy;
in string IN_WindowScroll;
in string IN_FieldScroll;
in string IN_Seed_Value;
inout anonymous INOUT_Return_Field;
```

An example script is:

```plaintext
local string l_string;
call with name "MBS_Form_Lookup" in dictionary 5261, Dict, Form, Window, Field, FieldSortBy, WindowScroll, FieldScroll, l_string, Return_Field;
```
MBS_Form_Lookup_Parameter

This call is used to open a lookup and return the selected value to a field. It uses the Custom Form Lookup from Parameter Lists. You need to provide a Parameter List Parameter ID and the Position for parameter that is set up for a Custom Lookup (Form) mode. The other parameters are the seed value and return field (usually the same window field).

The parameter list for this call is:

in string IN_Parameter_ID;
in integer IN_Position;
in string IN_Seed_Value;
inout anonymous INOUT_Return_Field;

An example script is:

local string l_string;
call with name "MBS_Form_Lookup_Parameter" in dictionary 5261, "XXXX", Y, l_string, Return_Field;
MBS_Project_Start

This call is used to activate all Automatic Trigger Mode Triggers belonging to the specified Project ID. It is designed to be used with Non-logging triggers in the Automatic Trigger Mode.

The parameter list for this call is:

in string IN_ProjectID;

An example script is:

call with name "MBS_Project_Start" in dictionary 5261, "XXXX";
MBS_Project_Stop

This call is used to deactivate all Automatic Trigger Mode Triggers belonging to the specified Project ID. It is designed to be used with Non-logging triggers in the Automatic Trigger Mode.

The parameter list for this call is:

in string IN_ProjectID;

An example script is:

call with name "MBS_Project_Stop" in dictionary 5261, "XXXX";
MBS_Script_Substitute

Use this call to replace all instances of a placeholder in a script with a value in script. It can be used to manually perform Parameter List substitution. Use this function with the MBS_Parameter_String function and MBS_Parameter_Number, MBS_Parameter_Currency, MBS_Parameter_Boolean, MBS_Parameter_Date and MBS_Parameter_Time functions.

The parameter list for this call is:

inout text INOUT_Text;
in string IN_Placeholder;
in string IN_Value;

An example script is:

local text MBS_Text_Field;
local string MBS_Placeholder;
local string MBS_Value;

call with name "MBS_Script_Substitute" in dictionary 5261,
   MBS_Text_Field, MBS_Placeholder, MBS_Value;
MBS_Parameter_Placeholder

Use this call to obtain the Parameter List placeholder to use with the MBS_Script_Substitute function.

The parameter list for this call is:

in integer IN_Type; { 1 = String, 2 = Integer, 3 = Currency, 4 = Boolean, 5 = Date, 6 = Time }
in integer IN_Position;
in integer IN_From; { 0 = From, 1 = To }
in integer IN.Language; { 1 = Dex, 2 = SQL, 3 = C#, 4 = VB }
out string OUT_String;

An example script is:

local string MBS_Placeholder;

call with name "MBS_Parameter_Placeholder" in dictionary 5261, 1 {Type: String}, 1 {Position}, 0 {FromTo: From/Single}, 1 {Language: Dexterity sanScript}, MBS_Placeholder;
**MBS_Parameter_String**

Use this call to obtain the string representation of a string value to use with the MBS_Script_Substitute function.

The parameter list for this call is:

```plaintext
in string IN_Value;
in integer IN_Language; { 1 = Dex, 2 = SQL, 3 = C#, 4 = VB }
out string OUT_String;
```

An example script is:

```plaintext
local string MBS_Value;
local string MBS_Value_String;

MBS_Value_String = <Variable>;
call with name "MBS_Parameter_String" in dictionary 5261,
    MBS_Value_String, 1 {Language: Dexterity sanScript}, MBS_Value;
```
**MBS_Parameter_Number**

Use this call to obtain the string representation of a number value to use with the MBS_Script_Substitute function.

The parameter list for this call is:

```plaintext
in long IN_Value;
in integer IN_Language; { 1 = Dex, 2 = SQL, 3 = C#, 4 = VB }
out string OUT_String;
```

An example script is:

```plaintext
local string MBS_Value;
local long MBS_Value_Number;

MBS_Value_Number = <Variable>;
call with name "MBS_Parameter_Number" in dictionary 5261,
   MBS_Value_Number, 1 {Language: Dexterity sanScript}, MBS_Value;
```
MBS_Parameter_Currency

Use this call to obtain the string representation of a currency value to use with the MBS_Script_Substitute function.

The parameter list for this call is:

in currency IN_Value;
in integer IN_Language; { 1 = Dex, 2 = SQL, 3 = C#, 4 = VB };
out string OUT_String;

An example script is:

local string MBS_Value;
local currency MBS_Value_Currency;

MBS_Value_Currency = <Variable>;
call with name "MBS_Parameter_Currency" in dictionary 5261,
    MBS_Value_Currency, 1 {Language: Dexterity sanScript}, MBS_Value;
MBS_Parameter_Boolean

Use this call to obtain the string representation of a boolean value to use with the MBS_Script_Substitute function.

The parameter list for this call is:

\[
\begin{align*}
\text{in boolean IN_Value;} \\
\text{in integer IN_Language; \{ 1 = Dex, 2 = SQL, 3 = C#, 4 = VB \}} \\
\text{out string OUT_String;}
\end{align*}
\]

An example script is:

\[
\begin{align*}
\text{local string MBS_Value;} \\
\text{local boolean MBS_Value_Boolean;} \\
MBS_Value_Boolean = \langle \text{Variable}\rangle; \\
call with name "MBS_Parameter_Boolean" in dictionary 5261, \\
\quad MBS_Value_Boolean, 1 \{\text{Language: Dexterity sanScript}\}, MBS_Value;
\end{align*}
\]
MBS_Parameter_Date

Use this call to obtain the string representation of a date value to use with the MBS_Script_Substitute function.

The parameter list for this call is:

in date IN_Value;
in integer IN_Language; { 1 = Dex, 2 = SQL, 3 = C#, 4 = VB }
out string OUT_String;

An example script is:

local string MBS_Value;
local date MBS_Value_Date;

MBS_Value_Date = <Variable>;
call with name "MBS_Parameter_Date" in dictionary 5261,
    MBS_Value_Date, 1 {Language: Dexterity sanScript}, MBS_Value;
**MBS_Parameter_Time**

Use this call to obtain the string representation of a time value to use with the MBS_Script_Substitute function.

The parameter list for this call is:

```
in time IN_Value;
in integer IN_Language; { 1 = Dex, 2 = SQL, 3 = C#, 4 = VB }
out string OUT_String;
```

An example script is:

```
local string MBS_Value;
local time MBS_Value_Time;

MBS_Value_Time = <Variable>;
call with name "MBS_Parameter_Time" in dictionary 5261,
   MBS_Value_Time, 1 (Language: Dexterity sanScript), MBS_Value;
```
MBS_Parameter_Load

Use this call to a Parameter List Parameter ID with its default values. Use this command before using the MBS_Parameter_Set_String, MBS_Parameter_Set_Number, MBS_Parameter_Set_Boolean, MBS_Parameter_Set_Date, MBS_Parameter_Set_Time, MBS_Parameter_Get_Number, MBS_Parameter_Get_Boolean, MBS_Parameter_Get_Date, and MBS_Parameter_Get_Time functions.

The parameter list for this call is:

in string IN_ParameterID;
out integer OUT_Status;

An example script is:

local integer MBS_Status;

call with name "MBS_Parameter_Load" in dictionary 5261, "XXXX", MBS_Status;
MBS_Parameter_Set_String

Use this call to set the value of a string parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

- in integer IN_Position;
- in integer IN_From; { 0 = From, 1 = To }
- in string IN_Value;

An example script is:

```
local string MBS_Value_String;

MBS_Value_String = <Variable>;
call with name "MBS_Parameter_Set_String" in dictionary 5261,
  1 (Position: "XXXX"), 0 {FromTo: From/Single}, MBS_Value_String;
```
MBS_Parameter_Set_Number

Use this call to set the value of a number parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

- **IN_Position**: integer
- **IN_From**: integer (0 = From, 1 = To)
- **IN_Value**: long

An example script is:

```plaintext
local long MBS_Value_Number;
MBS_Value_Number = <Variable>;
call with name "MBS_Parameter_Set_Number" in dictionary 5261,
    1 {Position: "XXXX"}, 0 {FromTo: From|Single}, MBS_Value_Number;
```
MBS_Parameter_Set_Currency

Use this call to set the value of a currency parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

- in integer IN_Position;
- in integer IN_From; { 0 = From, 1 = To }
- in currency IN_Value;

An example script is:

```gp
local currency MBS_Value_Currency;

MBS_Value_Currency = <Variable>;
call with name "MBS_Parameter_Set_Currency" in dictionary 5261,
   1 {Position: "XXXX"}, 0 {FromTo: From/Single},
   MBS_Value_Currency;
```
MBS_Parameter_Set_Boolean

Use this call to set the value of a boolean parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

- in integer IN_Position;
- in integer IN_From; { 0 = From, 1 = To }
- in boolean IN_Value;

An example script is:

```plaintext
local boolean MBS_Value_Boolean;
MBS_Value_Boolean = <Variable>;
call with name "MBS_Parameter_Set_Boolean" in dictionary 5261,
   1 {Position: "XXXX"}, 0 {FromTo: From/Single}, MBS_Value_Boolean;
```
MBS_Parameter_Set_Date

Use this call to set the value of a date parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

- in integer IN_Position;
- in integer IN_From; { 0 = From, 1 = To }
- in date IN_Value;

An example script is:

```plaintext
local date MBS_Value_Date;
MBS_Value_Date = <Variable>;
call with name "MBS_Parameter_Set_Date" in dictionary 5261,
   1 (Position: "XXXX"), 0 {FromTo: From/Single}, MBS_Value_Date;
```
MBS_Parameter_Set_Time

Use this call to set the value of a time parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

in integer IN_Position;
in integer IN_From; { 0 = From, 1 = To }
in time IN_Value;

An example script is:

local time MBS_Value_Time;
MBS_Value_Time = <Variable>;
call with name "MBS_Parameter_Set_Time" in dictionary 5261,
   1 {Position: "XXXX"}, 0 {FromTo: From/Single}, MBS_Value_Time;
MBS_Parameter_Get_String

Use this call to get the value of a string parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

\[
\begin{align*}
\text{in integer IN\_Position;} \\
\text{in integer IN\_From; \{ 0 = From, 1 = To \}} \\
\text{out string OUT\_Value;} \\
\end{align*}
\]

An example script is:

local string MBS\_Value\_String;

call with name "MBS\_Parameter\_Get\_String" in dictionary 5261, 
  1 \{Position: "XXXX"\}, 0 \{FromTo: From/Single\}, MBS\_Value\_String;
**MBS_Parameter_Get_Number**

Use this call to get the value of a number parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

```plaintext
in integer IN_Position;
in integer IN_From; { 0 = From, 1 = To }
out long OUT_Value;
```

An example script is:

```plaintext
local long MBS_Value_Number;
call with name "MBS_Parameter_Get_Number" in dictionary 5261,
   1 {Position: "XXXX"}, 0 {FromTo: From/Single}, MBS_Value_Number;
```
**MBS_Parameter_Get_Currency**

Use this call to get the value of a currency parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

- in integer IN_Position;
- in integer IN_From; { 0 = From, 1 = To }
- out currency OUT_Value;

An example script is:

```plaintext
local currency MBS_Value_Currency;

call with name "MBS_Parameter_Get_Currency" in dictionary 5261,
    1 {Position: "XXXX"}, 0 {FromTo: From/Single},
    MBS_Value_Currency;
```
**MBS_Parameter_Get_Boolean**

Use this call to get the value of a boolean parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

```plaintext
in integer IN_Position;
in integer IN_From; { 0 = From, 1 = To }  
out boolean OUT_Value;
```

An example script is:

```plaintext
local boolean MBS_Value_Boolean;

call with name "MBS_Parameter_Get_Boolean" in dictionary 5261,  
1 {Position: "XXXX"}, 0 {FromTo: From/Single}, MBS_Value_Boolean;
```
MBS_Parameter_Get_Date

Use this call to get the value of a date parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

\[
\begin{align*}
\text{in integer IN\_Position;} & \\
\text{in integer IN\_From; \{ 0 = From, 1 = To \}} & \\
\text{out date OUT\_Value;} &
\end{align*}
\]

An example script is:

\[
\begin{align*}
\text{local date MBS\_Value\_Date;} & \\
\text{call with name "MBS\_Parameter\_Get\_Date" in dictionary 5261,} & \\
\quad 1 \{\text{Position: "XXXX"}\}, 0 \{\text{FromTo: From/Single}\}, \text{MBS\_Value\_Date;} &
\end{align*}
\]
MBS_Parameter_Get_Time

Use this call to get the value of a time parameter into the parameter list memory. The MBS_Parameter_Load must be executed before using this function to initialize the parameter list in memory.

The parameter list for this call is:

```
in integer IN_Position;
in integer IN_From; { 0 = From, 1 = To }
out time OUT_Value;
```

An example script is:

```
local time MBS_Value_Time;

call with name "MBS_Parameter_Get_Time" in dictionary 5261,
   1 {Position: "XXXX"}, 0 {FromTo: From/Single}, MBS_Value_Time;
```


**MBS_getmsg**

Use this call to get a message resource from any installed dictionary.

The parameter list for this call is:

```plaintext
in integer IN_DictID;
in integer IN_ID;
out string OUT_Message;
```

An example script is:

```plaintext
local string l_message;

call with name "MBS_getmsg" in dictionary 5261, 0, 1765,
   l_message;
warning l_message;
```
MBS_Token

Use this call break a character separated string into individual string values.

The parameter list for this call is:

in string IN_string;
in string IN_token;
in integer IN_position;
out string OUT_string;

An example script is:

local string l_parameters;
local string l_value;

call with name "MBS_Token" in dictionary 5261,
    l_parameters, CH_COMMA, l_value;
MBS_Field_ParseText

Use this call to break a text field into lines, similar to Field_ParseText() Dexterity function, but can return text variables longer than 255 characters.

The parameter list for this call is:

```plaintext
in text IN_Text;
in integer IN_Characters;
inout integer INOUT_Position;
out text OUT_Text;
```

An example script is:

```plaintext
inout text MBS_InText;
local integer MBS_StartPos;
local text MBS_Text;

repeat
    call with name "MBS_Field_ParseText" in dictionary 5261,
        MBS_InText, 32767, MBS_StartPos, MBS_Text;
until MBS_StartPos = 0;
```
MBS_Security_Form_Check

Use this call to manually check security to identify if additional fields are available on an alternate or modified window.

The parameter list for this call is:

in integer dictid;
in string resname;
out boolean access;
out integer altdictid;
out boolean modified;

An example script is:

local boolean access, modified;
local integer altdictid;

call with name "MBS_Security_Form_Check" in dictionary 5261,
    DYNAMICS, "PM_Vendor_Maintenance", access, altdictid, modified;
if access and altdictid = DYNAMICS and modified then
    end if;

Chapter 9: RW Functions

GP Power Tools has support for six generic Report Writer functions in the core Dynamics.dic dictionary which can be used in the Report Writer with any report as a user defined function in a calculated field.

To use the functions with GP Power Tools, the first two parameters passed in for each of the functions will be the Dictionary ID for GP Power Tools (5261) and the Script ID of a Runtime Execute Setup script.

The Dexterity sanScript code contained in the Script ID will then be executed allowing for the development of custom Report Writer functions. Use the Select Custom Script Purpose option on the Runtime Execute Setup window to automatically add the template code to handle the parameter passing into and out of the script using the MBS_Param_Get and MBS_Param_Set helper functions.

Below are the details of the RW Functions available from the system series:

- `rw_ReportStart`
- `rw_ReportEnd`
- `rw_TableHeaderString`
- `rw_TableHeaderCurrency`
- `rw_TableLineString`
- `rw_TableLineCurrency`

You can use these functions to capture information off a report and store it in the log using the MBS_Auto_Log helper function. For example: you can capture the values of legends, calculated fields, report fields or values from any of the attached tables.

While the Report Writer Functions were designed to work with the report start and end events and with a header and line type document such as seen in Sales Order Processing, you can use the functions and parameters as desired to achieve the results required.

More detail on these functions is available from Knowledge Base (KB) article 888884:

http://support.microsoft.com/kb/888884
rw_ReportStart

This report writer function can be used in the Report Writer as a user defined function in a calculated field. The first two parameters passed in need to be the Dictionary ID for GP Power Tools (5261) and the Script ID of the Runtime Execute Setup script to be executed.

The returned value for this report writer function is a `string` and the input parameter list for this report writer function is:

\[
\begin{align*}
\text{in integer } & \text{dict_id; \{Dictionary ID\}} \\
\text{in string } & \text{script_id; \{Script ID\}}
\end{align*}
\]

An example of how it would be called from the Report Writer for a calculated field with a Result Type of string is:

```plaintext
FUNCTION_SCRIPT( rw_ReportStart 5261 "Script ID")
```

The template Runtime Execute Setup script added by the Helper Function Assistant window is:

```plaintext
local string MBS_Status;
MBS_Status = "";

{ Add your code below here }

{ Add your code above here }

call with name "MBS_Param_Set" in dictionary 5261, "ReportStart", MBS_Status;
```
**rw_ReportEnd**

This report writer function can be used in the Report Writer as a user defined function in a calculated field. The first two parameters passed in need to be the Dictionary ID for GP Power Tools (5261) and the Script ID of the Runtime Execute Setup script to be executed.

The returned value for this report writer function is a string and the input parameter list for this report writer function is:

```
in integer dict_id; {Dictionary ID}
in string script_id; {Script ID}
```

An example of how it would be called from the Report Writer for a calculated field with a Result Type of string is:

```
FUNCTION_SCRIPT( rw_ReportEnd 5261 "Script ID"
)
```

The template Runtime Execute Setup script added by the Helper Function Assistant window is:

```
local string MBS_Status;

MBS_Status = "";

{ Add your code below here }

{ Add your code above here }

call with name "MBS_Param_Set" in dictionary 5261, "ReportEnd", MBS_Status;
```
rw_TableHeaderString

This report writer function can be used in the Report Writer as a user defined function in a calculated field. The first two parameters passed in need to be the Dictionary ID for GP Power Tools (5261) and the Script ID of the Runtime Execute Setup script to be executed.

The returned value for this report writer function is a string and the input parameter list for this report writer function is:

in integer dict_id; {Dictionary ID}
in string script_id; {Script ID}
in string sNumber; {control field 1}
in integer sType; {control field 2}
in integer iControl; {which piece of data to return}

An example of how it would be called from the Report Writer for a calculated field with a Result Type of string is:

FUNCTION_SCRIPT( rw_TableHeaderString 5261 "Script ID"
SOP_HDR_WORK.SOP Number  SOP_HDR_WORK.SOP Type  1 )

The template Runtime Execute Setup script added by the Helper Function Assistant window is:

local string MBS_TableHeaderString;
local string MBS_Number;
local integer MBS_Type;
local integer MBS_Control;
local string MBS_String;

call with name "MBS_Param_Get" in dictionary 5261, "Number", MBS_Number;
call with name "MBS_Param_Get" in dictionary 5261, "Type", MBS_String;
MBS_Type = integer(value(MBS_String));
call with name "MBS_Param_Get" in dictionary 5261, "Control", MBS_String;
MBS_Control = integer(value(MBS_String));
MBS_TableHeaderString = "";

{ Add your code below here }

{ Add your code above here }

call with name "MBS_Param_Set" in dictionary 5261, "TableHeaderString", MBS_TableHeaderString;
**rw_TableHeaderCurrency**

This report writer function can be used in the Report Writer as a user defined function in a calculated field. The first two parameters passed in need to be the Dictionary ID for GP Power Tools (5261) and the Script ID of the Runtime Execute Setup script to be executed.

The returned value for this report writer function is a **currency** and the input parameter list for this report writer function is:

- `in integer dict_id;` {Dictionary ID}
- `in string script_id;` {Script ID}
- `in string sNumber;` {control field 1}
- `in integer sType;` {control field 2}
- `in integer iControl;` {which piece of data to return}

An example of how it would be called from the Report Writer for a calculated field with a Result Type of currency is:

```
FUNCTION_SCRIPT( rw_TableHeaderCurrency 5261 "Script ID"
SOP_HDR_WORK.SOP Number  SOP_HDR_WORK.SOP Type 1 )
```

The template Runtime Execute Setup script added by the Helper Function Assistant window is:

```
local currency MBS_TableHeaderCurrency;
local string MBS_Number;
local integer MBS_Type;
local integer MBS_Control;
local string MBS_String;

call with name "MBS_Param_Get" in dictionary 5261, "Number", MBS_Number;
call with name "MBS_Param_Get" in dictionary 5261, "Type", MBS_String;
MBS_Type - integer(value(MBS_String));
call with name "MBS_Param_Get" in dictionary 5261, "Control", MBS_String;
MBS_Control - integer(value(MBS_String));
MBS_TableHeaderCurrency = 0.0000;

{ Add your code below here }

{ Add your code above here }

call with name "MBS_Param_Set" in dictionary 5261, "TableHeaderCurrency", str(MBS_TableHeaderCurrency);
```
rw_TableLineString

This report writer function can be used in the Report Writer as a user defined function in a calculated field. The first two parameters passed in need to be the Dictionary ID for GP Power Tools (5261) and the Script ID of the Runtime Execute Setup script to be executed.

The returned value for this report writer function is a string and the input parameter list for this report writer function is:

in integer dict_id; {Dictionary ID}
in string script_id; {Script ID}
in string sNumber; {control field 1}
in integer sType; {control field 2}
in currency cSequenceOne; {control field 3}
in currency cSequenceTwo; {control field 4}
in integer iControl; {which piece of data to return}

To use the rw_TableLineString report writer function we need to be able to pass the two sequence fields as currency data type. So to use the Sales Order Processing fields SOP_LINE_WORK.Line Item Sequence and SOP_LINE_WORK.Component Sequence, we will need to create two calculated fields to convert them from a long integer to a currency data type.

Calculated Field (C) Line Item Sequence is defined as result type currency with the expression of SOP_LINE_WORK.Line Item Sequence * 1.00000.

Calculated Field (C) Component Sequence is defined as result type currency with the expression of SOP_LINE_WORK.Component Sequence * 1.00000.

An example of how it would be called from the Report Writer for a calculated field with a Result Type of string is:

FUNCTION_SCRIPT( rw_TableLineString 5261 "Script ID"
SOP_LINE_WORK.SOP Number SOP_LINE_WORK.SOP Type (C) Line Item Sequence (C) Component Sequence 1 )
The template Runtime Execute Setup script added by the Helper Function Assistant window is:

```plaintext
local string MBS_TableLineString;
local string MBS_Number;
local integer MBS_Type;
local currency MBS_SequenceOne;
local currency MBS_SequenceTwo;
local integer MBS_Control;
local string MBS_String;

call with name "MBS_Param_Get" in dictionary 5261, "Number", MBS_Number;
call with name "MBS_Param_Get" in dictionary 5261, "Type", MBS_String;
MBS_Type = integer(value(MBS_String));
call with name "MBS_Param_Get" in dictionary 5261, "SequenceOne", MBS_String;
MBS_SequenceOne = currency(value(MBS_String));
call with name "MBS_Param_Get" in dictionary 5261, "SequenceTwo", MBS_String;
MBS_SequenceTwo = currency(value(MBS_String));
call with name "MBS_Param_Get" in dictionary 5261, "Control", MBS_String;
MBS_Control = integer(value(MBS_String));
MBS_TableLineString = "";

{ Add your code below here }

{ Add your code above here }

call with name "MBS_Param_Set" in dictionary 5261, "TableLineString", MBS_TableLineString;
```
**rw_TableLineCurrency**

This report writer function can be used in the Report Writer as a user defined function in a calculated field. The first two parameters passed in need to be the Dictionary ID for GP Power Tools (5261) and the Script ID of the Runtime Execute Setup script to be executed.

The returned value for this report writer function is a currency and the input parameter list for this report writer function is:

```
in integer dict_id; {Dictionary ID}
in string script_id; {Script ID}
in string sNumber; {control field 1}
in integer sType; {control field 2}
in currency cSequenceOne; {control field 3}
in currency cSequenceTwo; {control field 4}
in integer iControl; {which piece of data to return}
```

To use the `rw_TableLineCurrency` report writer function we need to be able to pass the two sequence fields as currency data type. So to use the Sales Order Processing fields `SOP_LINE_WORK.Line Item Sequence` and `SOP_LINE_WORK.Component Sequence`, we will need to create two calculated fields to convert them from a long integer to a currency data type.

Calculated Field (C) **Line Item Sequence** is defined as result type currency with the expression of `SOP_LINE_WORK.Line Item Sequence * 1.00000`.

Calculated Field (C) **Component Sequence** is defined as result type currency with the expression of `SOP_LINE_WORK.Component Sequence * 1.00000`.

An example of how it would be called from the Report Writer for a calculated field with a Result Type of currency is:

```
FUNCTION_SCRIPT(  rw_TableLineCurrency  5261  "Script ID"
SOP_LINE_WORK.SOP Number  SOP_LINE_WORK.SOP Type  (C) Line Item
Sequence  (C) Component Sequence  1  )
```
The template Runtime Execute Setup script added by the Helper Function Assistant window is:

```plaintext
call with name "MBS_Param_Get" in dictionary 5261, "Number", MBS_Number;
call with name "MBS_Param_Get" in dictionary 5261, "Type", MBS_Type;
MBS_Type = integer(value(MBS_Type));
call with name "MBS_Param_Get" in dictionary 5261, "SequenceOne", MBS_SequenceOne;
MBS_SequenceOne = currency(value(MBS_SequenceOne));
call with name "MBS_Param_Get" in dictionary 5261, "SequenceTwo", MBS_SequenceTwo;
MBS_SequenceTwo = currency(value(MBS_SequenceTwo));
call with name "MBS_Param_Get" in dictionary 5261, "Control", MBS_Control;
MBS_Control = integer(value(MBS_Control));
MBS_TableLineCurrency = 0.00000;
{ Add your code below here }
{ Add your code above here }
call with name "MBS_Param_Set" in dictionary 5261, "TableLineCurrency", str(MBS_TableLineCurrency);
```
Chapter 10: Service Procedures

GP Power Tools has five Service Procedures which can be called to perform custom actions within Microsoft Dynamics GP. These Service Procedures can be called from Dexterity, Visual Studio (Visual C# or Visual Basic.Net) or from Web Services (for version 14.00 or later).

To use the Service Procedures with GP Power Tools, the first parameter passed in will be the Script ID of a Runtime Execute Setup script.

The Dexterity sanScript code contained in the Script ID will then be executed allowing for the development of custom Service Procedures. Use the Select Custom Script Purpose option on the Runtime Execute Setup window to automatically add the template code to handle the parameter passing into and out of the script using listbox fields or collections of strings (for version 14.00 or later).

Due to limitations in Dexterity, the maximum size of data passed into and out of GP Power Tools Service Procedures is limited to the maximum size of a text variable being 32K (32767 bytes).

Below are the details of the Service Procedures system series:

- **ServiceCreateCustom**
- **ServiceDeleteCustom**
- **ServiceGetCustom**
- **ServiceUpdateCustom**
- **ServicePostCustom**
ServiceCreateCustom

This service procedure can be used to make custom “Create” code which can be called as a web service or from Visual Studio or Dexterity. The first parameter passed in needs to be the Script ID of the Runtime Execute Setup script to be executed.

Web Service (version 14.00 or later)
Details for calling the service procedure as a web service are:

Name: ServiceCreateCustom

URI Template: /Custom/Create({ScriptID})

Header Value: GP-Custom-Action=Post

Request Type: Custom (POST)

Parameters:
- ScriptID (String) Runtime Execute Setup Script ID

Payload:
- List<InStringList> (List<String>) Passed in data as collection of strings

Returns:
- Status (Short) Returned Status Code

URL Example: https://domain.com/gpservice/Tenants(DefaultTenant)/Companies(Fabrikam,%20Inc.)/GPPowerTools/Custom/Create({ScriptID})

Visual Studio Call (version 14.00 or later)
Details for calling the service procedure using Visual Studio are:

Name: ServiceCreateCustom

Qualified Name: GPPowerTools.Procedures.ServiceCreateCustom

Parameters:
- ScriptID (String) Runtime Execute Setup Script ID
- List<InStringList> (List<String>) Passed in data as collection of strings
- Status (out Short) Returned Status Code

Invoke Example:
Application.GPPowerTools.Procedures.ServiceCreateCustom.Invoke(ScriptID, InStringList, Status;)

377 G P P O W E R T O O L S
CHAPTER 9     RW FUNCTIONS

Visual Studio Call (versions 11.00 or 12.00)
Details for calling the service procedure using Visual Studio are:

Name: ServiceCreateCustom
Qualified Name: GPPowerTools.Procedures.ServiceCreateCustom
Parameters:
- ScriptID (String) Runtime Execute Setup Script ID
- InStringList (ListBox) Passed in data as Listbox
- Status (out Short) Returned Status Code

Invoke Example:
Application.GPPowerTools.Procedures.ServiceCreateCustom.Invoke(ScriptID, InStringList, Status);

Dexterity Call (version 14.00 or later)
Details for calling the service procedure using Dexterity are:

Name: ServiceCreateCustom
Parameters:
using System.Collections;
using System.Collections.Generic;
in string ScriptID;
in List<System.String> InStringList;
out integer Status;

Call Example:
call with name "ServiceCreateCustom" in dictionary 5261, ScriptID, InStringList, Status;

Dexterity Call (versions 11.00 or 12.00)
Details for calling the service procedure using Dexterity are:

Name: ServiceCreateCustom
Parameters:
in string ScriptID;
in LISTBOX InStringList;
out integer Status;

Call Example:
call with name "ServiceCreateCustom" in dictionary 5261, ScriptID, InStringList, Status;
ServiceDeleteCustom

This service procedure can be used to make custom “Delete” code which can be called as a web service or from Visual Studio or Dexterity. The first parameter passed in needs to be the Script ID of the Runtime Execute Setup script to be executed.

Web Service (version 14.00 or later)
Details for calling the service procedure as a web service are:

Name: ServiceDeleteCustom

URI Template: /Custom/Delete({ScriptID})

Header Value: GP-Custom-Action=Post

Request Type: Custom (POST)

Parameters:
- ScriptID (String) Runtime Execute Setup Script ID

Payload:
- List<InStringList> (List<String>) Passed in data as collection of strings

Returns:
- Status (out Short) Returned Status Code

URL Example:
https://domain.com/gpservice/Tenants(DefaultTenant)/Companies(Fabrikam,%20Inc.)/GPPowerTools/ Custom/Delete({ScriptID})

Visual Studio Call (version 14.00 or later)
Details for calling the service procedure using Visual Studio are:

Name: ServiceDeleteCustom

Qualified Name: GPPowerTools.Procedures.ServiceDeleteCustom

Parameters:
- ScriptID (String) Runtime Execute Setup Script ID
- List<InStringList> (List<String>) Passed in data as collection of strings
- Status (out Short) Returned Status Code

Invoke Example:
Application.GPPowerTools.Procedures.ServiceDeleteCustom.Invoke(ScriptID, InStringList, Status;)}
**Visual Studio Call (versions 11.00 or 12.00)**
Details for calling the service procedure using Visual Studio are:

Name: ServiceDeleteCustom

Qualified Name: GPPowerTools.Procedures.ServiceDeleteCustom

Parameters:
- ScriptID (String)  
Runtime Execute Setup Script ID
- InStringList (ListBox)  
Passed in data as Listbox
- Status (out Short)  
Returned Status Code

Invoke Example:
```
Application.GPPowerTools.Procedures.ServiceDeleteCustom.Invoke(ScriptID, InStringList, Status);
```

**Dexterity Call (version 14.00 or later)**
Details for calling the service procedure using Dexterity are:

Name: ServiceDeleteCustom

Parameters:
```
using System.Collections;
using System.Collections.Generic;
```  
in string ScriptID;  
in List<System.String> InStringList;  
out integer Status;
```

Call Example:
```
call with name "ServiceDeleteCustom" in dictionary 5261, ScriptID, InStringList, Status;
```

**Dexterity Call (versions 11.00 or 12.00)**
Details for calling the service procedure using Dexterity are:

Name: ServiceDeleteCustom

Parameters:
in string ScriptID;  
in LISTBOX InStringList;  
out integer Status;
```

Call Example:
```
call with name "ServiceDeleteCustom" in dictionary 5261, ScriptID, InStringList, Status;
```
ServiceGetCustom

This service procedure can be used to make custom “Get” code which can be called as a web service or from Visual Studio or Dexterity. The first parameter passed in needs to be the Script ID of the Runtime Execute Setup script to be executed.

Web Service (version 14.00 or later)

Details for calling the service procedure as a web service are:

- **Name:** ServiceGetCustom
- **URI Template:** `/Custom/Get({ScriptID})`
- **Header Value:** GP-Custom-Action=Post
- **Request Type:** Custom (POST)
- **Parameters:**
  - `ScriptID` (String) - Runtime Execute Setup Script ID
  - `List<InStringList>` (List<String>) - Passed in data as collection of strings
  - `List<OutStringList>` (List<String>) - Passed out data as collection of strings
  - `Status` (out Short) - Returned Status Code
- **URL Example:**
  https://domain.com/gpservice/Tenants(DefaultTenant)/Companies(Fabrikam,%20Inc.)/GPPowerTools/Custom/Get({ScriptID})

Visual Studio Call (version 14.00 or later)

Details for calling the service procedure using Visual Studio are:

- **Name:** ServiceGetCustom
- **Qualified Name:** GPPowerTools.Procedures.ServiceGetCustom
- **Parameters:**
  - `ScriptID` (String) - Runtime Execute Setup Script ID
  - `List<InStringList>` (List<String>) - Passed in data as collection of strings
  - `List<OutStringList>` (List<String>) - Passed out data as collection of strings
  - `Status` (out Short) - Returned Status Code
- **Invoke Example:**

Visual Studio Call (versions 11.00 or 12.00)
Details for calling the service procedure using Visual Studio are:

Name: ServiceGetCustom

Qualified Name: GPPowerTools.Procedures.ServiceGetCustom

Parameters:
- ScriptID (String) Runtime Execute Setup Script ID
- InStringList (ListBox) Passed in data as ListBox
- OutStringList (ListBox) Passed out data as ListBox
- Status (out Short) Returned Status Code

Invoke Example:

Dexterity Call (version 14.00 or later)
Details for calling the service procedure using Dexterity are:

Name: ServiceGetCustom

Parameters:
using System.Collections;
using System.Collections.Generic;
in string ScriptID;
in List<System.String> InStringList;
out List<System.String> OutStringList;
out integer Status;

Call Example:
call with name "ServiceGetCustom" in dictionary 5261, ScriptID, InStringList, OutStringList, Status;

Dexterity Call (versions 11.00 or 12.00)
Details for calling the service procedure using Dexterity are:

Name: ServiceGetCustom

Parameters:
in string ScriptID;
in LISTBOX InStringList;
out LISTBOX OutStringList;
out integer Status;

Call Example:
call with name "ServiceGetCustom" in dictionary 5261, ScriptID, InStringList, OutStringList, Status;
ServiceUpdateCustom

This service procedure can be used to make custom “Update” code which can be called as a web service or from Visual Studio or Dexterity. The first parameter passed in needs to be the Script ID of the Runtime Execute Setup script to be executed.

**Web Service (version 14.00 or later)**

Details for calling the service procedure as a web service are:

**Name:** ServiceUpdateCustom

**URI Template:** /Custom/Update{ScriptID}

**Header Value:** GP-Custom-Action=Post

**Request Type:** Custom (POST)

**Parameters:**
- ScriptID (String)  
  Runtime Execute Setup Script ID
- List<InStringList> (List<String>)  
  Passed in data as collection of strings

**Returns:**
- Status (Short)  
  Returned Status Code

**URL Example:**
https://domain.com/gpservice/Tenants(DefaultTenant)/Companies(Fabrikam,%20Inc.)/GPPowerTools/Custom/Update{ScriptID}

**Visual Studio Call (version 14.00 or later)**

Details for calling the service procedure using Visual Studio are:

**Name:** ServiceUpdateCustom

**Qualified Name:** GPPowerTools.Procedures.ServiceUpdateCustom

**Parameters:**
- ScriptID (String)  
  Runtime Execute Setup Script ID
- List<InStringList> (List<String>)  
  Passed in data as collection of strings
- Status (out Short)  
  Returned Status Code

**Invoke Example:**
Application.GPPowerTools.Procedures.ServiceUpdateCustom.Invoke(ScriptID, InStringList, Status;)}
CHAPTER 9

Visual Studio Call (versions 11.00 or 12.00)
Details for calling the service procedure using Visual Studio are:

Name: ServiceUpdateCustom

Qualified Name: GPPowerTools.Procedures.ServiceUpdateCustom

Parameters:
- ScriptID (String)
- InStringList (ListBox)
- Status (out Short)

Runtime Execute Setup Script ID
Passed in data as ListBox
Returned Status Code

Invoke Example:
```csharp
```

Dexterity Call (version 14.00 or later)
Details for calling the service procedure using Dexterity are:

Name: ServiceUpdateCustom

Parameters:
- using System.Collections;
- using System.Collections.Generic;
- in string ScriptID;
- in List<System.String> InStringList;
- out integer Status;

Call Example:
```csharp
call with name "ServiceUpdateCustom" in dictionary 5261, ScriptID, InStringList, Status;
```

Dexterity Call (versions 11.00 or 12.00)
Details for calling the service procedure using Dexterity are:

Name: ServiceUpdateCustom

Parameters:
- in string ScriptID;
- in LISTBOX InStringList;
- out integer Status;

Call Example:
```csharp
call with name "ServiceUpdateCustom" in dictionary 5261, ScriptID, InStringList, Status;
```
**ServicePostCustom**

This service procedure can be used to make custom “Post” code which can be called as a web service or from Visual Studio or Dexterity. The first parameter passed in needs to be the Script ID of the Runtime Execute Setup script to be executed.

**Web Service (version 14.00 or later)**

Details for calling the service procedure as a web service are:

- **Name:** ServicePostCustom
- **URI Template:** /Custom/Post({ScriptID})
- **Header Value:** GP-Custom-Action=Post
- **Request Type:** Custom (POST)
- **Parameters:**
  - ScriptID (String) Runtime Execute Setup Script ID
  - List<InStringList> (List<String>) Passed in data as collection of strings
  - List<OutStringList> (List<String>) Passed out data as collection of strings
  - Status (out Short) Returned Status Code

**URL Example:**

https://domain.com/gpservice/Tenants(DefaultTenant)/Companies(Fabrikam,%20Inc.)/GPPowerTools/Custom/Post({ScriptID})?GP-Custom-Action=Post

**Visual Studio Call (version 14.00 or later)**

Details for calling the service procedure using Visual Studio are:

- **Name:** ServicePostCustom
- **Qualified Name:** GPPowerTools.Procedures.ServicePostCustom
- **Parameters:**
  - ScriptID (String) Runtime Execute Setup Script ID
  - List<InStringList> (List<String>) Passed in data as collection of strings
  - List<OutStringList> (List<String>) Passed out data as collection of strings
  - Status (out Short) Returned Status Code

**Invoke Example:**

Visual Studio Call (versions 11.00 or 12.00)
Details for calling the service procedure using Visual Studio are:

Name: ServicePostCustom
Qualified Name: GPPowerTools.Procedures.ServicePostCustom

Parameters:
- **ScriptID** (String)  
  Runtime Execute Setup Script ID
- **InStringList** (ListBox)  
  Passed in data as Listbox
- **OutStringList** (ListBox)  
  Passed out data as Listbox
- **Status** (out Short)  
  Returned Status Code

Invoke Example:
```csharp
```

Dexterity Call (version 14.00 or later)
Details for calling the service procedure using Dexterity are:

Name: ServicePostCustom

Parameters:
```csharp
using System.Collections;
using System.Collections.Generic;
in string ScriptID;
in List<System.String> InStringList;
out List<System.String> OutStringList;
out integer Status;
```

Call Example:
```csharp
call with name "ServicePostCustom" in dictionary 5261, ScriptID, InStringList, OutStringList, Status;
```

Dexterity Call (versions 11.00 or 12.00)
Details for calling the service procedure using Dexterity are:

Name: ServicePostCustom

Parameters:
- **in string ScriptID**;  
- **in LISTBOX InStringList**;
- **out LISTBOX OutStringList**;
- **out integer Status**;

Call Example:
```csharp
call with name "ServicePostCustom" in dictionary 5261, ScriptID, InStringList, OutStringList, Status;
```
Chapter 11: Developer APIs

GP Power Tools has several external APIs available for use by other developers.

Below are the details of the Developer APIs:

- *MBS_Email_API*
- *MBS_WindowPositionCheck*
- *MBS_WindowPositionMemory*
- *MBS_WindowPositionMemoryResize*
MBS_Email_API

This Developer API can be called from another Dexterity product to send emails using the GP Power Tools email engine.

The parameter list for this call is:

- in string IN_EmailFrom;
- in string IN_EmailTo;
- in string IN_EmailCC;
- in string IN_EmailBCC;
- in string IN_EmailSubject;
- in text IN_EmailBody;
- in text IN_EmailSignature;
- in boolean IN_EmailSignatureDefault;
- in text IN_EmailAttachments;
- in boolean IN_EmailPreview;
- in boolean IN_EmailAutoSend;

An example script is:

```plaintext
local string l_EmailFrom;
llocal string l_EmailTo;
llocal string l_EmailCC;
llocal string l_EmailBCC;
llocal string l_EmailSubject;
llocal text l_EmailBody;
llocal text l_EmailSignature;
llocal boolean l_EmailSignatureDefault;
llocal text l_EmailAttachments;
llocal boolean l_EmailPreview;
llocal boolean l_EmailAutoSend;
l_EmailTo = "email@domain.com";
l_EmailSubject = "Email API Test";
l_EmailBody = "This is a test of the Email API"+char(13);
l_EmailSignatureDefault = true;
l_EmailAttachments = l_EmailAttachments + "C:\Dex1000\Data\Dex.ini"+char(13);
l_EmailAttachments = l_EmailAttachments + "C:\Dex1100\Data\Dex.ini"+char(13);
l_EmailPreview = false;
l_EmailAutoSend = false;

call with name "MBS_Email_API" in dictionary 5261,
l_EmailFrom, l_EmailTo, l_EmailCC, l_EmailBCC, l_EmailSubject,
l_EmailBody, l_EmailSignature, l_EmailSignatureDefault,
l_EmailAttachments, l_EmailPreview, l_EmailAutoSend;
```
MBS_WindowPositionCheck

This Developer API can be called from another Dexterity product to temporarily disable the Window Position Check which prevents windows from opening outside of the visible desktop.

The parameter list for this call is:

in boolean IN_Active;

An example script is:

call with name "MBS_WindowPositionCheck" in dictionary 5261,  
    false;

Remember to re-enable the feature with a second call after your code has completed.
This Developer API can be called from another Dexterity product to temporarily disable the Window Position Memory feature which can move windows to their previous location on the desktop.

The parameter list for this call is:

in boolean IN_Active;

An example script is:

```plaintext
call with name "MBS_WindowPositionMemory" in dictionary 5261,
    false,
    false;
```

Remember to re-enable the feature with a second call after your code has completed.
MBS_WindowPositionMemoryResize

This Developer API can be called from another Dexterity product to temporarily disable the Window Position Memory Resizing feature which can resize windows to their previous size on the desktop.

The parameter list for this call is:

in boolean IN_Active;

An example script is:

call with name "MBS_WindowPositionMemoryResize" in dictionary 5261,  
false;

Remember to re-enable the feature with a second call after your code has completed.
GP Power Tools Index

A

About GP Power Tools, 26, 30
Check for Updates, 30
GP Power Tools Modules, 27
Info, 27
Reinstall, 27
Uninstall, 26
Accelerator Key, 201
Access Denied, 99
Account Framework, 264, 276
Actions Tab, 192, 202
Activate Company based Color Schemes, 126
Activate Debug Font Logging for the Report Writer, 57
Activate Debug Logging for the Report Writer, 57
Activate Word Template Processing Engine Logging, 57
Active Profile, 163, 164
Active SQL Profile Traces, 39, 40
ActiveX Data Objects, 35, 216
Add Application Details to GPPTools_<User>_<Company> Log, 55
Add Attachment Button, 47
Add Button, 47, 154, 167, 185, 231
Add Exemption Button, 266
Add extra width to company name drop down list on Company Login window, 132
Add Field Context Menu, 213
Add Form Menu, 213
Add Menu Below Entry, 195
Add Menu to Bottom, 195
Add Menu to Top, 195
Add settings to target, 120
Additional Administrator Features, 176
Security Resource Descriptions, 176
SUPERUSER Security Task and Role, 176
SUPERUSER Workflow Setup, 176
User Company Access Fix, 176
User Setup Additional Information, 176
Additional Developer Features, 255
Macro Play Fast, 255
Open Script Debugger on Startup, 256
Resource Information Context, 256
Runtime Execute Context, 256
Script Debugger Context, 255
Additional System Features, 81
Exit After Processes, 81
Find a Window, 81
Login Remember User, 81
Raise All Windows, 81
Remember Last Company, 81
Transaction being Edited, 82
User Preferences Apply, 81
Adds Allowed, 207, 302, 304, 306, 308
Administration Button, 199
Administrator Controlled Shared Folder Location, 64
Administrator Controlled Shared Folder Location for logs and export files, 64
Administrator Email, 47, 75, 203, 204
Administrator Password, 32, 62, 133
Administrator Password Setup, 62, 63
Administrator Password, 62
Challenge 'sa' user with Administrator password on login, 63
Don’t ask for users who have access to this window, 62
GP Power Tools Administrator Password, 62
Password Fields, 62
System Password, 62
Use separate password instead of System Password, 62
Activate Company based Color Schemes, 126
Add extra width to company name drop down list on Company Login window, 132
Administrator Email, 203, 204
After Login warn user when password is due to expire, 132
After X Minutes, 140
Allow selection of users for Company based Schemes, 128
Apply Button, 127
Attempt to save changes on open windows when logging out, 141
Automatic Logout, 160
Automatic Open Mode, 101, 134
Change User Setup Additional Information User Defined Field Prompts, 139
Change Window Titles in Windows Start Bar, 131
Check for SQL activity before logging out inactive users, 140
Colors Tab, 126
Company Colors Lookup, 127
Company Colors Users, 128
Custom Color Themes, 126, 129
Days to keep daily Max User Count data for, 139
Disable Automatic Logout warning dialog taking focus, 141
Disable Automatic Logout warning dialog when logging out, 141
Disable logging of Security Errors and Warnings, 135
Disable updating Security Privilege warning to include form name, 135
Disable User Setup Additional Information window automatically opening, 137
Disable Window Position Memory feature, 137
Display dialog on login for this company, 130
Enable a second level of Automatic Logout, 140
Enable Automatic Logout of inactive users, 139
Enable Security Activity Tracking, 110, 114, 135
Enable Security Activity Tracking when opening Smartlist, 136
Enable User Activity Tracking, 139, 160
Features Tab, 134
Field Background Color, 126
Include Current Launch File, 43, 134, 204
Include Dex.ini Settings File, 43, 134, 204
Include info for all databases, 43, 134, 204
Include User Dex.ini Settings File, 43, 134, 204
Number of days prior to password expiry to start warning, 132
OK Button, 127
Only require System or Administrator Password to be entered once per session, 133
Options Tab, 137, 159
Prevent application windows from opening outside of the visible desktop area, 131, 135
Reset Buttons, 127
Scrolling Window Line Color, 126
Select Buttons, 127
Select Theme, 126, 129
Settings Applied Message, 134
Spinner Controls, 127
SQL Profile Trace Settings, 215
Test Button, 141
Theme Group, 126, 129
Theme Name, 126, 129
Usability Tab, 130
User Setup Additional Information, 176
When only X% of licenses available, 140
Window Background Color, 126
Window Heading Color, 126
Window Toolbar Color, 126
Windows Start Bar, 131
Administrator Tools, 1, 2, 83, 84, 98, 102, 110, 114, 117, 118, 122, 124, 126, 146, 151, 155, 160, 163, 167, 171, 176
Additional Administrator Features, 176
Administrator Settings, 126
Company Login Filter, 151
Deny Based Security, 117
Dex.ini Configuration, 142
Dictionary Control, 146
Dynamic Product Selection, 171
Enhanced Security, 118
Launch File Configuration, 167
Login Limits, 163
Resource Information, 84
Security Analyzer, 114
Security Denied, 122
Security Hidden, 124
Security Information, 102
Security Log, 110
Security Profiler, 98
User Activity Log, 160
Window Position Memory, 155
ADO, 35, 216
Advanced Mode, 1, 8, 32, 36, 37, 38, 61, 62, 64, 75, 79, 80, 99, 126, 142, 146, 151, 155, 160, 163, 167, 171, 182, 187, 217, 225, 236, 243, 251, 258, 261, 263, 281, 284, 285, 320, 321, 322
.Net Execute Setup, 181, 236, 243, 322
Access, 32
Administrator Password, 32
Administrator Password Setup, 62
Administrator Settings, 99, 110, 114, 126, 160
Automatic Trigger Mode, 187
Company Login Filter, 151
Configuration Maintenance, 79, 113
Database Validation, 263
Development Project, 61
Dex.ini Configuration, 142, 285
Dictionary Control, 146
Dynamic Product Selection, 171
Dynamic Trigger Logging, 251
Email Settings, 75
Launch File Configuration, 167, 285
Logging Settings, 36, 37, 38, 64
Login Limits, 163
Parameter Lists, 243
Project Setup, 61, 182
Runtime Execute Setup, 178, 217, 243, 284, 320
Setup Backup and Restore, 80
SQL Execute Setup, 179, 225, 243, 321
SQL Login Maintenance, 281
<table>
<thead>
<tr>
<th>SQL Server, 32</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Password, 32</td>
</tr>
<tr>
<td>User Activity Log, 160</td>
</tr>
<tr>
<td>Window Position Memory, 155</td>
</tr>
<tr>
<td>XML Table Export, 258</td>
</tr>
<tr>
<td>XML Table Import, 261</td>
</tr>
<tr>
<td>Advanced Security, 170</td>
</tr>
<tr>
<td>After Logging In, 196</td>
</tr>
<tr>
<td>After Login Event, 196</td>
</tr>
<tr>
<td>After Login on Day X, 196</td>
</tr>
<tr>
<td>After Login on DOW, 196</td>
</tr>
<tr>
<td>After Login warn user when password is due to expire, 132</td>
</tr>
<tr>
<td>After Logout Event, 196</td>
</tr>
<tr>
<td>After Menu Selected, 196</td>
</tr>
<tr>
<td>After Original, 196</td>
</tr>
<tr>
<td>After Table Event, 196</td>
</tr>
<tr>
<td>After Time XX XX, 196</td>
</tr>
<tr>
<td>After Timed Event, 196</td>
</tr>
<tr>
<td>After X Minutes, 140</td>
</tr>
<tr>
<td>alias keyword, 229</td>
</tr>
<tr>
<td>All Except Disabled, 188</td>
</tr>
<tr>
<td>All Traces on SQL Server, 39</td>
</tr>
<tr>
<td>All Triggers for select Project, 188</td>
</tr>
<tr>
<td>All Users, 39</td>
</tr>
<tr>
<td>All Users and Companies, 147, 174, 198, 221, 230, 239</td>
</tr>
<tr>
<td>Allow selection of users for Company based Schemes, 128</td>
</tr>
<tr>
<td>AllowWrongDex, 293</td>
</tr>
<tr>
<td>Alternate, 98, 146</td>
</tr>
<tr>
<td>Alternate/Modified Forms and Reports, 104, 116</td>
</tr>
<tr>
<td>Alternate/Modified Status, 146</td>
</tr>
<tr>
<td>Always allow access to this Company, 165</td>
</tr>
<tr>
<td>Application Level Menu, 194, 195, 196, 201</td>
</tr>
<tr>
<td>Application Level Security, 98, 99</td>
</tr>
<tr>
<td>Application Menus, 13</td>
</tr>
<tr>
<td>Application Tools Menu, 11</td>
</tr>
<tr>
<td>Application Window Position, 55</td>
</tr>
<tr>
<td>Application Window Size, 55</td>
</tr>
<tr>
<td>Application.GpPowerTools.dll, 7</td>
</tr>
<tr>
<td>Application.GpPowerTools.xml, 7</td>
</tr>
<tr>
<td>ApplicationName, 54, 292</td>
</tr>
<tr>
<td>Apply Advanced SQL Server options, 282</td>
</tr>
<tr>
<td>Apply Button, 127, 283</td>
</tr>
<tr>
<td>Apply User Status, 282</td>
</tr>
<tr>
<td>Area Page, 14, 26, 28, 31, 41, 45, 49, 51, 60, 62, 64, 75, 79, 80, 84, 98, 102, 110, 114, 118, 122, 124, 126, 142, 146, 151, 155, 160, 163, 167, 171, 178, 179, 181, 182, 189, 192, 217, 225, 236, 243, 251, 258, 261, 263, 281</td>
</tr>
<tr>
<td>Associated Tables Button, 86</td>
</tr>
<tr>
<td>Attachments, 47</td>
</tr>
<tr>
<td>Attempt to save changes on open windows when logging out, 141</td>
</tr>
<tr>
<td>Authentication, 77</td>
</tr>
<tr>
<td>Authentication Mode, 68</td>
</tr>
<tr>
<td>Auto select if only one Company, 153</td>
</tr>
<tr>
<td>Auto Send, 48, 77</td>
</tr>
<tr>
<td>AutoInstallChunks, 54, 292</td>
</tr>
<tr>
<td>Automated Diagnostics, 290</td>
</tr>
<tr>
<td>MBS_Debug_Automatic_AutoSave, 290</td>
</tr>
<tr>
<td>MBS_Debug_Automatic_Script, 290</td>
</tr>
<tr>
<td>MBS_Debug_Automatic_Status, 290</td>
</tr>
<tr>
<td>Automatic Logout, 139, 140, 141, 160</td>
</tr>
<tr>
<td>Automatic Open Mode, 101, 134</td>
</tr>
<tr>
<td>Automatic Start Only, 188</td>
</tr>
<tr>
<td>Automatic Trigger Mode, 182, 187</td>
</tr>
<tr>
<td>.Net Execute Setup, 206, 236</td>
</tr>
<tr>
<td>Accelerator Key, 201</td>
</tr>
<tr>
<td>Actions Tab, 192, 202</td>
</tr>
<tr>
<td>Administration Button, 199</td>
</tr>
<tr>
<td>All Except Disabled, 188</td>
</tr>
<tr>
<td>All Triggers for selected Project, 188</td>
</tr>
<tr>
<td>All Users and Companies, 198</td>
</tr>
<tr>
<td>Automatic Start Only, 188</td>
</tr>
<tr>
<td>Capture Dexterity Script Log, 215</td>
</tr>
<tr>
<td>Capture Dexterity Script Profile, 215</td>
</tr>
<tr>
<td>Capture Macro Recording, 215</td>
</tr>
<tr>
<td>Capture Screenshots to default logging folder or email, 204</td>
</tr>
<tr>
<td>Capture SQL Log, 215</td>
</tr>
<tr>
<td>Capture SQL Profile Trace, 215</td>
</tr>
<tr>
<td>Change Start Mode Button, 199</td>
</tr>
<tr>
<td>Change State Button, 199</td>
</tr>
<tr>
<td>Check Security, 205</td>
</tr>
<tr>
<td>Check Syntax, 210</td>
</tr>
<tr>
<td>Conditional Script, 187, 190, 202, 206, 214</td>
</tr>
<tr>
<td>DEFAULT, 79, 188, 192</td>
</tr>
<tr>
<td>Default Button, 206</td>
</tr>
<tr>
<td>DEFAULT only, 188</td>
</tr>
<tr>
<td>Dialog Message, 203</td>
</tr>
<tr>
<td>Disable trigger after Condition met, 215</td>
</tr>
<tr>
<td>Disabled, 197</td>
</tr>
<tr>
<td>Display Message, 190, 203</td>
</tr>
<tr>
<td>Display Message to screen using desktop alert, 202</td>
</tr>
<tr>
<td>Display Message to screen using simple system dialog instead of text box dialog, 203</td>
</tr>
<tr>
<td>Display Message to screen using system dialog, 202</td>
</tr>
<tr>
<td>Do not activate Logging Mode, 197</td>
</tr>
<tr>
<td>Do not run missed event on next login, 214</td>
</tr>
<tr>
<td>Duplicate Button, 197</td>
</tr>
<tr>
<td>Email Address, 203</td>
</tr>
<tr>
<td>Email Screenshots using Administrator Email or Email Address below, 204</td>
</tr>
<tr>
<td>End Date, 213</td>
</tr>
<tr>
<td>Entry, 201</td>
</tr>
<tr>
<td>Error Handling, 214</td>
</tr>
<tr>
<td>Exclude Selected Users and Companies rather than include them, 198</td>
</tr>
<tr>
<td>Execution Mode, 214</td>
</tr>
<tr>
<td>Export Current Table Record to XML, 203</td>
</tr>
<tr>
<td>Export Entire Table to XML restricted by Where Clause, 203</td>
</tr>
<tr>
<td>Export Record, 190</td>
</tr>
</tbody>
</table>
Export Table, 190
Field, 201
Field Name, 201, 203
Find …, 210
Find Next, 210
Font Size, 211
Font Style, 211
Form, 200
Form Name, 200
Function, 201
Function Name, 201
Generate Dexterity Pass Through, 211
Goto Line …, 210
GP Power Tools Setup, 327, 328, 330, 331, 340, 341
GP Power Tools Setup, 284
Help Button, 206
Helper Button, 207
Helper Function Assistant, 207
How to Setup, 187
If less than X MB, 203
Include Current Launch File, 204
Include Dex.ini Settings File, 204
Include info for all databases, 204
Include User Dex.ini Settings File, 204
Include zipped log files, 203
Insert Button, 207
Insert Helper Function, 207
Introduction, 187
Issue Reject Record, 203
Issue Reject Script, 203
Keep Focus on Field, 204
Log File, 191
Mark To Delete Button, 199
Message, 206
Minimize Log Entries, 197
Modified, 200, 205
Names Button, 208
Names Button Uses Clipboard, 212
Non Logging All Except Disabled, 188
Non Logging Automatic Start Only, 188
Non Logging Triggers, 188, 189, 197, 203, 204, 213, 215, 243, 328, 329, 330, 331, 340, 341
Notes Button, 193
Number of execution logs to keep, 214
Old Field Value, 189
Only restart selected logs when trigger fires, 215
Open Window Hidden, 203
Optional Where Clause, 203
Options, 211
Options Tab, 192, 213
OUT_Condition, 206
Parameter ID, 206, 213
Parameter Lists, 206, 213
Parameter Placeholder, 206, 220, 228, 238, 243
Parameters Button, 206
Perform actions when fired and condition not met, 202
Perform actions when fired regardless of condition, 202
Procedure, 201
Procedure Name, 201
Product Name, 200
Project ID, 213
Register, 189
Registration, 188
Release Notes, 193
Replace …, 210
Replace and Find Next, 210
Report Explorer, 209
Resource Explorer, 200, 208
Resource Tab, 192, 200
Restore Field Value, 204
Restriction of Scope, 216
Runtime Execute Setup, 206, 217
Save and Continue, 210
Script Context, 205
Script Explorer, 209
Script Menu, 210
Script Tab, 192, 205
Selected Users and Companies, 198
Send Email using Administrator Email or Email Address below, 203
Setup, 192, 243, 284, 327, 328, 329, 330, 331, 340, 341
SQL Execute Setup, 206, 225
SQL Profile Trace Mode, 215
Start Date, 213
Start Trigger Automatically on Login, 52, 187, 197
Start Trigger Automatically on Login for Users, 198
Stop Trigger after Condition met, 215
Syntax Errors, 206
Table, 187, 200
Table Explorer, 200, 208
Table Name, 200
Technical Name, 200, 201
Timestamp Button, 193
Trigger, 217, 225, 236
Trigger Administration, 199
Trigger Attach, 196
Trigger Description, 193
Trigger Event, 187, 194, 200, 206
Trigger ID, 79, 187, 188, 190, 192, 193, 197
Trigger Setup, 192, 243
Trigger Setup Scheduled Log, 214
Trigger Status, 189, 197
Trigger Type, 194, 196, 200, 206, 216
Triggering, 190
Unregister, 189, 197
Users Button, 197, 198
Visual Studio Integration Toolkit, 195
Window, 201
Window Name, 201
WinthropDC.GpPowerToolsVB.dll, 200, 205
Automatic Update Check, 30
Automatically check for updated keys, 29
Automatically check for updates, 30
Automatically Generate Passwords, 282
Automatically Install Chunk Files without displaying dialog, 54
Automatically open GP Power Tools main window after login, 51
Automatically open Logging Control window after login, 53

B
Back Button, 85
Back Up Button, 95
Backup Button, 80
Bcc Button, 47
Bcc Field, 47
Before Original, 196, 203, 204
Bitmap Scaling, 59
Body, 47, 76
### G P  P O W E R  T O O L S  I N D E X

**Body Text**, 47, 76  
**Bottom Button**, 148, 153, 154, 168, 173, 231

#### C

**Calculator**, 49, 289  
**Cancel Button**, 44, 48, 95, 155, 253, 283  
**Capture Dexterity Script Log**, 65, 215  
**Capture Dexterity Script Profile**, 65, 215  
**Capture Macro Recording**, 65, 215  
**Capture Screenshots to default logging folder or email**, 204  
**Capture SQL Log**, 65, 215  
**Capture SQL Profile Trace**, 65, 215  
**Case Sensitive**, 86  
**Cc Button**, 47  
**Cc Field**, 47  
**CDO**, 76  
**Challenge 'sa' user with Administrator password on login**, 63  
**Change Password Next Login**, 282  
**Change Start Mode Button**, 199  
**Change State Button**, 199  
**Change User Setup Additional Information User Defined Field Prompts**, 139  
**Change Window Titles in Windows Start Bar**, 131  
**Check for SQL activity before logging out inactive users**, 140  
**Check for Updates**, 30  
**Check Security**, 205  
**Check Syntax**, 210, 222, 232, 241  
**Clear Button**, 61, 79, 86, 100, 253  
**Collaboration Data Objects**, 76  
**Comma Delimited**, 95, 107, 112, 115, 123, 125, 162, 180, 230  
**Company**, 103, 111, 120, 123, 125, 161  
**Company Colors Lookup**, 127  
**Company Colors Users**, 128  
**Company Display Sort Order**, 153  
**Company ID**, 111, 123, 125, 161  
**Company Login Filter**, 151, 286  
**Add Button**, 154  
**Auto select if only one Company**, 153  
**Bottom Button**, 153, 154  
**Company Display Sort Order**, 153  
**Delete Button**, 154  
**Display Company Database**, 153  
**Down Button**, 153, 154  
**Duplicate Button**, 153  
**Edit Button**, 154  
**Enable current Profile on this workstation**, 152  
**Prefix for Disabled Companies**, 153  
**Profile ID**, 152, 153, 291  
**Profile Name**, 152  
**Show Disabled Companies**, 153  
**Top Button**, 153, 154  
**Up Button**, 153, 154  
**Conditional Script**, 187, 190, 202, 206, 214  
**Configuration**, 6  
**Configuration Export/Import**, 60, 80, 182, 260, 288  
**Clear Button**, 61  
**Export Button**, 60  
**File Name**, 61  
**Import Button**, 60  
**Import Settings File**, 60  
**Transfer User and Company details**, 61  
**Configuration File Path**, 184, 185  
**Configuration Maintenance**, 79, 113, 288  

- **Clear Button**, 79  
- **Redisplay Button**, 79  
- **Contact Details**, 28  
- **Context Menu**, 194, 195  
- **Convert References**, 232  
- **Copy Button**, 119  
- **Copy SQL Profile Trace files to Logs and Export files location**, 74  
- **Copy to current User in other Companies**, 120  
- **Copy to other Users in current Company**, 119  
- **Create SQL Profile Trace SQL Components**, 74  
- **Create/Update Security Task**, 100  
- **Create/Update Security Task from Log**, 111  
- **Create/Update Security Task from selected rows**, 111  
- **Current Project**, 183  
- **Current User only**, 39  
- **Custom Color Themes**, 126, 129  
- **Customization Status**, 146, 150  
- **Customization Tools**, 84, 98  

#### D

**DAG.EXE**, 90  
**Daily Event**, 195  
**Data Source Name**, 55  
**Database**, 227  
**Database Maintenance**, 273  
**Database Tools**, 1, 3, 257, 258, 261, 263, 281  
**Database Validation**, 263  
**SQL Login Maintenance**, 281  
**XML Table Export**, 258  
**XML Table Import**, 261  
**Database Validation**, 137, 263, 275, 281, 287  
**Account Framework**, 264, 276  
**Add Exemption Button**, 266  
**Database Maintenance**, 273  
**Database Validation Email Settings**, 269  
**Database Validation Exemptions**, 267  
**Dynamics GP Utilities**, 264, 271  
**Email Settings**, 269  
**Exemptions**, 266, 267  
**Exemptions Button**, 267  
**Fix Account Framework**, 271, 276  
**Fix Framework Button**, 271  
**Fix Tables**, 271, 280  
**Fix Tables Button**, 271  
**Fix Users and Databases**, 269, 276  
**Fix Users Button**, 269  
**Fix Utilities**, 270, 276  
**Fix Utilities Button**, 270  
**Legend**, 268  
**Legend Button**, 268  
**OK Button**, 265  
**Only include SQL Table & Views which have a DEX_ROW_ID column**, 273  
**Only Show Tables with Account Fields**, 273  
**Options Menu**, 275, 281  
**Override to Convert Table Structures without using Dynamics Utilities**, 271  
**Print Button**, 268  
**Print Report**, 268  
**Process Button**, 265  
**Redisplay Button**, 265  
**Remove Exemption Button**, 267  
**Reset User SQL Logins and Passwords**, 275, 281
Show Structure Errors Button, 274
Table Structure Errors, 274
User Email Address, 137
User Setup Additional Information, 270
Using Database Validation, 276
Validate Button, 266
Database Validation Email Settings, 269, 282
Database Validation Exemptions, 267
Exemption Mode, 267
Object Mode, 267
Remove All, 267
Remove Selected, 267
View Mode, 267
Database Validation Exemptions, 267
Database Validation Users and Databases, 264
Days to keep daily Max User Count data for, 139
Debug Menu, 52
Debug Menu Product, 52
Debug Tab, 51
DebugFonts, 57, 294
Debugger.xml, 7, 15, 80, 284
DebugLog.txt, 57, 294
DebugRW, 57, 293
DebugRW.txt, 57, 293
DEFAULT, 79, 188, 192
Default Body Text, 47
Default Body Text for Send Email window, 76
Default last Company used on login, 55
Default last User ID used on login, 55
Default maximum sessions per User, 164
DEFAULT only, 188
Default Site ID, 138
Invoice Entry, 138
Item Inquiry, 138
Item Transaction Entry, 138
Item Transfer Entry, 138
Sales Transaction Entry, 138
Default Subject, 47, 75
Default LastCompany, 55, 285
Delete Button, 122, 124, 148, 154, 168, 253
Delete Record, 194
Deny Based Security, 104, 117, 118, 120, 122, 123, 124, 125
Enhanced Security, 104, 118, 120, 123, 125
Security Denied, 104, 120, 122
Security Hidden, 104, 120, 124
Description, 172
Description of Modified/Alternate Resource, 173
Detail Format, 114
details Button, 112
Developer, 187
Developer APIs, 387
MBS_Email_API, 388
MBS_WindowPositionCheck, 389
MBS_WindowPositionMemory, 390
MBS_WindowPositionMemoryResize, 391
Developer Tools, 1, 3, 177, 178, 179, 181, 182, 187, 217, 225, 236, 243, 251, 255
.Net Execute Setup, 236
.Net Execute, 181
Additional Developer Features, 255
Automatic Trigger Mode, 187
Dynamic Trigger Logging, 251
Parameter Lists, 243
Project Setup, 182
Runtime Execute Setup, 217
Runtime Executer, 178
SQL Execute Setup, 225
SQL Executer, 179
Development Project, 61
Dex.chm, 7
Dex.dic, 293
DEX.DIC, 284
Dex.ini, 43, 51, 134, 142, 143, 151, 153, 204, 284
Global, 43, 51, 134, 143, 204
User, 43, 51, 134, 143, 204
Dex.ini Configuration, 142, 143, 285, 288
Dex.ini, 142
Dex.ini Settings Inspector, 145
Display Dex.ini Settings, 143
Do not update any Dex.ini settings automatically, 143
Edit Dex.ini Button, 145
Info Button, 143
Log, 142
Search Mode, 142, 143
Setting or Search String, 142, 143
Settings List, 142
Silent, 142
Target Dex.ini, 143
Value, 142
Dex.ini Settings, 26, 37, 51, 64, 158, 190, 284
Activate Debug Font Logging for the Report Writer, 57
Activate Debug Logging for the Report Writer, 57
Activate Word Template Processing Engine Logging, 57
Add Application Details to GPPTools_<User>_<Company> Log, 55
AllowWrongDex, 293
Application Window Position, 55
Application Window Size, 55
ApplicationName, 54, 292
AutoInstallChunks, 54, 292
Automatically Install Chunk Files without displaying dialog, 54
Automatically open GP Power Tools main window after login, 51
Automatically open Logging Control window after login, 53
Debug Tab, 51
DebugFonts, 57, 294
DebugRW, 57, 293
Default last Company used on login, 55
Default last User ID used on login, 55
Default maximum sessions per User, 164
DEFAULT only, 188
Default Site ID, 138
Invoice Entry, 138
Item Inquiry, 138
Item Transaction Entry, 138
Item Transfer Entry, 138
Sales Transaction Entry, 138
Default Subject, 47, 75
Default LastCompany, 55, 285
Delete Button, 122, 124, 148, 154, 168, 253
Delete Record, 194
Deny Based Security, 104, 117, 118, 120, 122, 123, 124, 125
Enhanced Security, 104, 118, 120, 123, 125
Security Denied, 104, 120, 122
Security Hidden, 104, 120, 124
Description, 172
Description of Modified/Alternate Resource, 173
Detail Format, 114
details Button, 112
Developer, 187
Developer APIs, 387
MBS_Email_API, 388
MBS_WindowPositionCheck, 389
MBS_WindowPositionMemory, 390
MBS_WindowPositionMemoryResize, 391
Developer Tools, 1, 3, 177, 178, 179, 181, 182, 187, 217, 225, 236, 243, 251, 255
.Net Execute Setup, 236
.Net Execute, 181
Additional Developer Features, 255
Automatic Trigger Mode, 187
Dynamic Trigger Logging, 251
Parameter Lists, 243
<table>
<thead>
<tr>
<th>GP Power Tools Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnableWCRibbons, 55, 294</td>
</tr>
<tr>
<td>Export Body Section as One Line, 56</td>
</tr>
<tr>
<td>ExportLinesPerPage, 56, 293</td>
</tr>
<tr>
<td>ExportOneLineBody, 56, 293</td>
</tr>
<tr>
<td>ExportPDFLinesPerPage, 56, 293</td>
</tr>
<tr>
<td>Folder location for logs and export files, 9, 53</td>
</tr>
<tr>
<td>GP Power Tools Settings, 284</td>
</tr>
<tr>
<td>KeepTemplateTempFiles, 57, 294</td>
</tr>
<tr>
<td>MaxSWScrollbarSize, 59, 294</td>
</tr>
<tr>
<td>MBS_Debug_Automate_File, 290</td>
</tr>
<tr>
<td>MBS_Debug_Automate_Script, 290</td>
</tr>
<tr>
<td>MBS_Debug_Automate_Status, 290</td>
</tr>
<tr>
<td>MBS_DebugAutoOpen, 53, 284</td>
</tr>
<tr>
<td>MBS_Debug_Break, 256, 291</td>
</tr>
<tr>
<td>MBS_Debug_CompanyFilter, 152, 154, 291</td>
</tr>
<tr>
<td>MBS_Debug_CompanySwitchWidth, 132, 285</td>
</tr>
<tr>
<td>MBS_Debug_ConfigurationOverride, 143, 285</td>
</tr>
<tr>
<td>MBS_Debug_DisableScreenOutputMemory, 57, 289</td>
</tr>
<tr>
<td>MBS_Debug_DisableSplitters, 290</td>
</tr>
<tr>
<td>MBS_Debug_LastRunSystem, 290</td>
</tr>
<tr>
<td>MBS_Debug_LastRunUser, 291</td>
</tr>
<tr>
<td>MBS_Debug_LaunchConfigurationOverride, 169, 285</td>
</tr>
<tr>
<td>MBS_Debug_LogAppDetails, 55, 285</td>
</tr>
<tr>
<td>MBS_Debug_LogListPath, 253, 254, 291</td>
</tr>
<tr>
<td>MBS_Debug_LogOnStartup, 53, 284</td>
</tr>
<tr>
<td>MBS_Debug_LogWinData, 291</td>
</tr>
<tr>
<td>MBS_Debug_NameUseClipboard, 291</td>
</tr>
<tr>
<td>MBS_Debug_Path, 53, 64, 284</td>
</tr>
<tr>
<td>MBS_Debug_RuntimeCheck, 284</td>
</tr>
<tr>
<td>MBS_Debug_SetupMode, 52, 284</td>
</tr>
<tr>
<td>MBS_Debug_ShowRuntime, 284</td>
</tr>
<tr>
<td>MBS_Debug_SkipVersionChecks, 290</td>
</tr>
<tr>
<td>MBS_Debug_UpdateLastUserOnExit, 55, 285</td>
</tr>
<tr>
<td>MBS_Debug_VBADisableReset, 290</td>
</tr>
<tr>
<td>MBS_Debug_WinActivityLog, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinActivityLogMaxUser, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinAdminSettings, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinCalculator, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinCompanyFilter, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinConfigSettings, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinConfigurationExportImport, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinConfigurationMaintenance, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinDatabaseValidation, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinDebugger, 285</td>
</tr>
<tr>
<td>MBS_Debug_WinDebuggerSetups, 285</td>
</tr>
<tr>
<td>MBS_Debug_WinDebuggerStatus, 285</td>
</tr>
<tr>
<td>MBS_Debug_WinDictionaryControl, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinEmailSettings, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinFieldLookup, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinKeyLookup, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinLaunchFileConfig, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinLoggingSettings, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinLoginLimits, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinLoginMaintenance, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinMenuExplorer, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinNetExecute, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinNetExplorer, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinObjectExplorer, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinParameterMaintenance, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinProductSelection, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinProjectSetup, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinReportExplorer, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinResourceExplorer, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinResourceInformation, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinRuntimeExecute, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinRuntimeExecutor, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinScreenOutput, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinScreenOutput, 57</td>
</tr>
<tr>
<td>MBS_Debug_WinScreenShot, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinScriptExplorer, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinSecurityAnalyzer, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinSecurityDeny, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinSecurityEnhanced, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinSecurityHide, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinSecurityInfo, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinSecurityInfoResource, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinSecurityLog, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinSecurityLogDetail, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinSecurityProfile, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinSendEmail, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinSQLExecute, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinSQLExecutor, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinSQLResults, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinTableExplorer, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinTableLookup, 289</td>
</tr>
<tr>
<td>MBS_Debug_WinTriggerListMaintenance, 288</td>
</tr>
<tr>
<td>MBS_Debug_WinWindowsMemory, 286</td>
</tr>
<tr>
<td>MBS_Debug_WinXMLTableExport, 287</td>
</tr>
<tr>
<td>MBS_Debug_WinXMLTableImport, 287</td>
</tr>
<tr>
<td>MouseWheel, 294</td>
</tr>
<tr>
<td>Name shown on Application title bar during initial loading, 54</td>
</tr>
<tr>
<td>Number of Lines Per Page when Exporting Reports (inc. PDF), 56</td>
</tr>
<tr>
<td>OLEClose, 59, 295</td>
</tr>
<tr>
<td>Open Application Maximized on next login, 55</td>
</tr>
<tr>
<td>Other Tab, 58</td>
</tr>
<tr>
<td>Pathname location for Debugger Setup files, exports and logs, 37, 64, 190</td>
</tr>
<tr>
<td>Pathname location for SQL Log file, 52</td>
</tr>
<tr>
<td>QueueMoreInfo, 59, 294</td>
</tr>
<tr>
<td>RememberUser, 81</td>
</tr>
<tr>
<td>Rename DEXSQL.LOG at the beginning of each day, 52</td>
</tr>
<tr>
<td>Reports Tab, 56</td>
</tr>
<tr>
<td>Reset Window Positions, 53, 158</td>
</tr>
<tr>
<td>SAMPLEDATEMSG, 54, 293</td>
</tr>
<tr>
<td>Script Editor Settings, 296</td>
</tr>
<tr>
<td>ScriptCommentColor, 296</td>
</tr>
<tr>
<td>ScriptDebugger, 52, 292</td>
</tr>
<tr>
<td>ScriptDebuggerProduct, 52, 292</td>
</tr>
<tr>
<td>ScriptEditorFontName, 296</td>
</tr>
<tr>
<td>ScriptEditorFontSize, 296</td>
</tr>
<tr>
<td>ScriptEditorSyntaxColoring, 296</td>
</tr>
<tr>
<td>ScriptErrorColor, 296</td>
</tr>
<tr>
<td>ScriptIdentifierColor, 296</td>
</tr>
<tr>
<td>ScriptKeywordColor, 296</td>
</tr>
<tr>
<td>ScriptNumberColor, 296</td>
</tr>
<tr>
<td>ScriptOperatorColor, 296</td>
</tr>
<tr>
<td>ScriptStringColor, 296</td>
</tr>
<tr>
<td>Show Advanced Macro Menu, 58</td>
</tr>
<tr>
<td>Show All Menu Items, 58</td>
</tr>
<tr>
<td>Show Debug Messages on next login, 52</td>
</tr>
</tbody>
</table>
ShowAdvancedMacroMenu, 58, 293
ShowAllMenuItems, 58, 293
SkipVersionChecks, 293
SQL Logging, 53
SQLLastCompany, 81, 285
SQLLogAllODBCMessages, 292
SQLLogODBCMessages, 52, 292
SQLLogPath, 52, 292
SQLLogRename, 52, 285
SQLLogSQLStmt, 52, 292
Start Logging on next startup only, 38, 53
Startup Tab, 54
Suppress Date Change Dialog, 58
Suppress Sample Company Date Warning, 54
SuppressSound, 58, 293
System Settings, 292
TPELogging, 57, 294
Update User ID and Company on exit, 55
Use SQL Login Compatibility Mode, 54
VBADisable, 149, 294
WindowHeight, 55, 294
WindowMax, 55, 294
WindowPosX, 55, 294
Windows Bitmap Font Registry Settings, 59
Windows Bitmap Scaling Settings, 59
WindowWidth, 55, 294
Dex.ini Settings Inspector, 145
DEXSQL.LOG, 35, 37, 52, 292
DEXSQL_<Date>_ <Time>.LOG, 37
Dexterity, 10, 35, 56, 65, 84, 85, 90, 146, 178, 187, 188, 189, 190, 192, 196, 207, 215, 216, 217, 220, 228, 229, 284
Customization Status, 146
Debug Menu, 52
Debug Menu Product, 52
Developer, 187
Dexterity Script Logging, 35, 65, 215
Dexterity Script Profiling, 35, 65, 215
Dictionary, 84
Display Name, 84, 229
Enable Enhanced Script Log on next login, 52
Field, 84, 187, 201, 229
Form, 84, 200
Function, 84, 201
Macro Recording, 35, 65, 215
Physical Name, 84, 229
Procedure, 84, 201
Report, 84
Resource ID, 84
Resources, 84
Sanscript, 178, 187, 207, 217, 220, 228, 284
Script, 84
Show Debug Messages on next login, 52
Table, 84, 187, 200, 229
Table Group, 84
Technical Name, 84, 200, 201
Trigger, 146, 187, 188, 189, 190, 192, 196, 216
Window, 84, 201
Dexterity Call, 378, 380, 382, 384, 386
ServiceCreateCustom, 378
ServiceDeleteCustom, 380
ServiceGetCustom, 382
ServicePostCustom, 386
ServiceUpdateCustom, 384
Dexterity Debug Menu, 52
Dexterity Debug Menu Product, 52
Dexterity Profile, 53
Dexterity Script, 53
Dexterity Script Logging, 35, 65, 215
Dexterity Script Profiling, 35, 65, 215
Diagnostics, 290
MBS_Debug_Automate_File, 290
MBS_Debug_Automate_Script, 290
MBS_Debug_Automate_Status, 290
Dialog Message, 203
Dialog mode when selecting product, 173
Dictionary, 84
Dictionary Assembly, 90
Dictionary Control, 146, 286
All Users and Companies, 147
Alternate/Modified Status, 146
Bottom Button, 148
Customization Status, 150
Delete Button, 148
Disable Visual Basic for Applications (VBA) on next login, 149
Disable Visual Studio Tools (VST) Addins on next login, 149
Disabled After Login for Users, 147
Down Button, 148
Enable Visual Basic for Applications after one login, 149
Enable Visual Studio Tools Addins after one login, 149
Exclude Selected Users and Companies rather than include them, 147
Field Level Security, 150
Info Button, 150
Selected Users and Companies, 147
Show Launch File, 150
Top Button, 148
Trigger Status, 146
Up Button, 148
Dictionary ID, 252
Disable Automatic Logout warning dialog taking focus, 141
Disable Automatic Logout warning dialog when logging out, 141
Disable closing of the OLE Contain.exe on exit, 59
Disable logging of Security Errors and Warnings, 135
Disable Ribbons for workstation on next login, 55
Disable Screen Output window position memory, 57
Disable trigger after Condition met, 215
Disable updating Security Privilege warning to include form name, 135
Disable User Setup Additional Information window automatically opening, 137
Disable Visual Basic for Applications (VBA) on next login, 149
Disable Visual Studio Tools (VST) Addins on next login, 149
Disabled After Login for Users, 147
Display, 203
Display Company Database, 153
Display Dex.ini Settings, 143
Display dialog on login for this company, 130
Display Excluded and Missing Resources, 120
Display Keys Button, 88
| Display Message, 190, 203 |
| Display Message to screen using desktop alert, 202 |
| Display Message to screen using system dialog, 202 |
| Display Mode, 110, 123, 125, 160 |
| Display More Info button on Process Monitor, 59 |
| Display Name, 84, 229 |
| Display only Selected Users, 121 |
| Display Parameters, 90 |
| Display Parameters Button, 90 |
| Display Security Tasks and Roles, 108 |
| Divider Adjustment Buttons, 228, 238 |
| Do not activate Logging Mode, 197 |
| Do not run missed event on next login, 214 |
| Do not update any Dex.ini settings automatically, 143 |
| Do not update the Launch File automatically, 169 |
| Document Access, 84, 98 |
| Don’t ask for users who have access to this window, 62 |
| Down Button, 148, 153, 154, 168, 173, 231, 249 |
| DSN, 55 |
| DUOS, 323, 324, 325, 326, 332, 333, 334, 335 |
| Duplicate Button, 153, 165, 173, 184, 197, 220, 229, 238, 249, 259 |
| Duplicate Records, 262 |
| Dynamic Product Selection, 171, 287 |
| All Users and Companies, 174 |
| Alternate, 171 |
| Alternate/Modified Forms and Report ID, 171 |
| Bottom Button, 173 |
| Description, 172 |
| Description of Modified/Alternate Resource, 173 |
| Dialog mode when selecting product, 173 |
| Down Button, 173 |
| Duplicate Button, 173 |
| Enabled for Users, 173, 174 |
| Exclude Selected Users and Companies rather than include them, 174 |
| Modified Alternate, 171 |
| Modified/Alternate ID, 171, 173 |
| Modifiedl, 171 |
| Original, 171 |
| Resource Tree, 173 |
| Resource Type, 173 |
| Selected Users and Companies, 174 |
| Selection List, 173 |
| Short Description used for dialog buttons, 173 |
| Top Button, 173 |
| Up Button, 173 |
| User Button, 173 |
| User Security, 171 |
| Users Button, 173, 174 |
| Dynamic Trigger Logging, 251 |
| Cancel Button, 253 |
| Clear Button, 253 |
| Delete Button, 253 |
| Dictionary ID, 252 |
| Field Name, 252 |
| File Path, 252, 291 |
| Form Name, 252 |
| OK Button, 253 |
| Product Dictionary, 252 |
| Redisplay Button, 253 |
| Trigger Mode, 252 |
| Trigger Type, 252 |
| Window/Table/Procedure/Function Name, 252 |
| Dynamics GP Utilities, 264, 271 |
| Dynamics Trigger Logging, 288 |
| Dynamics.exe.config, 149 |
| Dynamics.set, 15, 26, 43, 54, 147, 148, 150, 151, 167, 204 |
| E |
| eConnect, 216 |
| Edit Button, 154 |
| Edit Dex.ini Button, 145 |
| Edit SQL Profile Trace Settings, 66 |
| Email, 29 |
| Email Address, 203 |
| Email Button, 44 |
| Email Mode, 76 |
| Email Screenshots using Administrator Email or Email Address below, 204 |
| Email Settings, 76, 77, 78 |
| Authentication, 77 |
| Auto Send, 77 |
| Exchange Web Services, 77 |
| MAPI Compliant Client, 77 |
| Outlook, 76 |
| Password, 78 |
| Preview, 77 |
| Send HTML, 77 |
| Sender’s Email, 77 |
| SMTP Server, 77 |
| SMTP Server Port, 77 |
| SMTP Server via CDO, 76 |
| Terminal Server, 76 |
| User ID, 78 |
| Email Settings, 44, 46, 47, 48, 75, 76 |
| Administrator Email, 47, 75 |
| Auto Send, 48 |
| Body Text, 46 |
| Default Body Text, 47 |
| Default Body Text for Send Email window, 76 |
| Default Subject, 46, 47, 75 |
| Email Email, 46 |
| Email Mode, 76 |
| Microsoft Outlook Client, 76 |
| Preview, 46, 48 |
| Sender’s Email, 47 |
| Standard Signature to add to all emails, 76 |
| Subject, 46 |
| Email Settings, 95 |
| Email Settings, 99 |
| Email Settings, 107 |
| Email Settings, 112 |
| Email Settings, 115 |
| Email Settings, 123 |
| Email Settings, 125 |
| Email Settings, 162 |
| Email Settings, 180 |
| Email Settings, 230 |
| Email Settings, 269 |
| Email Settings, 282 |
| Email Settings, 288 |
| Email Settings, 289 |
| Email Settings, 269 |
| Employee ID, 137 |
| Enable a second level of Automatic Logout, 140 |
| Enable Automatic Logout of inactive users, 139 |
| Enable current Profile on this workstation, 152 |
| Enable Debugger Setup Mode, 51 |
Enable Dexterity Debug Menu on next login, 52
Enable Enhanced Script Log on next login, 52
Enable GP Power Tools Setup Mode, 52
Enable Individual Logging Modes, 38, 65
Enable Scrollbar width override, 59
Enable Security Activity Tracking, 110, 114, 135
Enable Security Activity Tracking when opening Smartlist, 136
Enable selection of Data Server on Login, 55
Enable SQL Logging on next login, 52
Enable User Activity Tracking, 139, 160
Enable Visual Basic for Applications after one login, 149
Enable Visual Studio Tools Addins after one login, 149
Enabled for Users, 173, 174
EnableServerDropDown, 55, 294
EnableWCRibbons, 55, 294
End Date, 213
Enforce Password Expiration, 282
Enforce Password Policy, 282
Enhanced Security, 104, 118, 121, 123, 125, 286
Add settings to target, 120
Company, 120
Copy Button, 119
Copy to current User in other Companies, 120
Copy to other Users in current Company, 119
Deny Based Security, 120
Display Excluded and Missing Resources, 120
Display only Selected Users, 121
Enhanced Security Legend, 119
Legend Button, 119
OK Button, 118
Options Menu, 121
Redisplay Button, 119
Refresh Application Navigation, 121
Reset target before copying, 120
Resource Info Button, 120
Scan for missing Menu Entries, 121
Security Button, 120
Security Information, 120, 123
Show Table Groups, 120
User ID, 120
Enhanced Security Legend, 119
Error Handling, 214
Every 1 Minute, 195
Every 10 Minutes, 195
Every 15 Minutes, 195
Every 30 Minutes, 195
Every 5 Minutes, 195
Every 60 Minutes, 195
Examples of use, 4
Exception Error Dialog, 220, 228
Exchange Web Services, 77
Exclude Selected Users and Companies rather than include them, 147, 174, 198, 221, 230, 239
Excluded from Security, 111
Execute, 223, 233, 241
Execute Button, 178, 179, 181, 220, 228, 238
Execute Change Script, 207, 300
Execute Dexterity SanScript code in the context of Product, 219
Execute Query in which SQL Database, 227
Execute Script for all Companies, 227
Execute Selection, 220, 228
Execution Mode, 214
Exempt user from system maximum sessions limit, 164
Exemptions, 266, 267
Exemption Mode, 267
Object Mode, 267
Remove All, 267
Remove Selected, 267
View Mode, 267
Exemptions Button, 267
Exit After Processes, 11, 81
Expanded Fields, 95
Expansion Button, 227
Export Body Section as One Line, 56
Export Button, 60, 95, 99, 107, 112, 115, 123, 125, 162, 180, 184, 230, 259
Export Current Table Record to XML, 203
Export Entire Table to XML restricted by Where Clause, 203
Export Mode, 95, 107, 112, 115, 123, 125, 162, 180, 230
Export Path, 259, 260, 262
Export Record, 190
Export Table, 190
ExportLinesPerPage, 56, 293
ExportOneLineBody, 56, 293
ExportPDFLinesPerPage, 56, 293
Extender Resources, 84, 98

F
Features Tab, 134
Feedback Survey, 31
Field, 84, 187, 201, 229
Change, 194, 195
Post, 194, 195
Pre, 194, 195
Field Background Color, 126
Field Context, 195
Field Context Menu, 196
Field Context Menu, 194, 195
Field Information, 97
field keyword, 229
Field Level Security, 150
Field Lookup, 89, 289
Field Name, 201, 203, 252
FileName, 61
File Path, 252, 291
Filter Menus, 109
Filter Tables having field, 86
Find …, 210, 222, 231, 240
Find a Window, 13, 81
Find Next, 210, 222, 232, 240
Fix Account Framework, 271, 276
Fix Framework Button, 271
Fix Tables, 271, 280
Fix Tables Button, 271
Fix Users and Databases, 269, 276
Fix Users Button, 269
Fix Utilities, 270, 276
Fix Utilities Button, 270
Focus Event, 194, 196, 203, 204, 213
Focus Event with Table, 194, 195, 196
Folder location for logs and export files, 9, 53
Folder on local drive on SQL Server, 73
Font Size, 211, 223, 233, 241, 296
Font Style, 211, 223, 233, 241, 296
Form, 84, 98, 200
Level, 194
Post, 194, 195
G

Generate Dexterity Pass Through, 211, 223, 233
Global
  Level, 194
Global Dex.ini, 43, 51, 134, 143, 204
GO Statement, 229
Go To Button, 102, 104, 115
Goto Line …, 210, 222, 232, 241
Gotos Button, 180, 231
GP Power Tools Administrator Password, 62, 133
GP Power Tools Area Page, 14, 26, 28, 31, 41, 45, 49, 51, 60,
  84, 98, 102, 110, 114, 118, 122, 124, 151, 155, 160, 163,
  171, 178, 179, 181, 189
GP Power Tools Feedback Survey, 31
GP Power Tools Logging Control, 12, 13
GP Power Tools Menus, 13
GP Power Tools Modules, 27
GP Power Tools Navigation Pane, 13
GP Power Tools Portal, 5
GP Power Tools Registration, 28
  Automatically check for updated keys, 29
  Contact Details, 28
  Email, 29
  Privacy Policy, 28
  Product Key, 29
  Trial Key, 29
  Update Keys, 29
GP Power Tools Settings, 284
GP Power Tools Setup, 284, 285, 327, 328, 329, 330, 331,
  340, 341
GP Power Tools Traces only, 39
GP Power Tools Update Check, 30
  Automatically check for updates, 30
GPPTools.cnk, 7
GPPTools.log, 37, 55, 66, 135, 191, 327
GPPTools.pdf, 7
GPPTools.txt, 7
GPPTools_<User>_ <Company>_<log, 37, 55, 66, 135, 191,
  327
GPPTools_<User>_ <Company>_ <Date>_<log, 37, 66

H

Help Button, 206, 220
Helper Button, 207, 220, 238
Helper Function Assistant, 207, 220, 238, 297, 368, 369, 370,
  371, 373, 375
  Adds Allowed, 207, 302, 304, 306, 308
  Execute Change Script, 207, 300
  Key Fields, 207
Helper Functions, 206, 207, 220, 238, 297
  Adds Allowed, 302, 304, 306, 308
DUOS, 323, 324, 325, 326, 332, 333, 334, 335
  Execute Change Script, 300
  MBS_Auto_Log, 327, 367
MBS_DUOS_Del, 334
MBS_DUOS_DelAll, 335
MBS_DUOS_Get, 332, 333
MBS_DUOS_Set, 332, 333, 334, 335
MBS.Export_SQL_Query_To_File, 314
MBS_Field_ParseText, 365
MBS_FormLookup, 338
MBS_FormLookup_Parameter, 339
MBS_Get_Table_Value1, 301
MBS_Get_Table_Value2, 303
MBS_Get_Table_Value3, 305
MBS_Get_Table_Value4, 307
MBS_Get_Window_Value, 299
MBS_getmsg, 363
MBS_Logging_Start, 330
MBS_Logging_Stop, 331
MBS_Net_Execute, 318, 322
MBS_Param_Del, 325
MBS_Param_DelAll, 326
MBS_Param_Get, 323, 324, 367
MBS_Param_Get_Boolean, 347
MBS_Param_Currency, 346
MBS_Param_Date, 348
MBS_Param_Get_boolean, 360
MBS_Param_Get_Currency, 359
MBS_Param_Get_Date, 361
MBS_Param_Set, 323, 324, 325, 326, 367
MBS_Parameter/bower, 323, 324, 367
MBS_Parameter_Currency, 346
MBS_Parameter_Date, 348
MBS_Parameter_Get, 360
MBS_Parameter_Get_Boolean, 360
MBS_Parameter_Get_Currency, 359
MBS_Parameter_Get_Date, 361
MBS_Parameter_Get_Number, 358
MBS_Parameter_Get_String, 357
MBS_Parameter_Get_Time, 357
MBS_Parameter_Load, 350
MBS_Parameter_Number, 345
MBS_Parameter_Reset, 343
MBS_Parameter_ResetBool, 343
MBS_Parameter_Reset_Currency, 354
MBS_Parameter_Reset_Date, 355
MBS_Parameter_Reset_Number, 352
MBS_Parameter_Reset_String, 351
MBS_Parameter_Reset_Time, 356
MBS_Parameter_Reset, 344
MBS_Parameter_Time, 349
MBS_Project_Start, 340
MBS_Project_Stop, 341
MBS_Runtime_Execute, 309, 320, 323, 324, 325, 326
MBS_Runtime_Execute_Background, 311
MBS_Runtime_Execute_Delayed, 312
MBS_Runtime_Execute_Modified, 310
MBS_Script_Load_Dex, 320
MBS_Script_Load_Net, 322
MBS_Script_Load_SQL, 321
MBS_Script_Substitute, 342
MBS_Security_Form_Check, 366
MBS_Set_Table_Value1, 302
MBS_Set_Table_Value2, 304
MBS_Set_Table_Value3, 306
MBS_Set_Table_Value4, 308
MBS_Set_Window_Value, 300
MBS_SQL_Chek_Exists, 313, 321
MBS_SQL_Goto_Get_Data, 317
MBS_SQL_Lookup, 336
MBS_SQL_Lookup_Parameter, 337
MBS_SQL_Results, 315
MBS_SQL_Results_Get_Data, 317
MBS_Secure_Wizard, 364
MBS_Trigger_Start, 328
MBS_Trigger_Stop, 329
SY_User_Object_Store, 323, 324, 325, 326, 332, 333, 334, 335
SY90000, 323, 324, 325, 326, 332, 333, 334, 335
Hidden Forms, 95, 158
Home Page, 13, 14, 42, 45, 49
Quick Links, 13, 14, 42, 45, 49
How to Setup, 187
HTML Table, 95, 107, 112, 115, 125, 162, 180, 230
If less than X MB, 203
Import Button, 60, 99, 184, 261
Import Path, 260, 261, 262
Import Settings File, 60
Import Utility, 84, 98
Inactive, 103
Include, 123
Include Current Launch File, 43, 134, 204
Include Dex.ini Settings File, 43, 134, 204
Include info for all databases, 43, 134, 204
Include sessions for all user types instead of just Full user, 164
Include User Dex.ini Settings File, 43, 134, 204
Include zipped log files, 203
Individual Logging Control, 38, 65
Info, 27
Info Button, 43, 143, 150
Information Button, 185
Insert Button, 207, 220, 238
Insert Helper Function, 207, 220, 238
Add Allowed, 207
Execute Change Script, 207
Key Fields, 207
Installation, 6, 7
Integration Manager, 216
Introduction, 1, 187
Issue Reject Record, 203
Issue Reject Script, 203
Keep Focus on Field, 204
KeepTemplateTempFiles, 57, 294
Key Fields, 207
Large SQL Profile Trace, 65, 215
Launch Configuration, 167, 168
Launch File, 15, 26, 43, 54, 147, 148, 150, 151, 153, 154, 167, 204
Launch File Configuration, 167, 168, 169, 285, 287
Add Button, 167
Advanced Security, 170
Bottom Button, 168
Delete Button, 168
Do not update the Launch File automatically, 169
Down Button, 168
Dynamics.set, 167
Launch File, 167
Launch File Configuration Additional Files, 168
Launch File Configuration Preview, 169
Launch File Rule, 167, 168
Preview, 169
Preview Button, 169
Rule Fields, 168
Rule List, 167
Smartlist, 170
Top Button, 168
Up Button, 168
Launch File Configuration Additional Files, 168
Launch File Configuration Preview, 169
Launch File Rule, 167, 168
Legend, 268
Legend Button, 104, 107, 119, 268
Letters, 84, 98
License.doc, 7
Limit results set to fixed number of lines, 227
Limited Users, 103
Link to Dexterity Script Debugger, 256
Linked Table, 87
List, 228
Log, 142
Log File, 191
Logging Options, 38, 65
Logging Password, 38, 64
Logging Settings, 20, 25, 35, 36, 37, 38, 39, 64, 65, 66, 68, 69, 71, 72, 73, 74, 288
Administrator Controlled Shared Folder Location, 64
Administrator Controlled Shared Folder Location for logs and export files, 64
Authentication Mode, 68
Capture
Maximum email attachment file size for zipped log files, 66
Number of days to keep logs, 66
Rename log each day, 66
Capture Dexterity Script Log, 65
Capture Dexterity Script Profile, 65
Capture Macro Recording, 65
Capture SQL Log, 65
Capture SQL Profile Trace, 65
Copy SQL Profile Trace files to Logs and Export files location, 74
Create SQL Profile Trace SQL Components, 74
Edit SQL Profile Trace Settings, 66
Enable Individual Logging Modes, 38
Enable Individual Logging Modes, 65
Folder on local drive on SQL Server, 73
Logging Password, 38, 64
Macro Recording Settings, 25
Maximum number of Trace files, 72
Maximum Trace file size, 72
Multi User Authentication Mode, 68
Process Multi User Mode SQL Server Action, 71
Process Single User Mode SQL Server Action, 71
Remove SQL Profile Trace SQL Components, 74
Single User Authentication Mode, 68
SQL Profile Trace Mode, 65
SQL Profile Trace Settings, 20, 35, 39, 65, 66, 67
UNC Network shared path to above Folder, 74
When Manual Logging is stopped, 65
Windows Administrator User ID, 69
Login Event, 195
Login Limits, 163, 287
Active Profile, 163, 164
Always allow access to this Company, 165
Default maximum sessions per User, 164
Duplicate Button, 165
Exempt user from system maximum sessions limit, 164
Include sessions for all user types instead of just Full user, 164
Maximum number of sessions for this Company, 165
Override maximum sessions per User, 164
Profile ID, 164, 165
Profile Name, 164
Login Remember User, 81
Login/Logout Event, 194, 195, 196
Macro Play Fast, 255
Macro Recording, 25, 35, 65, 215
Macro Recording Configuration, 25
Macro_recording_configuration, 25
Macro.mac, 35, 37
Macro_<User>_<Company>_<Date>_ <Time>.mac, 37
Manifest File, 59
Manual Logging Mode, 35, 36, 53, 64, 65, 66, 190, 330, 331
MAPI Compliant Client, 77
MBS_Auto_Log, 327, 367
MBS_Debug_Automate File, 290
MBS_Debug_Automate_Script, 290
MBS_Debug_Automate_Status, 290
MBS_Debug_LastRunSystem, 290
MBS_Debug_LastRunUser, 291
MBS_Debug_LaunchConfigurationOverride, 169, 285
MBS_Debug_LogAppDetails, 55, 285
MBS_Debug_LogListPath, 253, 254, 291
MBS_Debug_LogOnStartup, 53, 284
MBS_Debug_LogWinData, 291
MBS_Debug_LogWritePosition, 291
MBS_Debug_NamesUseClipboard, 291
MBS_Debug_Path, 53, 64, 284
MBS_Debug_RuntimeCheck, 284
MBS_Debug_SetupMode, 52, 284
MBS_Debug_ShowRuntime, 284
MBS_Debug_skipVersionChecks, 290
MBS_Debug_UpdateLastUserOnExit, 55, 285
MBS_Debug_VBADisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableReset, 149
MBS_Debug_VSTDisableRese...
Comma Delimited, 95
Expanded Fields, 95
Export Button, 95
Export Mode, 95
Filter Menus, 109
HTML Table, 95
Menu Command Details, 109
OK Button, 95
Tab Delimited, 95
Menu Explorer
  Missing Resources, 95
Message, 206
Microsoft Dynamics GP Import, 84, 98
Microsoft Outlook Client, 76
Minimize Log Entries, 197
Missing Resources, 95
Modal Dialog, 194, 195
Modified, 98, 146, 148, 200, 205, 219
Modified Alternate, 98
Modified/Alternate ID, 171, 173
Modified, 85, 200, 219, 310
Module
  Administrator Tools, 1, 2, 83, 84, 98, 110, 114, 117,
  118, 122, 124, 126, 142, 146, 151, 155, 160, 163, 167,
  171, 176
  Database Tools, 1, 3, 257, 258, 261, 263, 281
  Developer Tools, 1, 3, 177, 178, 179, 181, 182, 187, 217,
  225, 236, 243, 251, 255
  Preview Mode, 3
  System Module, 1, 34, 35, 41, 45, 49, 51, 60, 62, 64, 75,
  79, 80, 81
  Monthly Event, 195
  Mouse Explorer, 294
  Multi User Authentication Mode, 68

N
Name shown on Application title bar during initial loading,
  54
Names Button, 208, 220, 228, 238
Names Button Uses Clipboard, 212, 224, 234, 242, 291
Navigation, 11
  Application Menus, 13
  Application Tools Menu, 11
  Area Page, 14, 26, 28, 31, 41, 45, 49, 51, 60, 62, 64, 75,
  79, 80, 84, 98, 102, 110, 114, 118, 122, 124, 126, 142,
  146, 151, 155, 160, 163, 167, 171, 178, 181, 182,
  189, 192, 217, 225, 236, 243, 251, 258, 261, 263, 281
  GP Power Tools Area Page, 14, 26, 28, 31, 41, 45, 49, 51,
  60, 84, 98, 102, 110, 114, 118, 122, 124, 151, 155, 160,
  163, 171, 178, 181, 189
  GP Power Tools Menus, 13
  GP Power Tools Navigation Pane, 13
Navigation Pane, 13
  Options Button, 12, 26, 28, 31, 41, 45, 49, 51, 60, 62, 64,
  75, 79, 80, 84, 98, 102, 110, 114, 118, 122, 124, 126,
  142, 146, 151, 155, 160, 163, 167, 171, 178, 181, 182,
  189, 192, 217, 225, 236, 243, 251, 258, 261, 263, 281
Quick Links, 13
Standard Toolbar, 41, 45, 49
Tools Menu, 11, 12, 41, 45, 49
Web Client, 14
  Window Tools Menu, 12
Navigation Lists, 84, 98
Navigation Pane, 13
Net Execute, 288
Net Executer, 288
Non Logging All Except Disabled, 188
Non Logging Automatic Start Only, 188
Non Logging Triggers, 188, 189, 197, 203, 204, 213, 215, 243, 328, 329, 330, 331, 340, 341
Notes Button, 183, 193, 218, 226, 237, 244
Number of days prior to password expiry to start warnings, 132
Number of days to keep logs, 66
Number of execution logs to keep, 214
Number of Lines Per Page when Exporting Reports (inc. PDF), 56

ODBC, 55
OK Button, 85, 95, 99, 103, 107, 111, 115, 118, 122, 124, 127, 155, 161, 253, 265
Old Field Value, 189
OLEClose, 59, 295
Only include SQL Table & Views which have a DEX_ROW_ID column, 273
Only include tables which contain data, 89
Only require System or Administrator Password to be entered once per session, 133
Only restart selected logs when trigger fires, 215
Only show Service Enabled Procedures, 94, 95
Only Show Tables with Account Fields, 273
Open Application Maximized on next login, 55
Open Button, 86, 100, 111, 115, 122, 124, 161
Open Database Connectivity, 55
Open Script Debugger on Startup, 256
Open Window Hidden, 203
Open Windows, 43
Optional Where Clause, 203, 260
Options, 211, 223, 233, 241
Options Button, 12, 26, 28, 31, 41, 45, 49, 51, 60, 62, 64, 75, 79, 80, 84, 98, 102, 110, 114, 118, 122, 124, 126, 142, 146, 151, 155, 160, 163, 167, 171, 178, 191, 181, 182, 189, 192, 217, 225, 236, 243, 251, 258, 261, 263, 281
Options Menu, 186, 250
Save and Continue, 186, 250
Options Tab, 137, 159, 192, 213
Other SQL Profile Trace, 65, 215
Other Tab, 58
OUT_Condition, 206
Outlook, 76
Override maximum sessions per User, 164
Override to Convert Table Structures without using Dynamics Utilities, 271
Overwrite Duplicate Records, 262
Overwrite Table Contents, 262

Parameter Active, 245
Parameter Description, 244
Parameter Expansion Button, 247
Parameter From Value, 249
Parameter Hidden, 245
Parameter ID, 206, 213, 219, 220, 227, 228, 238, 244, 249
Parameter Instructions, 245
Parameter Length/Decimal, 246
Parameter List Dialog, 243, 245, 250
Parameter List Drop Down List Maintenance, 247
Parameter List Drop Down List SQL Script, 247
Parameter List Lookup Form Definition, 248
Parameter List Lookup SQL Script, 248
Parameter List Maintenance, 288
Parameter List Maintenance Additional Information, 247
Parameter Lists, 182, 206, 213, 219, 220, 227, 228, 238, 243
Down Button, 249
Duplicate Button, 249
Notes Button, 244
Options Menu, 250
Parameter Active, 245
Parameter Description, 244
Parameter Expansion Button, 247
Parameter From Value, 249
Parameter Hidden, 245
Parameter ID, 206, 213, 219, 220, 227, 228, 238, 244, 249
Parameter Instructions, 245
Parameter Length/Decimal, 246
Parameter List Dialog, 243, 245, 250
Parameter List Drop Down List Maintenance, 247
Parameter List Drop Down List SQL Script, 247
Parameter List Lookup Form Definition, 248
Parameter List Lookup SQL Script, 248
Parameter List Maintenance Additional Information, 247
Parameter Lists, 206, 213, 219, 220, 227, 228, 238
Parameter Maximum Value, 249
Parameter Minimum Value, 249
Parameter Mode, 245
Parameter Options, 246
Parameter Placeholder, 206, 220, 228, 238, 243
Parameter Prompt, 245
Parameter Single Value, 249
Parameter Title, 245
Parameter To Value, 249
Parameter Type, 245
Parameters Button, 206, 220, 228, 238
Preview Button, 250
Project ID, 245
Release Notes, 244
Save and Continue, 250
SQL Execute Setup, 247, 248
Timestamp Button, 244
Up Button, 249
Parameter Maximum Value, 249
Parameter Minimum Value, 249
Parameter Mode, 245
Parameter Options, 246
Parameter Placeholder, 206, 220, 228, 238, 243
Parameter Prompt, 245
Parameter Single Value, 249
Parameter Title, 245
Parameter To Value, 249
Parameter Type, 245
Parameters Button, 206, 220, 228, 238
Password, 78
Password Fields, 62
Pathname for Debugger.xml file, 80
Pathname location for Debugger Setup files, exports and logs, 37, 64, 190
Pathname location for SQL Log file, 52
Perform actions when fired and condition not met, 202
Perform actions when fired regardless of condition, 202

Performance SQL Profile Trace, 65, 215
Physical Name, 84, 229
POWERUSER Security Role, 176
Prefix for Disabled Companies, 153
Prevent application windows from opening outside of the visible desktop area, 131, 155
Preview, 48, 77
Preview Button, 169, 250
Preview Mode, 3
Print Button, 101, 108, 123, 125, 268
Print Report, 268
Privacy Policy, 28
Procedure, 84, 194, 196, 201
Procedure Name, 201
Process Button, 265
Process Monitor, 59, 139, 294
Process Multi User Mode SQL Server Action, 71
Process Single User Mode SQL Server Action, 71
Product Dictionary, 252
Product Key, 29
Product Name, 200, 219
Profile ID, 152, 153, 164, 165, 258, 259, 291
Profile Name, 152, 164, 259
Profile.txt, 35, 37
Profile_<User>_Company_Date_Time.txt, 37
Progress Window, 260, 262
Project Component List, 184
Project Description, 183
Project ID, 182, 184, 213, 219, 227, 238, 245
Project Setup, 61, 182, 287
.Net Execute Setup, 182
Add Button, 185
Automatic Trigger Mode, 182
Configuration File Path, 184, 185
Current Project, 183
Duplicate Button, 184
Export Button, 184
Import Button, 184
Information Button, 185
Notes Button, 183
Options Menu, 186
Parameter Lists, 182
Project Component List, 184
Project Description, 183
Project ID, 182, 184, 213, 219, 227, 238, 245
Release Notes, 183
Remove Project objects not being imported, 184
Reset Path Button, 185
Runtime Execute Setup, 182
Save and Continue, 186
SQL Execute Setup, 182
Start Button, 185
Stop Button, 185
Timestamp Button, 183
Transfer User and Company details, 184
Publish Script for Users, 221, 230, 239
Published to Executer Window, 178, 179, 181, 219, 227, 238

Q
Query Analyzer, 216, 225
QueueMoreInfo, 59, 294
Quick Links, 13, 14, 42, 45, 49

R
Raise All Windows, 11, 81
Read Record, 194
Recommended Configuration, 9, 15
Record.xml, 190
Record_User_Company_Date_Time.xml, 190
Redisplay Button, 79, 103, 107, 111, 115, 119, 122, 124, 161, 253, 265, 283
References, 240, 241
References Button, 240
Refresh Button, 43, 44
Register, 189
Registration, 28, 188
Registry, 59
Re-install, 27
Release Notes, 183, 193, 218, 226, 237, 244
Remember Last Company, 81
RememberUser, 81
Remove Attachment Button, 47
Remove Button, 47
Remove Exemption Button, 267
Remove Project objects not being imported, 184
Remove SQL Profile Trace SQL Components, 74
Rename DEXSQL.LOG at the beginning of each day, 52
Rename log each day, 66
Replace ..., 210, 222, 232, 240
Replace and Find Next, 210, 222, 232, 240
Report, 84, 98
Report Explorer, 93, 289
Back Up Button, 95
Comma Delimited, 95
Export Button, 95
Export Mode, 95
HTML Table, 95
OK Button, 95
Splitter, 95
Tab Delimited, 95
Screen Output, 57
ScreenOutput, 289
Report Writer Functions, 219, 367
ReportExplorer, 209
Reports Tab, 56
Reset Button, 157
Reset Buttons, 127
Reset Path Button, 185
Reset target before copying, 120
Reset User Passwords, 282
Reset User SQL Logins and Passwords, 275, 281
Reset Window Memory Settings, 157
Reset Window Position Memory Settings, 157
Reset Window Positions, 53
Resource Explorer, 91, 200, 208, 284, 289
Back Up Button, 95
Comma Delimited, 95
Expanded Fields, 95
Export Button, 95
Export Mode, 95
Hidden Forms, 95
HTML Table, 95
OK Button, 95
Only show Service Enabled Procedures, 94, 95
Tab Delimited, 95
Resource ID, 84
Save and Continue, 222
Script, 219
Script ID, 218, 220, 231, 320
Script Language, 224
Script Menu, 222
Script Name, 218
Select Custom Script Purpose, 219, 367, 376
Selected Users and Companies, 221
Service Enabled Procedure, 219
SmartList Builder Goto, 219
SQL Execute Setup, 225
SQL Gotos, 219
SQLExecuteGotoHandler, 231
Syntax Errors, 219
Timestamp Button, 218
Transact SQL, 219
URL Drill Backs, 219
Users Button, 221
WinthropDC.GpPowerToolsVB.dll, 219, 224

Runtime Executer, 178, 219, 287
Execute Button, 178
Script ID, 178
RW Functions, 219, 367
Runtime Execute Setup, 367, 368, 369, 370, 371, 372, 374, 376
rw_ReportEnd, 369
rw_ReportStart, 368
rw_TableHeaderCurrency, 371
rw_TableHeaderString, 370
rw_TableLineCurrency, 374
rw_TableLineString, 372
rw_ReportEnd, 369
rw_ReportStart, 368
rw_TableHeaderCurrency, 371
rw_TableHeaderString, 370
rw_TableLineCurrency, 374
rw_TableLineString, 372

S

SAMPLEDATEMSG, 54, 293
Sanscript, 178, 187, 207, 217, 220, 228, 284
Save and Continue, 186, 210, 222, 232, 241, 250
Save Button, 44
Save Path, 43, 44
Save Record, 194
Scheduled Event, 194, 195, 196, 214
Screen Output, 57, 289
ScreenShot, 41, 43, 134, 204, 288
Cancel Button, 44
Email Button, 44
Include Current Launch File, 43, 134, 204
Include Dex.ini Settings File, 43, 134, 204
Include info for all databases, 43, 134, 204
Include User Dex.ini Settings File, 43, 134, 204
Info Button, 43
Mark All Button, 44
Open Windows, 43
Refresh Button, 43, 44
Save Button, 44
Save Path, 43, 44
System Status, 41, 43, 44, 204
Unmark All Button, 44
Script, 84, 219, 227, 238
Script Context, 205

Script Debugger Context, 255
Script Editor Settings, 296
Script Explorer, 94, 209, 289
Only show Service Enabled Procedures, 94, 95
Script ID, 178, 179, 181, 218, 220, 226, 229, 237, 238, 320, 321, 322
Script Language, 224, 238
Visual Basic.Net, 238
Visual C#, 238
Script Menu, 210, 222, 231, 240
Check Syntax, 210, 221, 232, 241
Convert References, 232
Execute, 223, 233, 241
Find …, 210, 222, 231, 240
Find Next, 210, 222, 232, 240
Font Size, 211, 223, 233, 241, 296
Font Style, 211, 223, 233, 241, 296
Generate Dexterity Pass Through, 211, 223, 233
Goto Line …, 210, 222, 232, 241
Names Button Uses Clipboard, 212, 224, 234, 242, 291
Options, 211, 223, 233, 241
References, 241
Replace …, 210, 222, 232, 240
Replace and Find Next, 210, 222, 232, 240
Save and Continue, 210, 222, 232, 241
Syntax Highlighting, 296
Script Name, 218, 226, 237
Script Tab, 192, 205
Script.log, 35, 37
Script_<User>_<Company>_<Date>_<Time>.log, 37
ScriptCommentColor, 296
ScriptDebugger, 52, 292
ScriptDebuggerProduct, 52, 292
ScriptEditorFontName, 296
ScriptEditorFontSize, 296
ScriptEditorSyntaxColoring, 296
ScriptErrorColor, 296
ScriptIdentifierColor, 296
ScriptKeywordColor, 296
ScriptLogEnhanced, 52, 292
ScriptNumberOfColor, 296
ScriptOperatorColor, 296
ScriptStringColor, 296
Scroll
Change, 194, 195
Delete, 194, 195
Fill, 194, 195
Insert, 194, 195
Post, 194, 195
Pre, 194, 195
Scroll Fill, 203
Scrolling Window Line Color, 126
Search Again Button, 85
Search Mode, 86
Search Results, 85
Security, 8, 104, 116, 120
Alternate/Modified Forms and Reports, 104, 116
Security Role Setup, 104, 116
Security Roles, 10
Security Task Setup, 104, 116
User Security Setup, 10, 104, 116
Security Analyzer, 114, 116, 286
Comma Delimited, 115
Detail Format, 114, 116
Export Button, 115
Export Mode, 115
Go To Button, 115
HTML Table, 115
OK Button, 115
Open Button, 115
Options Menu, 116
Redisplay Button, 115
Refresh Resource Information Table, 116
Right click enabled, 116
Security Button, 115
Splitter, 116
Summary Format, 114
SY09400, 114, 116
syCurrentResources, 114, 116
System Level Queries, 114
Tab Delimited, 115
Users & Companies Queries, 114
Security Button, 86, 102, 104, 115, 120, 123, 125
Security Button Drop List, 100, 102
Security Denied, 104, 120, 122, 123, 125
Comma Delimited, 123
Company, 123
Company ID, 123
Delete Button, 122
Display Mode, 123
Export Button, 123
Export Mode, 123
HTML Table, 123
Include, 123
Mark All Button, 123
OK Button, 122
Open Button, 122
Options Menu, 123
Print Button, 123
Redisplay Button, 122
Refresh Application Navigation, 123
Security Button, 123
Sort Mode, 123
Tab Delimited, 123
Unmark All Button, 123
User ID, 123
Security Hidden, 104, 124
Comma Delimited, 125
Company, 125
Company ID, 125
Delete Button, 124
Display Mode, 125
Export Button, 125
Export Mode, 125
HTML Table, 125
Mark All Button, 125
OK Button, 124
Open Button, 124
Options Menu, 125
Print Button, 125
Redisplay Button, 124
Refresh Application Navigation, 125
Security Button, 125
Sort Mode, 125
Tab Delimited, 125
Unmark All Button, 125
User ID, 125
Security Information, 86, 100, 102, 108, 111, 115, 120, 123, 286
Company, 103
Deny Based Security, 104
Filter Menus, 109
Go To Button, 102, 104
Inactive, 103
Legend Button, 104
Menu Command Details, 109
Menu Explorer, 109
OK Button, 103
Options Menu, 108
Redisplay Button, 103
Refresh Resource Information Table, 108
Resource Info Button, 104
Security Button, 104
Security Information Legend, 104
Security Information Resources, 104, 106
Security Information SQL Role Views, 105
Show All SQL Users & Databases, 106
Show only Selected, 103
Show Resources Button, 104
Splitter, 105
SUPERUSER Security Role, 108
SUPERUSER Security Task, 108
SY09400, 108
syCurrentResources, 108
User ID, 103
Security Information Legend, 104, 107
Security Information Resources, 104, 106, 286
Comma Delimited, 107
Display Security Tasks and Roles, 108
Export Button, 107
Export Mode, 107
HTML Table, 107
Legend Button, 107
OK Button, 107
Print Button, 108
Redisplay Button, 107
Security Information Legend, 107
Show Series, 107
Tab Delimited, 107
Unmark All Button, 112
Security Information SQL Role Views, 105
Security Log, 79, 110, 135, 286
Comma Delimited, 112
Company, 111
Company ID, 111
Create/Update Security Task from Log, 111
Create/update Security Task from selected rows, 111
Details, 112
Details Button, 112
Display Mode, 110
Excluded from Security, 111
Export Button, 112
Export Mode, 112
HTML Table, 112
Mark All Button, 112
OK Button, 111
Open Button, 111
Redisplay Button, 111
Right click enabled, 113
Security Button Drop List, 111
Security Log Details, 112
Sort Mode, 111
Tab Delimited, 112
Unmark All Button, 112
User ID, 111
Security Log Details, 112, 286
Security Object Explorer, 93, 289
Security Objects, 84, 98
  Customization Tools, 84, 98
  Document Access, 84, 98
  Extender Resources, 84, 98
  Import Utility, 84, 98
Letters, 84, 98
  Microsoft Dynamics GP Import, 84, 98
Navigation Lists, 84, 98
Security Object Explorer, 93
Series Posting Permissions, 84, 98
SmartList Builder Permissions, 84, 98
SmartList Objects, 84, 98
Unknown Objects, 84, 94, 98
Security Privileges, 98
Security Profiler, 98, 101, 134, 135, 286
Access Denied, 99
Alternate, 98
Application Level Security, 98, 99
Automatic Open Mode, 101, 134
Clear Button, 100
Create/Update Security Task, 100
Customization Tools, 98
Document Access, 98
Export Button, 99
Extender Resources, 98
Form, 98
Import Button, 99
Import Utility, 98
Letters, 98
  Microsoft Dynamics GP Import, 98
Modified, 98
Modified Alternate, 98
Navigation Lists, 98
OK Button, 99
Open Button, 100
Options Menu, 101
Print Button, 101
Refresh Application Navigation, 101
Report, 98
Right click enabled, 101, 102
Security Button, 102
Security Button Drop List, 100
Security Objects, 98
Security Privileges, 99
Security Profiler Log, 99
Series Posting Permissions, 98
SmartList Builder Permissions, 98
SmartList Objects, 98
SQL Server Security, 99
Start Capture of Resources and Security Objects, 100
Stop Capture and create/update Security Task, 100
Table, 98
Unknown Objects, 98
Windows Level Security, 99
Security Profiler Log, 99
Security Resource Descriptions, 176
Security Role Setup, 104, 116
Security Roles, 10
Security Task Setup, 104, 116
Select Associated Table
  Filter Tables having field, 86
Select Associated Table, 86
Select Buttons, 127
Select Custom Script Purpose, 219, 367, 376
Select Table Containing Field, 89
  Only include tables which contain data, 89
Select Theme, 126, 129
Selected Users and Companies, 147, 174, 198, 221, 230, 239
Selection List, 173
Send Button, 47, 77
Send Email, 45, 77, 289
  Add Attachment Button, 47
  Add Button, 47
  Administrator Email, 47
  Attachments, 47
  Bcc Button, 47
  Bcc Field, 47
  Body, 47, 76
  Body Text, 47, 76
  Cancel Button, 48
  Cc Button, 47
  Cc Field, 47
  Default Body Text, 47
  Default Subject, 47
  From Field, 47
  Remove Attachment Button, 47
  Remove Button, 47
  Send Button, 47, 77
  Sender’s Email, 47
  Subject, 47, 75
  To Button, 47
  To Field, 47
Send Email using Administrator Email or Email Address below, 203
Send HTML, 77
Send Password changed emails, 281
Sender’s Email, 47, 77
Series Posting Permissions, 84, 98
Service Enabled Procedure, 94, 102, 219
Service Enabled Procedures, 376
  Runtime Execute Setup, 377, 379, 381, 383, 385
  ServiceCreateCustom, 377
  ServiceDeleteCustom, 379
  ServiceGetCustom, 381
  ServicePostCustom, 385
  ServiceUpdateCustom, 383
Service Procedures, 376
  Runtime Execute Setup, 377, 379, 381, 383, 385
  ServiceCreateCustom, 377
  ServiceDeleteCustom, 379
  ServiceGetCustom, 381
  ServicePostCustom, 385
  ServiceUpdateCustom, 383
ServiceCreateCustom, 377
ServiceDeleteCustom, 379
ServiceGetCustom, 381
ServicePostCustom, 385
ServiceUpdateCustom, 383
Setting or Search String, 142, 143
Settings Applied Message, 134
Settings List, 142
Setup, 192, 284
Setup Backup and Restore, 80
  Pathname for Debugger.xml file, 80
  Restore Button, 80
  Setup Button, 80
Setup Mode, 52, 284
Short Description used for dialog buttons, 173
Show Advanced Macro Menu, 58
Show All Menu Items, 58
Show All SQL Users & Databases, 106
Show currently selected Window and Field information, 86
Show Debug Messages on next login, 52
Show Dexterity Technical Name Syntax Button, 228
Show Disabled Companies, 153
Show keyword, 229
Show Launch File, 150
Show only Selected, 103
Show Resources Button, 104
Show Series, 107
Show SQL Profile Traces, 39
Show Structure Errors Button, 274
Show AdvancedMacroMenu, 58, 293
ShowAllMenuItems, 58, 293
ShowDebugMessages, 52, 292
Silent, 142
Single User Authentication Mode, 68
SkipVersionChecks, 293
Small SQL Profile Trace, 65, 215
Smartlist, 136, 170
SmartList Builder Goto, 219
SmartList Builder Permissions, 84, 98
SmartList Objects, 84, 98
SMTP Server, 77
SMTP Server Port, 77
SMTP Server via CDO, 76
Sort Mode, 111, 123, 125, 161
Spinner Controls, 127
Splitter, 95, 105, 116
SQL Database, 227
SQL Execute
   Goto Line …, 232
SQL Execute Setup, 179, 182, 225, 243, 247, 248, 287, 321
   alias keyword, 229
   All Users and Companies, 230
   Check Syntax, 232
   Comma Delimited, 230
   Convert References, 232
   Database, 227
   Display Name, 229
   Divider Adjustment Buttons, 228
   Duplicate Button, 229
   Exception Error Dialog, 228
   Exclude Selected Users and Companies rather than include
   them, 230
   Execute, 233
   Execute Button, 228
   Execute Query in which SQL Database, 227
   Execute Script for all Companies, 227
   Execute Selection, 228
   Expansion Button, 227
   Export Button, 230
   Export Mode, 230
   field keyword, 229
   Find …, 231
   Find Next, 232
   Font Size, 233
   Font Style, 233
   Generate Dexterity Pass Through, 233
   GO Statement, 229
   Gotos Button, 231
   HTML Table, 230
   Limit results set to fixed number of lines, 227
   List, 228
   Names Button, 228
   Names Button Uses Clipboard, 234
   Notes Button, 226
   Options, 233
   Parameter ID, 227, 228
   Parameter Lists, 227, 228
   Parameters Button, 228
   Physical Name, 229
   Project ID, 227
   Publish Script for Users, 230
   Published to Executer Window, 179, 227
   Query Analyzer, 225
   Release Notes, 226
   Replace …, 232
   Replace and Find Next, 232
   Save and Continue, 232
   Script, 227
   Script ID, 226, 229, 321
   Script Menu, 231
   Script Name, 226
   Selected Users and Companies, 230
   Show Dexterity Technical Name Syntax Button, 228
   show keyword, 229
   SQL Database, 227
   SQL Execute Setup Gotos, 231
   SQL Gotos, 180, 231
   SQLExecuteGotoHandler, 231
   Tab Delimited, 230
   Table Explorer, 228
   Text, 228
   Timestamp Button, 226
   Transact SQL, 179, 225, 227, 229
   Users Button, 230
   WinthropDC.GpPowerToolsVB.dll, 234
SQL Execute Setup Gotos, 231
Add Button, 231
Bottom Button, 231
Down Button, 231
Top Button, 231
Up Button, 231
SQL Execute, 179, 227, 238
   Comma Delimited, 180
   Execute Button, 179
   Export Button, 180
   Export Mode, 180
   Gotos Button, 180
   HTML Table, 180
   Script ID, 179
   Tab Delimited, 180
   SQL Gotos, 180, 219, 231, 316, 317
   SQL Logging, 35, 53, 65, 215
SQL Login Maintenance, 275, 281, 287
   Apply Advanced SQL Server options, 282
   Apply Button, 283
   Apply User Status, 282
   Automatically Generate Passwords, 282
   Cancel Button, 283
   Change Password Next Login, 282
   Database Validation Email Settings, 282
   Email Settings, 282
   Enforce Password Expiration, 282
   Enforce Password Policy, 282
   Mark All Button, 283
Redisplay Button, 283
Reset User Passwords, 282
Send Password changed emails, 281
Unmark All Button, 283
User List, 281
User Password, 282
User Status, 282
SQL Native Client, 55
SQL Profile Trace Application, 39
SQL Profile Trace Mode, 65, 215
SQL Profile Trace Settings, 67
SQL Profile Trace User, 39
SQL Profile Traces, 20, 39, 40
Active SQL Profile Traces, 39, 40
All Traces on SQL Server, 39
All Users, 39
Current User only, 39
GP Power Tools Traces only, 39
Show SQL Profile Traces, 39
SQL Profile Trace Application, 39
SQL Profile Trace User, 39
SQL Profile Tracing Configuration, 20
Stop SQL Profile Trace, 39
Stranded SQL Profile Traces, 39
SQL Profile Tracing, 20, 35, 39, 65, 67, 215
Large, 65, 215
Medium, 65, 215
Other, 65, 215
Performance, 65, 215
Small, 65, 215
SQL Profile Tracing Configuration, 20
SQL Results, 288, 315, 316, 317
SQL Gotos, 316, 317
SQL Server, 32, 35, 65, 67, 85, 215
SQL Logging, 35, 65, 215
SQL Profile Tracing, 35, 65, 67, 215
SQL Server Security, 99
SQLEnexecuteGotoHandler, 231
SQLLastCompany, 81, 285
SQLLogAllODBCMessages, 292
SQLLoginCompatibilityMode, 54, 293
SQLLogODBCMessages, 52, 292
SQLLogPath, 52, 292
SQLLogRename, 52, 285
SQLLogStmt, 52, 292
Standard Mode, 1, 8, 35, 37, 38, 39, 41, 45, 49, 51, 60, 64, 65, 84, 98, 102, 110, 114, 118, 122, 124, 135, 178, 179, 181, 190, 260, 330, 331
.Net Executer, 181, 238
Calculator, 49
Configuration Export/Import, 60, 80, 260
Dex.ini Settings, 37, 51, 64, 190
Enhanced Security, 118
Individual Logging Control, 38, 65
Logging Options, 38, 65
Manual Logging Mode, 35, 330, 331
Resource Information, 84
Runtime Executer, 178, 219
ScreenShot, 41, 204
Security Analyzer, 114
Security Denied, 122
Security Hidden, 124
Security Information, 86, 100, 102, 111, 115
Security Log, 79, 110, 135
Security Profiler, 98, 135
Send Email, 45
SQL Executer, 179, 227
SQL Profile Traces, 39
Standard Signature to add to all emails, 76
Standard Toolbar, 41, 45, 49
Start Button, 185
Start Capture of Resources and Security Objects, 100
Start Date, 213
Start Logging on next startup only, 38, 53
Dexterity Profile, 53
Dexterity Script, 53
SQL Logging, 53
Start Trigger Automatically on Login, 52, 187, 197
Start Trigger Automatically on Login for Users, 198
Startup Tab, 54
Static Values, 90
Stop Button, 185
Stop Capture and create/update Security Task, 100
Stop SQL Profile Trace, 39
Stop Trigger after Condition met, 215
Stranded SQL Profile Traces, 39
Subject, 47, 75
Summary Format, 114
SUPERUSER Security Role, 108, 176
SUPERUSER Security Task, 108, 176
SUPERUSER Security Task and Role, 176
SUPERUSER Workflow Setup, 176
Support, 5
Suppress Date Change Dialog, 58
Suppress Sample Company Date Warning, 54
Suppress Sound from Application, 58
SuppressChangeDateDialog, 58, 187, 197
SuppressSound, 58, 293
Survey, 31
SY_User_Object_Store, 323, 324, 325, 326, 332, 333, 334, 335
SY09400, 108, 114, 116
SY90000, 323, 324, 325, 326, 332, 333, 334, 335
syCurrentResources, 108, 114, 116
Syntax Errors, 206, 219, 238
Syntax Highlighting, 296
System Level Queries, 114
System Module, 1, 34, 35, 41, 45, 49, 51, 60, 62, 64, 75, 79, 80, 81
Additional System Features, 81
Administrator Password Setup, 62
Calculator, 49
Configuration Export/Import, 60
Configuration Maintenance, 79
Dex.ini Settings, 51
Email Settings, 75
Logging Settings, 64
Manual Logging Mode, 35
ScreenShot, 41
Send Email, 45
Setup Backup and Restore, 80
System Password, 32, 62, 133
System Settings, 292
System Status, 41, 43, 44, 204
T
Tab Delimited, 95, 107, 112, 115, 123, 125, 162, 180, 230
Table, 84, 98, 187, 194, 196, 200, 229
Table Descriptions, 96
Table Explorer, 92, 118, 200, 208, 228, 284, 289
  Back Up Button, 95
  Comma Delimited, 95
  Expanded Fields, 95
  Export Button, 95
  Export Mode, 95
  HTML Table, 95
  OK Button, 95
  Tab Delimited, 95
  Table Groups, 93
  Table Group, 84
  Table Groups, 93
  Table Keys, 88
  Table Keys Lookup, 88, 289
  Table Lookup, 259
  Table Name, 200
  Table Physical Name, 259
  Table restricted to Form, 194, 196
  Table Structure Errors, 274
  Table Technical Name, 259
  Table.xml, 190
  Table_<User>_<Company>_<Date>_<Time>.xml, 190
  Tables Containing Field Button, 89
  Target Dex.ini, 143
  Technical Name, 84, 200, 201
  Terminal Server, 76
  Test Button, 141
  Text, 228
  Theme Group, 126, 129
  Theme Name, 126, 129
  Third Party Dictionary, 146
  Timed Event, 194, 195, 196
  Timestamp Button, 183, 193, 218, 226, 237, 244
  To Button, 47
  To Field, 47
  Tools Menu, 11, 12, 41, 45, 49
  Top Button, 148, 153, 154, 168, 173, 231
  TPELogging, 57, 294
  Trace.trc, 35, 37
  Trace_<User>_<Company>_<Date>_<Time>_<Mode>.trc, 37
  Transact SQL, 179, 219, 225, 227, 229
  Transaction being Edited, 82
  Transfer User and Company details, 61, 184
  Trial Key, 29
  Trigger, 146, 187, 188, 189, 190, 192, 196, 216
  Trigger Administration, 199
    Change Start Mode Button, 199
    Change State Button, 199
    Mark To Delete Button, 199
  Trigger Attach, 196
    After Logging In, 196
    After Login Event, 196
    After Login on Day X, 196
    After Login on DOW, 196
    After Logout Event, 196
    After Menu Selected, 196
    After Original, 196
    After Table Event, 196
    After Time XX XX, 196
    After Timed Event, 196
    Before Original, 196, 203, 204
  Trigger Description, 193
  Trigger Event, 187, 194, 200, 206
    Add Menu Below Entry, 195
    Add Menu to Bottom, 195
    Add Menu to Top, 195
    Context Menu, 194, 195
    Daily Event, 195
    Delete Record, 194
    Every 1 Minute, 195
    Every 10 Minutes, 195
    Every 15 Minutes, 195
    Every 30 Minutes, 195
    Every 5 Minutes, 195
    Every 60 Minutes, 195
    Field Change, 194, 195
    Field Context, 195
    Field Post, 194, 195
    Field Pre, 194, 195
    Form Level, 194, 195
    Form Post, 194, 195
    Form Pre, 194, 195
    Global Level, 194
    Login Event, 195
    Logout Event, 195
    Modal Dialog, 194, 195
    Monthly Event, 195
    Read Record, 194
    Save Record, 194
    Scroll Change, 194, 195
    Scroll Delete, 194, 195
    Scroll Fill, 194, 195, 203
    Scroll Insert, 194, 195
    Scroll Post, 194, 195
    Scroll Pre, 194, 195
    Warning Dialog, 195
    Weekly Event, 195
    Window Activate, 194, 195
    Window Post, 194, 195
    Window Pre, 194, 195, 203
  Trigger ID, 79, 187, 188, 190, 192, 193, 197
  Trigger Mode, 252
  Trigger Setup, 192, 243
  Trigger Setup Scheduled Log, 214
  Trigger Status, 146, 189, 197, 285
    Register, 189
    Unregister, 189, 197
  Trigger Type, 194, 196, 200, 206, 216, 252
    Add Field Context Menu, 213
    Add Form Menu, 213
    Application Level Menu, 194, 195, 196, 201
    Field Context Menu, 194, 195, 196
    Focus Event, 194, 196, 203, 204, 213
    Focus Event with Table, 194, 195, 196
    Form Level Menu, 194, 195, 196
    Function, 194, 196
    Login/Logout Event, 194, 195, 196
    Procedure, 194, 196
    Scheduled Event, 194, 195, 196, 214
    Table, 194, 196
    Table restricted to Form, 194, 196
    Timed Event, 194, 195, 196
    Warning Dialog, 194, 195, 196
  Triggering, 190
U
UAC, 7, 27, 59, 148, 149
UNC Network shared path to above Folder, 74
Uninstall, 26
Unknown Objects, 84, 94, 98
Unmark All Button, 44, 112, 123, 125, 156, 283
Unregister, 189, 197
Up Button, 148, 153, 154, 168, 173, 231, 249
Update Check, 30
Update Keys, 29
Update last User ID and Company on exit, 55
URL Drill Backs, 219
Usability Tab, 130
Use separate password instead of System Password, 62
Use SQL Login Compatibility Mode, 54
User Account Control, 7, 27, 59, 148, 149
User Activity Log, 139, 141, 160, 287
Comma Delimited, 162
Company, 161
Company ID, 161
Days to keep daily Max User Count data for, 139
Display Mode, 160
Enable User Activity Tracking, 139
Export Button, 162
Export Mode, 162
HTML Table, 162
Maximum Users, 161
OK Button, 161
Open Button, 161
Redisplay Button, 161
Sort Mode, 161
Tab Delimited, 162
User Activity Log Maximum Users, 161
User ID, 160
User Setup, 161
User Activity Log Maximum Users, 161, 287
User Button, 173
User Company Access Fix, 176
User Defined Date, 138, 139
User Defined String, 138, 139
User Email Address, 137, 281
User ID, 78, 103, 111, 120, 123, 125, 160
User List, 281
User Password, 282
User Preferences, 157
User Preferences Apply, 81
User Security Setup, 8, 10, 104, 116
User Setup, 137, 138, 161, 176, 281
User Setup Additional Information, 137, 138, 139, 176, 270, 281
Default Site ID, 138
Employee ID, 137
User Defined Date, 138, 139
User Defined String, 138, 139
User Email Address, 137, 281
User Status, 282
Users & Companies Queries, 114
Users and Databases, 264
Users Button, 173, 174, 197, 198, 221, 230, 239
Using Database Validation, 276

V
Validate Button, 266
Value, 142
VBA, 35, 149, 216
VBADisable, 149, 294
Visual Basic for Applications, 35, 149, 216
Visual Basic.Net, 7, 149, 181, 236, 238, 318, 322, 376
Visual C#, 7, 149, 181, 236, 238, 318, 322, 376
Visual Studio Call, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386
ServiceCreateCustom, 377, 378
ServiceDeleteCustom, 379, 380
ServiceGetCustom, 381, 382
ServicePostCustom, 385, 386
ServiceUpdateCustom, 383, 384
Visual Studio Integration Toolkit, 195
Visual Studio Tools, 90, 149
VSTools, 149

W
Warning Dialog, 194, 195, 196
Warnings, 158
Web Client, 14, 33, 36, 42, 44, 45, 49, 77, 95, 103, 105, 116, 127, 131, 149, 150
Web Service, 377, 379, 381, 383, 385
ServiceCreateCustom, 377
ServiceDeleteCustom, 379
ServiceGetCustom, 381
ServicePostCustom, 385
ServiceUpdateCustom, 383
Weekly Event, 195
When Manual Logging is stopped, 65
When only X% of licenses available, 140
Window, 84, 201
Activate, 194, 195
Post, 194, 195
Pre, 194, 195
Window Background Color, 126
Window Descriptions, 96
Window Heading Color, 126
Window Name, 201
Window Position Control
  Reset Window Positions, 158
Window Position Memory, 131, 137, 155, 159, 286
Cancel Button, 155
Default Button, 156
Hidden Forms, 158
Mark All Button, 155
OK Button, 155
Reset Button, 157
Reset Window Memory Settings, 157
Reset Window Position Memory Settings, 157
Unmark All Button, 156
User Preferences, 157
Warnings, 158
Window Pre, 203
Window Toolbar Color, 126
Window Tools Menu, 12
Window/Table/Procedure/Function Name, 252
WindowHeight, 55, 294
WindowMax, 55, 294
WindowPosX, 55, 294
WindowPosY, 55, 294
GP POWER TOOLS INDEX

Windows Administrator User ID, 69
Windows Bitmap Font Registry Settings, 59
Windows Bitmap Scaling Settings, 59
Windows Level Security, 99
Windows Start Bar, 131
WindowWidth, 55, 294
WinthropDC.GpPowerToolsVB.dll, 7, 181, 200, 205, 212, 219, 224, 234, 236, 242
WinthropDC.GpPowerToolsVC.dll, 7, 181, 236

X

XML Table Export, 258, 287
  Duplicate Button, 259
  Export Path, 259, 260, 262

Optional Where Clause, 260
Profile ID, 258, 259
Profile Name, 259
Progress Window, 260, 262
Table List, 259
Table Physical Name, 259
Table Technical Name, 259
XML Table Import, 261, 287
  Duplicate Records, 262
  Import Button, 261
  Import Path, 260, 261, 262
  Overwrite Duplicate Records, 262
  Overwrite Tables Contents, 262
  Progress Window, 260, 262

** End of document - GPPTools.docx - DM - 3 January 2020 **