

OFFICIAL MICROSOFT LEARNING PRODUCT

10775A

**Administering Microsoft®
SQL Server® 2012 Database**

MCT USE ONLY. STUDENT USE PROHIBITED

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

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

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2012 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners

Product Number: 10775A

Part Number: X18-29125

Released: 05/2012

MICROSOFT LICENSE TERMS
OFFICIAL MICROSOFT LEARNING PRODUCTS
MICROSOFT OFFICIAL COURSE Pre-Release and Final Release Versions

These license terms are an agreement between Microsoft Corporation and you. Please read them. They apply to the Licensed Content named above, which includes the media on which you received it, if any. These license terms also apply to any updates, supplements, internet based services and support services for the Licensed Content, unless other terms accompany those items. If so, those terms apply.

BY DOWNLOADING OR USING THE LICENSED CONTENT, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO NOT DOWNLOAD OR USE THE LICENSED CONTENT.

If you comply with these license terms, you have the rights below.

1. DEFINITIONS.

- a. "Authorized Learning Center" means a Microsoft Learning Competency Member, Microsoft IT Academy Program Member, or such other entity as Microsoft may designate from time to time.
- b. "Authorized Training Session" means the Microsoft-authorized instructor-led training class using only MOC Courses that are conducted by a MCT at or through an Authorized Learning Center.
- c. "Classroom Device" means one (1) dedicated, secure computer that you own or control that meets or exceeds the hardware level specified for the particular MOC Course located at your training facilities or primary business location.
- d. "End User" means an individual who is (i) duly enrolled for an Authorized Training Session or Private Training Session, (ii) an employee of a MPN Member, or (iii) a Microsoft full-time employee.
- e. "Licensed Content" means the MOC Course and any other content accompanying this agreement. Licensed Content may include (i) Trainer Content, (ii) sample code, and (iii) associated media.
- f. "Microsoft Certified Trainer" or "MCT" means an individual who is (i) engaged to teach a training session to End Users on behalf of an Authorized Learning Center or MPN Member, (ii) currently certified as a Microsoft Certified Trainer under the Microsoft Certification Program, and (iii) holds a Microsoft Certification in the technology that is the subject of the training session.
- g. "Microsoft IT Academy Member" means a current, active member of the Microsoft IT Academy Program.
- h. "Microsoft Learning Competency Member" means a Microsoft Partner Network Program Member in good standing that currently holds the Learning Competency status.
- i. "Microsoft Official Course" or "MOC Course" means the Official Microsoft Learning Product instructor-led courseware that educates IT professionals or developers on Microsoft technologies.

- j. "Microsoft Partner Network Member" or "MPN Member" means a silver or gold-level Microsoft Partner Network program member in good standing.
 - k. "Personal Device" means one (1) device, workstation or other digital electronic device that you personally own or control that meets or exceeds the hardware level specified for the particular MOC Course.
 - l. "Private Training Session" means the instructor-led training classes provided by MPN Members for corporate customers to teach a predefined learning objective. These classes are not advertised or promoted to the general public and class attendance is restricted to individuals employed by or contracted by the corporate customer.
 - m. "Trainer Content" means the trainer version of the MOC Course and additional content designated solely for trainers to use to teach a training session using a MOC Course. Trainer Content may include Microsoft PowerPoint presentations, instructor notes, lab setup guide, demonstration guides, beta feedback form and trainer preparation guide for the MOC Course. To clarify, Trainer Content does not include virtual hard disks or virtual machines.
2. **INSTALLATION AND USE RIGHTS.** The Licensed Content is licensed not sold. The Licensed Content is licensed on a one copy per user basis, such that you must acquire a license for each individual that accesses or uses the Licensed Content.
- 2.1 Below are four separate sets of installation and use rights. Only one set of rights apply to you.
- a. **If you are a Authorized Learning Center:**
- i. If the Licensed Content is in digital format for each license you acquire you may either:
 - 1. install one (1) copy of the Licensed Content in the form provided to you on a dedicated, secure server located on your premises where the Authorized Training Session is held for access and use by one (1) End User attending the Authorized Training Session, or by one (1) MCT teaching the Authorized Training Session, or
 - 2. install one (1) copy of the Licensed Content in the form provided to you on one (1) Classroom Device for access and use by one (1) End User attending the Authorized Training Session, or by one (1) MCT teaching the Authorized Training Session.
 - ii. You agree that:
 - 1. you will acquire a license for each End User and MCT that accesses the Licensed Content,
 - 2. each End User and MCT will be presented with a copy of this agreement and each individual will agree that their use of the Licensed Content will be subject to these license terms prior to their accessing the Licensed Content. Each individual will be required to denote their acceptance of the EULA in a manner that is enforceable under local law prior to their accessing the Licensed Content,
 - 3. for all Authorized Training Sessions, you will only use qualified MCTs who hold the applicable competency to teach the particular MOC Course that is the subject of the training session,
 - 4. you will not alter or remove any copyright or other protective notices contained in the Licensed Content,

5. you will remove and irretrievably delete all Licensed Content from all Classroom Devices and servers at the end of the Authorized Training Session,
6. you will only provide access to the Licensed Content to End Users and MCTs,
7. you will only provide access to the Trainer Content to MCTs, and
8. any Licensed Content installed for use during a training session will be done in accordance with the applicable classroom set-up guide.

b. If you are a MPN Member.

- i. If the Licensed Content is in digital format for each license you acquire you may either:
 1. install one (1) copy of the Licensed Content in the form provided to you on (A) one (1) Classroom Device, or (B) one (1) dedicated, secure server located at your premises where the training session is held for use by one (1) of your employees attending a training session provided by you, or by one (1) MCT that is teaching the training session, **or**
 2. install one (1) copy of the Licensed Content in the form provided to you on one (1) Classroom Device for use by one (1) End User attending a Private Training Session, or one (1) MCT that is teaching the Private Training Session.
- ii. You agree that:
 1. you will acquire a license for each End User and MCT that accesses the Licensed Content,
 2. each End User and MCT will be presented with a copy of this agreement and each individual will agree that their use of the Licensed Content will be subject to these license terms prior to their accessing the Licensed Content. Each individual will be required to denote their acceptance of the EULA in a manner that is enforceable under local law prior to their accessing the Licensed Content,
 3. for all training sessions, you will only use qualified MCTs who hold the applicable competency to teach the particular MOC Course that is the subject of the training session,
 4. you will not alter or remove any copyright or other protective notices contained in the Licensed Content,
 5. you will remove and irretrievably delete all Licensed Content from all Classroom Devices and servers at the end of each training session,
 6. you will only provide access to the Licensed Content to End Users and MCTs,
 7. you will only provide access to the Trainer Content to MCTs, and
 8. any Licensed Content installed for use during a training session will be done in accordance with the applicable classroom set-up guide.

c. If you are an End User:

You may use the Licensed Content solely for your personal training use. If the Licensed Content is in digital format, for each license you acquire you may (i) install one (1) copy of the Licensed Content in the form provided to you on one (1) Personal Device and install another copy on another Personal Device as a backup copy, which may be used only to reinstall the Licensed Content; or (ii) print one (1) copy of the Licensed Content. You may not install or use a copy of the Licensed Content on a device you do not own or control.

d. **If you are a MCT.**

- i. For each license you acquire, you may use the Licensed Content solely to prepare and deliver an Authorized Training Session or Private Training Session. For each license you acquire, you may install and use one (1) copy of the Licensed Content in the form provided to you on one (1) Personal Device and install one (1) additional copy on another Personal Device as a backup copy, which may be used only to reinstall the Licensed Content. You may not install or use a copy of the Licensed Content on a device you do not own or control.
- ii. **Use of Instructional Components in Trainer Content.** You may customize, in accordance with the most recent version of the MCT Agreement, those portions of the Trainer Content that are logically associated with instruction of a training session. If you elect to exercise the foregoing rights, you agree: (a) that any of these customizations will only be used for providing a training session, (b) any customizations will comply with the terms and conditions for Modified Training Sessions and Supplemental Materials in the most recent version of the MCT agreement and with this agreement. For clarity, any use of “customize” refers only to changing the order of slides and content, and/or not using all the slides or content, it does not mean changing or modifying any slide or content.

2.2 **Separation of Components.** The Licensed Content components are licensed as a single unit and you may not separate the components and install them on different devices.

2.3 **Reproduction/Redistribution Licensed Content.** Except as expressly provided in the applicable installation and use rights above, you may not reproduce or distribute the Licensed Content or any portion thereof (including any permitted modifications) to any third parties without the express written permission of Microsoft.

2.4 **Third Party Programs.** The Licensed Content may contain third party programs or services. These license terms will apply to your use of those third party programs or services, unless other terms accompany those programs and services.

2.5 **Additional Terms.** Some Licensed Content may contain components with additional terms, conditions, and licenses regarding its use. Any non-conflicting terms in those conditions and licenses also apply to that respective component and supplements the terms described in this Agreement.

3. **PRE-RELEASE VERSIONS.** If the Licensed Content is a pre-release (“beta”) version, in addition to the other provisions in this agreement, then these terms also apply:

- a. **Pre-Release Licensed Content.** This Licensed Content is a pre-release version. It may not contain the same information and/or work the way a final version of the Licensed Content will. We may change it for the final version. We also may not release a final version. Microsoft is under no obligation to provide you with any further content, including the final release version of the Licensed Content.
- b. **Feedback.** If you agree to give feedback about the Licensed Content to Microsoft, either directly or through its third party designee, you give to Microsoft without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft software, Microsoft product, or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its software, technologies, or products to third parties because we include your feedback in them. These rights

survive this agreement.

- c. **Term.** If you are an Authorized Training Center, MCT or MPN, you agree to cease using all copies of the beta version of the Licensed Content upon (i) the date which Microsoft informs you is the end date for using the beta version, or (ii) sixty (60) days after the commercial release of the Licensed Content, whichever is earliest ("**beta term**"). Upon expiration or termination of the beta term, you will irretrievably delete and destroy all copies of same in the possession or under your control.

4. **INTERNET-BASED SERVICES.** Classroom Devices located at Authorized Learning Center's physical location may contain virtual machines and virtual hard disks for use while attending an Authorized Training Session. You may only use the software on the virtual machines and virtual hard disks on a Classroom Device solely to perform the virtual lab activities included in the MOC Course while attending the Authorized Training Session. Microsoft may provide Internet-based services with the software included with the virtual machines and virtual hard disks. It may change or cancel them at any time. If the software is pre-release versions of software, some of its Internet-based services may be turned on by default. The default setting in these versions of the software do not necessarily reflect how the features will be configured in the commercially released versions. If Internet-based services are included with the software, they are typically simulated for demonstration purposes in the software and no transmission over the Internet takes place. However, should the software be configured to transmit over the Internet, the following terms apply:

- a. **Consent for Internet-Based Services.** The software features described below connect to Microsoft or service provider computer systems over the Internet. In some cases, you will not receive a separate notice when they connect. You may switch off these features or not use them. By using these features, you consent to the transmission of this information. Microsoft does not use the information to identify or contact you.
- b. **Computer Information.** The following features use Internet protocols, which send to the appropriate systems computer information, such as your Internet protocol address, the type of operating system, browser and name and version of the software you are using, and the language code of the device where you installed the software. Microsoft uses this information to make the Internet-based services available to you.
 - **Accelerators.** When you use click on or move your mouse over an Accelerator, the title and full web address or URL of the current webpage, as well as standard computer information, and any content you have selected, might be sent to the service provider. If you use an Accelerator provided by Microsoft, the information sent is subject to the Microsoft Online Privacy Statement, which is available at go.microsoft.com/fwlink/?linkid=31493. If you use an Accelerator provided by a third party, use of the information sent will be subject to the third party's privacy practices.
 - **Automatic Updates.** This software contains an Automatic Update feature that is on by default. For more information about this feature, including instructions for turning it off, see go.microsoft.com/fwlink/?LinkId=178857. You may turn off this feature while the software is running ("opt out"). Unless you expressly opt out of this feature, this feature will (a) connect to Microsoft or service provider computer systems over the Internet, (b) use Internet protocols to send to the appropriate systems standard computer information, such as your computer's Internet protocol address, the type of operating system, browser and name and version of the software you are using, and the language code of the device where you installed the software, and (c) automatically download and install, or prompt you to download and/or install, current Updates to the software. In some cases, you will not receive a separate notice before this feature takes effect.

By installing the software, you consent to the transmission of standard computer information and the automatic downloading and installation of updates.

- **Auto Root Update.** The Auto Root Update feature updates the list of trusted certificate authorities. you can switch off the Auto Root Update feature.
- **Customer Experience Improvement Program (CEIP), Error and Usage Reporting; Error Reports.** This software uses CEIP and Error and Usage Reporting components enabled by default that automatically send to Microsoft information about your hardware and how you use this software. This software also automatically sends error reports to Microsoft that describe which software components had errors and may also include memory dumps. You may choose not to use these software components. For more information please go to <http://go.microsoft.com/fwlink/?LinkID=196910>.
- **Digital Certificates.** The software uses digital certificates. These digital certificates confirm the identity of Internet users sending X.509 standard encrypted information. They also can be used to digitally sign files and macros, to verify the integrity and origin of the file contents. The software retrieves certificates and updates certificate revocation lists. These security features operate only when you use the Internet.
- **Extension Manager.** The Extension Manager can retrieve other software through the internet from the Visual Studio Gallery website. To provide this other software, the Extension Manager sends to Microsoft the name and version of the software you are using and language code of the device where you installed the software. This other software is provided by third parties to Visual Studio Gallery. It is licensed to users under terms provided by the third parties, not from Microsoft. Read the Visual Studio Gallery terms of use for more information.
- **IPv6 Network Address Translation (NAT) Traversal service (Teredo).** This feature helps existing home Internet gateway devices transition to IPv6. IPv6 is a next generation Internet protocol. It helps enable end-to-end connectivity often needed by peer-to-peer applications. To do so, each time you start up the software the Teredo client service will attempt to locate a public Teredo Internet service. It does so by sending a query over the Internet. This query only transfers standard Domain Name Service information to determine if your computer is connected to the Internet and can locate a public Teredo service. If you
 - use an application that needs IPv6 connectivity or
 - configure your firewall to always enable IPv6 connectivity

by default standard Internet Protocol information will be sent to the Teredo service at Microsoft at regular intervals. No other information is sent to Microsoft. You can change this default to use non-Microsoft servers. You can also switch off this feature using a command line utility named "netsh".

- **Malicious Software Removal.** During setup, if you select "Get important updates for installation", the software may check and remove certain malware from your device. "Malware" is malicious software. If the software runs, it will remove the Malware listed and updated at www.support.microsoft.com/?kbid=890830. During a Malware check, a report will be sent to Microsoft with specific information about Malware detected, errors, and other information about your device. This information is used to improve the software and other Microsoft products and services. No information included in these reports will be used to identify or contact you. You may disable the software's reporting functionality by following the instructions found at

www.support.microsoft.com/?kbid=890830. For more information, read the Windows Malicious Software Removal Tool privacy statement at go.microsoft.com/fwlink/?LinkId=113995.

- **Microsoft Digital Rights Management.** If you use the software to access content that has been protected with Microsoft Digital Rights Management (DRM), then, in order to let you play the content, the software may automatically request media usage rights from a rights server on the Internet and download and install available DRM updates. For more information, see go.microsoft.com/fwlink/?LinkId=178857.
- **Microsoft Telemetry Reporting Participation.** If you choose to participate in Microsoft Telemetry Reporting through a “basic” or “advanced” membership, information regarding filtered URLs, malware and other attacks on your network is sent to Microsoft. This information helps Microsoft improve the ability of Forefront Threat Management Gateway to identify attack patterns and mitigate threats. In some cases, personal information may be inadvertently sent, but Microsoft will not use the information to identify or contact you. You can switch off Telemetry Reporting. For more information on this feature, see <http://go.microsoft.com/fwlink/?LinkId=130980>.
- **Microsoft Update Feature.** To help keep the software up-to-date, from time to time, the software connects to Microsoft or service provider computer systems over the Internet. In some cases, you will not receive a separate notice when they connect. When the software does so, we check your version of the software and recommend or download updates to your devices. You may not receive notice when we download the update. You may switch off this feature.
- **Network Awareness.** This feature determines whether a system is connected to a network by either passive monitoring of network traffic or active DNS or HTTP queries. The query only transfers standard TCP/IP or DNS information for routing purposes. You can switch off the active query feature through a registry setting.
- **Plug and Play and Plug and Play Extensions.** You may connect new hardware to your device, either directly or over a network. Your device may not have the drivers needed to communicate with that hardware. If so, the update feature of the software can obtain the correct driver from Microsoft and install it on your device. An administrator can disable this update feature.
- **Real Simple Syndication (“RSS”) Feed.** This software start page contains updated content that is supplied by means of an RSS feed online from Microsoft.
- **Search Suggestions Service.** When you type a search query in Internet Explorer by using the Instant Search box or by typing a question mark (?) before your search term in the Address bar, you will see search suggestions as you type (if supported by your search provider). Everything you type in the Instant Search box or in the Address bar when preceded by a question mark (?) is sent to your search provider as you type it. In addition, when you press Enter or click the Search button, all the text that is in the search box or Address bar is sent to the search provider. If you use a Microsoft search provider, the information you send is subject to the Microsoft Online Privacy Statement, which is available at go.microsoft.com/fwlink/?linkid=31493. If you use a third-party search provider, use of the information sent will be subject to the third party’s privacy practices. You can turn search suggestions off at any time in Internet Explorer by using Manage Add-ons under the Tools button. For more information about the search suggestions service, see go.microsoft.com/fwlink/?linkid=128106.
- **SQL Server Reporting Services Map Report Item.** The software may include features that retrieve content such as maps, images and other data through the Bing Maps (or successor branded)

application programming interface (the “Bing Maps APIs”). The purpose of these features is to create reports displaying data on top of maps, aerial and hybrid imagery. If these features are included, you may use them to create and view dynamic or static documents. This may be done only in conjunction with and through methods and means of access integrated in the software. You may not otherwise copy, store, archive, or create a database of the content available through the Bing Maps APIs. you may not use the following for any purpose even if they are available through the Bing Maps APIs:

- Bing Maps APIs to provide sensor based guidance/routing, or
- Any Road Traffic Data or Bird’s Eye Imagery (or associated metadata).

Your use of the Bing Maps APIs and associated content is also subject to the additional terms and conditions at <http://www.microsoft.com/maps/product/terms.html>.

- **URL Filtering.** The URL Filtering feature identifies certain types of web sites based upon predefined URL categories, and allows you to deny access to such web sites, such as known malicious sites and sites displaying inappropriate or pornographic materials. To apply URL filtering, Microsoft queries the online Microsoft Reputation Service for URL categorization. You can switch off URL filtering. For more information on this feature, see <http://go.microsoft.com/fwlink/?LinkId=130980>
- **Web Content Features.** Features in the software can retrieve related content from Microsoft and provide it to you. To provide the content, these features send to Microsoft the type of operating system, name and version of the software you are using, type of browser and language code of the device where you run the software. Examples of these features are clip art, templates, online training, online assistance and Appshelp. You may choose not to use these web content features.
- **Windows Media Digital Rights Management.** Content owners use Windows Media digital rights management technology (WMDRM) to protect their intellectual property, including copyrights. This software and third party software use WMDRM to play and copy WMDRM-protected content. If the software fails to protect the content, content owners may ask Microsoft to revoke the software’s ability to use WMDRM to play or copy protected content. Revocation does not affect other content. When you download licenses for protected content, you agree that Microsoft may include a revocation list with the licenses. Content owners may require you to upgrade WMDRM to access their content. Microsoft software that includes WMDRM will ask for your consent prior to the upgrade. If you decline an upgrade, you will not be able to access content that requires the upgrade. You may switch off WMDRM features that access the Internet. When these features are off, you can still play content for which you have a valid license.
- **Windows Media Player.** When you use Windows Media Player, it checks with Microsoft for
 - compatible online music services in your region;
 - new versions of the player; and
 - codecs if your device does not have the correct ones for playing content.

You can switch off this last feature. For more information, go to www.microsoft.com/windows/windowsmedia/player/11/privacy.aspx.

- **Windows Rights Management Services.** The software contains a feature that allows you to create content that cannot be printed, copied or sent to others without your permission. For more information, go to www.microsoft.com/rms. you may choose not to use this feature

- **Windows Time Service.** This service synchronizes with time.windows.com once a week to provide your computer with the correct time. You can turn this feature off or choose your preferred time source within the Date and Time Control Panel applet. The connection uses standard NTP protocol.
 - **Windows Update Feature.** You may connect new hardware to the device where you run the software. Your device may not have the drivers needed to communicate with that hardware. If so, the update feature of the software can obtain the correct driver from Microsoft and run it on your device. You can switch off this update feature.
- c. **Use of Information.** Microsoft may use the device information, error reports, and malware reports to improve our software and services. We may also share it with others, such as hardware and software vendors. They may use the information to improve how their products run with Microsoft software.
- d. **Misuse of Internet-based Services.** You may not use any Internet-based service in any way that could harm it or impair anyone else's use of it. You may not use the service to try to gain unauthorized access to any service, data, account or network by any means.
5. **SCOPE OF LICENSE.** The Licensed Content is licensed, not sold. This agreement only gives you some rights to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the Licensed Content that only allows you to use it in certain ways. Except as expressly permitted in this agreement, you may not:
- install more copies of the Licensed Content on devices than the number of licenses you acquired;
 - allow more individuals to access the Licensed Content than the number of licenses you acquired;
 - publicly display, or make the Licensed Content available for others to access or use;
 - install, sell, publish, transmit, encumber, pledge, lend, copy, adapt, link to, post, rent, lease or lend, make available or distribute the Licensed Content to any third party, except as expressly permitted by this Agreement.
 - reverse engineer, decompile, remove or otherwise thwart any protections or disassemble the Licensed Content except and only to the extent that applicable law expressly permits, despite this limitation;
 - access or use any Licensed Content for which you are not providing a training session to End Users using the Licensed Content;
 - access or use any Licensed Content that you have not been authorized by Microsoft to access and use; or
 - transfer the Licensed Content, in whole or in part, or assign this agreement to any third party.
6. **RESERVATION OF RIGHTS AND OWNERSHIP.** Microsoft reserves all rights not expressly granted to you in this agreement. The Licensed Content is protected by copyright and other intellectual property laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Licensed Content. You may not remove or obscure any copyright, trademark or patent notices that appear on the Licensed Content or any components thereof, as delivered to you.
7. **EXPORT RESTRICTIONS.** The Licensed Content is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Licensed Content. These laws include restrictions on destinations, End Users and end use. For additional information, see www.microsoft.com/exporting.

8. **LIMITATIONS ON SALE, RENTAL, ETC. AND CERTAIN ASSIGNMENTS.** You may not sell, rent, lease, lend or sublicense the Licensed Content or any portion thereof, or transfer or assign this agreement.
9. **SUPPORT SERVICES.** Because the Licensed Content is “as is”, we may not provide support services for it.
10. **TERMINATION.** Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. Upon any termination of this agreement, you agree to immediately stop all use of and to irretrievably delete and destroy all copies of the Licensed Content in your possession or under your control.
11. **LINKS TO THIRD PARTY SITES.** You may link to third party sites through the use of the Licensed Content. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.
12. **ENTIRE AGREEMENT.** This agreement, and the terms for supplements, updates and support services are the entire agreement for the Licensed Content.
13. **APPLICABLE LAW.**
- United States. If you acquired the Licensed Content in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.
 - Outside the United States. If you acquired the Licensed Content in any other country, the laws of that country apply.
14. **LEGAL EFFECT.** This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the Licensed Content. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.
15. **DISCLAIMER OF WARRANTY. THE LICENSED CONTENT IS LICENSED "AS-IS," "WITH ALL FAULTS," AND "AS AVAILABLE." YOU BEAR THE RISK OF USING IT. MICROSOFT CORPORATION AND ITS RESPECTIVE AFFILIATES GIVE NO EXPRESS WARRANTIES, GUARANTEES, OR CONDITIONS UNDER OR IN RELATION TO THE LICENSED CONTENT. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT CORPORATION AND ITS RESPECTIVE AFFILIATES EXCLUDE ANY IMPLIED WARRANTIES OR CONDITIONS, INCLUDING THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.**
16. **LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. TO THE EXTENT NOT PROHIBITED BY LAW, YOU CAN RECOVER FROM MICROSOFT CORPORATION AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO USD\$5.00. YOU AGREE NOT TO SEEK TO RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES FROM MICROSOFT CORPORATION AND ITS RESPECTIVE SUPPLIERS.**

This limitation applies to

- anything related to the Licensed Content, services made available through the Licensed Content, or content (including code) on third party Internet sites or third-party programs; and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this Licensed Content is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce le contenu sous licence étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le contenu sous licence visé par une licence est offert « tel quel ». Toute utilisation de ce contenu sous licence est à votre seule risque et péril. Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 \$ US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.

Cette limitation concerne:

- tout ce qui est relié au le contenu sous licence , aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers ; et
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

Revised March 2012

Welcome!

Thank you for taking our training! We've worked together with our Microsoft Certified Partners for Learning Solutions and our Microsoft IT Academies to bring you a world-class learning experience—whether you're a professional looking to advance your skills or a student preparing for a career in IT.

- **Microsoft Certified Trainers and Instructors**—Your instructor is a technical and instructional expert who meets ongoing certification requirements. And, if instructors are delivering training at one of our Certified Partners for Learning Solutions, they are also evaluated throughout the year by students and by Microsoft.
- **Certification Exam Benefits**—After training, consider taking a Microsoft Certification exam. Microsoft Certifications validate your skills on Microsoft technologies and can help differentiate you when finding a job or boosting your career. In fact, independent research by IDC concluded that 75% of managers believe certifications are important to team performance¹. Ask your instructor about Microsoft Certification exam promotions and discounts that may be available to you.
- **Customer Satisfaction Guarantee**—Our Certified Partners for Learning Solutions offer a satisfaction guarantee and we hold them accountable for it. At the end of class, please complete an evaluation of today's experience. We value your feedback!

We wish you a great learning experience and ongoing success in your career!

Sincerely,

Microsoft Learning
www.microsoft.com/learning

Microsoft | Learning

¹ IDC, Value of Certification: Team Certification and Organizational Performance, November 2006

Acknowledgments

Microsoft Learning would like to acknowledge and thank the following for their contribution towards developing this title. Their effort at various stages in the development has ensured that you have a good classroom experience.

Design and Development

This course was designed and developed by SolidQ. SolidQ is a global provider of consulting, mentoring and training services for Microsoft Data Management, Business Intelligence and Collaboration platforms.

Greg Low – Lead Developer

Dr Greg Low is a SQL Server MVP, an MCT, and a Microsoft Regional Director for Australia. Greg has worked with SQL Server since version 4.2 as an active mentor, consultant and trainer. Greg describes himself as a SQL Server junkie and also describes himself as having been involved in development since dinosaurs roamed the Earth. He has been an instructor in the Microsoft SQL Server Masters certification program for several years and was one of the first two people to achieve the SQL Server 2008 Master certification. He is the author of a number of whitepapers on the Microsoft MSDN and TechNet web sites and the author of a number of SQL Server related books. Greg is based in Melbourne Australia.

Herbert Albert – Course Developer

Herbert Albert started his career in 1994. He works as a trainer, consultant, and author focusing on SQL Server technologies. Herbert is a mentor and the Central European CEO for SolidQ. He is based in Vienna, Austria. He has several Microsoft certifications including MCT which he has held since 1997. Herbert is a regular speaker at conferences and co-author of the SQL Server 2012 Upgrade Technical Reference Guide and SQL Server 2005 Step-by-Step Applied Techniques. Together with Gianluca Hotz, Herbert writes a regular column at the SolidQ Journal.

Mark Hions – Technical Reviewer

Mark's passion for computing and skill as a communicator were well suited to his position as instructor at Honeywell Canada, where he started working with minicomputers, mainframes and mature students in 1984. He first met Microsoft SQL Server when it ran on OS/2, and has delivered training on every version since. An independent MCT and consultant for many years, he is a highly-rated presenter at TechEd, has designed SQL Server exams for Microsoft, and has delivered deep dive courses through the Microsoft Partner Channel. Mark is now the Principal SQL Server Instructor and Consultant at DesTech, which is the largest provider of SQL Server training in the Toronto area.

Chris Barker – Technical Reviewer

Chris Barker is an MCT working in the New Zealand market and currently employed as a staff trainer at Auldhouse, one of New Zealand's major CPLS training centers in Wellington. Chris' background includes programming from the early 1970s—his first program was written in assembly language and debugged in binary (literally)! While focusing training on programming (mostly .NET) and databases (mostly Microsoft SQL Server) Chris has also been an infrastructure trainer and has both Novell and Microsoft networking qualifications.

Contents

Module 1: Introduction to SQL Server 2012 and Its Toolset

Lesson 1: Introduction to the SQL Server Platform	1-3
Lesson 2: Working with SQL Server Tools	1-14
Lesson 3: Configuring SQL Server Services	1-26
Lab 1: Introduction to SQL Server and Its Toolset	1-36

Module 2: Preparing Systems for SQL Server 2012

Lesson 1: Overview of SQL Server Architecture	2-3
Lesson 2: Planning Server Resource Requirements	2-17
Lesson 3: Pre-installation Testing for SQL Server	2-29
Lab 2: Preparing Systems for SQL Server	2-35

Module 3: Installing and Configuring SQL Server 2012

Lesson 1: Preparing to Install SQL Server	3-3
Lesson 2: Installing SQL Server	3-16
Lesson 3: Upgrading and Automating Installation	3-24
Lab 3: Installing and Configuring SQL Server	3-32

Module 4: Working with Databases

Lesson 1: Overview of SQL Server Databases	4-3
Lesson 2: Working with Files and Filegroups	4-15
Lesson 3: Moving Database Files	4-29
Lab 4: Working with Databases	4-39

Module 5: Understanding SQL Server 2012 Recovery Models

Lesson 1: Backup Strategies	5-3
Lesson 2: Understanding SQL Server Transaction Logging	5-12
Lesson 3: Planning a SQL Server Backup Strategy	5-22
Lab 5: Understanding SQL Server Recovery Models	5-32

Module 6: Backup of SQL Server 2012 Databases

Lesson 1: Backing up Databases and Transaction Logs	6-3
Lesson 2: Managing Database Backups	6-14
Lesson 3: Working with Backup Options	6-20
Lab 6: Backup of SQL Server Databases	6-26

Module 7: Restoring SQL Server 2012 Databases

Lesson 1: Understanding the Restore Process	7-3
Lesson 2: Restoring Databases	7-8
Lesson 3: Working with Point-in-time recovery	7-19
Lesson 4: Restoring System Databases and Individual Files	7-27
Lab 7: Restoring SQL Server 2012 Databases	7-34

Module 8: Importing and Exporting Data

Lesson 1: Transferring Data To/From SQL Server	8-3
Lesson 2: Importing & Exporting Table Data	8-15
Lesson 3: Inserting Data in Bulk	8-20
Lab 8: Importing and Exporting Data	8-29

Module 9: Authenticating and Authorizing Users

Lesson 1: Authenticating Connections to SQL Server	9-3
Lesson 2: Authorizing Logins to Access Databases	9-13
Lesson 3: Authorization Across Servers	9-22
Lab 9: Authenticating and Authorizing Users	9-30

Module 10: Assigning Server and Database Roles

Lesson 1: Working with Server Roles	10-3
Lesson 2: Working with Fixed Database Roles	10-12
Lesson 3: Creating User-defined Database Roles	10-18
Lab 10: Assigning Server and Database Roles	10-26

Module 11: Authorizing Users to Access Resources

Lesson 1: Authorizing User Access to Objects	11-3
Lesson 2: Authorizing Users to Execute Code	11-12
Lesson 3: Configuring Permissions at the Schema Level	11-21
Lab 11: Authorizing Users to Access Resources	11-28

Module 12: Auditing SQL Server Environments

Lesson 1: Options for Auditing Data Access in SQL	12-3
Lesson 2: Implementing SQL Server Audit	12-12
Lesson 3: Managing SQL Server Audit	12-26
Lab 12: Auditing SQL Server Environments	12-31

Module 13: Automating SQL Server 2012 Management

Lesson 1: Automating SQL Server Management	13-3
Lesson 2: Working with SQL Server Agent	13-11
Lesson 3: Managing SQL Server Agent Jobs	13-19
Lab 13: Automating SQL Server Management	13-26

Module 14: Configuring Security for SQL Server Agent

Lesson 1: Understanding SQL Server Agent Security	14-3
Lesson 2: Configuring Credentials	14-13
Lesson 3: Configuring Proxy Accounts	14-18
Lab 14: Configuring Security for SQL Server Agent	14-24

Module 15: Monitoring SQL Server 2012 with Alerts and Notifications

Lesson 1: Configuration of Database Mail	15-3
Lesson 2: Monitoring SQL Server Errors	15-11
Lesson 3: Configuring Operators, Alerts and Notifications	15-18
Lab 15: Monitoring SQL Agent Jobs with Alerts and Notifications	15-30

Module 16: Performing Ongoing Database Maintenance

Lesson 1: Ensuring Database Integrity	16-3
Lesson 2: Maintaining Indexes	16-12
Lesson 3: Automating Routine Database Maintenance	16-26
Lab 16: Performing Ongoing Database Maintenance	16-30

Module 17: Tracing Access to SQL Server 2012

Lesson 1: Capturing Activity using SQL Server Profiler and Extended Events Profiler	17-3
Lesson 2: Improving Performance with the Database Engine Tuning Advisor	17-17
Lesson 3: Working with Tracing Options	17-25
Lab 17: Tracing Access to SQL Server 2012	17-36

Module 18: Monitoring SQL Server 2012

Lesson 1: Monitoring Activity	18-3
Lesson 2: Capturing and Managing Performance Data	18-15
Lesson 3: Analyzing Collected Performance Data	18-23
Lab 18: Monitoring SQL Server 2012	18-32

Module 19: Managing Multiple Servers

Lesson 1: Working with Multiple Servers	19-3
Lesson 2: Virtualizing SQL Server	19-9
Lesson 3: Deploying and Upgrading Data-tier Applications	19-15
Lab 19: Managing Multiple Servers	19-22

Module 20: Troubleshooting Common SQL Server 2012 Administrative Issues

Lesson 1: SQL Server Troubleshooting Methodology	20-3
Lesson 2: Resolving Service-related Issues	20-7
Lesson 3: Resolving Login and Connectivity Issues	20-13
Lesson 4: Resolving Concurrency Issues	20-17
Lab 20: Troubleshooting Common Issues	20-25

Appendix A: Core Concepts in SQL Server High Availability and Replication

Lesson 1: Core Concepts in High Availability	A-3
Lesson 2: Core Concepts in Replication	A-11

Appendix: Lab Answer Keys

Module 1 Lab: Introduction to SQL Server and Its Toolset	L1-1
Module 2 Lab: Preparing Systems for SQL Server	L2-5
Module 3 Lab: Installing and Configuring SQL Server	L3-11
Module 4 Lab: Working with Databases	L4-17
Module 5 Lab: Understanding SQL Server Recovery Models	L5-23
Module 6 Lab: Backup of SQL Server Databases	L6-27
Module 7 Lab: Restoring SQL Server 2012 Databases	L7-31
Module 8 Lab: Importing and Exporting Data	L8-35
Module 9 Lab: Authenticating and Authorizing Users	L9-39
Module 10 Lab: Assigning Server and Database Roles	L10-41
Module 11 Lab: Authorizing Users to Access Resources	L11-43
Module 12 Lab: Auditing SQL Server Environments	L12-45
Module 13 Lab: Automating SQL Server Management	L13-49
Module 14 Lab: Configuring Security for SQL Server Agent	L14-53
Module 15 Lab: Monitoring SQL Server 2012 with Alerts and Notifications	L15-57
Module 16 Lab: Performing Ongoing Database Maintenance	L16-63
Module 17 Lab: Tracing Access to SQL Server	L17-67
Module 18 Lab: Monitoring SQL Server 2012	L18-71
Module 19 Lab: Managing Multiple Servers	L19-75
Module 20 Lab: Troubleshooting Common Issues	L20-79

MCT USE ONLY. STUDENT USE PROHIBITED

Module 17

Tracing Access to SQL Server 2012

Contents:

Lesson 1: Capturing Activity Using SQL Server Profiler and Extended Events Profiler	17-3
Lesson 2: Improving Performance with the Database Engine Tuning Advisor	17-17
Lesson 3: Working with Tracing Options	17-25
Lab 17: Tracing Access to SQL Server 2012	17-36

Module Overview

- Capturing Activity Using SQL Server Profiler and Extended Events Profiler
- Improving Performance with the Database Engine Tuning Advisor
- Working with Tracing Options

Microsoft® SQL Server® performs well with many types of SQL workloads but the performance of most SQL Server systems can be improved by a process of tuning queries and database structures. For most large organizations, performance tuning is not a task that is performed once and completed. Performance tuning typically involves a process of continuous incremental improvements.

It is important to spend most of your performance tuning efforts where the effort will provide the greatest benefit. Developers will often guess how their applications will be used and try to optimize the applications based on those guesses. The ability to trace activity against a SQL Server removes the guesswork from this process and allows you to focus your efforts on the areas of the applications that users are actually using.

SQL Server Profiler and Extended Events Profiler are used to capture traces of activity against SQL Server. The Database Engine Tuning Advisor can be used to analyze the traces captured by SQL Server Profiler and to suggest improvements that can be made, particularly in relation to indexes and statistics.

Objectives

After completing this lesson, you will be able to:

- Capture activity using SQL Server Profiler and Extended Events Profiler.
- Improve performance with the Database Engine Tuning Advisor.
- Work with tracing options.

Lesson 1

Capturing Activity Using SQL Server Profiler and Extended Events Profiler

- Overview of SQL Server Profiler
- Available Tracing Output Options
- Commonly Used Trace Events
- Commonly Used Trace Columns
- Filtering Traces
- Working with Trace Templates
- Demonstration 1A: Capturing Activity Using SQL Server Profiler
- Working with Extended Events Profiler
- Demonstration 1B: Capturing Activity Using Extended Events Profiler

SQL Server Profiler and Extended Events Profiler provide you with the ability to trace the activity that is occurring in the database engine within SQL Server. SQL Server Profiler can also trace activity against Analysis Services. (The tracing of Analysis Services activity is out of scope for this training).

Traces can be used for performance tuning, for troubleshooting and diagnostic purposes, and for replaying workloads. The ability to replay a workload allows you to test the impact of performance changes against test systems or to test application workloads against newer versions of SQL Server.

It is important to learn to configure these tools, to avoid creating excessive impacts from the tracing process itself.

Objectives

After completing this lesson, you will be able to:

- Use SQL Server Profiler.
- Describe available tracing output options.
- Detail commonly used trace events.
- Detail commonly used trace columns.
- Filter traces.
- Work with trace templates.
- Use Extended Events Profiler.

Overview of SQL Server Profiler

SQL Server Profiler is a tool used to trace activity against SQL Server

- Based on the SQL Trace programming interface
- Used in many scenarios such as debugging, performance monitoring, deadlock monitoring, and many others

Audit Logout		Microsoft SQL...	hal	r2d3\hal
SQL:BatchStarting	select value_in_use from sys.config...	Microsoft SQL...	hal	r2d3\hal
SQL:BatchCompleted	select value_in_use from sys.config...	Microsoft SQL...	hal	r2d3\hal
RPC:Completed	exec sp_executesql N' DECI...	Microsoft SQL...	hal	r2d3\hal
RPC:Completed	exec sp_executesql N' DECI...	Microsoft SQL...	hal	r2d3\hal
RPC:Completed	exec sp_executesql N'SELECT CAST(ha...	Microsoft SQL...	hal	r2d3\hal
Trace Stop				

```

        AS [Database_Status],
        dtb.compatibility_level AS [Database_CompatibilityLevel],
        ISNULL(dtm.mirroring_role,0) AS [Database_MirroringRole],
        ISNULL(dtm.mirroring_state + 1, 0) AS [Database_MirroringStatus],
        dtb.recovery_model AS [Recovery_Model],
        dtb.user_access AS [UserAccess],
        dtb.is_read_only AS [IsReadOnly],
        dtb.name AS [DatabaseName]
    FROM
        master.sys.databases dtb

```

Key Points

SQL Server Profiler is an important tool when tuning the performance of SQL Server queries. It captures the activity from client applications to SQL Server and stores it in a trace. These traces can then be analyzed.

SQL Server Profiler

SQL Server profiler captures data when events occur. Only events that have been selected are captured. A variety of information (shown as a set of columns) is available when each event occurs. The trace created contains only the selected columns for the selected events.

Rather than needing to select events and columns each time you run SQL Server Profiler, a set of existing templates are available. You can also save your own selections as a new template.

The captured traces are useful when tuning the performance of an application and when diagnosing specific problems that are occurring. When using traces for diagnosing problems, log data from the Windows Performance Monitor tool can be loaded. This allows relationships between system resource impacts and the execution of queries in SQL Server to be made.

The traces can also be replayed. The ability to replay traces is useful for load testing systems or for ensuring that upgraded versions of SQL Server can be used with existing applications. SQL Server Profiler also allows you to step through queries when diagnosing problems.

SQL Trace

SQL Server Profiler is a graphical tool and it is important to realize that it can have significant performance impacts on the server being traced, depending upon the options chosen. SQL Trace is a library of system stored procedures that can be used for tracing when minimizing the performance impacts of the tracing is necessary. Internally, SQL Server Profiler uses the programming interface that has been provided by SQL Trace.



Note The Extended Events system that was introduced in SQL Server 2008 also provides capabilities for tracing SQL Server activity and resources. The use of Extended Events for tracing activity is discussed later in this lesson. The use of Extended Events for other monitoring purposes is outside the scope of this course.

Question: Where would the ability to replay a trace be useful?

Available Tracing Output Options

- Profiler traces can be saved in tables and files
- For file output, ensure that you configure:
 - Maximum file size (default of 5MB is almost always too small)
 - Enable file rollover (if appropriate)

The screenshot shows the 'Save to file' section of the SQL Server Profiler configuration. The 'Save to file' checkbox is checked. The file name is 'L:\Traces\Trace20110115AM.trc'. The 'Set maximum file size (MB)' is set to 5000. The 'Enable file rollover' checkbox is checked. The 'Server processes trace data' checkbox is unchecked. The 'Save to table' section is collapsed, showing the 'Set maximum rows (in thousands)' set to 1.

Key Points

When a SQL Server Profiler trace is active, the captured events are loaded into a graphical grid within the SQL Server Profiler user interface.

In addition, SQL Server Profiler can send the captured event details to either operating system files or to database tables.

Capture to Files

Capturing to an operating system file is the most efficient option for SQL Server Profiler traces. When configuring file output, you need to supply a filename for the trace. The default file type for a trace file is ".trc". SQL Server Profiler defaults to a file size of 5MB which is far too small for most traces. A more realistic value on most large systems would be either 500MB or 5000MB, depending upon the volume of activity that needs to be recorded.

When the allocated file size is exhausted, SQL Server Profiler will open a new file based on a variation of the current file name and start writing to it, if the "Enable file rollover" option has been checked. It is considered good practice to work with a large maximum file size and to avoid the need for rollover files, unless there is a need to move the captured traces onto media such as DVDs or onto download sites that cannot work with larger files.

Selecting the "Server processes trace data" option causes the SQL Server service to write the output file instead of the client system that is running SQL Server Profiler. Using this option can enhance performance but requires access to the server's filesystem.

Capturing to Tables

SQL Server Profiler can also capture trace data to database tables. The underlying SQL Trace programming interface does not directly support output to tables. This means that the SQL Server Profiler program needs to retrieve the event data into its graphical grid and as batches of rows are received, SQL Server Profiler then writes those rows to the selected database table.



Note It is very important to avoid writing trace data directly back to the SQL Server system that is being monitored, unless a very low level of activity is expected.

SQL Server Profiler also provides an option for saving existing captured event data that is being displayed in the graphical grid into a database table.

Question: What might be an advantage of saving events in a SQL Server table?

Commonly Used Trace Events

Events:
The occurrence of an action within SQL Server

Categories:
Groups of related trace

Event	Description
SQL:BatchCompleted	A SQL Server batch completed
SQL:StmtCompleted	A T-SQL statement completed
RPC:Completed	A remote procedure call (stored procedure) completed
Audit Login / Audit Logout	Login or Logout occurred
Deadlock Graph	An XML description of a deadlock was captured

Key Points

The information recorded in a trace is divided into categories. Categories contain events, each of which has attributes further defined by columns.

Trace Categories

In SQL Server Profiler, a category is a group of related event classes. Event classes consist of types of events that can be traced. The event class contains all the data columns that can be reported by an event.

Events

An event is defined as the occurrence of an action within an instance of the SQL Server Database Engine. Events are further defined by their attributes, which are listed in data columns.

The most commonly traced events are as follows:

Event	Description
SQL:BatchCompleted	When a batch of T-SQL statements is completed, the SQL:BatchCompleted event is fired. Note that there is also an event raised when the batch is first started but the completed event contains more useful information such as details of the resources used during the execution of the batch.
SQL:StmtCompleted	If tracing at the SQL batch level is too coarse, it is possible to retrieve details of each individual statement that is contained within the batch.

(continued)

Event	Description
RPC: Completed	The RPC:Completed event is fired when a stored procedure finishes execution. There is a traceable event when the stored procedure starts but similar to the SQL:BatchCompleted event, the RPC:Completed event is useful as it contains details of the resources used during the execution of the stored procedure. You can see a statement by statement breakdown of resources used within the stored procedure via the SP:StmtCompleted event.
Audit Login / Audit Logout	You can include in your traces details of each login and logout event that occurs during the tracing activity.
Deadlock Graph	Unhandled deadlocks often lead to errors being passed to end users from applications. If your system is suffering from deadlocks, the Deadlock Graph event fires when deadlocks occur and captures details of what caused the deadlock. The details are captured into an XML document that can be viewed graphically within SQL Server Profiler.

Question: Why would events that are raised on completion of batches or statements often be more interesting than events that are raised when batches or statements start?

Commonly Used Trace Columns

Trace columns represent data (often attributes) that can be captured when an event occurs

- Avoid capturing too many columns, to minimize trace size
- Not all events provide data in every column
- Columns can be used to group event data

TextData	ApplicationName	NTUserName	LoginName	CPU	Reads	Writes	Duration
☒	☒	☒	☒				
	☒	☒	☒	☒	☒	☒	☒
☒	☒	☒	☒				
☐	☒	☒	☒	☒	☒	☒	☒

Key Points

Data columns contain the attributes of events. SQL Server Profiler uses data columns in the trace output to describe events that are captured when the trace runs.

SQL Server Profiler has a large set of potential columns but not every event writes values to all the possible columns. For example, in the SQL:BatchStarting event, the columns Reads, Writes, Duration, and CPU are not available because the values are not available at the time of the event. Those columns are available in the SQL:BatchCompleted event.



Note The output in the SQL Server Profiler graphical grid can be grouped based on column values.

You should minimize the number of columns that you capture when events occur, to help to minimize the overall size of the trace that is captured.

You can also organize columns into related groups by using the Organize Columns function.

Useful Columns Often Omitted

One of the more interesting columns is the TextData column. Many events do not include it by default yet the values contained in it are very useful. For example, in the RPC:Completed event, the TextData column contains the T-SQL statement that was used to execute the stored procedure.

Two other useful columns are the database ID and the database name. You will often need to create a filter based on the database that you want to trace activity against. Tracing by database ID is more efficient than tracing by database name. However, trace templates that filter by database ID are less portable than those that filter by database name because a database that is restored on another server will typically have a different database ID. You can also use wildcard values within the LIKE clause of a database name filter.

Question: What information would the TextData column return?

Filtering Traces

- Avoid capturing too many events, to minimize:
 - Trace size
 - Performance impact
 - Complexity during analysis
- Filter is only applied when the filtered column is supported by the selected event

☐ Show all events
☐ Show all columns

ApplicationName

BinaryData

ClientProcessID

CPU

Duration

EndTime

LoginName

NTUserName

Reads

SPID

StartTime

TextData

Writes

ApplicationName

Name of the client application that created the connection to SQL Server. This column is populated with the values passed by the application rather than the displayed name of the program.

Like

Not like

SQL Server Profiler - e89aad1f-5686-42

Exclude rows that do not contain values

Key Points

Filters can be set for each of the columns that are captured in a trace. It is important to ensure that you are only capturing events of interest by limiting the events via filters. Effective use of filters helps to minimize the overall size of the captured trace, helps to avoid overwhelming the server with tracing activity, and minimizes the number of events that are contained in the trace, which reduces complexity during analysis. Smaller traces are also typically faster to analyze.

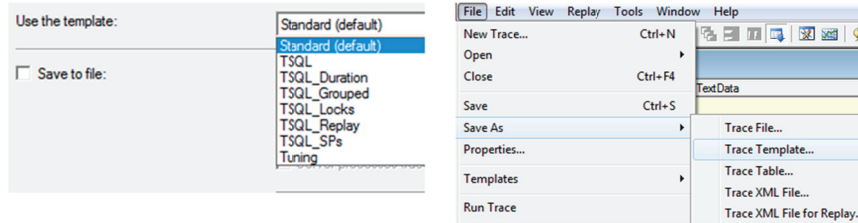
Filters are only applied if an event writes a particular column. For example if you set a filter for DatabaseName = AdventureWorks and define the event Deadlock Graph to be captured, all deadlock events will be shown as the DatabaseName column is not exposed by the Deadlock Graph event.

Question: What filter would you use to locate long running batches or statements?

Working with Trace Templates

Trace templates are predefined (or user-defined) sets of event classes and trace columns

- SQL Server provides a number of standard templates
- Users can save trace templates from existing trace configurations



Key Points

You can use SQL Server Profiler to create templates that define the event classes and data columns to include in traces. After you define and save the template, you can run a trace that records the data for each event class you selected.

Templates can be generated within SQL Server Profiler by creating a trace using the graphical interface, starting and stopping the trace at least once and then using File > Save as > Trace Template.

SQL Server Profiler offers predefined trace templates that allow you to easily configure the event classes that you need for specific types of traces. The Standard template, for example, helps you to create a generic trace for recording logins, logouts, batches completed, and connection information. You can use this template to run traces without modification or as a starting point for additional templates with different event configurations.

Question: How do SQL Server Profiler templates help you trace activity?

Demonstration 1A: Capturing Activity Using SQL Server Profiler

In this demonstration, you will see:

- How to create and run a trace using SQL Server Profiler
- How to use a trace to capture SQL Server activity
- How to save the trace to a file

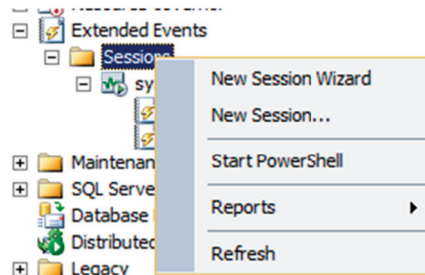
Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_17_PRJ\10775A_17_PRJ.ssmssl** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **11 – Demonstration 1A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Question: Why is there only one Batch Starting event and one Batch Completed event for the workload we ran?

Working with Extended Events Profiler

- Extended Events is a lightweight eventing infrastructure within SQL Server
 - Same underlying technology as used by SQL Server Audit
- Extended Events Profiler
 - Can capture activity for the database engine
 - Not yet supported for Analysis Services
 - Familiar concepts for SQL Server Profiler users
 - Embedded within SQL Server Management Studio



Key Points

SQL Server 2012 introduces a new tool for managing sessions that capture events as they occur, based upon the Extended Events infrastructure that was introduced in SQL Server 2008. It is the same technology that underpins SQL Server Audit.

Extended Events Profiler is integrated directly into SQL Server Management Studio and can be used to capture activity in the SQL Server database engine in much the same way that you can use SQL Server Profiler.

Over time, Extended Events Profiler will replace SQL Server Profiler as the tool of choice for tracing activity in SQL Server. In this version, the two tools have intersecting feature sets and both will likely continue to be used.

Demonstration 1B: Capturing Activity Using Extended Events Profiler

In this demonstration, you will see how to capture activity using Extended Events Profiler

Demonstration Steps

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_17_PRJ\10775A_17_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
1. Open the **12 – Demonstration 1B.sql** script file.
2. Follow the instructions contained within the comments of the script file.

Lesson 2

Improving Performance with the Database Engine Tuning Advisor

- Overview of Performance Tuning
- Available Options for Performance Tuning
- Introduction to the Database Engine Tuning Advisor
- Database Engine Tuning Advisor Options
- Demonstration 2A: Using the Database Engine Tuning Advisor

Tuning the performance of a SQL Server system is one of the most important tasks undertaken by database administrators. As mentioned earlier in this module, performance tuning tends to be an ongoing process, not a one-time task.

It is important to follow a standard methodology for performance tuning. SQL Server includes many tools that can help with performance tuning, including the Database Engine Tuning Advisor (DETA). It is important to know how to configure and use DETA.

Objectives

After completing this lesson, you will be able to:

- Explain performance tuning.
- Describe the available options for performance tuning.
- Use Database Engine Tuning Advisor.
- Describe the available DETA tuning options.

Overview of Performance Tuning



Key Points

Performance tuning is an iterative process of incremental improvement made on an ongoing basis.

First of all, it is important to define an initial goal based on any problems that might be apparent. "It needs to be faster" is not usually a well enough defined goal. Based on the chosen goal, you then need to select the appropriate tools to monitor the system and provide the metrics that will help you reach the goal. SQL Server provides a number of tools to help with this phase. (An overview of the available tools will be provided in the next topic).

In the chosen tools, you need to identify the metrics that you will use, from the list of available metrics. A baseline needs to be created via monitoring and a decision made about the changes to implement.

Based on the results, a strategy is then developed to overcome the issues that have been detected. The strategy is then implemented, the results are monitored and the process begins again.

Work On Causes Rather Than Symptoms

A key concept when performance tuning is that you need to try to find the underlying causes of problems, rather than constantly battling against the symptoms. For example, a client might ask you to help resolve a blocking issue. It is often the case that excessive blocking occurs because of long running queries. By fixing the queries and indexing so that the queries run quickly, the blocking issues often disappear.

As another common example, systems often appear to have I/O performance problems yet in many cases, this could be caused by the lack of available memory in the system.

Question: When would you know that performance tuning is complete?

Available Options for Performance Tuning

- Database Engine Tuning Advisor
- SSMS
 - Activity Monitor
 - Standard Reports
 - Custom Reports
- Dynamic Management Objects
- SQL Server Data Collection
- SQL Server Profiler
- SQL Trace
- SQL Server Extended Events
- Reliability and Performance Monitor

Key Points

SQL Server provides a number of tools that can be used to perform performance monitoring and tuning. Each of them is useful in certain scenarios and you will often need to combine several tools to achieve the desired outcome.

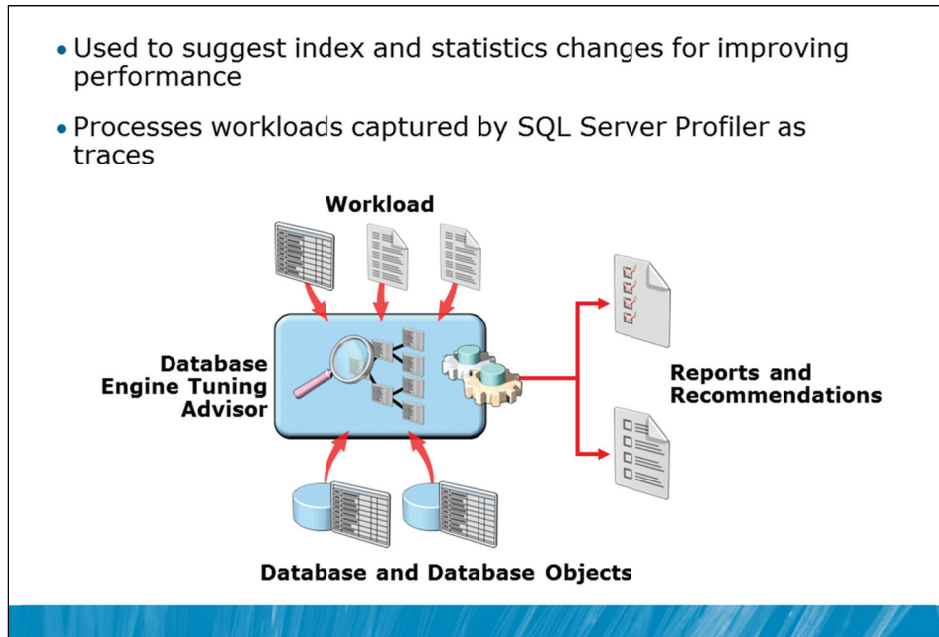
Tool	Description & Location
Database Engine Tuning Advisor	Tool provided with SQL Server for tuning indexes and statistics. DETA is discussed in this lesson.
SSMS	Core management tool provided with SQL Server. Aspects that are useful in performance tuning (activity monitor, standard and custom reports) are discussed in Module 18.
Dynamic Management Objects	Database objects that provide insight into internal SQL Server operations. Many useful DMOs are discussed in Module 18.

(continued)

Tool	Description & Location
SQL Server Data Collection	Automated system for collecting and storing performance data along with a set of standard reports. Data collection is discussed in Module 18.
SQL Server Profiler	Tracing and profiling tool that was discussed in the first lesson in this module.
SQL Trace	Programming interface for tracing SQL Server access that is discussed in the last lesson in this module.
SQL Server Extended Events	Light-weight eventing architecture. You have seen how to use Extended Events Profiler to capture server activity. Working directly with Extended Events (not via a GUI interface) is out of scope for this training.
Distributed Relay	Advanced tool for replaying workloads across a potentially distributed set of servers. Distributed Relay is out of scope for this training.
Reliability and Performance Monitor	Standard Windows performance tool that is discussed in Module 18.

Question: Which tools might be useful for real time monitoring?

Introduction to the Database Engine Tuning Advisor



Key Points

The Database Engine Tuning Advisor utility analyzes the performance effects of workloads run against one or more databases. Typically these workloads are obtained from traces captured by SQL Server Profiler or the SQL Trace facility. (SQL Trace is discussed later in the next lesson). After analyzing the effects of a workload on your databases, Database Engine Tuning Advisor provides recommendations for improving the performance of your system.

Database Engine Tuning Advisor

In SQL Server 2000 and earlier, a previous version of this tool was supplied. It was called the "Index Tuning Wizard". In SQL Server 2005, the name was changed as the tool evolved to be able to provide a broader range of recommendations.

Database Engine Tuning Advisor was further enhanced in SQL Server 2008 with improved workload parsing, integrated tuning, and the ability to tune multiple databases concurrently.

Workloads

A workload is a set of Transact-SQL statements that executes against databases that you want to tune. The workload source can be a file containing Transact-SQL statements, a trace file generated by SQL Profiler, or a table of trace information, again generated by SQL Profiler. SQL Server Management Studio also has the ability to launch Database Engine Tuning Advisor to analyze an individual statement.

Recommendations

The recommendations that can be produced include suggested changes to the database such as new indexes, indexes that should be dropped, and depending on the tuning options you set, partitioning recommendations. The recommendations that are produced are provided as a set of Transact-SQL statements that would implement the suggested changes. You can view the Transact-SQL and save it for later review and application, or you can choose to implement the recommended changes immediately.

Be careful of applying changes to a database without detailed consideration, especially in production environments. Also, ensure that any analysis that you perform is based on appropriately sized workloads so that recommendations are not made based on partial information.

Question: Why is it important to tune an entire workload rather than individual queries?

Database Engine Tuning Advisor Options

- Limit the time for analysis
 - Analysis can take a long time for large schemas and workloads
- Determine the types of recommendations that should be returned

☒ Limit tuning time

Stop at: Saturday, 15 January 2011 2:15 PM

Physical Design Structures (PDS) to use in database

☐ Indexes and indexed views ☐ Indexed views ☐ Include filtered indexes
☒ Indexes ☐ Nonclustered indexes
☐ Evaluate utilization of existing PDS only

Partitioning strategy to employ

☒ No partitioning ☐ Full partitioning
☐ Aligned partitioning

Physical Design Structures (PDS) to keep in database

☐ Do not keep any existing PDS ☐ Keep indexes only
☒ Keep all existing PDS ☐ Keep clustered indexes only
☐ Keep aligned partitioning

Key Points

DETA provides a rich set of configuration options that allow you to configure what analysis should be performed and how optimization recommendations should be made.

Executing DETA on large workloads can take a long time, particularly on systems that also have large numbers of database objects. You can configure DETA to limit the time it will spend on analysis and to return the results that it has obtained up to the time limit.

You can also configure which types of recommendations should be made, along with whether or not you wish to see recommendations that involve dropping existing objects.

Exploratory Analysis

Database administrators can use Database Engine Tuning Advisor also to perform exploratory analysis. Exploratory analysis involves a combination of manual tuning and tool-assisted tuning. To perform exploratory analysis with Database Engine Tuning Advisor, use the user-specified configuration feature. The user-specified configuration feature allows you to specify the tuning configurations for existing and hypothetical physical design structures, such as indexes, indexed views, and partitioning. The benefit of specifying hypothetical structures is that you can evaluate their effects on your databases without incurring the overhead of implementing them first.

You can create an XML configuration file to specify a hypothetical configuration. The configuration can then be used for analysis. The analysis can be performed either in isolation or relative to the current configuration. This type of analysis can also be performed using a command line interface.

Question: What is the disadvantage of limiting the analysis time?

Demonstration 2A: Using the Database Engine Tuning Advisor

In this demonstration you will see how to use the Database Engine Tuning Advisor to generate indexing and partitioning recommendations for a query

Demonstration Setup

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_17_PRJ\10775A_17_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **21 – Demonstration 2A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Question: Should you immediately apply the recommendations made by DETA to your server?

Lesson 3

Working with Tracing Options

- Overview of SQL Trace
- SQL Trace vs. SQL Server Profiler
- Demonstration 3A: Configuring SQL Trace
- Retrieving Trace Output
- Replaying Traces
- Default Trace
- Combining Traces with Reliability and Performance Monitor Logs
- Demonstration 3B: Combining Traces with Reliability and Performance Monitor Logs

In many larger organizations, you will not be able to use the graphical SQL Server Profiler tool in production environments as it places too high a load on the systems being profiled.

It is still possible, however, to use the SQL Trace programming interface that is based on system stored procedures, to create lightweight traces. It is important to know how to use these traces in place of SQL Server Profiler and how to use SQL Server Profiler to help you to configure SQL Trace configurations.

Objectives

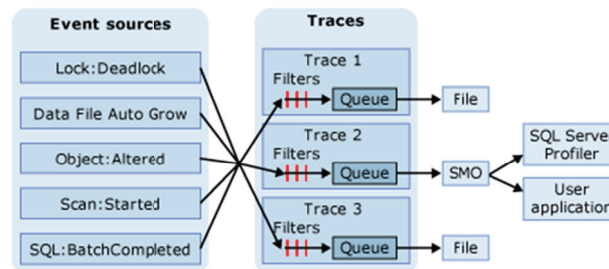
After completing this lesson, you will be able to:

- Describe SQL Trace.
- Describe the differences between SQL Trace and SQL Server Profiler.
- Retrieve trace output.
- Replay traces.
- Use the default trace.
- Combine traces with Reliability and Performance Monitor logs.

Overview of SQL Trace

SQL Trace is a programming interface that is based on system stored procedures and is used for capturing activity against SQL Server

- SQL Server Profiler can be used to script SQL traces
- Events can be sent to files or SMO objects
- SQL Server Profiler uses SMO to create and consume traces



Key Points

SQL Trace is a feature running within the database engine to create and run traces. Traces are managed using system stored procedures. Internally, SQL Server Profiler makes calls to the SQL Trace facility in SQL Server when SQL Server Profiler needs to create or manage traces.

Traces run in the process of SQL Server database engine and can write events to a file or to an application using SQL Server Management Objects (SMO) objects. The information you learned about how events, columns, and filtering work in SQL Server Profiler is directly applicable to how the same objects work within SQL Trace.

Implementing traces can at first appear difficult as you need to make many stored procedure calls to define and run a trace. However, the graphical interface in SQL Server Profiler can be used to create a trace and to then script the trace for use with SQL Trace. Very few changes typically need to be made to the SQL Trace script files that are created by SQL Server Profiler, such as the path to output files.

SQL Trace vs. SQL Server Profiler

SQL Trace	SQL Server Profiler
• Defined through procedures • Runs directly within the database engine • Writes events to files or SMO • Used for: • Long term monitoring • Performance-critical traces • Large traces	• Use via a Graphical tool • Utilizes SQL Trace • Write to files or database tables • Used for: • Debugging on test systems • Short term analysis • Small traces

Key Points

It is important to understand the differences between SQL Trace and SQL Server Profiler and to understand where each tool should be used.

- SQL Trace needs to be defined using a series of system stored procedure calls whereas SQL Server Profiler provides a graphical interface for configuration and for controlling the tracing activity.
- SQL Trace runs directly inside the database engine whereas SQL Server Profiler runs on a client system (or on the server) and communicates to the database engine by the use of the SQL Trace procedures.
- SQL Trace can write events to files or to applications based on SMO whereas SQL Server Profiler can write events to files or to database tables.
- SQL Trace is useful for long running, performance-critical traces, or for very large traces that would significantly impact the performance of the target system whereas SQL Server Profiler is more commonly used for debugging on test systems, for performing short term analysis, or for capturing small traces.



Note The option "Server processes trace data" in SQL Server Profiler is not the same option as scripting a trace and starting it directly through stored procedures. The option creates two traces: one trace that directly writes to a file and a second trace to send the events through SMO to SQL Server Profiler.

Traces do not automatically restart after the server instance restarts. If a trace needs to be run constantly, the trace needs to be scripted, launched from within a stored procedure, and the stored procedure marked as a startup procedure.

Question: Which option would have less impact on a traced system?

- a) SQL Trace with output to a file
- b) SQL Profiler with output to a table

Demonstration 3A: Configuring SQL Trace

In this demonstration, you will see:

- How to script a trace using SQL Server Profiler
- How to start a SQL Trace
- How to viewing a SQL Trace using SQL Server Profiler

Demonstration Steps

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_17_PRJ\10775A_17_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **31 – Demonstration 3A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Retrieving Trace Output

- SQL Server Traces are written to files
- Traces can be analyzed by:
 - Opening the trace in SQL Server Profiler
 - Importing the trace into a SQL Server table

```
CREATE TABLE dbo.tracetable
(
  TextData nvarchar(max) NULL,
  BinaryData varbinary(max) NULL,
  ...
);
INSERT INTO dbo.tracetable
SELECT * FROM
fn_trace_gettable('L:\Traces\adworks.trc',default);
```

Key Points

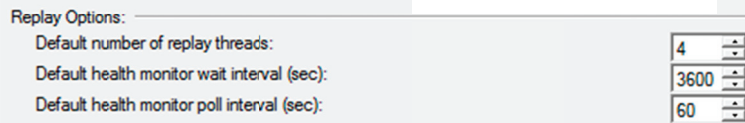
Traces that are written to trace files are not easy to read directly and are somewhat difficult to parse from within an application.

Two options are provided by SQL Server to make it easy to work with the contents of trace files:

- SQL Server Profiler can be used to open trace files. Within SQL Server Profiler the trace output can be filtered and grouped for analysis. SQL Server Profiler is especially useful for working with small trace files.
- Trace files can be imported into SQL Server using the `fn_trace_gettable` system function. Reading the files into a table is particularly useful when you need to analyze large volumes of trace data as the table can be indexed to improve the speed of common queries against the captured data. T-SQL queries can then be used to analyze and filter the data.

Replaying Traces

- Specific Events and Columns must be present in trace file
 - Use TSQL_Replay trace template
- SQL Server Profiler provides a playback engine to reproduce activity
 - Validate configuration changes
 - Test performance changes



- SQL Server 2012 provides a distributed replay utility
 - Simulate concurrent client computers
 - Simulate mission-critical workloads

Key Points

SQL Server Profiler provides the ability to replay trace files. The ability to replay trace files is useful for validating changes that you are considering making to a system or for testing a workload against new hardware, indexes, or physical layout changes.

The replay does not need to be performed against the same system that the trace events were captured on but the system must be configured in a very similar way. This particularly applies to objects such as databases and logins.

Distributed Replay

SQL Server 2012 introduced a Distributed Replay feature for traces captured by either SQL Server Profiler or SQL Trace. Distributed Replay is not limited to replaying the workload from a single computer and provides a more scalable solution than the replay option provided by SQL Server Profiler. This capability can allow improved simulation of mission-critical workloads.



Note In addition to the replay capabilities of SQL Server Profiler and the Distributed Replay Utility, a set of unsupported utilities from the SQL Server Product Support Group are available. These unsupported utilities are called the RML Utilities and are capable of replaying SQL Server workloads with advanced features such as synchronized replay through several clients. The RML utilities can also be used for analyzing trace files and are available for download from support.microsoft.com.

Before you can replay captured traces, you need to ensure that all columns that are required for replay have in fact been captured. To make this easier, SQL Server Profiler provides a template called TSQL Replay. You can use that template as a starting point but make sure you do not remove any columns that are included in the template.

Question: When might it be a good idea to test systems using the replay functionality?

Default Trace

- Traces mostly configuration changes for troubleshooting
 - Written to log directory of SQL Server
 - Uses rollover files
- Is on by default
 - Can be configured using sp_configure:

```
EXEC sp_configure 'show advanced options' , 1;
RECONFIGURE;

EXEC sp_configure 'default trace enabled' , 1;
RECONFIGURE;

EXEC sp_configure 'show advanced options' , 0;
RECONFIGURE;
```

Key Points

Beginning with SQL Server 2005, a default server side trace is started automatically whenever SQL Server starts. This trace is called the default trace and has the trace id of 1. The default trace is a lightweight trace that only keeps up to 5MB of data at any point in time. It captures the following events:

Object	Events
Database	Data file auto grow, Data file auto shrink, Database mirroring status change, Log file auto grow, Log file auto shrink
Errors and Warnings	Errorlog, Hash warning, Missing Column Statistics, Missing Join Predicate, Sort Warning
Full-Text	FT Crawl Aborted, FT Crawl Started, FT Crawl Stopped
Objects	Object Altered, Object Created, Object Deleted

(continued)

Object	Events
Security Audit	Audit Add DB user event Audit Add login to server role event Audit Add Member to DB role event Audit Add Role event Audit Add login event Audit Backup/Restore event Audit Change Database owner Audit DBCC event Audit Database Scope GDR event Audit Login Change Property event Audit Login Failed Audit Login GDR event Audit Schema Object GDR event Audit Schema Object Take Ownership Audit Server Starts and Stops
Server	Server Memory Change

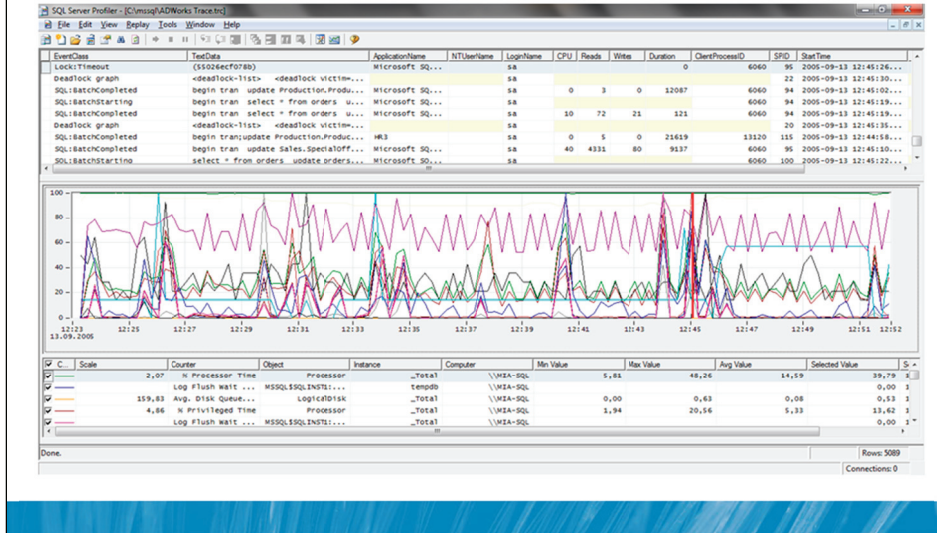
The default trace can be enabled or disabled using `sp_configure`. In the example shown in the slide, you can see how to re-enable the default trace if necessary.

The default trace cannot be modified and also the file location cannot be changed.

Question: Should the default trace be kept enabled?

Combining Traces with Performance Monitor Logs

- Correlate Performance Monitor with SQL Trace events in SQL Server Profiler



Key Points

Using SQL Server Profiler, you can open a Microsoft Windows® performance log, choose the counters you want to correlate with a trace, and display the selected performance counters alongside the trace in the SQL Server Profiler graphical user interface. When you select an event in the trace window, a vertical red bar in the System Monitor data window pane of SQL Server Profiler indicates the performance log data that correlates with the selected trace event.

To correlate a trace with performance counters, open a trace file or table that contains the StartTime and EndTime data columns, and then click Import Performance Data on the SQL Server Profiler File menu. You can then open a performance log, and select the System Monitor objects and counters that you want to correlate with the trace

Question: What is the main advantage of using this feature?

Demonstration 3B: Combining Traces with Performance Monitor Logs

In this demonstration, you will see how to combine a SQL Server trace with a performance monitor log

Demonstration Steps

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_17_PRJ\10775A_17_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **32 – Demonstration 3B.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lab 17: Tracing Access to SQL Server

- Exercise 1: Capture a Trace Using SQL Server Profiler
- Exercise 2: Analyze a Trace Using Database Engine Tuning Advisor
- Challenge Exercise 3: Configure SQL Trace (Only if time permits)

Logon information

Virtual machine	10775A-MIA-SQL1
User name	AdventureWorks\Administrator
Password	Pa\$\$w0rd

Estimated time: 45 minutes

Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_17_PRJ\10775A_17_PRJ.ssmssln**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Lab Scenario

The developers for the new marketing application are concerned about the performance of their queries. When the developers were testing the application they were working with small amounts of data and performance was acceptable. The developers are unsure that they have created appropriate indexes to support the application.

You will use SQL Server Profiler to capture traces of application execution. You will then analyze the traces using the Database Engine Tuning Advisor.

If you have time, you will configure traces using the SQL Trace system stored procedures.

Exercise 1: Capture a Trace Using SQL Server Profiler

Scenario

You have isolated the workload that causes the poor performance and you want to test the queries in a development environment. In this exercise, you will generate sample workload and you will capture database activity using SQL Server Profiler.

The main tasks for this exercise are as follows:

1. Create and start a suitable SQL Server Profiler trace.
2. Execute the workload.
3. Stop the trace.

► Task 1: Create and start a suitable SQL Server Profiler trace

- Create a SQL Server Profiler trace based upon the following settings:
 - Tuning template
 - Filtered to the MarketDev database only
 - Rollover files disabled
 - Maximum trace file size is 500 MB
- Start the trace.

► Task 2: Execute the workload

- Open and execute the file 51 – Lab Exercise 1.sql which is a workload file to be analyzed.



Note Ensure that you configure the workload window as per the instructions at the top of the script.

► Task 3: Stop the trace

- When the query completes, stop the trace.
- Close SQL Server Profiler.

Results: After this exercise, you should have captured a workload using SQL Server Profiler.

Exercise 2: Analyze a Trace Using Database Engine Tuning Advisor

Scenario

You want to take the results captured from the SQL Trace and identify any changes that could be made to improve performance. In this exercise you will analyze captured activity using the Database Engine Tuning Advisor.

The main tasks for this exercise are as follows:

1. Analyze the captured trace in Database Engine Tuning Advisor.
2. Review the suggested modifications.

- ▶ **Task 1: Analyze the captured trace in Database Engine Tuning Advisor**
 - Analyze the captured trace file using Database Engine Tuning Advisor.
- ▶ **Task 2: Review the suggested modifications**
 - Review the suggested modifications.
 - Close Database Engine Tuning Advisor.

Results: After this exercise, you should have analyzed the trace using Database Engine Tuning Advisor and reviewed the suggested modifications.

Challenge Exercise 3: Configure SQL Trace (Only if time permits)

Scenario

You have noticed that when SQL Profiler is running on the production server that performance is reduced. You want to capture the performance metrics and reduce the impact on the server. In this exercise, you will capture activity using the SQL Trace stored procedures to show how this can be used instead of SQL Profiler, to lessen the impact on the server.

The main tasks for this exercise are as follows:

1. Create a script that uses SQL Trace procedures to implement the same type of capture as you performed in Exercise 1 but with a different trace name.
 2. Test that the script works as expected by using the same workload.
 3. Analyze the new captured output and note if the suggested changes are identical to those suggested in Exercise 2.
- ▶ **Task 1: Create a script that uses SQL Trace procedures to implement the same type of capture as you performed in Exercise 1 but with a different trace name**
 - Create, start, and stop the same trace that you used in Exercise 1 but call the trace `ProsewareTrace2`.
 - Export the trace definition to a file.
 - ▶ **Task 2: Test that the script works as expected by using the same workload**
 - Configure and start the saved trace definition file. Make sure that you note the trace ID.
 - Re-execute the 51 – Lab Exercise 1.sql workload file.
 - ▶ **Task 3: Analyze the new captured output and note if the suggested changes are identical to those suggested in Exercise 2**
 - Stop the trace.
 - Open and review the captured trace file.

Results: After this exercise, you should have captured a trace using SQL Trace.

Module Review and Takeaways

- Review Questions
- Best Practices

Review Questions

1. What is the main purpose of Database Engine Tuning Advisor?
2. What can be used to test a workload after configuration changes?

Best Practices

1. Use SQL Server Profiler to perform short traces for debugging and other purposes.
2. Use SQL Trace for large and long running traces.
3. Use SQL Server Profiler to define traces and script them for SQL Trace.
4. Import trace data into a database table for advanced analysis.
5. Use Database Engine Tuning Advisor to analyze the database based on a workload rather than focusing on individual queries.

MCT USE ONLY. STUDENT USE PROHIBITED

Module 18

Monitoring SQL Server 2012

Contents:

Lesson 1: Monitoring Activity	18-3
Lesson 2: Capturing and Managing Performance Data	18-15
Lesson 3: Analyzing Collected Performance Data	18-23
Lab 18: Monitoring SQL Server 2012	18-32

Module Overview

- Monitoring Activity
- Capturing and Managing Performance Data
- Analyzing Collected Performance Data

The Microsoft® SQL Server® database engine is capable of running for very long periods without the need for administration. However, a process of ongoing monitoring of the activity that is occurring on the database server allows you to be proactive in dealing with potential issues before they arise.

SQL Server provides a number of tools that can be used for monitoring current activity and for recording details of previous activity. You need to become familiar with what each of the tools does and with how to use the tools.

When you record activity using monitoring tools, it is easy to become overwhelmed by the volume of output that a monitoring tool can provide. You also need to learn techniques for analyzing the output that is provided by the monitoring tools.

Objectives

After completing this lesson, you will be able to:

- Monitor current activity.
- Capture and manage performance data.
- Analyze collected performance data.

Lesson 1

Monitoring Activity

- Overview of Dynamic Management Views and Functions
- Viewing Activity Using Dynamic Management Views
- Demonstration 1A: Viewing Activity Using Dynamic Management Views
- Working with Activity Monitor in SQL Server® Management Studio
- Demonstration 1B: Working with Activity Monitor in SQL Server Management Studio
- Working with Performance Monitor
- Working with SQL Server Counters
- Demonstration 1C: Working with Performance Monitor

Dynamic management views (DMVs) and dynamic management functions (DMFs) provide insights directly into the inner operations of the SQL Server database engine and are useful for monitoring. It is important for SQL Server database administrators to become familiar with some of the more useful DMVs and DMFs as part of a process of ongoing server monitoring.

SQL Server Management Studio (SSMS) provides the Activity Monitor which can be used to investigate both current issues such as "Is one process being blocked by another process?" and recent historical issues such as "Which query has taken the most resources since the server was last restarted?" You should become familiar with the capabilities of Activity Monitor.

The SQL Server processes also expose a set of performance-related objects and counters to the Windows Performance Monitor. These objects and counters allow you to monitor SQL Server as part of monitoring the entire server.

Objectives

After completing this lesson, you will be able to:

- Explain Dynamic Management Views and Functions.
- View activity using Dynamic Management Views.
- Work with Activity Monitor in SSMS.
- Work with Performance Monitor.
- Work with SQL Server counters.

Overview of Dynamic Management Views and Functions

Dynamic Management Objects are virtual views and functions that provide state data for SQL Server systems

- Organized by category

Category	Description
sys.dm_exec_%	Execution and Connection
sys.dm_os_%	SQL OS related information
sys.dm_tran_%	Transaction Management
sys.dm_io_%	I/O related information
sys.dm_db_%	Database scoped information

- Many other categories available

Key Points

In earlier versions of SQL Server, database administrators often used third party tools to monitor the internal state of SQL Server. Most third party tools performed this monitoring by the use of extended stored procedures. Using extended stored procedures is not desirable as they operate within the memory space of the SQL Server process. Poorly-written programs that operate in these memory regions can cause instability or crashes of SQL Server.

SQL Server 2005 and later offer dynamic management objects to provide insight into the inner operation of the database engine without the need to use extended stored procedures. Some of the objects have been created as views and are called dynamic management views (DMVs). Other objects have been created as functions and are called dynamic management functions (DMFs).



Note The information exposed by DMVs and DMFs is generally not persisted in the database as is the case with catalog views. The views and functions are virtual objects that return state information. The state is cleared when the server instance is restarted.

DMVs and DMFs

DMVs and DMFs return server state information that can be used to monitor the health of a server instance, diagnose problems, and tune performance. There are two types of dynamic management views and functions:

- Server-scoped dynamic management views and functions.
- Database-scoped dynamic management views and functions.

All dynamic management views and functions exist in the sys schema and follow the naming convention dm_%. They are defined in the hidden resource database and are mapped to the other databases. The DMVs and DMFs are organized into a set of categories:

Category	Description
sys.dm_exec_%	These objects provide information about connections, sessions, requests, and query execution. For example, sys.dm_exec_sessions provides one row for every session that is currently connected to the server.
sys.dm_os_%	These objects provide access to SQL OS related information. For example, sys.dm_os_performance_counters provides access to SQL Server performance counters without the need to access them using operating system tools.
sys.dm_tran_%	These objects provide access to transaction management. For example, sys.dm_os_tran_active_transactions provides details of currently active transactions.
sys.dm_io_%	These objects provide information on I/O processes. For example, sys.dm_io_virtual_file_stats provides details of I/O performance and statistics for each database file.
sys.dm_db_%	These objects provide database-scoped information. For example, sys.dm_db_index_usage_stats provides information about how each index in the database has been used.

Required Permissions

To query a DMV or DMF requires SELECT permission on the object and VIEW SERVER STATE or VIEW DATABASE STATE permission, depending upon whether the object is server-scoped or database-scoped. This lets you selectively restrict access of a user or login to dynamic management views and functions. To control access for a user, first create the user in master (with any user name) and then deny the user SELECT permission on the DMVs or DMFs that you do not want them to access. After this, the user cannot select from these DMVs and DMFs, regardless of database context of the user because the DENY within that database context is processed first.

Question: Why would you want to use dynamic management views to view current activity?

Viewing Activity Using Dynamic Management Views

- Must be referenced using the sys schema
- Two basic types:
 - Current status information
 - Accumulated historical information

```
SELECT s.original_login_name, s.program_name,
       t.wait_type, t.wait_duration_ms
FROM sys.dm_os_waiting_tasks AS t
INNER JOIN sys.dm_exec_sessions AS s
ON t.session_id = s.session_id
WHERE s.is_user_process = 1
AND t.wait_duration_ms > 3000;
```

```
SELECT * FROM sys.dm_os_wait_stats
ORDER BY wait_time_ms DESC;
```

Key Points

You can see the list of available DMVs in Object Explorer in the System Views node for any database. The DMFs are shown in the Table Valued Functions node under the System Functions node that is under the master database.



Note Dynamic management views and functions need to be referenced in T-SQL statements by using the sys schema as a prefix. They cannot be referenced by one-part names.

There are two basic types of dynamic management objects:

- Objects that return real-time state information from the system
- Objects that return recent historical information

Objects That Return Real-Time State Information from the System

Most DMVs and DMFs are designed to provide information about the current state of the system. In the example on the slide, two DMVs are being joined. The sys.dm_exec_sessions view returns one row for each current user session. The sys.dm_os_waiting_tasks view returns a row for each task that is currently waiting on a resource. By joining the two views and adding a filter, you can locate a list of user tasks that have been waiting for longer than 3000 milliseconds (3 seconds).



Note Whenever a task has to wait for any resource, the task is sent to a waiting list. The task remains on that list until it receives a signal telling it that the requested resource is now available. The task is then returned to the running list, where it waits to be scheduled for execution again. This type of wait analysis is very useful when tuning system performance as it allows you to identify bottlenecks within the system.

In many cases, when a task is waiting, the cause of the wait will be some form of lock. Locks are discussed further in Module 20.

Objects That Return Historical Information

The second type of dynamic management object returns historical information. For example, you saw that the `sys.dm_os_waiting_tasks` view returned details of tasks that are currently waiting on resources. By comparison, the `sys.dm_os_wait_stats` view returns information about how often and how long any task had to wait for a specific wait_type since the SQL Server instance started.

Another useful example of a historical function is the `sys.dm_io_virtual_file_stats()` function that returns information about the performance of database files.

Question: What types of resources might SQL Server need to wait for?

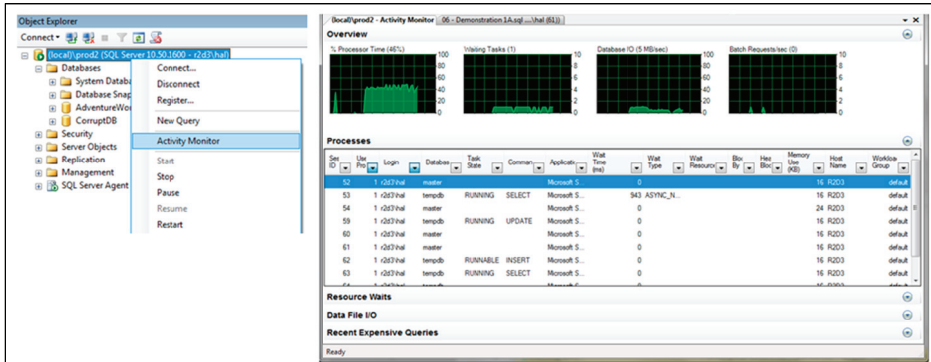
Demonstration 1A: Viewing Activity Using Dynamic Management Views

In this demonstration, you will see how to use DMVs to view performance information

Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_18_PRJ\10775A_18_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **11 – Demonstration 1A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Working with Activity Monitor in SQL Server® Management Studio



The screenshot shows the SQL Server Activity Monitor interface. On the left, the Object Explorer shows the server 'localhost\prod2' with the 'Activity Monitor' tool selected. The main window displays five sections: Overview (with four graphs for Processor Time, Waiting Tasks, Database I/O, and Batch Requests), Processes (a table of running processes), Resource Waits, Data File I/O, and Recent Expensive Queries.

SPID	User	Login	Database	Task State	Command	Application	Wait Time (ms)	Wait Resource	Wait Type	Wait Reason	Wait Class	Wait Subclass	Wait Description
52	1	rs27hal	master	RUNNING	SELECT	Microsoft S...	0						
53	1	rs27hal	tempdb	RUNNING	SELECT	Microsoft S...	343	ASYNC_N...					
54	1	rs27hal	master	RUNNING	UPDATE	Microsoft S...	0						
59	1	rs27hal	tempdb	RUNNING	UPDATE	Microsoft S...	0						
60	1	rs27hal	master	RUNNING	UPDATE	Microsoft S...	0						
61	1	rs27hal	master	RUNNING	INSERT	Microsoft S...	0						
62	1	rs27hal	tempdb	RUNNING	INSERT	Microsoft S...	0						
63	1	rs27hal	tempdb	RUNNING	SELECT	Microsoft S...	0						

- Shows information about SQL Server Processes, Waits, I/O and expensive Queries
- Accesses SQL Server through Dynamic Management Objects
- Requires VIEW SERVER STATE permission
- Can be used to kill processes that block others

Key Points

Activity Monitor is a tool in SQL Server Management Studio that shows information about processes, waits, I/O resource performance, and recent expensive queries. To start Activity Monitor, in SQL Server Management Studio, right-click the server name, and then click Activity Monitor.

Activity Monitor has five sections:

- The Overview section contains graphical information about processor usage, waiting tasks, database I/O, and batch requests per second.
- The Processes section includes detailed information on processes, their ID's, logins, databases, and commands. This section will also show details of processes that are blocking other processes.
- The Resource Waits section shows categories of processes that are waiting for resources, and information about the wait times.
- The Data File I/O section shows information about the physical database files in use, and their recent performance.
- The Recent Expensive Queries section shows detailed information about the most expensive recent queries, and resources consumed by those queries. You can right-click the queries in this section to view either the query or an execution plan for the query.

You can filter data by clicking column headings and choosing the parameter you want to view information for.

Question: Why is it important to monitor waits?

Demonstration 1B: Working with Activity Monitor in SQL Server Management Studio

In this demonstration, you will see:

- How to use Activity Monitor to view process information
- How to kill a process using Activity Monitor

Demonstration Steps

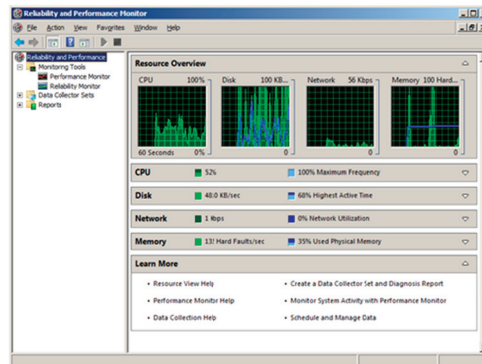
1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_18_PRJ\10775A_18_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **12 – Demonstration 1B.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Working with Performance Monitor

- Used to collect and view system metrics
- Provides real time monitoring of health and performance counters
- Creates data collector sets that capture data from different data collectors either on demand or scheduled

Key areas to monitor

- CPU
- Memory
- Disk System
- Network
- SQL Server Counters



Key Points

Windows Performance Monitor is a Microsoft Windows Server 2008 (and later) tool that brings together several previously disparate Microsoft Windows Server 2003 performance and monitoring tools. By consolidating several sources of information in one place, Windows Performance Monitor helps administrators obtain information needed to diagnose server performance and instability issues.

An important new feature in Windows Performance Monitor is the Data Collector Set, which groups data collectors into reusable elements for use with different performance monitoring scenarios. Once a group of data collectors are stored as a Data Collector Set, operations such as scheduling can be applied to the entire set through a single property change.

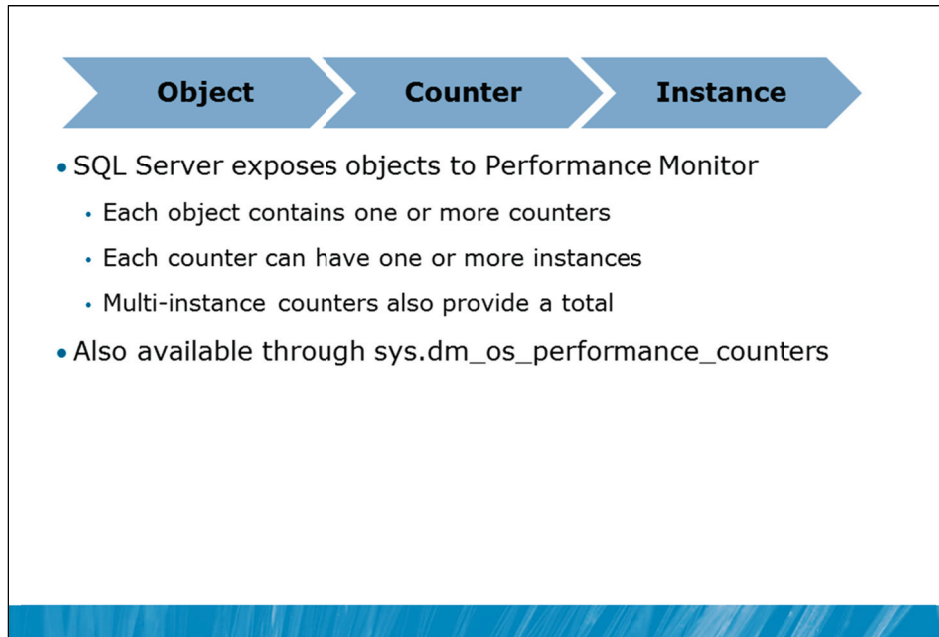
Windows Performance Monitor also includes a default Data Collector Set templates to help system administrators begin collecting performance data specific to a Server Role or monitoring scenario immediately.

Windows Performance Monitor is the key tool for monitoring Microsoft Windows® systems. Since SQL Server runs on the Windows operating system, it is important to monitor at the server level as well as at the database engine level as problems within the database engine might be caused by problems outside the database engine.

The main focus of the Windows Performance Monitor is on monitoring CPU, Memory, Disk System, and Network. After you install SQL Server, a number of SQL Server objects and counters are available within Windows Performance Monitor.

Question: Why is Performance Monitor useful when determining performance issues?

Working with SQL Server Counters



Key Points

It is important to understand some of the basic terminology used within Windows Performance Monitor:

- The term object is used to represent a resource that can be monitored.
- Every object exposes one or more counters.
- A counter might have several instances if more than one resource of that type exists.

As an example, there is an object called Processor that consists of several counters. The Processor object provides metrics related to the processors that are available on the server. One of the commonly used counters for the Processor object is the "% Processor Time" counter. That counter then offers a set of instances that represent the individual processor cores that exist on the system. In addition, a value called `_Total` represents all the instances combined.

Application Objects and Counters

Many applications expose application-related statistical data to Performance Monitor. SQL Server exposes a large number of objects and counters.

The SQL Server objects have the following naming convention:

Object Name Format	Usage
SQLServer:<object name>	Used for default instances
MSSQL\$<instance>:<object >	Used for named instances
SQLAgent\$<instance >:<object >	Used for SQL Server Agent

SQL Server also provides the same counter values through the `sys.dm_os_performance_counters` dynamic management view.

Question: Why might it be useful to query `sys.dm_os_performance_counters` to access SQL Server counters rather than using Performance Monitor?

Demonstration 1C: Working with Performance Monitor

In this demonstration, you will see how to use Windows Performance Monitor

Demonstration Steps

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_18_PRJ\10775A_18_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **13 – Demonstration 1C.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lesson 2

Capturing and Managing Performance Data

- Overview of Data Collector
- Designing a Data Collector Topology
- Configuring Data Collector
- Data Collector Security
- Monitoring Data Collector
- Demonstration 2A: Configuring Data Collector

You have seen that DMVs and DMFs provide useful information about the state of the system. The values that the DMVs and DMFs provide are not generally persisted in any way and only reside in memory while the server is running. When the server instance is restarted, these values are reset.

When DMVs and DMFs were introduced in SQL Server 2005, it was common for users to want to persist the values that the DMVs and DMFs provided. To this end, many users would create a database to hold the values, then create a job that would periodically collect and save the values.

The Data Collector system that was introduced with SQL Server 2008 formalizes this concept by creating a central warehouse for holding performance data, jobs for collecting and uploading the data to the warehouse, and a set of high quality reports that can be used to analyze the data. This lesson describes how to set up and configure the Data Collector. Lesson 3 describes the reports that are available from the data that has been collected by the Data Collector.

Objectives

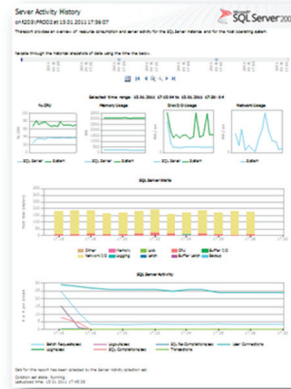
After completing this lesson, you will be able to:

- Explain the role of the Data Collector.
- Design a topology for use with the Data Collector.
- Configure the Data Collector.
- Configure security for the Data Collector.
- Monitor the Data Collector.

Overview of Data Collector

Data Collector is a component of SQL Server that collects capacity planning and performance data over time

- Low overhead data collection
- Persistence of diagnostics data
- Data retention
- Rich reporting
- Easy extensibility
- Central repository for several SQL Server instances



Key Points

Earlier in this module, you have seen how useful data can be obtained from the use of DMVs and DMFs. One disadvantage of using dynamic management objects is that most views and function only return real time data and the views and functions that return historic data are aggregations of the occurrences over time. However, to perform effective performance tuning and monitoring, an overview over time is needed, along with the ability to drill down to more detailed levels when investigating issues. The SQL Server Data Collector helps you to achieve this.

Data Collection Sets

When you configure the SQL Server Data Collector, a number of System Data Collection Sets are created. These sets define the data that needs to be collected, how often the data is uploaded to a central repository, and how long the data is retained in that repository.

The Data Collector can collect information from several locations:

- The Data Collector can query DMVs and DMFs to retrieve detailed information about the operation of the system.
- The Data Collector can retrieve performance counters that provide metrics about the performance of both SQL Server and the entire server.
- The Data Collector can also capture SQL Trace events that have occurred.

In addition to the System Data Collection Sets, the SQL Server Data Collector can be extended by the creation of user-defined Data Collection Sets. The ability to add user-defined Data Collection Sets allows users to specify the data that they wish to collect and to use the infrastructure that is provided by the SQL Server Data Collector to collect and centralize the data.

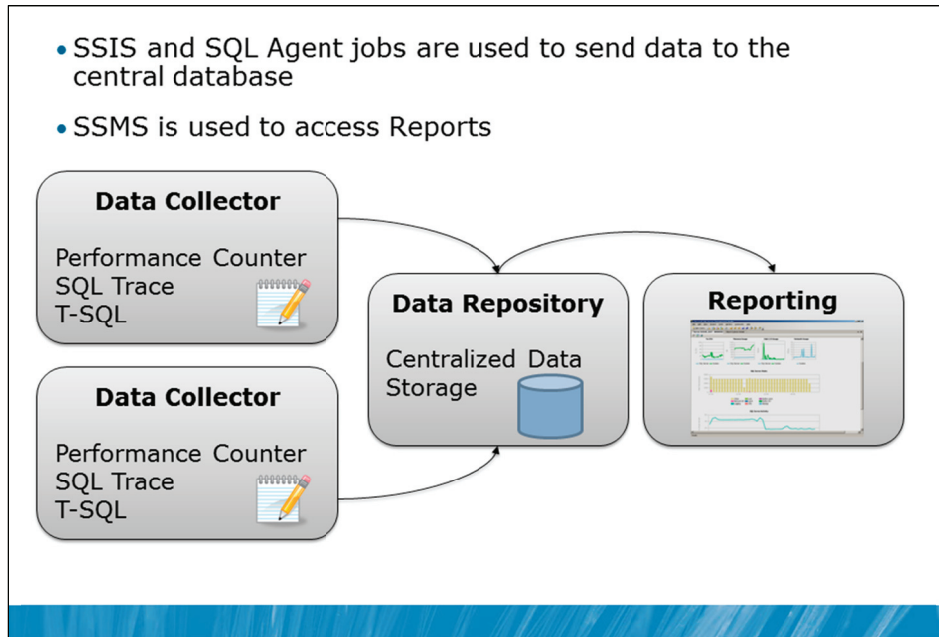
Management Data Warehouse and Reports

The data that is collected by the SQL Server Data Collector is stored in a central repository called a management data warehouse. The management data warehouse is a SQL Server database that is created by a wizard that is also used to configure data collection.

Three standard reports and a rich set of sub-reports has been provided with the Data Collector but it is possible to write your own reports that are based on either the data that is collected by the System Data Collection Sets or data that is collected by user-defined Data Collection Sets.

Question: Can Data Collector be used for real time monitoring?

Designing a Data Collector Topology



Key Points

There are three components that make up the data collection system in SQL Server:

- The Data Collector is made up of a set of jobs that run on the local server that collect data from dynamic management objects, performance counters, and SQL Trace events and upload that data to a central repository.
- The central repository that can consolidate data from multiple server instances.
- The rich reports which are available to analyze the data in the management data warehouse. The reports are accessed using SSMS.

A large enterprise should consider the use of a standalone system for the management data warehouse. There are two goals for creating a centralized management data warehouse:

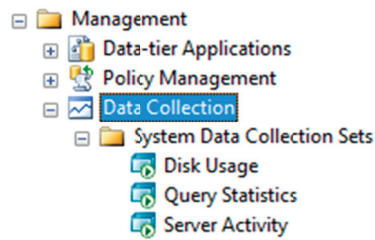
- You can access reports that combine information for all server instances in your enterprise.
- You can offload the need to hold collected data and to report on it from the production servers.

The data collector has two methods for uploading captured performance data into the central warehouse. Low volume information is sent immediately to the warehouse. Higher volume information is cached locally first and then uploaded to the warehouse using SSIS.

Question: Is it possible to configure all of the data collector features on a single instance?

Configuring Data Collector

- Wizard-based Setup
- Two-step process:
 1. Create the Management Data Warehouse
 2. Configure Data Collector on all instances to be monitored
- Plan for enough space in Management Data Warehouse
- Configuration creates System Data Collection Sets
 - Data to be collected
 - Frequency of collection
 - Retention period
- Custom Data Collection Sets can be added



Key Points

Installing the data collection system in SQL Server involves two steps:

- Configuring a central management data warehouse to hold the collected data.
- Configuring each server instance to collect and upload the required data.

SQL Server provides a wizard that is used for both these tasks. The first step when running the wizard is to create a management data warehouse. The management data warehouse can store collection data from many SQL Server instances.

Planning Space Requirements

Sufficient disk space needs to be available to support the needs of the management data warehouse. In a typical configuration, you should allow at least 300MB per day for each instance that is managed.

Impact on Local Systems

The only processes that are run on the local instances are the jobs that are used to collect and upload the data to the management data warehouse. Some data is collected very regularly and is cached locally and later uploaded using SSIS. Other data is captured infrequently and uploaded immediately.

System Data Collection Sets are created automatically during setup. They can be enabled and disabled as needed and both the frequency of collection and the retention periods for collected data can be customized for System Data Collection Sets as well as for user-defined Data Collection Sets.

It is also important to consider the security requirements for the Data Collector. The security requirements are discussed in the next topic.

Question: What might be interesting to include as a custom collection set?

Data Collector Security

Roles for the Management Data Warehouse:

Role	Description
mdw_admin	Full access to the management data warehouse
mdw_writer	Write and read access. Needed by data collectors
mdw_reader	Read access. Needed by users accessing Reports

Roles for Configuring the Data Collector:

Role	Description
dc_admin	Full administrator access to the configuration
dc_operator	Read and update access to the configuration
dc_proxy	Read access to the configuration

Key Points

The SQL Server Data Collector system has two security aspects that need to be considered:

- Who can access the Management Data Warehouse?
- Who can configure the Data collection?

Database roles exist for both these purposes as shown in the tables on the slide.



Note A user needs to be member of the mdw_reader role to be able to access the reports and the collected data.

As well as needing to be a member of the mdw_writer role to upload data, the jobs that collect the data need whatever permissions are required to access the data that they are collecting.

Question: If the SQL Server Agent service account does not have sufficient privileges to access the management data warehouse to upload the data, what configuration changes could you make to allow this?

Monitoring Data Collector

- Configuration and Log written to msdb
 - Implemented via stored procedures and SSIS logging features
- Three logging levels are available
- Retention is based on collection set retention
- Logs can be viewed using T-SQL or Log File Viewer

Date	Log ID	Source	Collection Set Name
9/08/2010 4:39:28 PM	26	Disk Usage	Disk Usage
9/08/2010 4:27:52 PM	3	Query Statistics	Query Statistics
9/08/2010 4:27:49 PM	2	Server Activity	Server Activity
9/08/2010 4:27:46 PM	1	Disk Usage	Disk Usage

Key Points

As with other SQL Server Agent jobs, the history of the jobs that are used by SQL Server Data Collector to capture and upload performance data are held in the SQL Server Agent job history tables and can be viewed using the standard SQL Server Agent job history viewer.

In addition to the standard job history, the Data Collector also logs configuration and other log information to tables in the msdb database. The Data Collector calls stored procedures to add the log information and also uses the SSIS logging features for the SSIS packages that it executes.

The data that is logged into the msdb database is kept with the same retention period settings as the Data Collection Sets that it relates to. The information that is retained can be viewed through the log file viewer or by querying the following views:

- fn_syscollector_get_execution_details()
- fn_syscollector_get_execution_stats()
- syscollector_execution_log
- syscollector_execution_log_full
- syscollector_execution_stats

Three levels of logging are available and can be set by calling the sp_syscollector_update_collection_set system stored procedure. The lowest level of logging records starts and stops of collector activity. The next level of logging adds execution statistics and progress reports. The highest level of logging adds detailed SSIS package logging.

Question: Why is it better to use the Data Collector logs than the data collector job history for troubleshooting Data Collector?

Demonstration 2A: Configuring Data Collector

In this demonstration you will see how to configure data collector

Demonstration Steps

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_18_PRJ\10775A_18_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **21 – Demonstration 2A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lesson 3

Analyzing Collected Performance Data

- Overview of Data Collector Reports
- Disk Usage Report
- Demonstration 3A: Viewing the Disk Usage Report
- Server Activity Report
- Demonstration 3B: Viewing the Server Activity Report
- Query Statistics Report
- Demonstration 3C: Viewing the Query Statistics Report

Once performance data has been collected from a number of server instances and consolidated into a management data warehouse, the data can then be analyzed. You can write your own custom reports using SQL Server Reporting Services, via custom reports in SSMS, or via T-SQL queries. Most users will find the standard reports that are supplied with SQL Server to be sufficient, without the need to write additional reports. It is important to be familiar with the information that is contained in the standard reports and with how to navigate within the reports.

Objectives

After completing this lesson, you will be able to:

- Explain the available Data Collector reports.
- Use the Disk Usage report.
- Use the Server Activity report.
- Use the Query Statistics report.

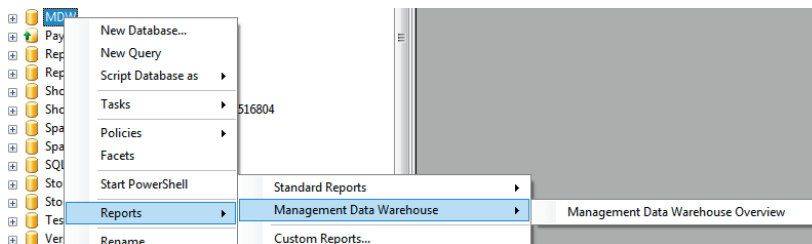
Overview of Data Collector Reports

Data Collector provides a series of cross-linked reports on historical data. The reports are visible from within SQL Server Management Studio

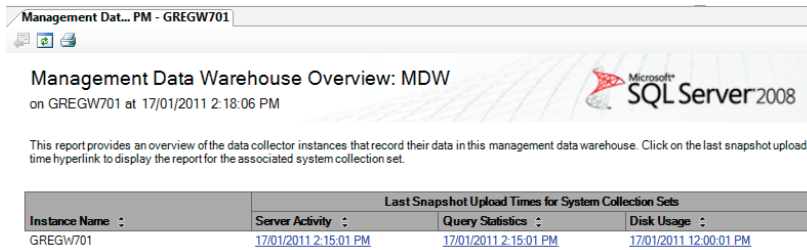
- Disk Usage Summary
 - Trends and details on disk and file usage
- Query Statistics History
 - Most expensive queries ranked by: CPU, Duration, Reads, Writes
- Server Activity History
 - CPU, Memory, Disk and Network I/O
 - SQL Server Waits and SQL Server Activity

Key Points

The SQL Server Data Collector provides a series of standard reports that are based on the data that is collected from the System Data Collection Sets and that has been consolidated into a centralized management data warehouse. The reports are accessed from within SQL Server Management Studio as a right-click option from the management data warehouse database as shown in the screen shot below:



The overview page for the Management Data Warehouse reports provides links to the main report areas as shown in the following screenshot:



Last Snapshot Upload Times for System Collection Sets			
Instance Name	Server Activity	Query Statistics	Disk Usage
GREGW701	17/01/2011 2:15:01 PM	17/01/2011 2:15:01 PM	17/01/2011 12:00:01 PM

From this overview page, there are three main report areas that are linked:

- Disk Usage Summary
- Query Statistics History
- Server Activity History



Note Each report can be printed and exported to PDF or Microsoft Excel® files for further analysis.

Question: What period would the Data Collector reports cover?

Disk Usage Report

- Based on Disk Usage System Data Collection Set
 - T-SQL collector type
 - Not cached, runs every 6 hours by default
 - Stored for 730 days by default

Database Name :	Database				Log			
	Start Size (MB)	Trend	Current Size (MB)	Average Growth (MB/Day)	Start Size (MB)	Trend	Current Size (MB)	Average Growth (MB/Day)
AdventureWorks	176.75		250.00	73.25	18.00		50.00	32
CorruptDB	4.38		15.00	10.625	1.00		2.00	1
master	4.00		4.00	0	1.25		1.25	0
mdw	100.00		100.00	0	10.00		10.00	0
model	2.25		2.25	0	0.75		0.75	0
msdb	16.25		17.88	1.625	3.38		3.38	0
tempdb	8.00		8.00	0	2,110.31		2,110.31	0

Key Points

The disk usage reports are based on the Disk Usage System Data Collection Set. By default, the collection set gathers disk usage data every six hours and keeps the data for two years.

Note that by default, the Disk Usage System Collection Set retains data for much longer than other System Data Collection Sets. However, as the amount of data collected by this collection set is quite small, this should not be a concern and the value of being able to track individual file space usage over time warrants the use of this longer retention period.

This report also includes a number of hyperlinks that lead to a set of linked sub-reports for drilling into the usage data.

Question: What might be a good custom report that could be created based on disk usage data?

Demonstration 3A: Viewing the Disk Usage Report

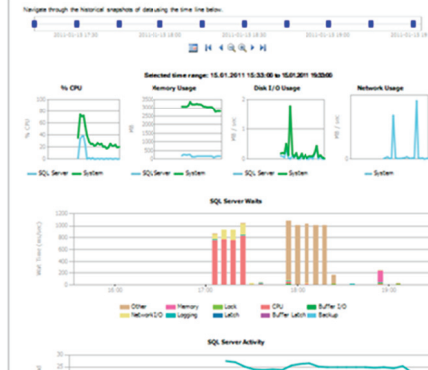
In this demonstration, you will see the information that is available in a disk usage report

Demonstration Steps

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the **virtual machine**, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the **Connect to Server** window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_18_PRJ\10775A_18_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
 - Open and follow the instructions in the **21 – Demonstration 2A.sql** script file.
2. Open the **31 – Demonstration 3A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Server Activity Report

- Based on Server Activity System Data Collection Set
 - DMV snapshot and performance counter collector types
 - Runs every 60 seconds and uploaded every 15 minutes by default
 - Retained for 14 days by default
- Provides many sub-reports with detailed information on SQL Server and Windows processes
- Crossed linked to Query Statistics Reports



Key Points

The Server Activity Report is based on the Server Activity System Data Collection Set. The collection set gathers a lot of SQL Server related statistics such as waits, locks, latches, and memory statistics that are accessed using DMVs. In addition, the collection set gathers Windows and SQL Server Performance counters to retrieve information such as CPU and memory usage from the system and from the processes that are running on the system. The collection runs every sixty seconds and is uploaded every fifteen minutes by default. The history is retained for fourteen days by default.

This report has a large number of linked sub-reports that provide much deeper information than is provided on the initial summary. The initial report is a dashboard that provides an overview. If you investigate this report, you will find that almost every item displayed is a possible drill-through point. For example, you can click on a trend line in a graph to find out the values that make up the trend.

Question: Why is it important that data collection also tracks the memory usage of other processes running on the system?

Demonstration 3B: Viewing the Server Activity Report

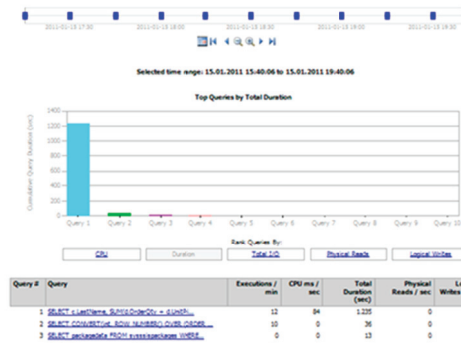
In this demonstration, you will see the information that is available in a server activity report

Demonstration Steps

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_18_PRJ\10775A_18_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
 - Open and follow the instructions in the **21 – Demonstration 2A.sql** script file.
2. Open the **32 – Demonstration 3B.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Query Statistics Report

- Based on Query Statistics System Data Collection Set
 - Query activity collector type
 - Runs every 10 seconds and uploaded every 15 minutes by default
 - Retained for 14 days by default
- Expensive queries are gathered including both the query text and the query plan



Key Points

The Query Statistics Report is based on the Query Statistics System Data Collection Set that retrieves details of expensive queries. The collection set is the most intensive default collection system and, by default, runs every ten seconds. To avoid the overhead of constant uploading, the data that is collected by this collection set is cached in the local file system and uploaded using SSIS every fifteen minutes. The data that is collected is retained for fourteen days by default but this value can be extended.

SQL Server determines expensive queries based upon:

- Elapsed time
- Worker time
- Logical reads
- Logical writes
- Physical reads
- Execution count

This report also includes a large number of linked sub-reports that can be used to drill-through to higher levels of detail. As an example, it is possible to retrieve query plans from the expensive queries that were in memory at the time the capture was performed.

Question: What is meant by the term "expensive queries"?

Demonstration 3C: Viewing the Query Statistics Report

In this demonstration, you will see the information that is available in a query statistics report

Demonstration Steps

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_18_PRJ\10775A_18_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
 - Open and follow the instructions in the **21 – Demonstration 2A.sql** script file.
2. Open the **33 – Demonstration 3C.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lab 18: Monitoring SQL Server 2012

- Exercise 1: Investigating DMVs
- Exercise 2: Configure Management Data Warehouse
- Exercise 3: Configure Instances for Data Collection
- Challenge Exercise 4: Work with Data Collector Reports (Only if time permits)

Logon information

Virtual machine	10775A-MIA-SQL1
User name	AdventureWorks\Administrator
Password	Pa\$\$w0rd

Estimated time: 45 minutes

Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_18_PRJ\10775A_18_PRJ.ssmssl**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Lab Scenario

The earlier versions of SQL Server that you have worked with did not include dynamic management functions and views. You have recently read about these and are interested to see how they might be used for collecting performance information.

Rather than collecting information separately for each SQL Server instance, you have decided to collect all the performance information to a central server. This will help when dealing with issues that were not reported to the helpdesk at the time they occurred.

Exercise 1: Investigating DMVs

Scenario

You have been provided with a series of sample queries that can be used to provide information from Dynamic Management Views and to provide an insight into SQL Server activity. In this exercise, you will execute these queries to gain an understanding of the potential uses for DMVs.

The main task for this exercise is as follows:

1. Investigate the use of Dynamic Management Views and Functions.

► Task 1: Investigate the use of Dynamic Management Views and Functions

- Open the script file 51 - Lab Exercise 1.sql and follow the instructions contained in the script file.

Results: After this exercise, you should have investigated the use of Dynamic Management Views and Functions.

Exercise 2: Configure Management Data Warehouse

Scenario

You want to be able to centrally collect performance information so that you can analyze the all the Adventure Works performance data a later stage. In this exercise, you will configure a server to hold the management data warehouse to collect the performance data.

The main task for this exercise is as follows:

1. Create a management data warehouse for central collection of performance data.

► Task 1: Create a management data warehouse for central collection of performance data

- Create a management data warehouse for central collection of performance data. Create the management data warehouse with a database name of MDW and on the Proseware instance.

Results: After this exercise, you should have created a management data warehouse.

Exercise 3: Configure Instances for Data Collection

Scenario

You want to configure each Adventure Works SQL Server instance so that it sends performance information to a central server. In this exercise, you will configure each of the SQL Server instances to send performance data to the newly configured data warehouse.

The main tasks for this exercise are as follows:

1. Configure data collection on each instance.
2. Configure data collection security for the AdventureWorks instance.

► **Task 1: Configure data collection on each instance**

- Configure data collection on the Proseware instance.
- Set the target for the performance data to the management data warehouse created in Exercise 2.
- Configure data collection on the AdventureWorks instance.

Results: After this exercise, you should have configured both instances for data collection.

Challenge Exercise 4: Work with Data Collector Reports (Only if time permits)

Scenario

You want to create a performance report that summarizes the performance information for all of the Adventure Works SQL Server instances. In this exercise, you will review the reports provided by the data collection system.



Note Performance data takes some time to begin to be collected. To assist with this exercise, you have been provided with a backup of another management data warehouse to use for testing reports.

The main tasks for this exercise are as follows:

1. Disable data collectors on both instances.
2. Restore a backup of the MDW database.
3. Review the available reports.

► **Task 1: Disable data collectors on both instances**

- Disable data collection on the Proseware instance.
- Disable data collection on the AdventureWorks instance.

► **Task 2: Restore a backup of the MDW database**

- Restore the MDW database from the backup file
D:\10775A_Labs\10775A_18_PRJ\10775A_18_PRJ\MDW.bak.

► **Task 3: Review the available reports**

- Create a server activity report.
- Create a disk space usage report.
- Create a query statistics report.

Results: After this exercise, you should have worked with Management Data Warehouse Reports.

Module Review and Takeaways

- Review Questions
- Best Practices

Review Questions

1. Why is it better to have a central management data warehouse for data collection, than local installations?
2. What are Dynamic Management Objects?

Best Practices

1. Use Dynamic Management Objects to perform real-time monitoring and troubleshooting.
2. Use Activity Monitor for easy access to the most relevant information.
3. Use Performance Monitor to gather metrics for Windows and SQL Server.
4. Create a central Management Data Warehouse to hold historical performance information.
5. Set up data collection to gather performance data of your SQL Server Instances.

MCT USE ONLY. STUDENT USE PROHIBITED

Module 19

Managing Multiple Servers

Contents:

Lesson 1: Working with Multiple Servers	19-3
Lesson 2: Virtualizing SQL Server	19-9
Lesson 3: Deploying and Upgrading Data-tier Applications	19-15
Lab 19: Managing Multiple Servers	19-22

Module Overview

- Working with Multiple Servers
- Virtualizing SQL Server
- Deploying and Upgrading Data-tier Applications

Database administrators that work with Microsoft® SQL Server® are increasingly being required to manage larger numbers of servers. The need to manage multiple servers provides additional challenges. Microsoft has been making substantial investments in SQL Server tooling to assist in the management of multiple servers.

While the virtualization of operating systems and applications is not new, virtualized SQL Server systems have been less common than other types of systems. However, virtualized SQL Server systems are now rapidly being deployed. It is important that you understand the core concepts involved with the virtualization of SQL Server systems, the tools that are commonly used to implement the virtualization, and with how such systems are managed.

In SQL Server 2012, Microsoft introduced a new form of database development project known as data-tier applications. While database administrators do not necessarily need to understand how to create data-tier applications, database administrators need to know how to deploy and upgrade data-tier applications.

Objectives

After completing this lesson, you will be able to:

- Work with multiple servers.
- Describe options for virtualizing SQL Server.
- Deploy and upgrade Data-Tier Applications.

Lesson 1

Working with Multiple Servers

- Overview of Central Management Servers
- Executing Multi-server Queries
- Demonstration 1A: Executing Multi-server Queries

One of the first tools that you need to become familiar with when you need to manage multiple SQL Server systems is the central management server (CMS). The CMS was first introduced in SQL Server 2008 and allows you to define groups of servers and to execute queries and apply policies against the defined groups of servers.

It is important that you are aware of how to configure a CMS, to define server groups, and to execute queries against the groups of servers. It is also important that you are aware of the limitations related to multi-server queries.

Objectives

After completing this lesson, you will be able to:

- Configure central management servers.
- Execute multi-server queries.

Overview of Central Management Servers

Central Management Servers are used to hold details of servers within an organization and to organize the servers within groups

- Configuration requires:
 - Define CMS server (typically only one server)
 - Create server groups
 - Register servers
- Group actions permitted:
 - Execute T-SQL statements
 - Evaluate and Import Policy-Based Management policies
 - Browse Object Explorer
- Usage requires Windows Authentication and membership of ServerGroupReaderRole

Key Points

For many product versions, SQL Server Management Studio has provided the ability for users to define groups of servers in the Registered Servers tool window. The typical use for this capability was to group together servers in different environments. For example, you could create a group of development servers, a group of test servers, a group of staging servers, and a group of production servers.

No actions were permitted against entire groups of servers and a key challenge with this capability was that the definition of the server groups was stored in an XML file on the local client computer that was executing SSMS.

If a user moved to a different computer, the groups of servers would no longer exist and would need to be recreated. If the list of servers in a group changed, the groups would need to be modified on every computer that was executing SSMS, if access to the groups was needed.

Central Management Server

A central management server is a server that is configured to maintain a centralized list of groups of servers that needs to be defined only once and that can be accessed by all authorized users. The centralized list is stored in tables within the msdb database of the CMS.

Users that need to access the list of servers must be members of the ServerGroupReaderRole in the msdb database. All SQL Server editions and versions can be registered through the CMS but all registrations require the use of Windows Authentication. The server that provides the CMS needs to be SQL Server 2008 or later.

While the ability to maintain central lists is useful, SQL Server 2008 and later versions provide the ability to perform actions against entire groups of computers. By right-clicking a group in the Registered Servers tool window, users can execute T-SQL statements against all servers within the group at the same time. Users can also use Object Browser to browse multiple servers concurrently.

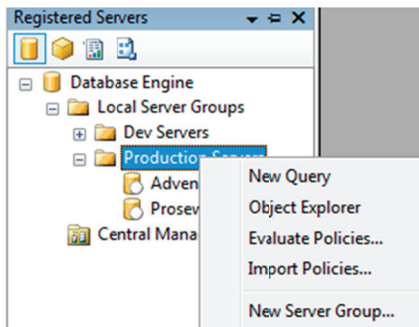


Note Users can also import and evaluate policies that are part of SQL Server Policy-Based Management (PBM) but PBM is an advanced topic that is out of scope for this course.

Question: What is the main advantage of using a central management server instead of standard registered servers in SSMS?

Executing Multi-server Queries

- T-SQL queries can be executed on multiple servers at the same time
- Results can be merged
- Additional columns are provided for source server and login information



Different permissions might apply on each server instance !

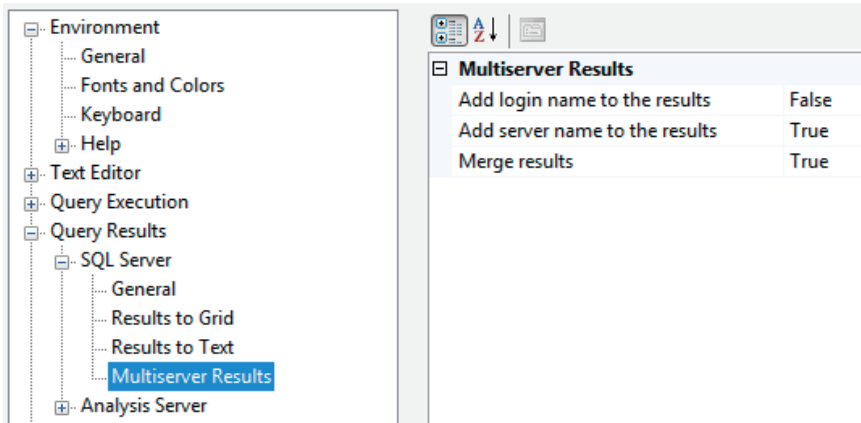
Key Points

When you open a multi-server query window, SSMS changes the color of the status bar to indicate to you that the query is not a standard query. In addition, instead of showing a server name at the bottom of the window, SSMS shows the name of the server group. The following screenshot shows the result of executing the following command against a group containing both the AdventureWorks and Proseware servers:

```
SELECT SERVERPROPERTY('IsFullTextInstalled') AS IsFullTextInstalled;
```

Results		
Messages		
	Server Name	IsFullTextInstalled
1	AdventureWorks	1
2	Proseware	1

Note the additional column called Server Name that is included in the results but was not included in the SELECT statement. For the execution of multi-server queries, SSMS provides options that determine how the results of the queries should be presented. From the Tools menu, click Options, click Query Results, click SQL Server, and then click Multi-server Results as shown in the following screenshot:



By default, SSMS will merge the results from each server into a single result set and insert a first column that contains the name of the server. Optionally, you can remove this server name, add the login name, or request that the results are not merged.

Question: What happens if the user has no permissions to execute a statement on one of the servers of a multi-server query?

Demonstration 1A: Executing Multi-server Queries

In this demonstration, you will see:

- How to create a Central Management Server
- How to execute multi-server queries

Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_19_PRJ\10775A_19_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **11 – Demonstration 1A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Lesson 2

Virtualizing SQL Server

- Discussion: Advantages and Disadvantages of Virtualizing SQL Server
- Overview of SQL Server Virtualization
- Common Virtualization Scenarios
- Considerations for Virtualizing SQL Server
- Overview of System Center Virtual Machine Manager

Virtualization of servers has been common for many years, particularly in relation to server consolidation efforts. SQL Server virtualization has been the exception. It has been much less common to see SQL Server systems virtualized. Mostly, this has been because of concerns about the intense I/O requirements that are common with SQL Server systems. This situation is now rapidly changing and it is very important that database administrators who work with SQL Server are aware of the techniques for virtualizing SQL Server and of the advantages and disadvantages associated with SQL Server virtualization.

Objectives

After completing this lesson, you will be able to:

- Discuss the advantages and disadvantages of virtualizing SQL Server.
- Describe virtualization.
- Describe available virtualization options.
- Explain considerations for virtualizing SQL Server.
- Describe System Center Virtual Machine Manager.

Discussion: Advantages and Disadvantages of Virtualizing SQL Server

- Do you use virtualization of SQL Server in your environment?
- If you are using virtualization, why are you using it?
- What are the main advantages provided by virtualization?
- What are the main concerns with virtualizing SQL Server?



Discussion Points

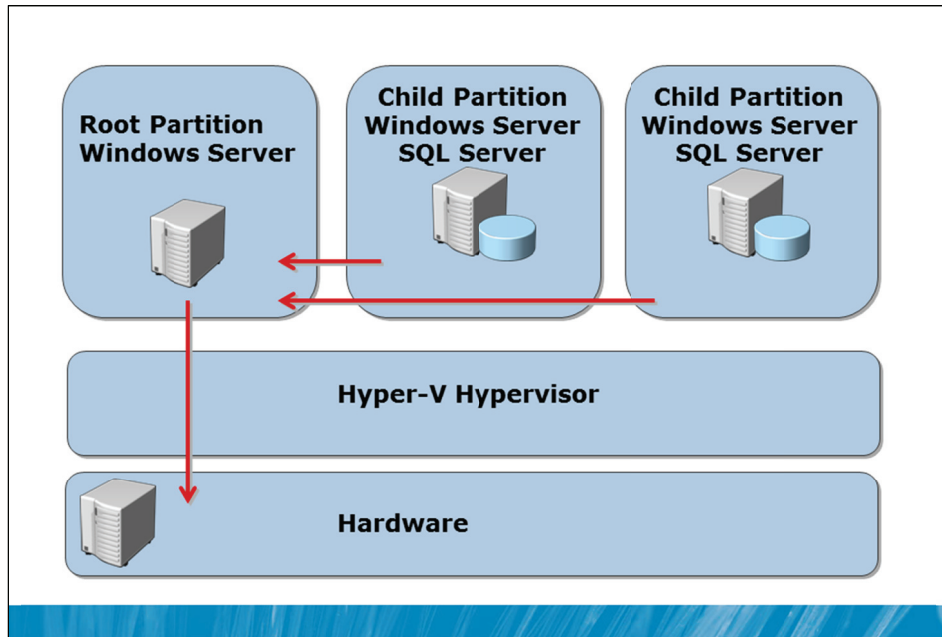
Question: Do you use virtualization of SQL Server in your environment?

Question: If you are using virtualization, why are you using it?

Question: What are the main advantages provided by virtualization?

Question: What are the main concerns with virtualizing SQL Server?

Overview of SQL Server Virtualization



Key Points

Virtualization is a technology that allows you to create and manage workloads running in separate virtual machines rather than separate physical computers. Microsoft Hyper-V® is a feature of Windows Server® 2008 (R2) x64 and is installed as an operating system role.

Hyper-V supports isolation through a concept of partitions. A Hyper-V partition is a logical unit that executes an operating system. Hardware access is performed through the root partition that is initially created as part of the operating system. None of the child partitions has direct access to the hardware. Child partitions access resources through the hypervisor that controls the whole system.

Every machine must have a root partition with Windows Server running. The root partition is used to manage the system, along with any child partitions.

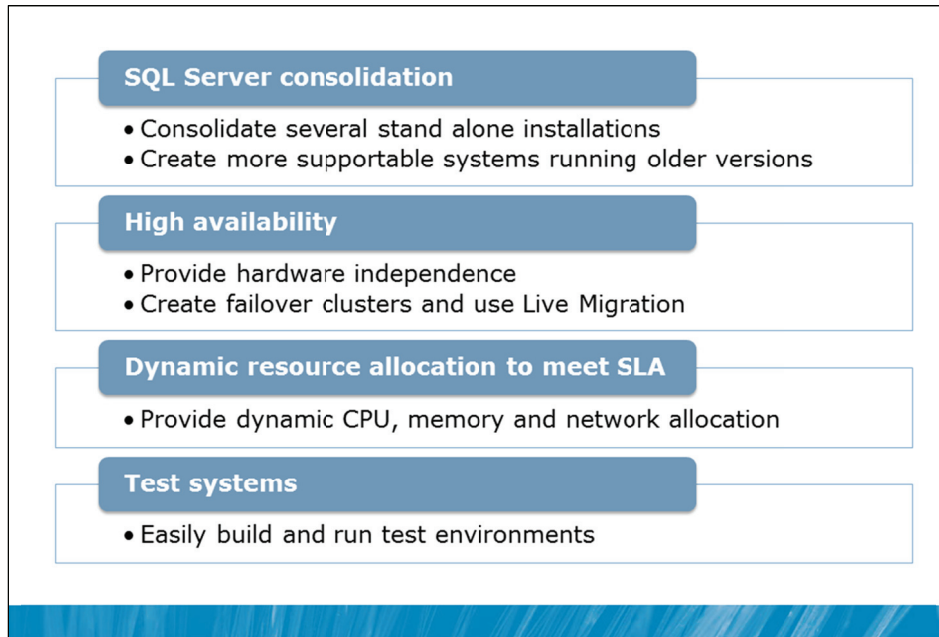
The virtualization that is provided by using Hyper-V technology not only provides better use of hardware resources, but also enables easier management and high availability through the ability of Live Migration, Snapshots, and many other features.

Live Migration allows the movement of a child partition from one computer to the child partition of another computer, while maintaining user connections throughout the process.

Snapshots allow the creation of named points in time that entire virtual machines can be reverted to on demand.

Question: What advantage would Live Migration provide over other options such as SQL Server Failover Clustering?

Common Virtualization Scenarios



Key Points

There are many reasons for virtualizing a SQL Server installation:

- One of the primary reasons for virtualizing SQL Server is to consolidate a number of standalone servers onto a single more powerful physical server. Consolidation can bring a number of benefits, particularly in terms of physical space and power consumption.
- Another core reason for virtualizing SQL Server is to move older versions of SQL Server and older server operating systems that need to be retained for application compatibility reasons, onto newer hardware platforms that are more supportable. Many older servers cannot be repaired when they fail. Virtualizing the servers alleviates much of this issue. PTV tools (physical to virtual tools) are available to help with this migration.
- Virtual servers are independent of the underlying hardware and are much easier to move to alternate hardware.
- Hyper-V based virtualization supports the creation of Hyper-V clusters across multiple physical servers. SQL Server failover clusters can be constructed above the Hyper-V clusters to provide high availability options. Live Migration can be used for manual failover of virtual machines.
- It is often difficult to balance the resource requirements of different applications. By separating the applications into separate virtual servers, dynamic balancing of resources between the virtual machines is possible.
- Test environments are easy to create and tear down when they are based on virtualization technology. Snapshots provide the ability to roll back the effect of application installations or test executions.

Question: Why would virtualization be very useful when testing applications?

Considerations for Virtualizing SQL Server

- Use POC to ensure that virtualization meet your needs
- Provide appropriate I/O resources
 - Use pass-through disks
 - Apply best practice as for non-virtualized systems
 - Use SQLIO for pre-deployment testing
- Provide appropriate CPU resources
 - Provide additional CPU for network intensive systems
- Avoid the use of emulated devices
 - Install integration components for Hyper-V that provide synthetic devices

Key Points

Before virtualizing a SQL Server (or any) workload, you should carry out an appropriate proof of concept (POC), making sure to use typical database sizes and typical workloads.

A very common mistake that is often made when virtualizing SQL Server systems is to neglect to provide sufficient I/O resources for the virtual servers. There is a common misconception that a set of virtualized servers requires fewer resources than the individual servers. Pass-through disks route I/O calls to the host system for execution and add minimal overhead to the calls. Pass-through disks need to be used in all cases. Emulated disks introduce substantial overhead and should always be avoided.

The SQLIO utility should be used to stress test the I/O subsystem before deploying SQL Server, as should be done with a non-virtualized SQL Server system. The testing should be performed from within the virtual system that will be used to run SQL Server so that all software layers are tested. The same suggestions regarding the separation of data and log drives onto separate physical drives that applies to non-virtualized servers also apply to virtualized servers.

Virtualized servers tend to make more use of CPU resources than non-virtualized servers. Network intensive systems require even greater CPU resources.

Hyper-V provides a set of integration components that increase the performance of the virtual system by replacing devices that are emulated in software with synthetic versions of the devices.

Question: What differences would exist between the needs of a virtualized SQL Server and a non-virtualized SQL Server in terms of I/O requirements?

Overview of System Center Virtual Machine Manager

System Center Virtual Machine Manager (SCVMM) is a tool that allows you to manage resource allocations between multiple virtual machines

- Component of the System Center suite
- Often used in conjunction with System Center Operations Manager (SCOM)
- Allows balancing resources across multiple virtual machines
 - Avoids much of the need to balance resources within each machine

Key Points

SQL Server 2008 introduced the SQL Server Resource Governor to help with balancing resource requirements for multiple applications on a single server. While the use of Resource Governor can assist somewhat with CPU and memory requirements that relate to the database engine and certain types of queries, users often want to balance resources at a higher level of granularity and across other SQL Server components.

One common option is to install multiple virtual machines that are each running SQL Server and to separate the applications onto the separate virtual machines. The role of System Center Virtual Machine Manager (SCVMM) is to then balance the resources that are allocated to each of the virtual machines.

SCVMM is a component of the System Center suite, along with other components such as System Center Operations Manager (SCOM) which provides an overall management framework for an organization. SCVMM is often used in conjunction with SCOM.

Lesson 3

Deploying and Upgrading Data-tier Applications

- Data-tier Application Overview
- Deploying Data-tier Applications
- Upgrading Data-tier Applications
- Extracting Data-tier Applications
- Demonstration 3A: Working with Data-tier Applications

Creating .NET based applications in Microsoft Visual Studio® is a well-refined process. The creation of projects to support the database-related aspects of the projects has been less well defined. Since Visual Studio 2005, it has been possible to create database projects in Visual Studio. Database projects could form part of solutions that include other types of Visual Studio projects. Database projects continue to be used for line of business or large mission-critical applications.

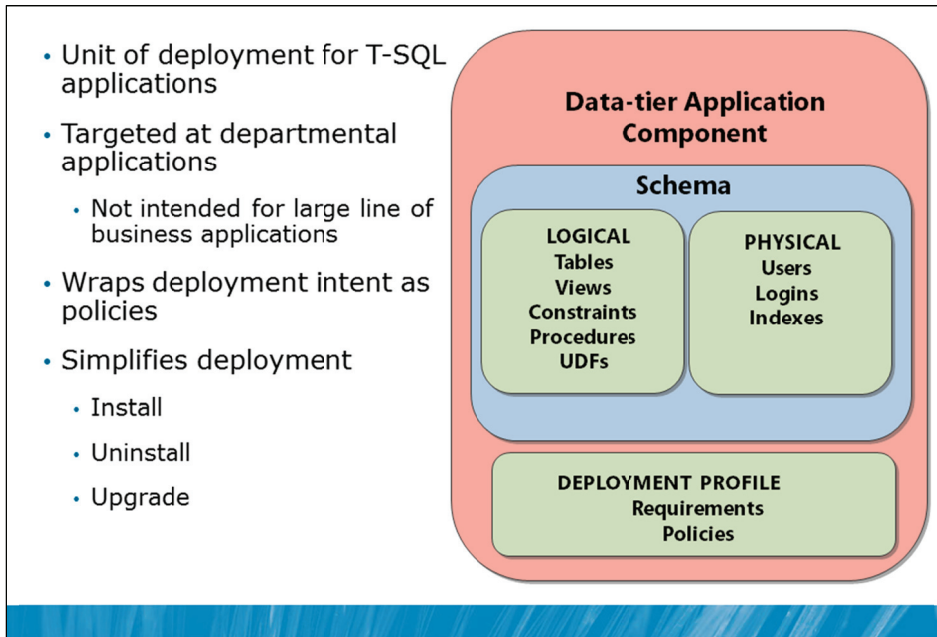
Data-tier applications are another form of database project that offer many of the advantages of database projects but are much easier to create, deploy, and upgrade. While database administrators do not necessarily need to know how to create data-tier applications, it is important that database administrators know how to deploy and upgrade data-tier applications that have been provided to them by developers.

Objectives

After completing this lesson, you will be able to:

- Describe data-tier application.
- Deploy data-tier applications.
- Upgrade data-tier applications.
- Extract data-tier applications.

Data-tier Application Overview



Key Points

Data-tier applications provide a mechanism for deploying and upgrading simple departmental applications.

Problems with Previous Approaches

Consider how most database applications are installed. The application itself is normally installed using a Windows setup program. The setup program asks for configuration information and then installs all necessary files, registers components, adds necessary registry entries, and so on. Upgrades also usually work in a very similar way.

Now consider how the databases that are required by the applications are installed and upgraded. The majority of database projects today are installed via T-SQL scripts. Some applications provide installers that execute the T-SQL commands automatically. Script-based installation has many disadvantages.

Developers have had very little support in their development environments to assist in creating the required setup scripts. The scripts often need to be edited manually. These manual processes invariably lead to errors. In addition, it is difficult for a developer to view the overall schema of the database they are working against, without building a database based on the schema. In many cases, this will mean retrieving and executing a "seed" script that creates a new database and then applying a large number of upgrade scripts to bring the database to the required level.

The database administrator often needs to run installation scripts manually and often has to perform other tasks such as creating logins and other objects. If these tasks are not performed perfectly, the installation and upgrade scripts will often fail and might leave the database in an inconsistent (or even unknown) state.

Data-tier Applications

Data-tier applications are designed to provide a similar experience for installing and upgrading database applications as occurs with Windows applications. The developer creates a data-tier application using Visual Studio, with all required objects, and defines policies that limit how the application can be installed. For example, a deployment policy could indicate that an application can only be installed on SQL Server versions 10.5 and above.

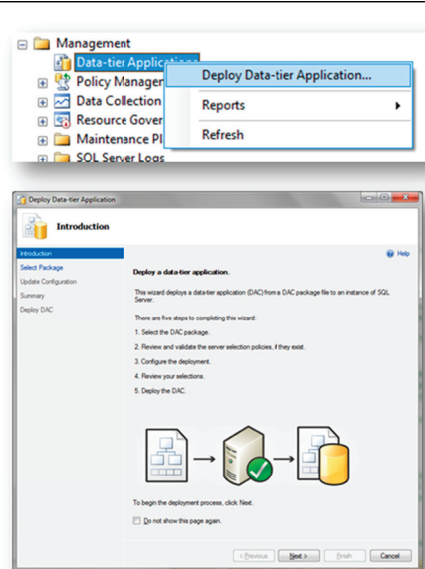
When the data-tier application project is built, the output is a .dacpac file that can be delivered to the database administrator. A single .dacpac file can be used for both installing and upgrading an application and is portable across different environments such as development, test, staging, and production.

Installation and upgrade of data-tier applications is automated. Data-tier applications are not designed for large line of business applications. The intention is to make it easy to install and upgrade large numbers of simpler applications.

Question: What is a disadvantage of deploying upgrades via T-SQL scripts?

Deploying Data-tier Applications

- Wizard-based deployment
- Requires target database name
- Checks server-selection policy
- Creates database and objects
 - Default database settings are used
 - Recovery model from the model database
- Review and test .dacpac files before production deployment



Key Points

Deploying a data-tier application has been designed to be very straightforward and automated. SSMS includes a data-tier application installation wizard. The wizard begins by asking for a database name and the location (file paths) for the database files.

Before making any changes, the wizard checks any policies that are included. In SQL Server 2012, the only policy that is available is a server selection policy that determines which versions of SQL Server that the application can be installed on. Later versions of data-tier applications might include other types of policy such as a policy that details failover or high availability requirements.

Before you deploy a .dacpac file as a data-tier application, make sure you trust the source of the .dacpac file. Otherwise, you should inspect the .dacpac files before deploying them. It is recommended, as with any type of application, that you install the data-tier applications on a test environment before installing them in a production environment.

You can view the contents of a .dacpac file by importing the file into a data-tier application project in Visual Studio. If Visual Studio is not available, a right-click "Unpack" option is available for .dacpac files on systems where SQL Server 2012 and later has been installed.

Question: Why is it important to review a dacpac file before deployment?

Upgrading Data-tier Applications

- Wizard-based upgrade process
 - Used to change schema and properties of deployed applications
 - Not T-SQL script-based
 - Application name must match currently deployed file
- Database is upgraded
 - Not the same as earlier version where data was migrated
 - Space usage must be considered
- Process can be managed via PowerShell

Key Points

Upgrades to data-tier applications are also automated. A right-click option on an existing deployed data-tier application in SSMS provides an option to upgrade the application. Upgrades are also wizard based and designed to be easy to apply.

In earlier versions of SQL Server, when you upgraded a data-tier application, a new database was created and all data in the database that you were upgrading was migrated across to the new database. In SQL Server 2012, databases are upgraded in place instead of being migrated.

The disk space requirements for the upgraded database must also be considered.

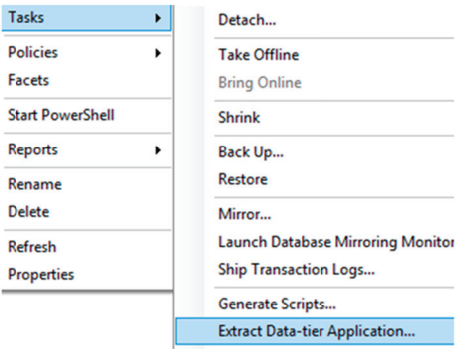
All aspects of data-tier applications that can be managed from within SSMS can also be managed by the use of Windows PowerShell® scripting.

One key advantage of data-tier application upgrades is that when errors occur, the upgrade is rolled back. Unlike script based upgrades where errors often lead to databases in inconsistent or unknown states, upgrades to data-tier applications either work or they do not work.

Question: What is the main advantage over .dacpac based upgrades compared to script-based upgrade?

Extracting Data-tier Applications

- Data-tier applications can be extracted from many existing databases
- Not all databases can be extracted
- Reverse engineering process:
 - Extract through SSMS
 - Create Visual Studio Project
 - Develop new application version
 - Compile and build
 - Upgrade application to new version



The screenshot shows the 'Tasks' menu in SQL Server Enterprise Manager. The menu items are: Tasks (selected), Policies, Facets, Start PowerShell, Reports, Rename, Delete, Refresh, and Properties. The 'Reports' sub-menu is open, showing: Detach..., Take Offline, Bring Online, Shrink, Back Up..., Restore, Mirror..., Launch Database Mirroring Monitor, Ship Transaction Logs..., Generate Scripts..., and Extract Data-tier Application... (highlighted in blue).

Key Points

Not all applications are started from scratch. There is a common need to work to further develop existing applications.

SSMS provides two options for working with existing applications:

- You can register an existing database as a data-tier application.
- You can extract a data-tier application (via a .dacpac file) from an existing database.

Note that data-tier applications do not support all SQL Server objects at this time. For example, XML schema collections and SQL CLR based objects are not supported. For this reason, not all databases can be either registered as data-tier applications or extracted to .dacpac files. When SQL Server is unable to perform a registration or extraction, the objects that are not supported are displayed within the wizard.

Question: What is the main purpose of extracting data-tier applications?

Demonstration 3A: Working with Data-tier Applications

In this demonstration, you will see:

- How to deploy a data-tier application
- How to upgrade a data-tier application

Demonstration Steps

1. If Demonstration 1A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
 - In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_19_PRJ\10775A_19_PRJ.ssmssln** and click **Open**.
 - From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
2. Open the **31 – Demonstration 3A.sql** script file.
3. Follow the instructions contained within the comments of the script file.

Lab 19: Managing Multiple Servers

- Exercise 1: Configure CMS and Execute Multi-server Queries
- Exercise 2: Deploy a Data-tier Application
- Exercise 3: Register and Extract a Data-tier Application
- Challenge Exercise 4: Upgrade a Data-tier Application (Only if time permits)

Logon information

Virtual machine	10775A-MIA-SQL1
User name	AdventureWorks\Administrator
Password	Pa\$\$w0rd

Estimated time: 45 minutes

Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_19_PRJ\10775A_19_PRJ.ssmssln**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Lab Scenario

You need to configure a solution that allows you to easily manage multiple SQL Server instances. You have noticed that on each computer that you connect to your SQL Server network, a different set of servers has been configured. You decide to configure a central management server to provide a consistent list of server groups. Your developers have begun using Data-Tier applications for some of their development. You need to deploy one of these applications to the new server, register an existing database as a Data-Tier application and, if time permits, upgrade a Data-Tier application.

Exercise 1: Configure CMS and Execute Multi-server Queries

Scenario

You want to create a list of SQL Server instances so that each database administrator has a consistent list of SQL Server instances in SQL Server Management studio. You want to manage this list of servers centrally. In this exercise, you will configure a central management server and execute multi-server queries.

The main tasks for this exercise are as follows:

1. Create a central management server.
2. Create a server group within the CMS.
3. Execute a command to find all databases on any core server in full recovery model.

► Task 1: Create a central management server

- Create a central management server on the Proseware instance.

► Task 2: Create a server group within the CMS

- Create a server group called Core Servers.
- Register the Proseware instance in the CoreServers group.
- Register the AdventureWorks instance in the CoreServers group.

► Task 3: Execute a command to find all databases on any core server in full recovery model

- Open a multi-server query window.
- Execute a multi-server query to find all databases in full recovery model.
- Hint: Look for the value FULL in the recovery_model_desc column in the sys.databases view.

Results: After this exercise, you should have created a centrally managed server group and executed a multi-server query.

Exercise 2: Deploy a Data-tier Application

Scenario

You want to deploy a data-tier application that has been supplied by your application development group. This data-tier application contains all of the database and instance objects used by an application. In this exercise, you will use SQL Server Management Studio to deploy a data-tier application.

The main task for this exercise is as follows:

1. Deploy the data-tier application using SSMS.

► Task 1: Deploy the data-tier application using SSMS

- Deploy the data-tier application D:\10775A_Labs\10775A_19_PRJ\10775A_19_PRJ\CityCode.dacpac as database name CityCode on the Proseware instance.
- Test the data-tier application by executing the script file 61 – Lab Exercise 2.sql.

Results: After this exercise, you should have deployed a data-tier application.

Exercise 3: Register and Extract a Data-tier Application

Scenario

You want to encapsulate all of the database and instance objects used by an existing application into a single package that you can email. In this exercise, you will register an existing database as a data-tier application. You will also extract a data-tier application to send to the development team.

The main tasks for this exercise are as follows:

1. Register the existing database as a data-tier application.
2. Extract a dacpac from the database to send to the development team.

► Task 1: Register the existing database as a data-tier application

- Register the Research database as a data-tier application.

► Task 2: Extract a dacpac from the database to send to the development team

- Extract a dacpac from the Research database, to the file D:\MKTG\Research.dacpac.

Results: After this exercise, you should have registered the Research database as a data-tier application and extracted a DAC package file.

Challenge Exercise 4: Upgrade a Data-tier Application (Only if time permits)

Scenario

The application development group has provided you with a new version of the database schema as a data-tier application. You want to deploy this new version of the schema. In this exercise, you will upgrade the existing data-tier application.

Note: Two versions of the updated data-tier application have been supplied. The first (CityCode_v2) will cause an error when you attempt to deploy it. The intention is to allow you to see that the system is not left in an unknown state. The second version (CityCode_v3) should deploy successfully.

The main tasks for this exercise are as follows:

1. Attempt to deploy the v2 upgrade.
2. Deploy the v3 upgrade.

► Task 1: Attempt to deploy the v2 upgrade

- Attempt to deploy the v2 upgrade from the file
D:\10775A_Labs\10775A_19_PRJ\10775A_19_PRJ\CityCode_v2.dacpac.

► Task 2: Deploy the v3 upgrade

- Deploy the v3 upgrade from the file
D:\10775A_Labs\10775A_19_PRJ\10775A_19_PRJ\CityCode_v3.dacpac.

Results: After this exercise, you should have upgraded the data-tier application.

Module Review and Takeaways

- Review Questions
- Best Practices

Review Questions

1. What must be considered, when executing multi-server queries?
2. What is the purpose of Live Migration as part of Hyper-V?

Best Practices

1. Use Central Management Server to maintain a central configuration of registered servers and groups.
2. Use multi-server queries to run batches against several servers concurrently.
3. Consider virtualization for high availability, consolidation, better hardware utilization, and test environments.
4. Consider the use of data-tier application for departmental database applications.

Module 20

Troubleshooting Common SQL Server 2012 Administrative Issues

Contents:

Lesson 1: SQL Server Troubleshooting Methodology	20-3
Lesson 2: Resolving Service-related Issues	20-7
Lesson 3: Resolving Login and Connectivity Issues	20-13
Lesson 4: Resolving Concurrency Issues	20-17
Lab 20: Troubleshooting Common Issues	20-25

Module Overview

- SQL Server Troubleshooting Methodology
- Resolving Service-related Issues
- Resolving Login and Connectivity Issues
- Resolving Concurrency Issues

An important role that database administrators that work with Microsoft® SQL Server® need to fulfill is that of a troubleshooter when issues arise, particularly those issues that are preventing users from working.

It is important to have a solid methodology for resolving issues in general, and to be familiar with the most common issues that can arise when working with SQL Server systems.

Objectives

After completing this lesson, you will be able to:

- Explain SQL Server troubleshooting methodology.
- Resolve service-related issues.
- Resolve login and connectivity issues.
- Resolve concurrency issues.

Lesson 1

SQL Server Troubleshooting Methodology

- Discussion: Characteristics of Good Troubleshooters
- Applying a Troubleshooting Methodology

Before starting to try to resolve any issue, it is important to apply a logical troubleshooting methodology and to apply it in a consistent manner. Troubleshooting is often regarded as somewhat of an art as much as it is regarded as a science. There are a number of characteristics that are common to all good troubleshooters. You should aim to develop or emulate those characteristics to ensure that you are successful when troubleshooting issues.

Objectives

After completing this lesson, you will be able to:

- Discuss characteristics of good troubleshooters.
- Apply a troubleshooting methodology.

Discussion: Characteristics of Good Troubleshooters

- What characteristics have you observed in people that you consider to be good troubleshooters?
- What characteristics do you notice in people that are poor troubleshooters?



Discussion Points

Question: What characteristics have you observed in people that you consider to be good troubleshooters?

Question: What characteristics do you notice in people that are poor troubleshooters?

Applying a Troubleshooting Methodology

- Investigate
 - Clearly define the issue as perceived by the user. How would you know if it is resolved?
 - What works? What doesn't work?
 - Did it ever work? When did it last work? What else changed?
- Analyze
 - Brainstorm all potential causes. Which potential causes are likely? How could they be tested for and eliminated?
- Implement
 - Eliminate potential causes in descending order of likelihood
- Validate
 - Ensure that the issue really is resolved
- Document for future learning

Key Points

You have seen in the discussion in the last topic that a key characteristic of good troubleshooters is that they follow a clear methodology in a logical manner. There are many different methodologies that are often applied to troubleshooting but the following list describes four phases that are common to most methodologies:

Investigation Phase

This is a critical phase. Too many people shortcut this step and start to jump directly into finding solutions. Before you can solve any problem, you need to be very clear in your understanding of the problem that you are solving.

One very important concept in this phase is that the issue needs to be defined from the point of view of the user that is affected, not from an assumed perspective of an IT person. For example, there is no point telling a user that a system is working if the user cannot use it for any reason, regardless of how your IT-based perspective might tell you that the system is working.

You need to understand what works and what doesn't work. A common mistake in this phase is to assume that when a user complains that something doesn't work, that it did ever work. Make sure that there was a time when the issue didn't exist and find out when that was. Also find out about anything, no matter how unrelated it might seem at this point, that has changed since that time.

Finally, you need to know how the user would decide that the issue is resolved. A common troubleshooting error is to find a problem, to assume that it is the cause of the issue, to resolve that problem, and to assume that the original issue is now resolved.

Analysis Phase

In the analysis phase, you need to determine all possible causes of the issue that you are trying to resolve. At this point, it is important to avoid excluding any potential causes, no matter how unlikely you consider them to be.

A brainstorming session with another person is often useful in this phase, particularly if that person is capable of constantly providing alternative viewpoints during discussions. (In many countries, this person would be described as being good at playing the Devil's advocate). The analysis phase often benefits from two types of people, one of whom has excellent technical knowledge of the product, and another that constantly requires the first person to justify their thoughts and to think both logically and laterally.

Implementation Phase

In the implementation phase, you need to eliminate each potential cause. This process of elimination usually returns the best results when the potential causes are eliminated in order from the most likely cause to the least likely cause.

The critical aspect of the implementation phase is to make sure that your reasons for eliminating potential causes are logically valid.

If you reach the end of your list of potential causes and have not yet found a solution to the issue, you need to return to the analysis phase and recheck your thinking. If you cannot find a problem in your analysis, you might even need to go back to recheck your initial assumptions in the investigation phase.

Validation Phase

Too many people, particularly those that are new to troubleshooting, assume that problems are resolved when they are not. Do not assume that because you have found and resolved a problem that it was the original problem that you have solved.

In the investigation phase, you should have determined how the user would decide if the issue is resolved. In the validation phase, you need to apply that test to see if the issue really is resolved.

Documentation

After the problem is resolved, it is really important to make sure that all in the organization learn from what happened.

Question: How do you know that a problem is resolved?

Lesson 2

Resolving Service-related Issues

- Troubleshooting Service-related Issues
- SQL Server Error Log
- Windows Event Logs
- Demonstration 2A: Troubleshooting Service-related Issues

In the remainder of this module, you will see how to approach common types of issues that can arise when working with SQL Server systems. This lesson covers the first of these types of issue that involves problems with SQL Server services.

SQL Server comprises a series of Windows services. The troubleshooting of issues with these services shares much in common with the troubleshooting of issues with other Windows services but there are some SQL Server specific considerations.

Objectives

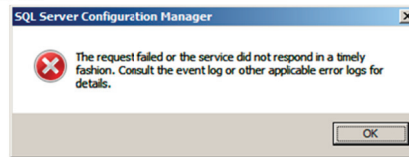
After completing this lesson, you will be able to:

- Troubleshoot service-related issues.
- Use SQL Server error log.
- Use Windows event logs.

Troubleshooting Service-related Issues

Typical issue: One or more SQL Server services will not start or cannot be accessed

- Check Windows and SQL error logs
- If SQL Server can be started but not accessed:
 - Check for network related issues
 - Try to access SQL Server via the DAC
- If SQL Server will not start:
 - Check the Windows system log
 - Check master and model databases for corruption
 - Check that the paths to tempdb files are accessible
 - Try to start the service from the command prompt



Key Points

The most common service-related problem is that one of the SQL Server services will not start or cannot be accessed. As with all Windows services, the SQL Server services are not permitted to interact directly with the system console, so this limits the options that are available to the service to advise you of problems. Also, no user may be logged on at the console to receive notifications even if they were permitted, as services run in the background and are mostly started automatically, even before any user logs in.

Windows and SQL Server Logs

The most common cause of a service not being able to start is that the service account logon is failing for some reason. This could be caused by an incorrect or expired password, or the account could be locked out. Logon failures for services typically appear in the Windows system event log. More complex issues such as missing files cause entries to be written to the SQL Server logs.

Check both the Windows and SQL Server logs as the first step in resolving service startup issues.

Other Service-related Issues

If SQL Server does start but cannot be accessed, the problem often relates to network issues that are discussed later in this module. In rare cases, the SQL Server scheduler might hang and stop accepting new connections. In this situation, and in similar situations, you should also try to connect to SQL Server using the Dedicated Administration Connection (DAC). You might need to use the DAC to access the instance and analyze the problem. On occasions, you might need to kill individual sessions that are causing the problem.

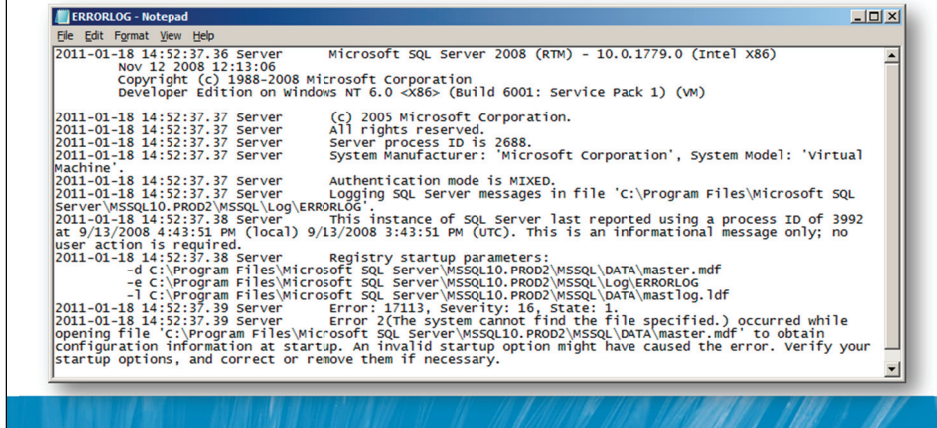
If SQL Server will not start but the issue isn't a logon issue, check the following:

- Check whether the SQL Server log files indicate that the master or model database is corrupt. If either of these databases is corrupt, follow the procedures to recover the databases that were described in Module 7.
- Check whether or not the file paths to the tempdb database files are accessible. The tempdb database is recreated each time the server instance starts but the path to the database files (as configured in the master database) must exist and be accessible.
- Try to start the instance using the command prompt. If starting SQL Server from a command prompt does work, check the configuration of the service and make sure that the permission requirements are met.

Question: What should be reviewed first when a service-related issue happens with SQL Server?

SQL Server Error Log

- Log File Viewer cannot be used when the instance is not started
 - Use editor to open the current log file and review it
 - Also review previous logs as the problem might have occurred before



Key Points

The SQL Server log will often provide detailed information about startup issues. The log can be viewed using the log viewer as part of SSMS.

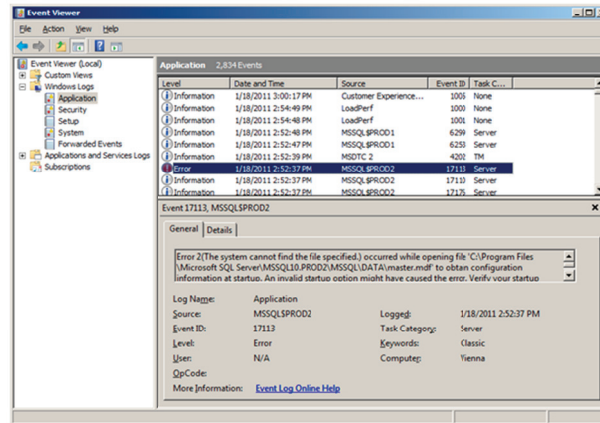
However, if SQL Server cannot start, you cannot use the log viewer in SSMS to see the contents of the SQL Server logs. You can, however, open these logs in an editor such as NotePad, as shown in the example on the slide.

The SQL Server Error log was discussed in more detail in Module 15. While the latest error log usually provides the most useful information, SQL Server keeps a set of archive log files and you should review those files as well, as problems may have been occurring for some time.

Question: What is the problem with the instance startup in the example shown in the slide?

Windows Event Logs

- System Log to review Windows related information
- Application Log to find application related messages



Key Points

Windows services are not permitted to interact directly with the console. The main place they can use to tell you about issues is the Windows system event log.

In addition, as with other applications, SQL Server writes quite an amount of information to the Windows application log over time.

Both these logs should be checked periodically as not all errors that are logged in them produce symptoms that are apparent to users.

Question: Which Windows log would logon failures for the SQL Server service appear in?

Demonstration 2A: Troubleshooting Service-related Issues

In this demonstration you will see how to troubleshoot a service-related issue

Demonstration Steps

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_20_PRJ\10775A_20_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **21 – Demonstration 2A.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Lesson 3

Resolving Login and Connectivity Issues

- Troubleshooting Connectivity Issues
- Troubleshooting Login Failures

The final type of issues that database administrators who work with SQL Server are commonly called upon to resolve are connectivity and login issues. Users may have issues making connections to SQL Server, in relation to network names, protocols, and ports and the users may have issues with passwords that need to be resolved. In general, you should try to create logins that are based on Windows group membership. This removes the need for you as a database administrator to deal with most password and Windows account related issues.

Objectives

After completing this lesson, you will be able to:

- Troubleshoot connectivity issues.
- Troubleshoot login failures.

Troubleshooting Connectivity Issues

Typical issue: Cannot connect to SQL Server

- Try to access using Shared Memory on the server
 - If no access via Shared Memory, troubleshoot the login and the service
- Test the network connectivity
 - Can the server name be resolved?
 - Can the network and the server be reached?
 - Is the Browser Service Running for Named Instances that are not using fixed ports?
 - Is the client configured to use the right protocol and settings?
 - Is a firewall blocking connectivity?

Key Points

The most common issue of this type that you might need to resolve is that users cannot connect to SQL Server at all.

Eliminate the Network

The first step in trying to resolve these issues is to determine whether or not the issue is related to network connectivity. You can often eliminate the network from consideration by attempting to connect via a shared memory connection while local to the server. If you cannot access SQL Server via a shared memory connection, you will need to troubleshoot the login or the service. Login troubleshooting is discussed in the next topic. Service troubleshooting was discussed in Lesson 2.

Network Related Issues

If the login using shared memory succeeds, the problem is almost always network related. In rare cases, you could see a problem with incompatible network libraries on the client and the server but most problems are much less subtle.

Issue	Actions
Can the server name be resolved?	The client system needs to be able to find the network address of the server. In TCP/IP networks, this will usually involve a DNS lookup. Make sure you can ping the server that SQL Server is installed on by name and that you see an IP address returned by the ping command.

(continued)

Issue	Actions
Can the network and the server be reached?	While a server name might be successfully resolved to a network address, no connectivity may be possible between the client and server systems. On TCP/IP networks, make sure that you can successfully ping the server that SQL Server is installed on, assuming that it does not have the ability to echo ping commands disabled. In many cases, problems in this area will relate to incorrect subnet or default gateway/router addresses.
Is the Browser Service Running for Named Instances that are not using fixed ports?	If you are using named instances of SQL Server, the client systems need to be able to resolve the names of the instances to a port number. By default, this is achieved by connecting to the SQL Browser service on UDP port 1434. If the SQL Browser service is disabled, the named instances need to be configured to use fixed ports and the client systems need to be configured to access those ports, potentially by the creation of aliases. Make sure that the aliases are created for the correct network library such as 32 bit vs. 64 bit.
Is the client configured to use the right protocol and settings?	Is the protocol that the client is attempting to use to connect to the server, actually enabled on the server? Are the protocol settings appropriate?
Is a firewall blocking connectivity?	Check to make sure that there is no firewall that is blocking the ports that you are trying to connect over. If a firewall is blocking your access, an exception or rule will likely need to be configured in the firewall.

Question: Why should a local connection be tested first?

Troubleshooting Login Failures

Typical issue: Cannot logon to SQL Server

- Windows Logins
 - Is the Domain Controller available?
 - Can SQL Server communicate with the Domain Controller?
- SQL Server Logins
 - Is SQL Server configured for Mixed Mode?
 - Is the password correct?
 - Is the login locked or is there a requirement to change password?
- General Considerations
 - Is the login enabled and does it have CONNECT permission?
 - Is the default/requested database available and access permitted?

Key Points

The typical issue with this type of problem is that a user can establish a network connection to SQL Server but cannot log in. The troubleshooting actions that you need to perform depend upon the type of login that is being used:

- For Windows Logins, make sure that SQL Server can connect to a Domain Controller to process the logins. To review potential issues, inspect the Windows Logs.
- For SQL Server Logins, make sure that SQL Server is configured for SQL Server Authentication. SQL Server Logins can be created and can also be enabled, even when SQL Server is configured for Windows Authentication only. The most common error that is returned in this situation says that a trusted connection is not available.

Also when working with SQL Server logins, make sure that the supplied password is correct and that the login has not been locked out by account policy. Another common problem for SQL Server logins is that the login might expire but the application that is trying to connect to SQL Server does not understand password expiry. This situation is common with applications that were designed for previous versions of SQL Server that did not implement account policy for SQL Server logins.

For both login types, make sure that the login has permission to connect to SQL Server and that permission has been granted to the login to access the database that the login is attempting to connect to. This check should include the default database for the login.

If a login problem is happening with a large number of different users, check to make sure that there isn't a logon trigger that is failing. When a logon trigger prevents users from connecting, the error message that is returned to interactive users indicates that a logon trigger prevented the connection.

Question: Can a SQL Server Login be created and enabled when SQL Server is configured for Windows Authentication only?

Lesson 4

Resolving Concurrency Issues

- Core Concurrency Concepts
- Troubleshooting Blocking
- Demonstration 4A: Troubleshooting Blocking
- Troubleshooting Deadlocks
- Demonstration 4B: Troubleshooting Deadlocks

SQL Server attempts to provide each user with the illusion that they are the only user on the system. It is not possible for SQL Server to provide this illusion in all circumstances and the actions of one user will impact on the actions of other users.

It is important to understand the core concepts surrounding concurrency and to then see how to troubleshoot the more common blocking and deadlock issues.

Objectives

After completing this lesson, you will be able to:

- Describe core concurrency concepts.
- Troubleshoot blocking issues.
- Troubleshoot deadlock issues.

Core Concurrency Concepts

- SQL Server has to ensure ACID properties of transactions
- Locking is the mechanism used to synchronize access by multiple users to the same data at the same time
- Isolation levels define the type and length of locks used
- Two main types of lock:
 - **Shared locks** – Allow others to read but not write
 - **Exclusive locks** – Stop others from reading or writing
- Locks prevent update conflicts
 - Locking ensures that transactions are serialized
 - Locking is automatic
 - Locks enable safe concurrent access to data

Key Points

Consider your expectations when you transfer money from one bank account to another bank account. You would expect the same amount of money to come out of one account as is added to the other account. You would also expect that the entire operation either occurs or does not occur, even if some form of failure happens during your transfer. You would not expect the funds to be removed from one account and not to be added to the other account.

Further consider what you would expect to see if you viewed the state of the two accounts while the transfer was occurring. You would expect to always see the accounts in a consistent state, based on either the balances of the accounts prior to the transfer or based on the balances of the accounts after the transfer was complete. You would not expect to ever be able to see the accounts in an intermediate state.

One final issue to consider is what you expect to see when you viewed the state of the accounts on the day following the successful transaction. You would expect to see that the transaction was still shown as completed.

ACID Properties

The expectations that you have closely relate to a set of properties that the IT industry refers to as ACID properties that are expected of transactions:

- **Atomicity** - A transaction has a defined start and end. It runs completely or is rolled back completely.
- **Consistency** - The database is not left in an intermediate state after the transaction. All logical constraints are enforced.
- **Isolation** - Transactions are prevented from interfering with each other.
- **Durability** - After a transaction is committed, the details of the transaction are persisted in the system and will survive restarts.

To achieve the isolation requirements, SQL Server uses a locking scheme. Locking is a mechanism used by the database engine to synchronize access by multiple users to the same piece of data at the same time. The aim is to maximize concurrency while maintaining the ACID properties.



Note Concurrency is discussed in detail in course 10776A: Developing Microsoft SQL Server 2012 Databases.

Locking Behavior

Before a transaction acquires a dependency on the current state of a data element, such as by reading or modifying the data, it must protect itself from the effects of another transaction that attempts to modify the same data. The transaction does this by requesting a lock on the data element. Locks have different modes, such as shared or exclusive. The locking mode defines the level of dependency the transaction has on the data.

No transaction can be granted a lock that would conflict with the mode of a lock that has already been granted to another transaction on that same data. If a transaction requests a lock mode that conflicts with a lock that has already been granted on the same data, the database engine will block the requesting transaction until the first lock is released.

For UPDATE operations, SQL Server always holds locks to the end of the transaction. For SELECT operations, it holds the lock protecting the row for a period that depends upon the transaction isolation level setting. All locks that are still held when a transaction completes are released, whether the transaction commits or rolls back.

Locking is crucial for transaction processing and is normal behavior for the system. Problems only occur when locks are held for too long and other transactions are blocked for too long because of the locks that are being held.

Question: Why would SELECT statements acquire locks in most cases?

Troubleshooting Blocking

Typical issue: Applications don't seem to be doing anything but also aren't returning errors

- Processes are blocked when they wait for locked resources
 - Normal behavior, essential for the system
 - Resource is shown as waiting to acquire a lock
 - Only a problem when the wait duration is lengthy
- Monitor long lasting blocking scenarios
 - SQL Server Activity Monitor
 - Dynamic Management Views
 - SQL Server Data Collector

Key Points

The most common issue that occurs with blocking is that applications appear to be hung. The symptoms of this are that the application is not busy and has work that needs to be done, yet no work is occurring. This situation generally indicates that the application is blocked while waiting for a resource of some type.

You saw in the last topic that locking is the action of taking and holding locks that is used to implement concurrency control and that blocking is what happens to one process while it needs to wait for a resource that another process has locked. Blocking is a normal occurrence for systems and is not an issue. Excessive blocking can be an issue.

Blocking can be monitored in real-time using SQL Server Activity Monitor and through Dynamic Management Views. You will see how to perform this monitoring in Demonstration 3A.

The SQL Server data collector that was discussed in Module 18 can be used to collect data about blocking scenarios including information about all the processes that are involved with the blocking. The reports that are provided with the data collector via the management data warehouse can be useful when investigating blocking issues that have happened earlier.

Question: What do you suspect that "excessive" blocking might refer to?

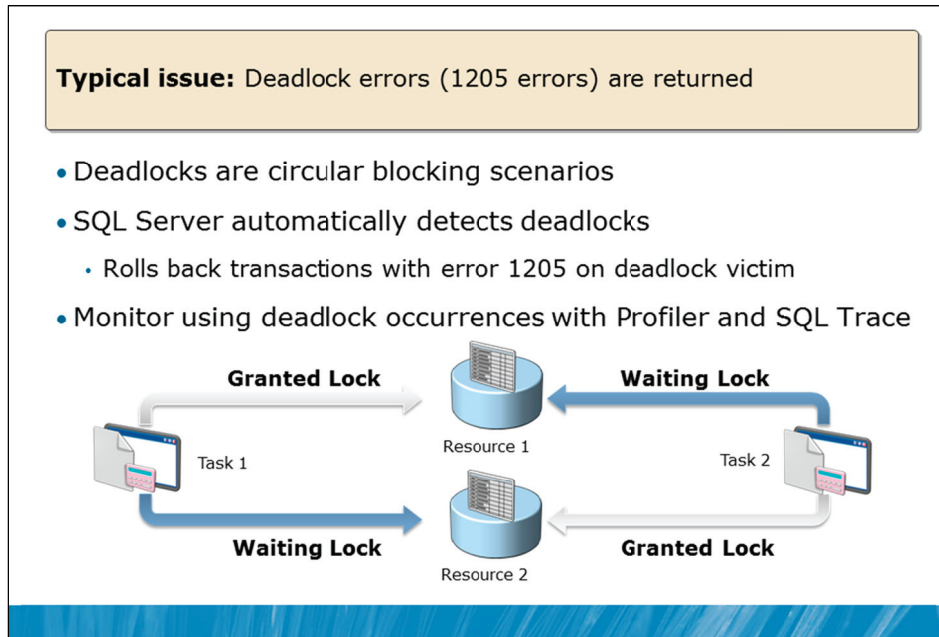
Demonstration 4A: Troubleshooting Blocking

In this demonstration, you will see how to troubleshoot a blocking issue

Demonstration Steps

1. If Demonstration 2A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_20_PRJ\10775A_20_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **41 – Demonstration 4A.sql** script file from within **Solution Explorer**.
5. Follow the instructions contained within the comments of the script file.

Troubleshooting Deadlocks



Key Points

Deadlock errors are a special type of blocking error where SQL Server needs to intervene as there would otherwise never be any locks released. The most common form of deadlock occurs when two transactions have locks on separate objects and each transaction requests a lock on the other transaction's object. For example:

- Task 1 holds a shared lock on row 1.
- Task 2 holds a shared lock on row 2.
- Task 1 requests an exclusive lock on row 2, but it cannot be granted until Task 2 releases the shared lock.
- Task 2 requests an exclusive lock on row 1, but it cannot be granted until Task 1 releases the shared lock.
- Each task must wait for the other to release the lock, which will never happen.

A deadlock can occur when several long-running transactions execute concurrently in the same database. A deadlock also can occur as a result of the order in which the optimizer processes a complex query, such as a join.

How SQL Server Ends a Deadlock

SQL Server ends a deadlock by automatically terminating one of the transactions. SQL Server does the following:

- Chooses a deadlock victim. SQL Server gives priority to the process that has the highest deadlock priority. If both processes have the same deadlock priority, SQL Server rolls back the transaction that is the least costly to rollback.
- Rolls back the transaction of the deadlock victim.
- Notifies the deadlock victim's application (with message number 1205).
- Allows the other transaction to continue.



Important In a multi-user environment, each client should check for message number 1205, which indicates that the transaction was rolled back. If message 1205 is found, the application should reconnect and try the transaction again.

Working with Deadlocks

Deadlocks are normally not logged. The only indication that a deadlock has occurred is that an error message is returned to the client that has been selected as the victim.

To monitor deadlocks use SQL Server Profiler and/or SQL Trace. There are several deadlock events available. The event with most information and that is easiest to read is the Deadlock Graph event. You will see an example of a deadlock graph in Demonstration 3B.

Note that there are other tools that can help with deadlocks that are out of scope for this course:

- Trace flags can be set to write deadlock information to the SQL Server Error Log.
- Event Notifications can be generated.
- Extended events can be tracked.

Question: Have you experienced deadlocking problems in your current environment? If so, how did you determine that deadlocks were a problem, and how was it resolved?

Demonstration 4B: Troubleshooting Deadlocks

In this demonstration, you will see:

- How to trace a deadlock using SQL Server Profiler
- How to extract deadlock events from SQL Server Profiler
- How to open deadlock files using SSMS

Demonstration Steps

1. If Demonstration 2A was not performed:
 - Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **SQL Server Management Studio**. In the Connect to Server window, type **Proseware** and click **Connect**. From the **File** menu, click **Open**, click **Project/Solution**, navigate to **D:\10775A_Labs\10775A_20_PRJ\10775A_20_PRJ.ssmssln** and click **Open**.
3. From the **View** menu, click **Solution Explorer**. Open and execute the **00 – Setup.sql** script file from within Solution Explorer.
4. Open the **42 – Demonstration 4B.sql** script file.
5. Follow the instructions contained within the comments of the script file.

Lab 20: Troubleshooting Common Issues

- Exercises 1 – 5: Troubleshoot and Resolve SQL Server Administrative Issues

Logon information

Virtual machine	10775A-MIA-SQL1
User name	AdventureWorks\Administrator
Password	Pa\$\$w0rd

Estimated time: 45 minutes

Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_20_PRJ\10775A_20_PRJ.ssmssl**.
7. From the **View** menu, click **Solution Explorer**. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Lab Scenario

You need to be able to resolve common issues with SQL Server processes and services at the time they are occurring. There are five exercises that create issues. You should attempt to troubleshoot and resolve as many of these issues as possible.

Supporting Documentation

Exercise 1

Users of the Promote application are complaining that the application can no longer connect to the server. The application connects using the SQL Login PromoteApp.

Exercise 2

A junior DBA created a backup of the production AdminDB database to send to the development team for testing. Now users of that database are complaining they can no longer connect to it.

Exercise 3

Users are complaining that the job Get File List in SQL Server Agent is failing to execute.

Exercise 4

The performance of the AdminDB database has been steadily reducing since the database was deployed. Queries that once took seconds are now taking minutes.

Exercise 5

A strange situation is occurring with the CityDetails database. Most databases work slower as more users are added. However, the CityDetails database performs the worst when only a single user connects to it. The connection time for just this database can be very long and sometimes times out. Other databases are OK.

Exercises 1 - 5: Troubleshoot and Resolve SQL Server Administrative Issues

Scenario

A number of issues have been reported by the users of the Proseware server. You need to troubleshoot and resolve the issues.

The main tasks for each exercise are as follows:

1. Read the supporting documentation for the exercise.
2. Troubleshoot and resolve the issue.

► Task 1: Read the supporting documentation for the exercise

- Read the supporting documentation for the exercise.

► Task 2: Troubleshoot and resolve the issue

- Troubleshoot the issue.
- Resolve the issue.
- If you have difficulty, check the solution in the LAK.

Results: After these exercise, you should have investigated and resolved a number of issues.

Module Review and Takeaways

- Review Questions
- Best Practices

Review Questions

1. What can be used to monitor blocking issues?
2. What happens when a deadlock occurs?

Best Practices

1. Monitor your system and store historical data using Data Collector for example for easier troubleshooting.
2. Clearly identify underlying problems rather than fighting symptoms.
3. Apply deadlock monitoring using SQL Trace.

Course Review and Evaluation



Your evaluation of this course will help Microsoft understand the quality of your learning experience.

Please work with your training provider to access the course evaluation form.

Microsoft will keep your answers to this survey private and confidential and will use your responses to improve your future learning experience. Your open and honest feedback is valuable and appreciated.

Appendix A

Core Concepts in SQL Server High Availability and Replication

Contents:

Lesson 1: Core Concepts in High Availability

A-3

Lesson 2: Core Concepts in Replication

A-11

Appendix Overview

- Core Concepts in High Availability
- Core Concepts in Replication

As you become a more experienced database professional, you are likely to need to implement more advanced technologies that are available within the Microsoft SQL Server platform. Two core areas are high availability and replication.

In this appendix, you will be introduced to the core concepts surrounding each of these technologies. In particular, you will learn when to use each of these technologies.

Objectives

After reading this lesson, you will be able to:

- Explain the different high availability options available for Microsoft SQL Server and know when to use each.
- Explain the different replication options available for Microsoft SQL Server and know when to use each.

Lesson 1

Core Concepts in High Availability

- Discussion: Previous Experience with High Availability
- Database Mirroring
- Failover Clustering
- AlwaysOn Availability Groups
- Log Shipping
- Choosing Between High Availability Options

SQL Server provides a wide variety of options for ensuring the highest availability of servers and databases. It is important that you understand the core concepts involved and know how to choose between the available options.

Objectives

After completing this lesson, you will be able to:

- Describe database mirroring.
- Describe failover clustering.
- Explain the role of AlwaysOn technologies.
- Describe log shipping.
- Choose appropriate use cases for each high availability option.

Discussion: Previous Experience with High Availability

- What do you understand by the term "High Availability"?
- What techniques have you used to try to achieve high availability?
- How successful have these attempts been?



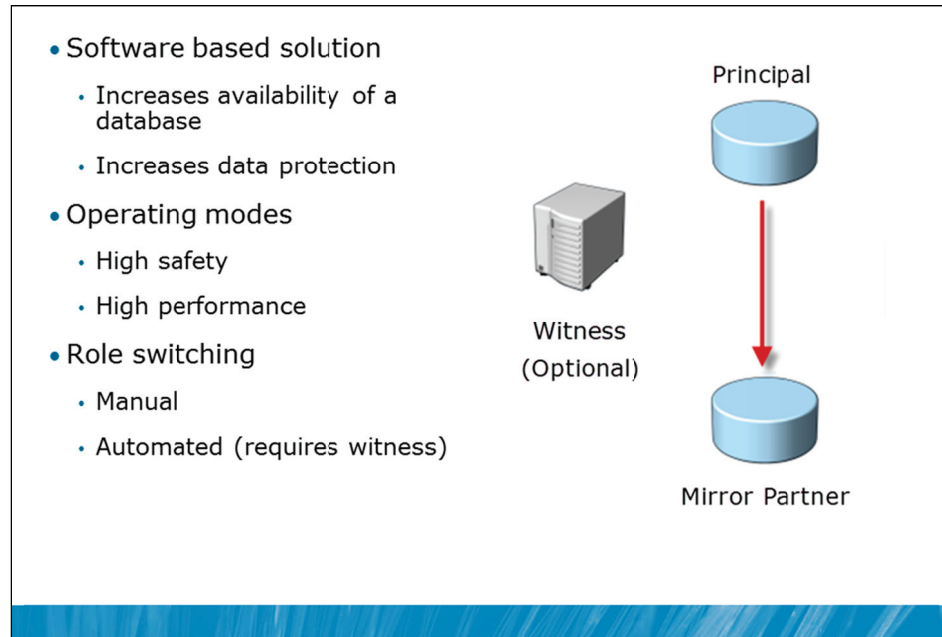
Key Points

In this discussion, members of your class will share their experiences with high availability strategies. In particular, consider (but do not limit yourself to) the following questions:

- What do you understand by the term "High Availability"?
- What techniques have you used to try to achieve high availability?
- How successful have these attempts been?

High Availability is more than technology. It also involves defined processes, documentation, knowledge of the people and much more. If the staff involved do not know what to do when issues occur, the best technology will not help. A good backup strategy is also part of a high availability solution.

Database Mirroring



Key Points

Database mirroring is a software based solution for achieving high availability of databases that was introduced in Microsoft SQL Server 2005. The transaction log from a database on one server is continuously applied to a copy of the database residing on another server. Databases to be mirrored must be in full recovery model.

Because an entirely separate server is used for the mirror partner, the chance of having a single point of failure is greatly reduced. While not generally desirable, the servers used in database mirroring could also be implemented using quite different hardware architectures. Database mirroring is, however, easier to maintain when the servers involved are similar, and when the disk layout of each server is identical.

Two modes of operation are available. In high safety mode, the principal server sends update information to both its own transaction log and to the transaction log of the mirror partner, and then waits for the writes to occur on the partner's transaction log, as well as waiting for its own writes to occur. In high performance mode, the principal server sends the update information to both logs but only waits for its own log to be written. High performance mode sacrifices data consistency for performance.

When a failure occurs, the mirror partner can take over the role of the principal. This can occur in two ways. The database administrator can initiate a manual failover or an automated failover can occur. Automated failovers require the following:

- High safety mode.
- The mirror partner must have been synchronized with the primary at the time of the failover. (This might not be the case if the mirror partner was catching up after a failure or network issue.)
- A witness server has been configured.

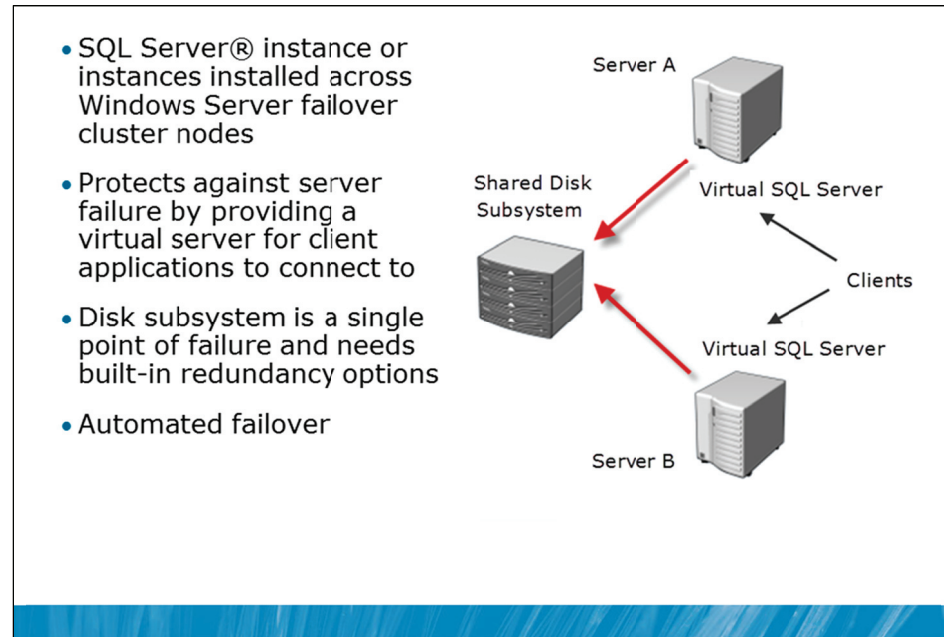
A witness server can be any edition of SQL Server including the Express edition, and should reside on another physical server, separate to both the principal server and the mirror partner. The witness server is used to provide a quorum with the mirror partner server, where they can agree that the principal server has failed and that a failover should occur. During a failure, if an automated failover cannot occur, it is possible to "force service" by switching the mirror partner into the principal role, even though a loss of data could occur.

Note that system databases such as master, msdb, tempdb, and the model database cannot be mirrored. SQL Server 2008 improved the performance of database mirroring through the introduction of log stream compression. While only a single mirror could be provided for each database, many more databases could be concurrently mirrored.

Mirroring can improve the availability of databases, including times when routine maintenance such as applying service packs is needed. The data held on the mirror partner is not accessible unless a database snapshot is created on the mirror to provide read-only access to the data.

The biggest limitation of database mirroring is that databases often have external dependencies such as logins in the master database. Mechanisms or processes need to be put in place to ensure that these dependent objects are present on both the servers.

Failover Clustering



Key Points

SQL Server failover clustering is built on Windows Server failover clustering. It requires each server to be a Windows Failover cluster node. Failover clustering is designed to protect against the failure of a server. Clients connect to a virtual SQL Server rather than to either physical server.

Unlike database mirroring, when a failover occurs the entire instance fails over. This removes the complexity of maintaining external database dependencies because the system databases (such as the master database) are also failed over. Databases do not need to be in full recovery model.

Servers involved in failover clustering should use identical hardware. A Cluster Validation Tool (CVT) is provided to check the configuration of a failover cluster before the cluster is created. Passing the tests provided by the CVT is a prerequisite for obtaining Microsoft support on the cluster configuration.

Failover clustering uses a shared disk subsystem. It is important to note that only a single copy of the data is being held. This means that the disk subsystem needs to incorporate its own redundancy mechanisms.

A SQL Server cluster exposes a virtual server to client systems. Regardless of which server is currently operating, the clients continue to connect to the same virtual server name or address. Each server monitors the other servers in the cluster, so that failover action can be initiated when necessary. This communication is normally carried out across a private network connection (often called a heartbeat network). The private network is kept separate to the public network connections from each server.

At any point in time, each SQL Server instance is only running on a single server. When a server failover occurs, the SQL Server service for each instance on the secondary server must be restarted and databases recovered. This process can take some time, depending upon how long recovery takes for each database.

AlwaysOn Availability Groups

- Enterprise-level high availability alternative
- Provide options to improve database availability and improve resource usage
 - Supports one primary and up to four secondary replicas
 - Replicas can support alternative availability modes (asynchronous vs. synchronous)
 - Supports further failover options such as planned manual failover
 - Availability group attempts to failover rather than single database
- Secondary replicas can provide additional support
 - Read-only access
 - Backup operations

Key Points

SQL Server 2012 introduced the AlwaysOn technologies. One of these technologies, AlwaysOn Availability Groups, provides an enterprise-level alternative to database mirroring. AlwaysOn Availability Groups must be installed on a Windows cluster node and improves on database mirroring in the following ways:

Aspect	Database Mirroring	AlwaysOn Availability Groups
Applies to	Database	Group of databases
Replicas	One	Four
Replica Usage	Possible to make it available read-only via database snapshots but in practice not very useful	Each replica can be made read-only and can be used for backups
Replica Operation Mode	Synchronous or Asynchronous	Replicas can be synchronous or asynchronous. Up to two synchronous replicas are supported. Combinations of synchronous and asynchronous replicas are supported.
Client Connection	Options for automated redirection available	Easier options for automated redirection available

Log Shipping

- Mature and reliable technology
- Widely deployed
- No automation of failover supplied
- Core process
 - Backup transaction log
 - Copy logs to another server
 - Restore transaction log on second server

Key Points

Log shipping is a mature and reliable technology that was introduced in SQL Server 2000. Many third party organizations and users have also implemented their own variations on log shipping. In essence, log shipping is a simple repetitive process involving:

- Backup the transaction log on the primary server.
- Copy the transaction log to the secondary server.
- Restore the transaction log on the secondary server.

Each time the restore is carried out on the secondary server, the database restores are performed without the recovery option, so that additional transaction logs can be restored later.

When a failure occurs, placing the secondary server into production is a manual task. No automation is provided for doing this within the product.

Log shipping allows for multiple secondary servers.

Choosing Between High Availability Options

Aspect	Database Mirroring	Failover Clustering	Availability Groups	Log Shipping
Failover unit	Database	Server	Group of databases	Database
Automated failover	Yes, with witness	Yes	Yes	No
Data Replicas	1	0	4	Unlimited
Complexity	Medium	Medium	Medium	Low
Replica Availability	Read-only via snapshot	N/A	Read-only, backup	Read-only between restores

Key Points

The table above provides a comparison of a number of the aspects of each high availability option, as discussed in the previous four topics. Note that a high availability solution might comprise a combination of these technologies.

With Failover Clustering, even though a single copy of data is held, the shared drive subsystem should always be configured with redundancy options.

With Log Shipping, read-only access to the replica data is only available when Standby Mode has been chosen.

Lesson 2

Core Concepts in Replication

- SQL Server Replication Architecture
- Snapshot Replication
- Transactional Replication
- Peer-to-Peer Transactional Replication
- Merge Replication
- Choosing Between Replication Options

SQL Server provides a wide variety of options for replicating data between servers. It is important that you understand the core concepts involved and know how to choose between the available options.

Objectives

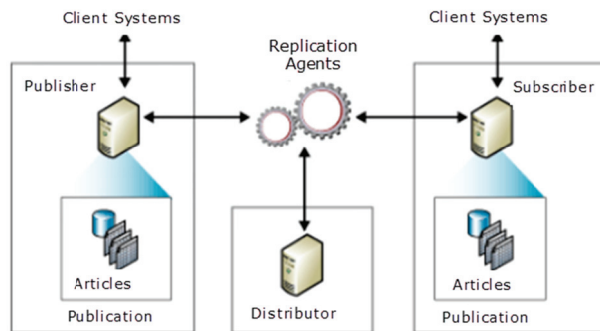
After completing this lesson, you will be able to:

- Explain the publisher/subscriber metaphor that replication is based upon.
- Describe snapshot replication.
- Describe transactional replication.
- Describe peer-to-peer transactional replication.
- Describe merge replication.
- Choose between the available replication options.

SQL Server Replication Architecture

- Can play a role in high availability but mostly targeted at data distribution
- Based on a magazine publisher metaphor

- Publisher
- Publication
- Article
- Distributor
- Subscriber



Key Points

While SQL Server replication can play a role in high availability (particularly in relation to peer-to-peer transactional replication), the major role that it plays is related to the distribution of data.

It is important to understand the terminology used by replication.

- A Publisher is an instance that is publishing data for subscribers to consume. A database needs to be configured to allow publication of data contained within it. (In the metaphor, this is the publishing house).
- An Article is the object being published. An article can comprise a table or a view. Either the entire table or view can be published, or an article can be based on vertical filtering (columns), horizontal filtering (rows) or both. (In the metaphor, this directly relates to the concept of an article within a magazine).
- A Publication is a group of articles that are being published. It is the unit of subscription. (In the metaphor, this is the magazine).
- A Distributor is used to move publications from publishers to subscribers. A distribution database is used to hold publications that need to be supplied to a subscriber. A distributor that is on the same local network as the publisher is called a Local Distributor. Other distributors (perhaps even in a different continent) are considered to be Remote Distributors.
- A Subscriber is an instance that receives the publications.
- An Agent is a service that performs operations needed by any of the replication roles.

A remote distributor might be used to minimize the network traffic between the distributor and a number of subscribers by placing the distributor close to the subscribers. In a push subscription, the distributor sends the data to the subscriber. In a pull subscription, the subscriber retrieves the data from the distributor.

Snapshot Replication

- Simplest form of replication
- All published data copied each time
- Use when:
 - Data does not change often
 - Out of date data at the subscriber is acceptable
 - Volume of data is relatively small
 - Large volume of changes occur over short time periods

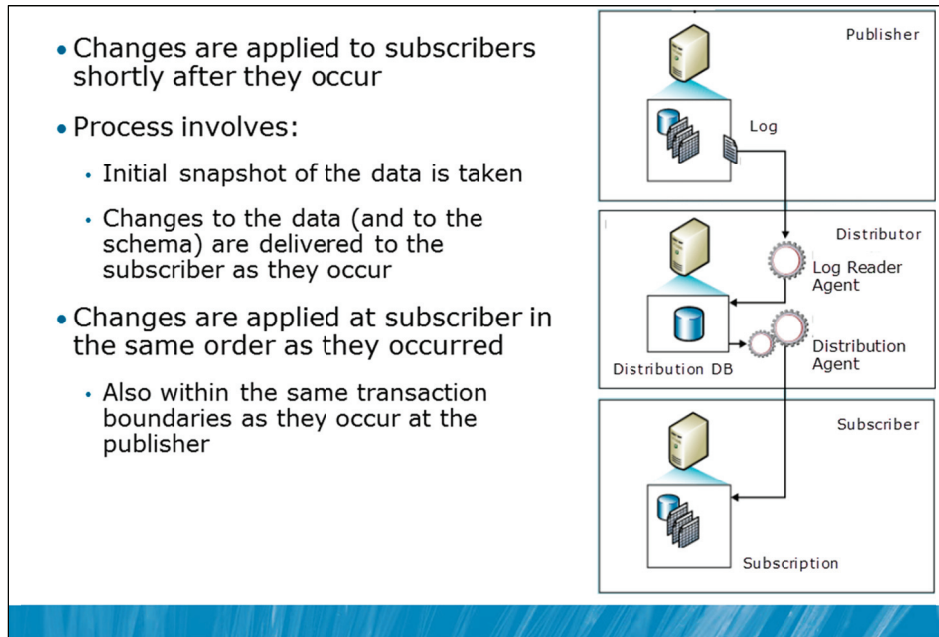
Key Points

Snapshot is the simplest form of replication. Periodically, the snapshot agent creates a copy of the published data. The entire copy of the data is then used to replace the data at the subscriber.

Snapshot replication is useful in several situations:

- If permitting out-of-date data at the subscriber is acceptable. Not all systems would allow this.
- If the data does not change often, then having out-of-date data at the subscriber is more likely to be acceptable.
- If the volume of data is very small, copying all the data is easier than trying to work out which data to copy.
- If the number of changes that occurs to the data is very large, it might be less costly to recopy the entire data rather than to copy the changes as the size of the changes might be larger than the size of the source data.
- The publisher is not a SQL Server database. (Publishing from an Oracle database is supported.)

Transactional Replication



Key Points

Transactional replication is used to send changes to subscribers shortly after the changes have occurred on the publishers. Transactional replication is initiated by first creating a snapshot of the data and applying the snapshot to the subscriber. There are several options available for creating this copy of the data, including taking a database backup and restoring it. Changes are applied in the same order and within the same transactional boundaries as they occurred at the publisher.

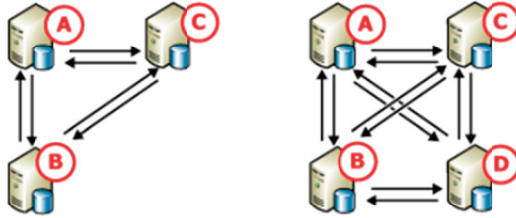
Transactional replication should typically be regarded as one-way replication. In general, changes made at the subscriber are not sent back to the publisher; however there are configuration options to allow for updates at the subscriber.

Transactional replication is useful in the following situations:

- Incremental changes to data need to be applied to the subscriber shortly after they occur.
- The application requires the delay between a change occurring and it being applied to be short. Note that this delay is configurable and might range from almost instantaneous to several hours.
- The subscriber needs to see each change to the data that occurs. (This allows the subscriber to take actions on each change such as firing triggers).
- A high volume of data modification activity occurs at the publisher.
- The publisher is not a SQL Server database. (Publishing from an Oracle database is supported.)

Peer-To-Peer Transactional Replication

- Configurable option within transactional replication
- Typically used with autonomous servers
 - Appropriately partitioned data is critical
 - Conflicts can be detected but need to be avoided



Key Points

"Allow peer-to-peer subscribers" is an option that can be configured on transactional replication publications. This can help to scale out a replication topology to allow read operations to occur on any node.

Reads and writes can occur on all nodes but to prevent conflicts, an appropriate data partitioning scheme needs to be used. Write operations on any individual rows are still single node. While conflicts can be detected, they cause the replication to fail or the changes to be held. For this reason, the most important aspect of peer-to-peer transactional replication is designing the partitioning of the data held by each server to avoid any chance of conflicts from occurring.

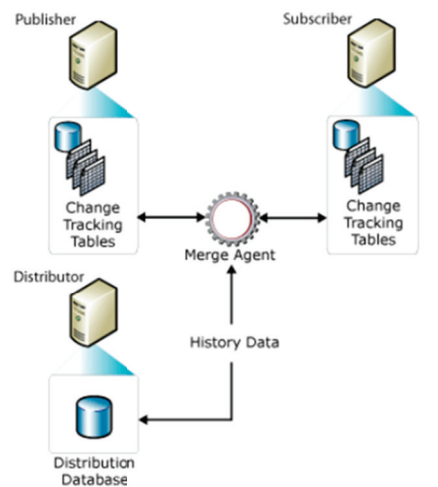
As a typical example, a company might have servers in London, Singapore, and Mumbai. Each row in a table would indicate which location the data has originated at. All rows are replicated to each server but a single row is never modified at more than one location.

Peer-to-peer transactional replication is useful in the following scenarios:

- Where a natural partitioning of the data exists.
- Where the partitioning of the data matches the topology of the servers.
- Where conflicts would rarely occur such as situations where each data row within a table is owned by a single server.

Merge Replication

- Commonly used for simple client-server applications
- Supports non-SQL Server nodes
- Not transactionally consistent
- Performance suffers with
 - Large amounts of data
 - Large numbers of subscribers



Key Points

Merge replication is commonly used in simple client-server applications. Merge replication works by applying a set of triggers to the source tables and using the triggers to insert data into change tracking tables whenever data modifications occur. The merge agent is used to copy those changes across to subscribers whenever synchronization occurs.

Merge replication is not transactionally consistent but does support non-SQL-Server nodes as part of the replication topology. The performance of merge replication usually suffers as the volume of data or the number of subscribers increases.

Merge replication is useful in the following situations:

- Multiple subscribers might update the data at the same time. Merge replication provides conflict detection and conflict correction.
- Subscribers need to make changes to data while disconnected.
- Conflicts occur during data changes on multiple subscribers.
- Applications do not depend upon tables being transactionally consistent.

Choosing Between Replication Options

Aspect	Snapshot	Transactional	Peer-To-Peer	Merge
Transactional	No	Yes	Yes	No
Complexity	Low	Medium	High	Medium
Subscribers can update	No	With limited options	With partitioned data	Yes
Conflict Detection	No	No	Yes	Yes
Conflict Resolution	No	No	No	Yes
Performance	Medium	High	High	Low

Key Points

The table above provides a comparison of a number of the aspects of each replication option, as discussed in the previous four topics.

Module 1: Introduction to SQL Server 2012 and Its Toolset

Lab 1: Introduction to SQL Server and Its Toolset

Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following step:

- Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.

Exercise 1: Verify SQL Server Component Installation

► Task 1: Check that Database Engine and Reporting Services have been installed for the MKTG instance

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **Configuration Tools**, and then click **SQL Server Configuration Manager**.
1. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
2. In the right-hand pane, ensure that the following services are listed for the MKTG instance:
 - SQL Server (MKTG)
 - SQL Full-text Filter Daemon Launcher (MKTG)
 - SQL Server Reporting Services (MKTG)
 - SQL Server Agent (MKTG)

► Task 2: Note the services that are installed for the default instance

1. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
2. In the right-hand pane, ensure that the following services are listed for the default instance:
 - SQL Server (MSSQLSERVER)
 - SQL Full-text Filter Daemon Launcher (MSSQLSERVER)
 - SQL Server Analysis Services (MSSQLSERVER)
 - SQL Server Agent (MSSQLSERVER)

► Task 3: Ensure that all required services including SQL Server Agent are started and set to autostart for both instances

1. Check that all the services for the MKTG instance have a **Start Mode** of **Automatic**. Ignore the SQL Full-text Filter Daemon Launcher service at this time.
2. Check that all the services for the default instance have a **Start Mode** of **Automatic**. Ignore the SQL Full-text Filter Daemon Launcher service at this time.

Exercise 2: Alter Service Accounts for New Instance

► Task 1: Change the service account for the MKTG database engine

1. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
2. In the right-hand pane, right-click **SQL Server (MTKG)**, and select **Properties**.
3. In the **Account Name** text box, type **AdventureWorks\PWService**.
4. In the **Password** text box, type **Pa\$\$w0rd**.
5. In the **Confirm Password** text box, type **Pa\$\$w0rd** and click **OK**.
6. In the **Confirm Account Change** window, click **Yes**.

► Task 2: Change the service account for the MKTG SQL Server Agent

1. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
2. In the right-hand pane, right-click **SQL Server Agent (MTKG)**, and select **Properties**.
3. In the **Account Name** text box, type:
 - **AdventureWorks\PWService**.
4. In the **Password** text box, type **Pa\$\$w0rd**.
5. In the **Confirm Password** text box, type **Pa\$\$w0rd** and click **OK**.
6. Right-click **SQL Server Agent (MTKG)** and select **Start** to restart the service.

Exercise 3: Enable Named Pipes Protocol for Both Instances

► Task 1: Enable the named pipes protocol for the default instance

1. In the left-hand pane of the SQL Server Configuration Manager window, expand **SQL Server Network Configuration** and then click **Protocols for MSSQLSERVER**.
2. In the right-hand pane, if the **Named Pipes** protocol is not enabled, right-click **Named Pipes** and select **Enable**. In the Warning window, click **OK**.

► Task 2: Enable the named pipes protocol for the MKTG instance

1. In the left-hand pane of the SQL Server Configuration Manager window, click **Protocols for MKTG**.
2. In the right-hand pane, if the **Named Pipes** protocol is not enabled, right-click **Named Pipes** and select **Enable**. In the Warning window, click **OK**.

► Task 3: Restart both database engine services

1. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
2. Right-click **SQL Server (MSSQLSERVER)** and select **Restart**.
3. Right-click **SQL Server (MTKG)** and select **Restart**.
4. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
5. In the toolbar, click the **Refresh** icon.

Exercise 4: Create an Alias for AdvDev

► Task 1: Create a 32-bit alias (AdvDev) for the MKTG instance

1. In the left-hand pane of the SQL Server Configuration Manager window, expand **SQL Native Client 11.0 Configuration (32bit)** and click **Client Protocols**.
2. Confirm that the **Named Pipes** protocol is **Enabled**.
3. In the left-hand pane, click **Aliases**.
4. In the left-hand pane, right-click **Aliases** and select **New Alias**.
5. In the Alias – New window, in the **Alias Name** text box, type **AdvDev**.
6. In the **Protocol** drop-down list box, select **Named Pipes**.
7. In the **Server** text box, type **.\MKTG** and click **OK**.

► Task 2: Create a 64-bit alias (AdvDev) for the MKTG instance

1. In the left-hand pane of the SQL Server Configuration Manager window, expand **SQL Native Client 11.0 Configuration** and click **Client Protocols**.
2. Confirm that the **Named Pipes** protocol is **Enabled**.
3. In the left-hand pane, click **Aliases**.
4. In the left-hand pane, right-click **Aliases** and select **New Alias**.
5. In the Alias – New window, in the **Alias Name** text box, type **AdvDev**.
6. In the **Protocol** drop-down list box, select **Named Pipes**.
7. In the **Server** text box, type **.\MKTG** and click **OK**.

► Task 3: Use SQL Server Management Studio to connect to the alias to ensure it works as expected

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and then click **SQL Server Management Studio**.
2. In the Connect to Server window, ensure that **Server Type** is set to **Database Engine**.
3. In the **Server name** text box, type **AdvDev**.
4. In the **Authentication** drop-down list, select **Windows Authentication**, and click **Connect**.
5. In Object Explorer, under **AdvDev** expand **Databases**.



Note: The databases that are present include at least the following: System Databases, Database Snapshots, ReportServer\$MKTG, and ReportServer\$MKTGTempDB.

6. Close SQL Server Management Studio.
7. Close SQL Server Configuration Manager.

Challenge Exercise 5: Ensure SQL Browser is Disabled and Configure a Fixed TCP/IP Port (Only if time permits)

► Task 1: Configure the TCP port for the MKTG database engine instance to 51550

1. In the Virtual Machine window, from the **Start** menu, click **All Programs**, click **Microsoft SQL Server 2012**, click **Configuration Tools**, and then click **SQL Server Configuration Manager**.
2. In the left-hand pane of the SQL Server Configuration Manager window, expand **SQL Server Network Configuration** and then click **Protocols for MKTG**.
3. Right-click the **TCP/IP** protocol and select **Properties**.
4. In the TCP/IP Properties window, click **IP Addresses** tab.
5. Scroll to the bottom of the screen, under the **IP All** section, clear the value for **TCP Dynamic Ports**.
6. For **TCP Port**, type **51550**, and click **OK**.
7. In the Warning window, click **OK**.
8. In the left-hand pane, click **SQL Server Services**.
9. Right-click **SQL Server (MKTG)** and select **Restart**.
10. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
11. In the toolbar, click the **Refresh** icon and make sure the service starts.

► Task 2: Disable the SQLBrowser service

1. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
2. In the right-hand pane, if the state for the **SQL Server Browser** is **Running**, right-click **SQL Server Browser** and click **Stop**.
3. Right-click the **SQL Server Browser** and click **Properties**.
4. In the SQL Server Browser Properties window, in the **Service** tab, set the **Start Mode** to **Disabled** and click **OK**.
5. Close SQL Server Configuration Manager.

Module 2: Preparing Systems for SQL Server 2012

Lab 2: Preparing Systems for SQL Server

Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A_Labs\10775A_02_PRJ\10775A_02_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

Exercise 1: Adjust Memory Configuration

► Task 1: Check total server memory

1. Click **Start**, right-click **Computer**, and click **Properties**.
2. Write down the value for **Installed memory (RAM)**.
3. Close the System window.

► Task 2: Check the memory allocated to the default instance

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
2. In the Connect to Server window, type **AdventureWorks** for the **Server name** and click **Connect**.
3. In Object Explorer, right-click the **AdventureWorks** server instance, and click **Properties**.
4. In the **Select a page** pane click **Memory**.
5. Write down the values for **Minimum server memory** (in MB) and **Maximum server memory** (in MB), and click **Cancel**.
6. In **SQL Server Management Studio**, from the **File** menu, click **Exit**.

► **Task 3: Check the memory allocated to the MKTG instance**

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
2. In the Connect to Server window, type **Proseware** for the **Server name** and click **Connect**.
3. In Object Explorer, right-click the **Proseware** server instance, and click **Properties**.
4. In the **Select a page** pane click **Memory**.
5. Write down the values for **Minimum server memory** (in MB) and **Maximum server memory** (in MB), and click **Cancel**.
6. In **SQL Server Management Studio**, from the **File** menu, click **Exit**.

► **Task 4: Decide if the memory allocation is appropriate. If not, make the required changes to the memory configuration**

1. Review the **Required Memory Configuration** from the Supporting Documentation in the Student Manual.
2. Calculate the **Maximum memory** for the **AdventureWorks** server instance as follows:
 - Example calculation (actual values depend upon VM configuration)
 - $\text{Max Memory} = (\text{Server Memory} - 1.5\text{GB}) * 0.6$
 - $\text{Max Memory} = (3.0 - 1.5) * 0.6$
 - $\text{Max Memory} = 0.9$ (approximate)
3. Calculate the **Maximum memory** for the **Proseware** server instance as follows:
 - Example calculation (actual values depend upon VM configuration)
 - $\text{Max Memory} = (\text{Server Memory} - 1.5\text{GB}) * 0.4$
 - $\text{Max Memory} = (3.0 - 1.5) * 0.4$
 - $\text{Max Memory} = 0.6$ (approximate)
4. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
5. In the Connect to Server window, type **AdventureWorks** for the **Server name** and click **Connect**.
6. In Object Explorer, right-click the **AdventureWorks** server instance, and click **Properties**.
7. In the **Select a page** pane click **Memory**.
8. Set the **Maximum server memory (in MB)** to the value **900**, and click **OK**.



Note: A more accurate value for 0.9GB would have been 921MB, but the value 900 has been used for simplicity.

9. In Object Explorer click **Connect**, and click **Database Engine**.
10. In the Connect to Server window, type **Proseware** for the **Server name** and click **Connect**.
11. In Object Explorer, right-click the **Proseware** server instance, and click **Properties**.
12. In the **Select a page** pane click **Memory**.
13. Set the **Maximum server memory (in MB)** to the value **600**, and click **OK**.



Note: A more accurate value for 0.6GB would have been 614MB, but the value 600 has been used for simplicity.

Exercise 2: Pre-installation Stress Testing

► Task 1: Configure SQLIOSIM

1. In the folder D:\10775A_Labs\10775A_02_PRJ, right-click **sqliosimx64.exe** and click **Run as administrator**.
2. In the Microsoft SQL Server IO Simulator window, note the agreement and click **Yes**.
3. In the WinZip Self-Extractor - SQLIOS~1.EXE window click **Unzip**.
4. In the WinZip Self-Extractor window click **OK**.
5. In the WinZip Self-Extractor - SQLIOS~1.EXE window click **Close**.
6. Open **Windows Explorer** and navigate to the C:\SQLIOSimX64 folder. Note that two versions of SQLIOSim are supplied: a command line version and a GUI version. Right-click **SQLIOSim.exe**, and click **Run as administrator**.
7. In the Files and Configuration window click on the line containing **C:\sqliosim.mdx**, then click **Remove**.
8. In the Files and Configuration window click on the line containing **C:\sqliosim.idx**, then click **Remove**.
9. In the Files and Configuration window click on the line containing **D:\sqliosim.idx**, then click **Remove**.
10. In the Files and Configuration window click on the line containing **L:\sqliosim.mdx**, then click **Remove**.
11. In the Files and Configuration window click on the line containing **D:\sqliosim.mdx**, then in the **Size (MB)** change the value from 4096 to **100**, also change the **Max Size** value from 8192 to **200**, and the **Increment** to **20**, then click **Apply**.
12. On the line containing **L:\sqliosim.idx**, set the **Size (MB)** to **50**, the **Max Size** to **100**, the **Increment** to **10**, and click **Apply**.
13. Change the **Cycle Duration (sec)** value from 300 to **60**.
14. Ensure that the **Delete Files at Shutdown** checkbox is checked, and click **OK** to complete the configuration.

► **Task 2: Execute SQLIOSIM**

1. In the SQLIOSIM window from the **Simulator** menu, click **Start**.
2. Wait for the test to complete in about 1 minute. While the test is running, note the information that is displayed.
3. In the SQLIOSim window click **OK**.

► **Task 3: Review the results from executing SQLIOSIM**

1. If any errors are returned in red, review the errors.
2. Locate the final summary for file **D:\sqliosim.mdx** and note the **Running Average IO Duration (ms)**.
3. Locate the final summary for file **L:\sqliosim.idx** and note the **Running Average IO Duration (ms)**.

Note: The value returned for each drive depends upon the speed of the hardware. A typical value for either drive would be 15.

4. In the SQLIOSim window from the **File** menu click **Exit**.

Challenge Exercise 3: Check Specific I/O Operations (Only if time permits)

► **Task 1: Install the SQLIO Utility**

1. In the folder **D:\10775A_Labs\10775A_02_PRJ**, right-click **SQLIO.msi** and click **Install**.
2. In the Welcome to the SQLIO Setup Wizard window click **Next**.
3. In the License Agreement window, note the agreement click the **I Agree** radio button, and click **Next**.
4. In the Select Installation Folder window click the **Everyone** radio button, and click **Next**.
5. In the Confirm Installation window click **Next**.
6. In the Installation Complete window click **Close**.

► **Task 2: Configure and Execute the SQLIO Utility**

1. Review the Supporting Documentation in the Student Manual for details of the SQLIO tests to be performed.
2. In Windows Explorer, navigate to the **C:\Program Files (x86)\SQLIO** folder.
3. Double-click the file **param.txt**. Place a **#** at the beginning of the line **c:\testfile.dat**. Remove the **#** from the line beginning **d:\testfile.dat**.
4. From the **File** menu click **Exit**.
5. In the Notepad window click **Save**.
6. In Windows Explorer, double-click the file **C:\Program Files (x86)\SQLIO\Using SQLIO.rtf**. Note the examples and descriptions of the available parameters for SQLIO.

7. Close the using SQLIO.rtf - WordPad window.
8. From the **Start** menu, click **All Programs**, click **Accessories**, right-click **Command Prompt** and click **Run as administrator**.
9. Type **CD \Program Files (x86)\SQLIO** and press Enter.
10. Type the following command and press Enter:

```
sqlio.exe -kR -s60 -fsequential -o8 -b64 -LS -Fparam.txt timeout /T 60
```

11. Look through the output of the command, note the values returned for IOs/sec and MBs/sec. These values are commonly referred to as IOPS and throughput. Note the minimum, maximum and average latency.
 12. Type the following command and press Enter:
- ```
sqlio.exe -kW -s60 -frandom -o8 -b8 -LS -Fparam.txt timeout /T 60
```
13. Look through the output of the command, note the values returned for IOs/sec and MBs/sec. These values are commonly referred to as IOPS and throughput. Note the minimum, maximum and average latency.
  14. Type **Exit** and press Enter.

**MCT USE ONLY. STUDENT USE PROHIBITED**



## Module 3: Installing and Configuring SQL Server 2012

# Lab 3: Installing and Configuring SQL Server

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_03\_PRJ\10775A\_03\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.
8. On the host system, in the **Virtual Machines** list in Hyper-V Manager, right-click the **10775A-MIA-SQL1** virtual machine and click **Settings**.
9. In the Settings for 10775A-MIA-SQL1 window, in the **Hardware** list expand **IDE Controller 1**, and click **DVD Drive**.
10. In the **DVD Drive** properties pane, click **Image file**, and click **browse**.
11. Navigate to the file **C:\Program Files\Microsoft Learning\10775\Drives\10775A-MIA-SQL1\Virtual Hard Disks\SQLFULL\_ENU.iso** and click **Open**.
12. In the Settings for 10775A-MIA-SQL1 window, click **OK**.

### Exercise 1: Review installation requirements

- **Task 1: Review the supporting documentation prior to installation**
  - Read the supporting documentation in the student manual.
- **Task 2: Create the folders that are required for the data and log files**
  - Using Windows Explorer, create the following folders:
    - D:\MKTGDEV
    - L\MKTGDEV

## Exercise 2: Install the SQL Server Instance

### ► Task 1: Based on the requirements reviewed in exercise 1, install another instance of SQL Server

1. In the Virtual Machine window, in the AutoPlay window (which should now have popped up) click **Run SETUP.EXE** and wait for SQL Server Setup to start. (If the AutoPlay window has not launched, or you have inadvertently closed it, double-click **SETUP.EXE** in the root of the new drive that has been mounted when you connected the ISO file to your virtual machine).
2. In the SQL Server Installation Center window, click on the **Installation** tab.
3. Click **New SQL Server stand-alone installation or add features to an existing installation** from the list of available options and wait for SQL Server setup to start.
4. In the Setup Support Rules window, click **Show details** and note that the list of rules that has been checked.
5. In the Setup Support Rules window, click **OK**.
6. Make sure that **Send feature usage data to Microsoft** is not selected and click **Next**.



**Note** In general, you are encouraged to choose this feature as it helps Microsoft produce a better product. In this case, we are deselecting it because we are in an isolated environment without network access within the virtual machine.

7. In the Product Updates window, uncheck **Include SQL Server product updates** and click **Next**. (Note the next screen might take some time to display.)
8. In the **Setup Support Rules** page, click **Show details** (if visible) and read the installation checklist and check to make sure the Status for items have **Passed**. Note any warnings that are listed and click **Next**.
9. In the **Installation Type** page, ensure the option button **Perform a new installation of SQL Server 2012** is selected and then click **Next**.
10. In the Product Key window, click **Next**.
11. In the **License Terms** page, note the Microsoft Software License Terms and check **I accept the license terms**.
12. In the **Setup Role** page, ensure that **SQL Server Feature Installation** is selected and click **Next**.
13. In the **Feature Selection** page, under the **Instance Features** check **Database Engine Services** and click **Next**.
14. In the **Installation Rules** page, click **Show details**, note the list of rules and the status of each rule and click **Next**.
15. In the **Instance Configuration** page, ensure that **Named instance** is checked and type **MKTGDEV** in the **Named instance** field and click **Next**.
16. In the **Disk Space Requirements** page, read the **Disk Usage Summary** and then click **Next**.

17. In the **Server Configuration** page, click the **Account Name** field for the **SQL Server Database Engine** row and select <<**Browse...**>> from the drop-down list.
18. In the **Select User, Computer, Service Account, or Group** window, in the **Enter the object name to select (Examples):** dialog box type **AdventureWorks\PWService**, click **Check Names** and click **OK**.
19. In the **Password** column for the **SQL Server Database Engine** row, type **Pa\$\$w0rd**.
20. In the **Server Configuration** page, click the **Account Name** field for the **SQL Server Agent** row and click <<**Browse...**>> from the drop-down list.
21. In the **Select User, Computer, Service Account, or Group** window, in the **Enter the object name to select (Examples):** dialog box type **AdventureWorks\PWService**, click **Check Names** and click **OK**.
22. In the **Password** column for the **SQL Server Agent** row, type **Pa\$\$w0rd**.
23. In the **Server Configuration** page, in the **Service Accounts** tab, and in the **Startup Type** drop-down list for the **SQL Server Agent** select **Automatic**.
24. Click the **Collation** tab, ensure that **SQL\_Latin1\_General\_CP1\_CI\_AS** is selected and click **Next**.
25. In the **Database Engine Configuration** page, on the **Server Configuration** tab, in the **Authentication Mode** section, ensure that **Mixed Mode (SQL Server authentication and Windows authentication)** is selected.
26. In the **Enter password** textbox, type **Pa\$\$w0rd**.
27. In the **Confirm password** textbox, type **Pa\$\$w0rd**.
28. Click **Add Current User**, this will add the user **ADVENTUREWORKS\administrator (Administrator)** to the list of Administrators.
29. Click the **Data Directories** tab, change the **User database directory** to **D:\MKTGDEV**.
30. Change the **User database log directory** to **L:\MKTGDEV**.
31. Change the **Temp DB directory** to **D:\MKTGDEV**.
32. Change the **Temp DB log directory** to **L:\MKTGDEV**.
33. Click the **FILESTREAM** tab, and ensure that **Enable FILESTREAM for Transact-SQL access** is not selected and click **Next**.
34. In the **Error Reporting** page, make sure that **Send Windows and SQL Server Error Reports to Microsoft or your corporate report server** is not checked, then click **Next**.
35. In the **Installation Configuration Rules** page, **Show details**, review the list of rules, and click **Next**.
36. In the **Ready to Install** page, review the summary and click **Install**.
37. In the **Complete** page, click **Close**.
38. Close the SQL Server Installation Center window.

### Exercise 3: Perform Post-installation Setup and Checks

► **Task 1: Check that the services for the new SQL Server instance are running**

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **Configuration Tools**, and then click **SQL Server Configuration Manager**.
2. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
3. In the right-hand pane, ensure that the following services are listed as running for the **MKTGDEV** instance:
  - SQL Server (MKTGDEV)
  - SQL Server Agent (MKTGDEV)

► **Task 2: Configure both 32 bit and 64 bit aliases for the new instance**

1. In the left-hand pane of the SQL Server Configuration Manager window, expand **SQL Server Network Configuration** and then click **Protocols for MKTGDEV**.
2. In the right-hand pane, right-click **Named Pipes** and select **Enable**.
3. In the Warning window, click **OK**.
4. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
5. Right-click **SQL Server (MKTGDEV)** and select **Restart**.
6. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
7. In the toolbar, click the **Refresh** icon.
8. In the left-hand pane of the SQL Server Configuration Manager window, expand **SQL Native Client 11.0 Configuration (32bit)** and click **Client Protocols**.
9. Confirm that the **Named Pipes** protocol is **Enabled**.
10. In the left-hand pane, right-click **Aliases** and select **New Alias**.
11. In the Alias – New window, in the **Alias Name** text box, type **PWDev**.
12. In the **Protocol** drop-down list box, select **Named Pipes**.
13. In the **Server** text box, type **.\MKTGDEV** and click **OK**.
14. In the left-hand pane of the SQL Server Configuration Manager window, expand **SQL Native Client 11.0 Configuration** and click **Client Protocols**.
15. Confirm that the **Named Pipes** protocol is **Enabled**.
16. In the left-hand pane, right-click **Aliases** and select **New Alias**.
17. In the Alias – New window, in the **Alias Name** text box, type **PWDev**.
18. In the **Protocol** drop-down list box, select **Named Pipes**.
19. In the **Server text** box, type **.\MKTGDEV** and click **OK**.
20. Close SQL Server Configuration Manager.

► **Task 3: Connect to the new instance using SSMS**

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and then click **SQL Server Management Studio**.
2. In the Connect to Server window, ensure that **Server Type** is set to **Database Engine**.
3. In the **Server name** text box, type **PWDev**.
4. In the **Authentication** drop-down list, select **Windows Authentication**, and click **Connect**.
5. In Object Explorer, under **PWDev** expand **Databases**.
6. Close SQL Server Management Studio.

### Challenge Exercise 4: Configure Server Memory (Only if time permits)

► **Task 1: Review the current memory available on the server**

1. Click **Start**, right-click **Computer**, and click **Properties**.
2. Write down the value for **Installed memory (RAM)**.

► **Task 2: Determine an appropriate memory allocation for each instance**

1. Review the **Required Memory Configuration** from the Supporting Documentation in the Student Manual.
2. Calculate the Maximum memory for the **AdventureWorks** server instance as follows:
  - Example calculation (actual values depend upon VM configuration)
  - $\text{Max Memory} = (\text{Server Memory} - 1.0\text{GB}) * 0.4$
  - $\text{Max Memory} = (3.0 - 1.0) * 0.4$
  - $\text{Max Memory} = 0.8$  (approximate)
3. Calculate the Maximum memory for the **Proseware** server instance as follows:
  - Example calculation (actual values depend upon VM configuration)
  - $\text{Max Memory} = (\text{Server Memory} - 1.0\text{GB}) * 0.3$
  - $\text{Max Memory} = (3.0 - 1.0) * 0.3$
  - $\text{Max Memory} = 0.6$  (approximate)
4. Calculate the Maximum memory for the **PWDev** server instance as follows:
  - Example calculation (actual values depend upon VM configuration)
  - $\text{Max Memory} = (\text{Server Memory} - 1.0\text{GB}) * 0.3$
  - $\text{Max Memory} = (3.0 - 1.0) * 0.3$
  - $\text{Max Memory} = 0.6$  (approximate)

► **Task 3: Configure each instance appropriately**

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
2. In the Connect to Server window, type **AdventureWorks** for the Server name and click **Connect**.
3. In Object Explorer, right-click the **AdventureWorks** server instance, and click **Properties**.
4. In the **Select a page** pane click **Memory**.
5. Set the **Maximum server memory (in MB)** to the value **800**, and click **OK**.



**Note** A more precise value could have been used but the value 800 has been used for simplicity.

6. In Object Explorer click **Connect**, and click **Database Engine**.
7. In the Connect to Server window, type **Proseware** for the **Server name** and click **Connect**.
8. In Object Explorer, right-click the **Proseware** server instance, and click **Properties**.
9. In the **Select a page** pane click **Memory**.
10. Set the **Maximum server memory (in MB)** to the value **600**, and click **OK**.



**Note** A more precise value for 1.0GB could have been used but the value 600 has been used for simplicity.

11. In Object Explorer click **Connect**, and click **Database Engine**.
12. In the Connect to Server window, type **PWDev** for the **Server name** and click **Connect**.
13. In Object Explorer, right-click the **PWDev** server instance, and click **Properties**.
14. In the **Select a page** pane click **Memory**.
15. Set the **Maximum server memory (in MB)** to the value **600**, and click **OK**.



**Note** A more accurate value for 1.0GB could have been used but the value 600 has been used for simplicity.

16. Close SQL Server Management Studio.

## Module 4: Working with Databases

# Lab 4: Working with Databases

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_04\_PRJ\10775A\_04\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Adjust tempdb Configuration

#### ► Task 1: Adjust the size of tempdb

1. Review the requirement for **tempdb size** in the Supporting Documentation.
2. In Object Explorer expand the **Proseware** server, expand **Databases**, and expand **System Databases**. Right-click the **master** database and click **New Query**.
3. Enter the following statements and click **Execute**:

```
USE master;
GO
ALTER DATABASE tempdb
 MODIFY FILE (NAME = tempdev, SIZE = 30MB);
GO
ALTER DATABASE tempdb
 MODIFY FILE (NAME = templog, SIZE = 10MB);
GO
```

#### ► Task 2: Check that the tempdb size is still correct after a restart

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **Configuration Tools**, and then click **SQL Server Configuration Manager**.
2. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
3. In the right-hand pane, right-click the service **SQL Server (MKTG)** and click **restart**.
4. In Object Explorer, right-click the **tempdb** database and click **Properties**.
5. In the Database Properties – tempdb window, click **Files** in the **Select a page** pane.
6. In the **Database files** list, check that the initial size of the data file is **30** and the initial size of the log file is **10**.

## Exercise 2: Create the RateTracking Database

### ► Task 1: Create the database

1. Review the supplied requirements in the supporting documentation for the exercise.
2. In Object Explorer expand the **Proseware** server, expand **Databases**, and expand **System Databases**. Right-click the **master** database and click **New Query**.
3. Enter the following statements and click **Execute**:

```
USE master;
GO

CREATE DATABASE RateTracking
ON
(NAME = RateTracking_dat,
 FILENAME = 'D:\MKTG\RateTracking.mdf',
 SIZE = 10MB, MAXSIZE = 100MB,
 FILEGROWTH = 10MB
)
LOG ON
(NAME = RateTracking_log,
 FILENAME = 'L:\MKTG\RateTracking.ldf',
 SIZE = 20MB,
 MAXSIZE = UNLIMITED,
 FILEGROWTH = 20MB
);
GO
```

### ► Task 2: Create the required filegroups and files

1. Review the supplied requirements in the supporting documentation for the required files and filegroups.
2. In Object Explorer expand the **Proseware** server, expand **Databases**, and expand **System Databases**. Right-click the **master** database and click **New Query**.
3. Enter the following statements and click **Execute**:

```
USE master;
GO

ALTER DATABASE RateTracking
ADD FILEGROUP USERDATA;
GO

ALTER DATABASE RateTracking
ADD FILE
(NAME = RateTracking_dat_1,
 FILENAME = 'D:\MKTG\RateTracking_1.ndf',
 SIZE = 20MB,
 MAXSIZE = 100MB,
 FILEGROWTH = 10MB
) TO FILEGROUP USERDATA;
GO
```

*(code continued on next page)*



```
ALTER DATABASE RateTracking
ADD FILE
(NAME = RateTracking_dat_2,
 FILENAME = 'D:\MKTG\RateTracking_2.ndf',
 SIZE = 20MB,
 MAXSIZE = 100MB,
 FILEGROWTH = 10MB
) TO FILEGROUP USERDATA;
GO

ALTER DATABASE RateTracking
ADD FILEGROUP ARCHIVE;
GO

ALTER DATABASE RateTracking
ADD FILE
(NAME = RateTracking_dat_3,
 FILENAME = 'D:\MKTG\RateTracking_3.ndf',
 SIZE = 200MB,
 MAXSIZE = 500MB,
 FILEGROWTH = 50MB
) TO FILEGROUP ARCHIVE;
GO

ALTER DATABASE RateTracking
ADD FILE
(NAME = RateTracking_dat_4,
 FILENAME = 'D:\MKTG\RateTracking_4.ndf',
 SIZE = 200MB,
 MAXSIZE = 500MB,
 FILEGROWTH = 50MB
) TO FILEGROUP ARCHIVE;
GO
```

► **Task 3: Change the default filegroup for the database**

1. Review the supplied requirements in the supporting documentation for the default filegroup.
2. In Object Explorer expand the **Proseware** server, expand **Databases**, and expand **System Databases**. Right-click the **master** database and click **New Query**.
3. Enter the following statements and click **Execute**:

```
USE master;
GO

ALTER DATABASE RateTracking MODIFY FILEGROUP USERDATA DEFAULT;
GO
```

### Exercise 3: Attach the OldProspects Database

#### ► Task 1: Copy the database files

- In Windows Explorer, copy the files as specified below:

| Filename         | Source Folder                | Destination Folder |
|------------------|------------------------------|--------------------|
| OldProspects.mdf | D:\10775A_Labs\10775A_04_PRJ | D:\MKTG            |
| OldProspects.ldf | D:\10775A_Labs\10775A_04_PRJ | L:\MKTG            |

#### ► Task 2: Attach the database to the MKTG instance

- In Object Explorer, expand the **Proseware** server and expand **Databases**.
- Right-click **Databases** and click **Attach**.
- Click **Add** and navigate to the file **D:\MKTG\OldProspects.mdf** and click **OK**.
- Click the ellipsis beside the **Current File Path** for the log entry and navigate to the file **L:\MKTG\OldProspects.ldf** and click **OK**.
- Click **OK** to attach the database and note that **OldProspects** now appears in the list of databases in **Object Explorer**.

### Challenge Exercise 4: Add Multiple Files to tempdb (Only if time permits)

#### ► Task 1: Review the tempdb file requirements

- In the Supporting Documentation review the tempdb Requirements From The Consultant section.

#### ► Task 2: Move existing files

- In Object Explorer expand the **Proseware** server, expand **Databases**, and expand **System Databases**. Right-click the **master** database and click **New Query**.
- Enter the following statements and click **Execute**:

```
USE master;
GO

ALTER DATABASE tempdb
MODIFY FILE (NAME = tempdev, FILENAME = 'D:\MKTG\tempdb.mdf');

ALTER DATABASE tempdb
MODIFY FILE (NAME = templog, FILENAME = 'L:\MKTG\templog.ldf');
GO
```

### ► Task 3: Add new files

1. In Object Explorer expand the **Proseware** server, expand **Databases**, and expand **System Databases**. Right-click the **master** database and click **New Query**.
2. Enter the following statements and click **Execute**:

```
USE master;
GO

ALTER DATABASE tempdb
ADD FILE (NAME = N'tempdev2',
 FILENAME = N'D:\MKTG\tempdb_file2.ndf' ,
 SIZE = 20MB,
 FILEGROWTH = 10MB,
 MAXSIZE = UNLIMITED);

ALTER DATABASE tempdb
ADD FILE (NAME = N'tempdev3',
 FILENAME = N'D:\MKTG\tempdb_file3.ndf' ,
 SIZE = 20MB,
 FILEGROWTH = 10MB,
 MAXSIZE = UNLIMITED);

ALTER DATABASE tempdb
ADD FILE (NAME = N'tempdev4',
 FILENAME = N'D:\MKTG\tempdb_file4.ndf' ,
 SIZE = 20MB,
 FILEGROWTH = 10MB,
 MAXSIZE = UNLIMITED);
GO
```

### ► Task 4: Restart the server and check file locations

1. Click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, click **Configuration Tools**, and then click **SQL Server Configuration Manager**.
2. In the left-hand pane of the SQL Server Configuration Manager window, click **SQL Server Services**.
3. In the right-hand pane, right-click the service **SQL Server (MKTG)** and click **restart**.
4. In Object Explorer, right-click the **tempdb** database and click **Properties**.
5. In the Database Properties – tempdb window, click **Files** in the **Select a page** pane.
6. Make sure that the files listed match the requirements.

**MCT USE ONLY. STUDENT USE PROHIBITED**

## Module 5: Understanding SQL Server 2012 Recovery Models

# Lab 5: Understanding SQL Server Recovery Models

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_05\_PRJ\10775A\_05\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Plan a Backup Strategy

#### ► Task 1: Review the business requirements

- Review the supplied business requirements in the supporting documentation for the exercise.

#### ► Task 2: Determine an appropriate backup strategy for each database

- There is no one correct answer. The options provided below are an example strategy that would meet the requirements.

For the MarketDev database: **Full Recovery Model**

| Type of Backup | Scheduled                                                              |
|----------------|------------------------------------------------------------------------|
| <b>Full</b>    | <b>7:00pm Daily</b>                                                    |
| <b>Log</b>     | <b>Every 20 minutes, starting at 8:20am and continuing till 6:00pm</b> |

#### Notes:

- Full database backup should complete in approximately 3.4 hours (20GB/100MB per minute). This means that you cannot employ only a full database backup strategy as it would not meet the RPO. The full database backup should complete by 10:24pm which is within the available time window for backups.

- Each log backup should complete in approximately 3.4 minutes (1GB per hour / 3 log backups per hour /100 MB per minute). This fits within the 20 minute interval and meets the RPO.
- Full database restore should complete in approximately 4.3 hours (20GB/80MB per minute). Log file restore should complete in approximately 2.1 hours (10 hours \* 1GB per hour /80MB per minute) and meets the RTO.

For the Research database: **Simple Recovery Model**

|                |                     |
|----------------|---------------------|
| Type of Backup | Scheduled           |
| <b>Full</b>    | <b>6:30pm Daily</b> |

**Notes:**

- Recovery to last full daily database backup complies with the RPO.
- Daily backup should complete in approximately 2 minutes (200MB/100MB per minute).
- Full restore should complete in approximately 2.5 minutes and complies with RTO (200MB/80MB per minute).

## Exercise 2: Configure Recovery Models

### ► Task 1: Review and adjust the current database recovery models

1. The recovery model for the MarketDev database is Simple and should be **Full**. Adjust the model by the following steps:
  1. In SSMS, in Object Explorer expand the **Proseware** server, right-click **Databases** and click **Refresh**.
  2. Expand **Databases**, right-click **MarketDev** and click **Properties**.
  3. In the **Select a page** pane, click **Options**.
  4. In the right hand pane, choose **Full** from the **Recovery model** drop down list and click **OK**.
2. The recovery model for the Research database is correct and doesn't need to be modified.

## Challenge Exercise 3: Review Recovery Models and Strategy (Only if time permits)

### ► Task 1: Review the RPO and RTO requirements for the databases

- The supporting documentation includes details of the business continuity requirements for the databases. You need to review this documentation.

### ► Task 2: Review the existing recovery models and backup strategies

- The supporting documentation also includes details of the backup strategy for the databases. You need to review this documentation.

► **Task 3: Indicate whether or not the strategy would be successful**

1. For the CreditControl database, a Full backup would take approximately 3.4 hours (20GB/100MB per minute). This would not satisfy the requirements for the time window as the Wednesday 6am Full backup would not complete before office hours start.
2. For the PotentialIssue database, the 15 minute log backups would meet the RPO. A Full restore should take approximately 24 minutes  $((200\text{MB} + (7 \text{ days} * 24 \text{ hours}) * 10\text{MB per hour}) / 80\text{MB per minute})$  which meets the RTO. The Full database backup would complete in approximately 2 minutes (200MB/100MB per minute) which means that the Full database would complete in the available time window. The backup strategy for the PotentialIssue database meets the business requirements.

**MCT USE ONLY. STUDENT USE PROHIBITED**



## Module 6: Backup of SQL Server Databases

# Lab 6: Backup of SQL Server Databases

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project D:\10775A\_Labs\10775A\_06\_PRJ\10775A\_06\_PRJ.ssmssln.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Investigate Backup Compression

#### ► Task 1: Create a database backup without compression

1. Using Windows Explorer create a new folder L:\SQLBackups.
2. In Object Explorer, in SQL Server Management Studio, expand the **Proseware** server, expand **Databases**, right-click the **MarketDev** database, click **Tasks**, and click **Back Up**.
3. In the Back Up Database – MarketDev window click **Remove**.
4. In the Back Up Database – MarketDev window click **Add**.
5. In the **File name** textbox type L:\SQLBackups\MarketDev\_Full\_Uncompressed.BAK and click **OK**.
6. In the **Select a page** pane, click **Options**.
7. In the **Set backup compression** drop-down list, click **Do not compress backup**, and click **OK**.
8. In the Microsoft SQL Server Management Studio window click **OK**.

#### ► Task 2: Create a database backup with compression

1. In Object Explorer, right-click the **MarketDev** database, click **Tasks**, and click **Back Up**.
2. In the Back Up Database – MarketDev window click **Remove**.
3. In the Back Up Database – MarketDev window click **Add**.
4. In the **File name** textbox type L:\SQLBackups\MarketDev\_Full\_Compressed.BAK and click **OK**.

5. In the **Select a page** pane, click **Options**.
6. In the **Set backup compression** drop-down list, click **Compress backup**, and click **OK**.
7. In the Microsoft SQL Server Management Studio window click **OK**.

► **Task 3: Compare the file sizes created**

1. In Windows Explorer, navigate to the folder L:\SQLBackups and note the size of the compress and uncompressed backup files.
2. Calculate the space savings provided by compression as:

$$\text{SpaceSavings} = (\text{Uncompressed size} - \text{Compressed size}) * 100 / \text{Uncompressed size}$$



**Note** A typical value would be  $(233-66)*100/233 = 72\%$  saving.

## Exercise 2: Transaction Log Backup

► **Task 1: Execute a script to introduce workload to the MarketDev database**

1. In Solution Explorer, double-click the file **61 – Workload File.sql** to open it.
2. On the **Toolbar** click **Execute**.

► **Task 2: Backup the transaction log on the MarketDev database**

1. In Object Explorer, right-click the **MarketDev** database, click **Tasks**, and click **Back Up**.
2. In the Back Up Database – MarketDev window click **Remove**.
3. In the Back Up Database – MarketDev window click **Add**.
4. In the **File name** textbox type **L:\SQLBackups\MarketDev\_Log\_Compressed.BAK** and click **OK**.
5. From the **Backup type** drop-down list box, select **Transaction Log**.
6. In the **Select a page** pane, click **Options**.
7. In the **Set backup compression** drop-down list, click **Compress backup**, and click **OK**.
8. In the Microsoft SQL Server Management Studio window click **OK**.

## Exercise 3: Differential Backup

► **Task 1: Execute a script to introduce workload to the MarketDev database**

1. In Solution Explorer, double-click the file **61 – Workload File.sql** to open it.
2. On the **Toolbar** click **Execute**.

► **Task 2: Create a differential backup of the MarketDev database**

1. In Object Explorer, right-click the **MarketDev** database, click **Tasks**, and click **Back Up**.
2. In the Back Up Database – MarketDev window click **Remove**.
3. In the Back Up Database – MarketDev window click **Add**.

4. In the **File name** textbox type **L:\SQLBackups\MarketDev\_Differential\_Compressed.BAK** and click **OK**.
5. From the **Backup type** drop-down list box, select **Differential**.
6. In the **Select a page** pane, click **Options**.
7. In the **Set backup compression** drop-down list, click **Compress backup**, and click **OK**.
8. In the Microsoft SQL Server Management Studio window click **OK**.
9. Using Windows Explorer, note the size of the differential backup **L:\SQLBackups\MarketDev\_Differential\_Compressed.BAK** compared to the full backup **L:\SQLBackups\MarketDev\_Full\_Compressed.BAK**.



**Note** A typical value for the Differential backup size would be 240KB. A typical value for the Full backup size would be 66 MB.

► **Task 3: Execute a script to introduce workload to the MarketDev database**

1. In Solution Explorer, double-click the file **71 – Workload File.sql** to open it.
2. On the **Toolbar** click **Execute**.

► **Task 4: Append a differential backup to the previous differential backup file**

1. In Object Explorer, right-click the **MarketDev** database, click **Tasks**, and click **Back Up**.
2. Ensure that the **Destination** shows **L:\SQLBackups\MarketDev\_Differential\_Compressed.BAK**.
3. From the **Backup type** drop-down list box, select **Differential**.
4. In the **Select a page** pane, click **Options**.
5. In the **Overwrite media** option ensure that **Append to the existing backup set** is selected.
6. In the **Set backup compression** drop-down list, click **Compress backup**, and click **OK**.
7. In the Microsoft SQL Server Management Studio window click **OK**.
8. Using Windows Explorer, note that the size of the **Differential backup** has increased. The file now contains two backups.



**Note** A typical value for the backup file size would be 54 MB.

## Exercise 4: Copy-only Backup

► **Task 1: Create a copy-only backup of the MarketDev database, ensuring to choose to verify the backup**

1. In Object Explorer, right-click the **MarketDev** database, click **Tasks**, and click **Back Up**.
2. In the Back Up Database – MarketDev window click **Remove**.
3. In the Back Up Database – MarketDev window click **Add**.
4. In the **File name** textbox type **L:\SQLBackups\MarketDev\_Copy\_Compressed.BAK** and click **OK**.

5. From the **Backup type** drop-down list box, select **Full**.
6. Ensure that **Copy-only Backup** is checked.
7. In the **Select a page** pane, click **Options**.
8. In the **Overwrite media** options, select **Back up to a new media set, and erase all existing backup sets**.
9. In the **New media set name** text box, type **MarketDev Copy Backup**.
10. In the **New media set description** text box, type **MarketDev Copy Backup for Integration Team**.
11. In the **Reliability** options, ensure that **Verify backup when finished** is checked.
12. In the **Set backup compression** drop-down list, click **Compress backup**, and click **OK**.
13. In the Microsoft SQL Server Management Studio window click **OK**.
14. Using Windows Explorer, note the size of the **Copy backup** compared to the **Full backup**.

### Challenge Exercise 5: Partial Backup (Only if time permits)

- **Task 1: Perform a backup of the read-write filegroups on the RateTracking database**
  1. In Solution Explorer, double-click the file **91 – Lab Exercise 5.sql** to open it.
  2. Review the script.
  3. On the **Toolbar** click **Execute**.

## Module 7: Restoring SQL Server 2012 Databases

# Lab 7: Restoring SQL Server 2012 Databases

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_07\_PRJ\10775A\_07\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.



**Note** The setup script for this module is intended to throw an error regarding missing files this is normal.

### Exercise 1: Determine a Restore Strategy

#### ► Task 1: Review the backups contained within the backup file

1. In Solution Explorer, double-click the file **51 – Lab Exercise 1 Task 1.sql** to open it.
2. Review the script.
3. On the **Toolbar** click **Execute**.
4. The BackUpTypeDescription contained in the file are (in order):
  - a. Database
  - b. Database
  - c. Database Differential
  - d. Transaction Log
  - e. Database Differential
  - f. Transaction Log
  - g. Transaction Log

► **Task 2: Determine how the restore should be performed**

1. Restoring a database requires:
  - Latest full database backup (File 2)
  - Most recent differential backup (File 5)
  - All log file backups since the latest differential backup (Files 6,7)
2. With **NORECOVERY** should be specified on all restores.

## Exercise 2: Restore the Database

► **Task 1: Restore the database**

1. In Object Explorer, expand the **Proseware** server, right-click **Databases**, and click **Restore Database**.
2. In the **Source** options, click **Device** then click on the ellipsis button to the right hand side of the **Device** textbox.
3. In the Select backup devices window, click **Add**.
4. Navigate to the file **D:\MSSQLServer\MarketYields.bak** and click **OK**.
5. In the Select backup devices window, click **OK**.
6. In the Restore Database - MarketYields window, in the **Select a page** pane, click **Files**.
7. In the **Restore database files as** options, check **Relocate all files to folder**.
8. In the **Select a page** pane, click **Options**.
9. From the **Recovery state** dropdown, select **RESTORE WITH STANDBY**.
10. In the **Standby file** textbox, type **L:\MKTG\Log\_Standby.bak** and click **OK**.



**Note** If the OK button is not visible you will need to scroll to the bottom of the window, or increase the resolution of your screen.

11. In the Microsoft SQL Server Management Studio window, click **OK**.
12. In Object Explorer expand the **Proseware** server, expand **Databases**, right-click **Databases** and click **Refresh**.
13. The **MarketYields** database should show as **Standby / Read-Only**.

### Challenge Exercise 3: Using STANDBY Mode (Only if time permits)

► **Task 1: Execute queries against the STANDBY database to ensure it is accessible**

1. In Object Explorer, right-click **Proseware** server, and click **New Query**.
2. Type the following query as shown in the snippet below:

```
SELECT * FROM MarketYields.dbo.LogData;
```

3. On the **Toolbar** click **Execute**.



**Note** 25000 rows should be returned.

4. Close the query window. (If prompted to Save changes click **No**).

► **Task 2: Restore another log file, leaving the database in STANDBY mode**

1. In Object Explorer, expand the **Proseware** server, expand **Databases**, right-click the **MarketYields** database and click **Tasks**, click **Restore**, click **Transaction Log**.
2. In the **Restore source** options, click **From file or tape**, click the **ellipsis** button at the right hand side of the **From file or tape** textbox.
3. In the Select backup devices window, click **Add**.
4. Navigate to the file **D:\MSSQLSERVER\MarketYields\_log.bak** and click **OK**.
5. In the Select backup devices window, click **OK**.
6. In the **Select a page** pane, click **Options**.
7. In the **Recovery state** options select **Leave the database in read-only mode**.
8. In the **Standby file** textbox, type **L:\MKTG\Log\_Standby.bak** and click **OK**.
9. In the Microsoft SQL Server Management Studio window, click **Yes**.
10. In the Microsoft SQL Server Management Studio window, click **OK**.
11. In Object Explorer expand the **Proseware** server, expand **Databases**, right-click **Databases** and click **Refresh**.
12. The **MarketYields** database should show as **Standby / Read-Only**.

**MCT USE ONLY. STUDENT USE PROHIBITED**



## Module 8: Importing and Exporting Data

# Lab 8: Importing and Exporting Data

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_08\_PRJ\10775A\_08\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Import the Excel Spreadsheet

#### ► Task 1: Import the data using the Import Wizard

1. In SQL Server Management Studio, in Object Explorer, expand the **Proseware** server, expand **Databases**, right-click the **MarketDev** database, click **Tasks**, and click **Import Data**.
2. In the SQL Server Import and Export Wizard window, click **Next**.
3. In the Choose a Data Source window, select **Microsoft Excel** from the **Data source** drop-down list, and click **Browse**.
4. Navigate to the file **D:\10775A\_Labs\10775A\_08\_PRJ\10775A\_08\_PRJ\Currency.xls** and click **Open**.
5. In the Choose a Data Source window, make sure that the **First row has column names** checkbox is checked, and click **Next**.
6. In the Choose a Destination window, make sure that the **Server name** is **Proseware** and the **Database name** is **MarketDev**, and click **Next**.
7. In the Specify Table Copy or Query window, click **Next**.
8. In the Select Source Tables and Views window, check the checkbox beside **`Currency`**, change the **Destination** to **[DirectMarketing].[Currency]** and click **Edit Mappings**.
9. In the Column Mappings window.
  - a. Change the **Type** for the **CurrencyID** row to **int**.
  - b. Uncheck the **Nullable** column in each row.
  - c. Change the **Size** for the **CurrencyCode** to **3**.
  - d. Change the **Size** for the **CurrencyName** to **50** and click **OK**.

10. In the Select Source Tables and Views window, click **Next**.
11. In the Review Data Type Mapping window, click **Next**.
12. In the Save and Run Package window, click **Next**.
13. In the Complete the Wizard window, click **Finish**.

 **Note** One warning about potential data truncation may occur. This is normal.

14. In the Execution is completed window, click **Close**.
15. In Object Explorer, right-click the **MarketDev** database and click **New Query**.
16. Type the code as shown in the snippet below:

```
SELECT * FROM DirectMarketing.Currency;
```

17. On the **Toolbar**, click **Execute**.

 **Note** 105 currencies should be returned.

## Exercise 2: Import the CSV File

### ► Task 1: Import the CSV file

1. In Solution Explorer, double-click the file **61 – Lab Exercise 2.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.
4. Wait for the query to complete and record the duration of the query. (The duration of the query is shown on the bottom right-hand side of the **Status bar**).

 **Note** A typical execution time on the VM would be approximately 3 minutes.

## Exercise 3: Create and Test an Extraction Package

### ► Task 1: Create and test an extraction package

1. In SQL Server Management Studio, in Object Explorer, expand the **Proseware** server, expand **Databases**, right-click the **MarketDev** database, click **Tasks**, and click **Export Data**.
2. In the SQL Server Import and Export Wizard window, click **Next**.
3. In the Choose a Data Source window, click **Next**.
4. In the Choose a Destination window, in the **Destination** drop down list, choose **Flat File Destination**.

In the **File name** text box type the following location: **D:\MKTG\ProspectsToContact.csv**.

5. Make sure that the **Column names in the first data row** checkbox is checked and click **Next**.

6. In the Specify Table Copy or Query window, click **Write a query to specify the data to transfer**, and click **Next**.
7. In the Provide a Source Query window, type the code as shown in the snippet below, and click **Parse**:

```
SELECT ProspectID, FirstName, LastName, CellPhoneNumber,
WorkPhoneNumber,EmailAddress, LatestContact
FROM Marketing.Prospect
WHERE LatestContact < DATEADD(MONTH,-1,SYSDATETIME())
OR LatestContact IS NULL
ORDER BY ProspectID;
```

8. In the SQL Server Import or Export Wizard window, make sure that the parsing succeeded and click **OK**.
9. In the Provide a Source Query window, click **Next**.
10. In the Configure Flat File Destination window, click **Next**.
11. In the Save and Run Package window, make sure that **Run immediately** and **Save SSIS Package** checkboxes are checked, and that **SQL Server** option button is selected, and click **Next**.
12. In the Save SSIS Package window, type **Weekly Extract of Prospects to Contact** in both the **Name** and **Description** textboxes, and click **Next**.
13. In the Complete the Wizard window, click **Finish**.



**Note** One warning about potential data truncation and one error if the table does not already exist may occur. This is normal.

14. In the **Execution is completed** window, click **Close**.

## Challenge Exercise 4: Compare Loading Performance (Only if time permits)

### ► Task 1: Re-execute load with indexes disabled

1. In Solution Explorer, double-click the file **81 – Lab Exercise 4.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.
4. Wait for the query to complete and record the duration of the query. (The duration of the query is shown on the bottom right-hand side of the **Status bar**).



**Note** A typical execution time on the VM would be approximately 1.2 minutes.

**MCT USE ONLY. STUDENT USE PROHIBITED**

## Module 9: Authenticating and Authorizing Users

# Lab 9: Authenticating and Authorizing Users

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_09\_PRJ\10775A\_09\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Create Logins

#### ► Task 1: Review the requirements

- Review the supplied security requirements in the supporting documentation. Determine the required Windows logins, Windows group logins and SQL logins.

#### ► Task 2: Create the required logins

1. In Solution Explorer, double-click the file **51 – Lab Exercise 1.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.

### Exercise 2: Correct an Application Login Issue

#### ► Task 1: Correct an Application Login Issue

1. In Solution Explorer, double-click the file **61 – Lab Exercise 2.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.

### Exercise 3: Create Database Users

► **Task 1: Review the requirements**

- Review the supplied security requirements in the supporting documentation. Determine the required Windows logins, Windows group logins and SQL logins.

► **Task 2: Create the required database users**

1. In Solution Explorer, double-click the file **71 – Lab Exercise 3.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.

### Challenge Exercise 4: Correct Access to Restored Database (Only if time permits)

► **Task 1: Correct Access to a Restored Database**

1. In Solution Explorer, double-click the file **81 – Lab Exercise 4.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.

## Module 10: Assigning Server and Database Roles

# Lab 10: Assigning Server and Database Roles

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_10\_PRJ\10775A\_10\_PRJ.ssmssl**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Assign Server Roles

#### ► Task 1: Review the requirements

- Review the supplied security requirements in the supporting documentation.

#### ► Task 2: Assign any required server roles

1. In Solution Explorer, double-click the file **51 – Lab Exercise 1.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.

### Exercise 2: Assign Fixed Database Roles

#### ► Task 1: Review the requirements

- Review the supplied security requirements in the supporting documentation.

#### ► Task 2: Assign any required fixed database roles

1. In Solution Explorer, double-click the file **61 – Lab Exercise 2.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.

### Exercise 3: Create and Assign User-defined Database Roles

- ▶ **Task 1: Review the requirements**
  - Review the supplied security requirements in the supporting documentation.
- ▶ **Task 2: Create and assign any required user-defined database roles**
  1. In Solution Explorer, double-click the file **71 – Lab Exercise 3.sql** to open it.
  2. Review the T-SQL script.
  3. On the **Toolbar** click **Execute**.

### Challenge Exercise 4: Check Role Assignments (Only if time permits)

- ▶ **Task 1: Check the role assignments for Darren Parker**
  1. In Solution Explorer, double-click the file **81 – Lab Exercise 4.sql** to open it.
  2. Review the T-SQL script.
  3. On the **Toolbar** click **Execute**.



## Module 11: Authorizing Users to Access Resources

# Lab 11: Authorizing Users to Access Resources

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_11\_PRJ\10775A\_11\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Assign Schema-level Permissions

#### ► Task 1: Review the security requirements that have been updated from the previous module

1. Review the supplied security requirements in the supporting documentation.
2. Determine the permissions that should be assigned at the schema level. (Note: that a sample solution is shown in Task 2).

#### ► Task 2: Assign the required permissions

1. In Solution Explorer, double-click the file **51 – Lab Exercise 1.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.

### Exercise 2: Assign Object-level Permissions

#### ► Task 1: Review the security requirements

1. Review the supplied security requirements in the supporting documentation.
2. Determine the permissions that should be assigned at the object level. (Note: that a sample solution is shown in Task 2).

► **Task 2: Assign the required permissions**

1. In Solution Explorer, double-click the file **61 – Lab Exercise 2.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.

**Challenge Exercise 3: Test Permissions (Only if time permits)**

► **Task 1: Design and execute a test**

1. In Solution Explorer, double-click the file **71 – Lab Exercise 3a.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.
4. In Solution Explorer, double-click the file **72 – Lab Exercise 3b.sql** to open it.
5. Review the T-SQL script.
6. On the **Toolbar** click **Execute**.



**Note** An error is returned as April.Reagan does not have permission to select the rows from the DirectMarketing.Competitor table.

## Module 12: Auditing SQL Server Environments

# Lab 12: Auditing SQL Server Environments

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_12\_PRJ\10775A\_12\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Determine Audit Configuration and Create Audit

#### ► Task 1: Review the requirements

- Review the supplied requirements in the supporting documentation for the exercise, noting requirements that relate to server audits.

#### ► Task 2: Create the server audit

1. Determine the configuration of the required server audit.
2. Using Windows Explorer, create a folder **C:\Audit**.
3. Using Windows Explorer, create a folder **C:\Audit\AuditLog**.
4. In Object Explorer, expand the **Proseware** server, and expand **Security**.
5. Right-click **Audits** and click **New Audit**.
6. In the Create Audit window, type **Proseware Compliance Audit** in the **Audit name** textbox.
7. In the **Queue delay (in milliseconds)** textbox, type **2000**.
8. For **Audit Log Failure**, select the **Shut down server** option.
9. In the **File path** textbox, type **C:\Audit\AuditLog**.
10. Uncheck the **Unlimited** checkbox for **Maximum file size**.
11. In the **Maximum file size** drop-down list type **1**.

12. Click the **GB** option button for **Maximum file size** and click **OK**.
13. In Object Explorer, expand **Audits**, right-click **Proseware Compliance Audit** and click **Enable Audit**.
14. In the Enable Audit window, click **Close**.

## Exercise 2: Create Server Audit Specifications

### ► Task 1: Review the requirements

- Review the supplied requirements in the supporting documentation for the exercise, noting requirements that relate to server audit specifications.

### ► Task 2: Create the server audit specifications

1. Determine the configuration of the required server audit specifications.
2. In Object Explorer, right-click **Server Audit Specifications**, and click **New Server Audit Specification**.
3. In the Create Server Audit Specification window, type **Proseware Compliance Server Audit Specification** in the **Name** textbox.
4. From the **Audit** drop-down list, select **Proseware Compliance Audit**.
5. In the **Actions** list, in row 1, click the **Audit Action Type** drop-down, and click **FAILED\_LOGIN\_GROUP**.
6. In the **Actions** list, in row 2, click the **Audit Action Type** drop-down, and click **SERVER\_PRINCIPAL\_CHANGE\_GROUP**.
7. In the **Actions** list, in row 3, click the **Audit Action Type** drop-down, and click **SERVER\_ROLE\_MEMBER\_CHANGE\_GROUP**.
8. In the **Actions** list, in row 4, click the **Audit Action Type** drop-down, and click **LOGIN\_CHANGE\_PASSWORD\_GROUP**, and click **OK**.
9. In Object Explorer, expand **Server Audit Specifications**, right-click **Proseware Compliance Server Audit Specification**, and click **Enable Server Audit Specification**.
10. In the Enable Server Audit Specification window, click **Close**.

## Exercise 3: Create database audit specifications

### ► Task 1: Review the requirements

- Review the supplied requirements in the supporting documentation for the exercise, noting requirements that relate to database audit specifications.

### ► Task 2: Create the database audit specifications

1. Determine the configuration of the required database audit specifications.
2. In Object Explorer, expand the **Proseware** server, expand **Databases**, expand the **MarketDev** database, expand **Security**, and expand **Database Audit Specifications**.
3. Right-click **Database Audit Specifications**, and click **New Database Audit Specifications**.

4. In the Create Database Audit Specification window, type **Proseware Compliance MarketDev Audit Specification** in the **Name** textbox.
5. From the **Audit** drop-down list, select **Proseware Compliance Audit**.
6. In the **Actions** list, in row 1, click the **Audit Action Type** drop-down, and click **BACKUP\_RESTORE\_GROUP**.
7. In the **Actions** list, in row 2, click the **Audit Action Type** drop-down, and click **DATABASE\_OWNERSHIP\_CHANGE\_GROUP**.
8. In the **Actions** list, in row 3, click the **Audit Action Type** drop-down, and click **DATABASE\_PERMISSION\_CHANGE\_GROUP**.
9. In the **Actions** list, in row 4, click the **Audit Action Type** drop-down, and click **DATABASE\_PRINCIPAL\_CHANGE\_GROUP**.
10. In the **Actions** list, in row 5, click the **Audit Action Type** drop-down, and click **DATABASE\_ROLE\_MEMBER\_CHANGE\_GROUP**.
11. In the **Actions** list, in row 6, click the **Audit Action Type** drop-down, and click **EXECUTE**.
12. In row 6, click the **Object Class** drop-down list and click **OBJECT**.
13. In row 6, click the **ellipsis** in the **Object Name** column.
14. In the Select Objects window, click **Browse**.
15. In the **Matching Objects** pane, check the box beside **[Marketing].[MoveCampaignBalance]**, and click **OK**.
16. In the Select Objects window, click **OK**.
17. In row 6, click the **ellipsis** in the **Principal Name** column.
18. In the Select Objects window, click **Browse**.
19. In the **Matching Objects** pane, check the box beside **[public]**, and click **OK**.
20. In the Select Objects window, click **OK**.
21. In the **Actions** list, in row 7, click the **Audit Action Type** drop-down, and click **UPDATE**.
22. In row 7, click the **Object Class** drop-down list and click **OBJECT**.
23. In row 7, click the **ellipsis** in the **Object Name** column.
24. In the Select Objects window, click **Browse**.
25. In the **Matching Objects** pane, check the box beside **[Marketing].[CampaignBalance]**, and click **OK**.
26. In the Select Objects window, click **OK**.
27. In row 7, click the **ellipsis** in the **Principal Name** column.
28. In the Select Objects window, click **Browse**.
29. In the **Matching Objects** pane, check the box beside **[public]**, and click **OK**.
30. In the Select Objects window, click **OK**.

31. In the Create Database Audit Specification window, click **OK**.
32. In Object Explorer, right-click **Proseware Compliance MarketDev Audit Specification** and click **Enable Database Audit Specification**.
33. In the Enable Database Audit Specification window, click **Close**.

### Challenge Exercise 4: Test Audit Functionality (Only if time permits)

#### ► Task 1: Execute the workload script

1. In Solution Explorer, double-click the file **81 – Lab Exercise 4a.sql** to open it.
2. Review the T-SQL script. Note any actions that you would expect to be audited. In Task 2 you will see the actual list of audited actions.
3. On the **Toolbar** click **Execute**.

#### ► Task 2: Review the captured audit details

1. In Object Explorer, expand the **Proseware** server, expand **Security**, expand **Audits**.
2. Right-click the **Proseware Compliance Audit**, and click **View Audit Logs**.
3. In the Log File Viewer – Proseware window, review the captured events and compare them to your list of expected events from Task 1.



**Note** Make sure you scroll to the list of events to the right to see all available columns.

4. In the Log File Viewer – Proseware window click **Close**.
5. In Solution Explorer, double-click the file **82 – Lab Exercise 4b.sql** to open it.
6. Review the T-SQL script.
7. On the **Toolbar** click **Execute**. A list of audited events will be returned.

## Module 13: Automating SQL Server 2012 Management

# Lab 13: Automating SQL Server Management

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_13\_PRJ\10775A\_13\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Create a Data Extraction Job

#### ► Task 1: Create the required job

1. In Object Explorer, expand the **Proseware** server, expand **SQL Server Agent**, right-click **Jobs** and click **New Job**.
2. In the New Job window, in the **Name** textbox type **Extract Uncontacted Prospects**.
3. In the **Select a page** pane, click **Steps**, and then click **New**.
4. In the **Step name** textbox, type **Execute Prospect Extraction Package**.
5. From the **Type** drop-down list, select **SQL Server Integration Services Package**.
6. In the **General** tab, from the **Package source** drop-down, select **SQL Server**.
7. In the **Server** textbox type **Proseware**, and click the **ellipsis** beside the **Package** textbox.
8. In the Select an SSIS Package window, click **Weekly Extract of Prospects to Contact** and click **OK**.
9. In the New Job Step window, click **OK**.
10. In the New Job window, click **OK**.

► **Task 2: Test that the job executes without error**

1. In Object Explorer, expand the **Proseware** server, expand **SQL Server Agent**, and expand **Jobs**.
2. Right-click **Jobs** and click **Refresh**.
3. Right-click **Extract Uncontacted Prospects**, and click **Start Job at Step**.
4. In the Start Jobs - Proseware window, make sure that the job succeeds and click **Close**.

## Exercise 2: Schedule the Data Extraction Job

► **Task 1: Schedule the data extraction job**

1. In Object Explorer, expand the **Proseware** server, expand **SQL Server Agent**, and expand **Jobs**.
2. Right-click **Jobs** and click **Refresh**.
3. Right-click **Extract Uncontacted Prospects**, and click **Properties**.
4. On the **Select a page** pane, click **Schedules**, and then click **New**.
5. In the New Job Schedule window, in the **Name** textbox type **Each Monday Morning at 8:30AM**.
6. Check the **Monday** checkbox; uncheck the **Sunday** checkbox.
7. In the **Occurs once at** textbox, change the time to **8:30:00AM** and press the Tab key. Note the contents of the **Description** textbox and click **OK**.
8. In the Job Properties – Extract Uncontacted Prospects window, click **New**.
9. In the New Job Schedule window, in the **Name** textbox type **Each Tuesday Evening at 6:30PM**.
10. Check the **Tuesday** checkbox; uncheck the **Sunday** checkbox.
11. In the **Occurs once at** textbox, change the time to **6:30:00PM** and hit the Tab key. Note the contents of the **Description** textbox and click **OK**.
12. In the Job Properties – Extract Uncontacted Prospects window, click **OK**.

## Challenge Exercise 3: Troubleshoot a Failing Job (Only if time permits)

► **Task 1: Troubleshoot the failing job**

1. In Object Explorer, expand the **Proseware** server, expand **SQL Server Agent**, and expand **Jobs**.
2. Right-click **Jobs** and click **Refresh**.
3. Right-click **Extract Long Page Loads**, and click **View History**.
4. In the Log File Viewer – Proseware window, click the **plus** sign in a row that has a failing execution to expand the job steps. Note that step ID **2** has failed.
5. Click on the row for step ID **2**, and scroll through the **Selected row details** pane to find the error. Note that the error is caused by a reference to an invalid object name **Marketing.RecentLongPageLoads**.
6. Click **Close** to close the Log File Viewer – Proseware window.
7. Right-click **Extract Long Page Loads**, and click **Properties**.
8. In the **Select a page** pane, click **Steps**.



9. In the **Job step** list, click the row for Step 2 and click **Edit**.
10. Review the command that is being executed and note that **Marketing.RecentLongPageLoads** appears to be a table or view that the web log rows are being inserted into.
11. In Object Explorer, expand **Databases**, expand **MarketDev**, and expand **Tables**. Note that the name of the table should be **Marketing.RecentLongPageLoad**.
12. In the Job Step Properties – Copy Recent Long Page Loads window change the name of the table in the **Command** textbox from **Marketing.RecentLongPageLoads** to **Marketing.RecentLongPageLoad** and click **OK**.
13. In the Job Properties – Extract Long Page Loads window, click **OK**.
14. In Object Explorer, right-click the **Extract Long Page Loads** job and click **Start Job at Step**.
15. In the Start Job on 'Proseware' window, click **Start**.
16. In the Start Job on 'Proseware' window, make sure the job executed successfully and click **Close**.
17. In Object Explorer, right-click **Extract Long Page Loads**, and click **Properties**.
18. In the **Select a page** pane, click **Schedules** and note the difference between the name of the schedule and the description of the schedule, then click **Edit**.
19. From the **Occurs** drop-down list box, click **Weekly**.
20. Check the **Monday** checkbox and review the remaining settings, then click **OK**.
21. In the **Schedule list**, note that the schedule name now relates to the schedule description.
22. In the Job Properties – Extract Long Page Loads window, click **OK**.

**MCT USE ONLY. STUDENT USE PROHIBITED**

## Module 14: Configuring Security for SQL Server Agent

# Lab 14: Configuring Security for SQL Server Agent

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_14\_PRJ\10775A\_14\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Troubleshoot Job Execution Failure

#### ► Task 1: Troubleshoot job execution failure

1. Review the previous actions taken by Terry Adams as detailed in the supporting documentation for the exercise.
2. In Object Explorer, expand the **Proseware** server, expand **SQL Server Agent**, expand **Jobs**, right-click **Extract Uncontacted Prospects** and click **View History**.
3. In the Log File Viewer - Proseware window, click the **plus** sign in a failing row to expand the job steps.
4. Click the row for Step ID **1**. In the **Selected row details** pane, scroll the pane until you locate the error.



**Note** The error indicates that Non-SysAdmins have been denied permission to run DTS Execution job steps without a proxy account. The step failed.

5. You have determined that a proxy account is needed for the job step. Close the Log File Viewer - Proseware window.

## Exercise 2: Resolve the Security Issue

### ► Task 1: Create and assign proxy account

1. In Object Explorer, expand the **Proseware** server, expand **Security**, and expand **Credentials**.
2. Right-click **Credentials** and click **New Credential**.
3. In the New Credential window, in the **Credential name** text box, type **ExtractIdentity**.
4. In the **Identity** textbox, type **MIA-SQL1\ExtractUser**.
5. In the **Password** and **Confirm password** textboxes, type **Pa\$\$w0rd**, and click **OK**.
6. In Object Explorer, expand **SQL Server Agent**, right-click **Proxies**, and click **New Proxy**.
7. In the New Proxy Account window, in the **Proxy name** textbox, type **ExtractionProxy**.
8. In the **Credential name** textbox, type **ExtractIdentity**.
9. In the **Active to the following subsystems** list, check the box beside **SQL Server Integration Services Package**.
10. In the **Select a page** pane, click **Principals**, and click **Add**.
11. In the **Available principals** list, check the box beside **PromoteApp** login, and click **OK**.
12. In the New Proxy Account window, click **OK**.
13. In Object Explorer, expand **Jobs**, right-click **Extract Uncontacted Prospects** and click **Properties**.
14. In the Job Properties - Extract Uncontacted Prospects window, in the **Select a page** pane click **Steps**, then click **Edit**.
15. In the **Run as** drop-down list, click **ExtractionProxy**, and click **OK**.
16. In the Job Properties - Extract Uncontacted Prospects window, click **OK**.

### ► Task 2: Test to see if all problems have been resolved

1. In Object Explorer, right-click the **Extract Uncontacted Prospects** job, and click **Start Job at Step**.
2. In the Start Jobs - Proseware window, note that the job still fails, and click **Close**.
3. In Object Explorer, right-click the **Extract Uncontacted Prospects** job, and click **View History**.
4. In the Log File Viewer - Proseware window, click the **plus** sign beside the top entry in the list to expand the job steps.
5. Click in the row for Step ID **1**, in the **Selected row details** pane, scroll down to find the error.

 **Note** A SQL Statement is now failing because of a login issue.

6. You have resolved the original problem. If you have time you should continue to Exercise 3 to resolve the remaining problem.

## Challenge Exercise 3: Perform Further Troubleshooting (Only if time permits)

### ► Task 1: Perform further troubleshooting

1. In the Log File Viewer - Proseware window, in the **Select row details** pane, read the detail of the error message, determine the cause of the error and close the window.



**Note** The most important error is Login failed for user 'MIA-SQL1\ExtractUser'. Even though a Windows credential is required for the SSIS job to access the file system to write the extracted file, the credential also needs to be able to connect to SQL Server to retrieve the data from the Marketing.Prospect table. You need to create a login for the Windows user, create a database user for the login and then assign SELECT permission on the Marketing.Prospect table to the credential.

2. In Solution Explorer, double-click the file **71 – Lab Exercise 3.sql** to open it.
3. Review the T-SQL script.
4. On the **Toolbar** click **Execute**.
5. In Object Explorer, right-click the **Extract Uncontacted Prospects** job, and click **Start Job at Step**.
7. In the Start Jobs - Proseware window, note that the job still fails, and click **Close**.
8. In Object Explorer, right-click the **Extract Uncontacted Prospects** job, and click **View History**.
9. In the Log File Viewer - Proseware window, click the **plus** sign beside the top entry in the list to expand the job steps.
10. Click in the row for Step ID **1**, in the **Selected row details** pane, scroll down to find the error.



**Note** ExtractUser does not have permissions on the D:\MKTG folder.

11. In Windows Explorer, navigate to the folder **D:\MKTG**, right-click the **MKTG** folder and click **Properties**.
12. In the MKTG Properties window, click on the **Security** tab, click **Edit**.
13. In the Permissions for MKTG window, click **Add**.
14. In the Select Users, Computers, Services Accounts, or Groups window, in the **Enter the object names to select** textbox type **MIA-SQL1\ExtractUser** and click **OK**.
15. In the Permissions for MKTG window, check the **Allow** checkbox for the **Modify** row and click **OK**.
16. In the MKTG Properties window, click **OK**.
17. In Object Explorer, right-click the **Extract Uncontacted Prospects** job, and click **Start Job at Step**.
18. In the Start Jobs - Proseware window, note that the job now completes successfully, and click **Close**.

**MCT USE ONLY. STUDENT USE PROHIBITED**

## Module 15: Monitoring SQL Server 2012 with Alerts and Notifications

### Lab 15: Monitoring SQL Agent Jobs with Alerts and Notifications

#### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_15\_PRJ\10775A\_15\_PRJ.ssmssln**.
7. In **Solution Explorer**, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

#### Exercise 1: Configure Database Mail

##### ► Task 1: Configure database mail

1. Review the database mail configuration parameters in the supporting documentation for the exercise.
2. In Object Explorer, expand the **Proseware** server, expand **Management**, right-click **Database Mail**, and click **Configure Database Mail**.
3. In the Welcome to Database Mail Configuration Wizard window, click **Next**.
4. In the Select Configuration Task window, click **Next**.
5. In the Microsoft SQL Server Management Studio window, click **Yes**.
6. In the New Profile window, in the **Profile name** textbox type **Proseware SQL Server Agent Profile**, and click **Add**.
7. In the New Database Mail Account window, in the **Account name** textbox, type **Proseware Administrator**.
8. In the **E-mail address** textbox, type **prosewaresqladmin@adventureworks.com**.
9. In the **Display name** textbox, type **Proseware SQL Server Administrator**.
10. In the **Reply e-mail** textbox, type **prosewaresqladmin@adventureworks.com**.
11. In the **Server name** textbox type **mailserver.adventureworks.com**, and click **OK**.

12. In the New Profile window, click **Add**.
13. In the New Database Mail Account window, in the **Account name** textbox, type **AdventureWorks Administrator**.
14. In the **E-mail address** textbox, type **adventureworkssqladmin@adventureworks.com**.
15. In the **Display name** textbox, type **AdventureWorks SQL Server Administrator**.
16. In the **Reply e-mail** textbox, type **adventureworkssqladmin@adventureworks.com**.
17. In the **Server name** textbox type **mailserver.adventureworks.com**, and click **OK**.
18. In the New Profile window, click **Next**.
19. In the Manage Profile Security window, check the box in the **Public** column and change the value in the **Default Profile** column to **Yes**.
20. Click on the **Private Profiles** tab, and from the **User name** drop-down list, select **ADVENTUREWORKS\pwservice**.
21. Check the box in the **Access** column, and change the value in the **Default Profile** column to **Yes**, then click **Next**.
22. In the Configure System Parameters window, change the **Maximum File Size (Bytes)** to **4194304**, and click **Next**.
23. In the Complete the Wizard window, click **Finish**.
24. In the Configuring window, click **Close**.

► **Task 2: Test that database mail operates**

1. In Object Explorer, right-click **Database Mail** and click **Send Test E-Mail**.
2. In the Send Test E-Mail from MIA-SQL1\MKTG window, in the **To** textbox type **prosewaresqladmin@adventureworks.com**, and click **Send Test E-Mail**.
3. In the Database Mail Test E-Mail window, note the description and click **OK**.
4. In Solution Explorer, double-click the file **51 – Lab Exercise 1.sql** to open it.
5. Review the T-SQL script.
6. On the **Toolbar** click **Execute**.



**Note** An email with the subject Database Mail Test should be returned by the query.

## Exercise 2: Implement Notifications

► **Task 1: Review the requirements**

- Review the supplied requirements in the supporting documentation for the exercise. In particular, note any required operators.



► **Task 2: Configure the required operators**

1. In Object Explorer, expand **SQL Server Agent**, right-click **Operators** and click **New Operator**.
2. In the New Operator window, in the **Name** textbox, type **IT Support Fail-safe Operator**.
3. In the **Pager e-mail name**, type **itsupport.pager@adventureworks.com**.
4. In the **Pager on duty schedule**, check **Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday**. Change the **Workday begin** time to **12:00:00 AM** and **Workday end** to **11:59:59 PM** for every row, and click **OK**.
5. In Object Explorer, right-click **Operators** and click **New Operator**.
6. In the New Operator window, in the **Name** textbox, type **Jeff Hay**.
7. In the **Pager e-mail name**, type **jeff.hay.pager@adventureworks.com**.
8. In the **Pager on duty schedule**, check **Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday**. Change the **Workday begin** time to **12:00:00 AM** and **Workday end** to **11:59:59 PM** for every row, and click **OK**.
9. In Object Explorer, right-click **Operators** and click **New Operator**.
10. In the New Operator window, in the **Name** textbox, type **Palle Petersen**.
11. In the **Pager e-mail name**, type **palle.petersen.pager@adventureworks.com**.
12. In the **Pager on duty schedule**, check **Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday**. Change the **Workday begin** time to **12:00:00 AM** and **Workday end** to **11:59:59 PM** for every row, and click **OK**.

► **Task 3: Configure SQL Server Agent Mail**

1. In Object Explorer, right-click **SQL Server Agent**, and click **Properties**.
2. In the **Select a page** pane click **Alert System**. In the **Mail session** options, click **Enable mail profile**.
3. From the **Mail profile** drop-down list, select **Proseware SQL Server Agent Profile**.
4. In the **Fail-safe operator** options, check **Enable fail-safe operator**.
5. In the **Operator** drop-down list, select **IT Support Fail-safe Operator**.
6. In the **Notify using options**, check **Pager**, and click **OK**.
7. In Object Explorer, right-click **SQL Server Agent** and click **Restart**.
8. In the SQL Server Management Studio window, click **Yes**.

► **Task 4: Configure and Test Notifications in SQL Server Agent Jobs**

1. In Object Explorer, expand **SQL Server Agent**, expand **Jobs**, right-click **Backup Log TestAlertDB** and click **Properties**.
2. In the **Select a page** pane, click **Notifications**.
3. In the **Actions to perform when the job completes** option, check **Page**.
4. In the **Page** drop-down list, select **Jeff Hay**.

5. In the **Page** actions drop-down list, select **When the job completes**, and click **OK**.
6. In Object Explorer, right-click **Job that Fails** and click **Properties**.
7. In the **Select a page** pane, click **Notifications**.
8. In the **Actions to perform when the job completes** option, check **Page**.
9. In the **Page** drop-down list, select **Palle Petersen**.
10. In the **Page** actions drop-down list, select **When the job fails**, and click **OK**.
11. In Object Explorer, right-click **Job that Succeeds** and click **Properties**.
12. In the **Select a page** pane, click **Notifications**.
13. In the **Actions to perform when the job completes** option, check **Page**.
14. In the **Page** drop-down list, select **Palle Petersen**.
15. In the **Page** actions drop-down list, select **When the job fails**, and click **OK**.
16. In Object Explorer, right-click **Backup Log TestAlertDB** and click **Start Job at Step**.
17. In the Start Jobs - Proseware window, click **Close**.
18. In Object Explorer, right-click **Job That Fails**, and click **Start Job at Step**.
19. In the Start Jobs - Proseware window, note that the job failed and click **Close**.
20. In Object Explorer, right-click **Job That Succeeds** and click **Start Job at Step**.
21. In the Start Jobs - Proseware window, click **Close**.
22. In Solution Explorer, double-click the file **51 – Lab Exercise 1.sql** to open it.
23. Review the T-SQL script.
24. On the **Toolbar** click **Execute**.



**Note** Two additional emails should now appear in the list, an email from the backup job sent to Jeff Hay and an email from the job that fails sent to Palle Petersen. No email should be sent for the job called "Job That Succeeds".

### Challenge Exercise 3: Implement Alerts (Only if time permits)

#### ► Task 1: Configure and Test Alerts

1. Review the supporting documentation for the alerting requirements.
2. In Object Explorer, right-click **Alerts** and click **New Alert**.
3. In the New Alert window, in the **Name** textbox, type **Error Severity 17 Alert**.
4. In the **Severity** drop-down list, click **017 – Insufficient Resources**.
5. In the **Select a page** pane, click **Response**.
6. Check **Notify operators**, check all checkboxes in the **Pager** column and click **OK**.
7. In Object Explorer, right-click **Alerts** and click **New Alert**.

8. In the New Alert window, in the **Name** textbox, type **Error Severity 18 Alert**.
9. In the **Severity** drop-down list, click **018 – Nonfatal Internal Error**.
10. In the **Select a page** pane, click **Response**.
11. Check **Notify operators**, check all checkboxes in the **Pager** column and click **OK**.
12. In Object Explorer, right-click **Alerts** and click **New Alert**.
13. In the New Alert window, in the **Name** textbox, type **Transaction Log Full Alert**.
14. Click the **Error number** option, and in the **Error number** textbox type **9002**.
15. In the **Select a page** pane, click **Response**.
16. Check **Notify operators**, check all checkboxes in the **Pager** column and click **OK**.
17. In Solution Explorer, double-click the file **71 – Lab Exercise 3.sql** to open it.
18. Review the T-SQL script.
19. On the **Toolbar** click **Execute**.



**Note** Executing this script will result in an error, indicating message 9002.

20. In Solution Explorer, double-click the file **51 – Lab Exercise 1.sql** to open it.
21. Review the T-SQL script.
22. On the **Toolbar** click **Execute**.



**Note** Additional emails should be listed related to the SQL Server alert system.

**MCT USE ONLY. STUDENT USE PROHIBITED**

## Module 16: Performing Ongoing Database Maintenance

# Lab 16: Performing Ongoing Database Maintenance

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_16\_PRJ\10775A\_16\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Check Database Integrity Using DBCC CHECKDB

#### ► Task 1: Check the consistency of the databases on the Proseware instance

1. In Solution Explorer, double-click the file **51 – Lab Exercise 1a.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.



**Note** Errors will be returned from the CoreAdmin database.

#### ► Task 2: Correct any issues found

1. In Solution Explorer, double-click the file **52 – Lab Exercise 1b.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.



**Note** Errors will be returned from the CoreAdmin database during this process.

## Exercise 2: Correct Index Fragmentation

- ▶ **Task 1: Review the fragmentation of indexes in the MarketDev database to determine which indexes should be defragmented and which indexes should be rebuilt**
  1. In Solution Explorer, double-click the file **61 – Lab Exercise 2a.sql** to open it.
  2. Review the T-SQL script.
  3. On the **Toolbar** click **Execute**.
- ▶ **Task 2: Defragment indexes as determined**
  1. In Solution Explorer, double-click the file **62 – Lab Exercise 2b.sql** to open it.
  2. Review the T-SQL script.
  3. On the **Toolbar** click **Execute**.
- ▶ **Task 3: Rebuild indexes as determined**
  1. In Solution Explorer, double-click the file **63 – Lab Exercise 2c.sql** to open it.
  2. Review the T-SQL script.
  3. On the **Toolbar** click **Execute**.

## Exercise 3: Create a Database Maintenance Plan

- ▶ **Task 1: Create the required database maintenance plan**
  1. In Object Explorer, expand **Proseware**, expand **Management**, right-click **Maintenance Plans**, and click **Maintenance Plan Wizard**.
  2. In the Maintenance Plan Wizard window, click **Next**.
  3. In the Select Plan Properties window, in the **Name** textbox, type **Proseware Weekly Maintenance**. Note the available scheduling options and click **Change**.
  4. In the New Job Schedule window, in the **Name** textbox, type **6PM Every Sunday**.
  5. In the **Occurs once at** textbox, change the time to **6PM**, and click **OK**.
  6. In the Select Plan Properties window, click **Next**.
  7. Review the available options and check **Rebuild index** and **Check Database Integrity**, and then click **Next**.
  8. In the Select Maintenance Task Order window, click **Next**.
  9. In the Define Database Check Integrity Task window, from the **Databases** drop-down list, select **All databases**, and click **OK** then click **Next**.
  10. In the Define Rebuild Index Task window, from the **Databases** drop-down list, check the **MarketDev** database, and click **OK**.
  11. In the **Free space options**, click **Change free space per page to:**.
  12. In the **Change free space per page to:** textbox, type **10**.

13. Check **Keep index online while reindexing**, and click **Next**.
14. In the Select Report Options window, in the **folder location** textbox, type **L:\MKTG**, and click **Next**.
15. In the Complete the Wizard window, click **Finish**.
16. In the Maintenance Plan Wizard Progress window, click **Close**.

### Challenge Exercise 4: Investigate Table Lock Performance (Only if time permits)

#### ► Task 1: Execute DBCC CHECKDB using database snapshots

1. In Solution Explorer, double-click the file **81 – Lab Exercise 4a.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.



**Note** Note the time taken to execute. A typical duration would be 42 seconds.

#### ► Task 2: Execute DBCC CHECKDB using table locks

1. In Solution Explorer, double-click the file **82 – Lab Exercise 4b.sql** to open it.
2. Review the T-SQL script.
3. On the **Toolbar** click **Execute**.



**Note** Note the time taken to execute. A typical duration would be 7 seconds.

**MCT USE ONLY. STUDENT USE PROHIBITED**



## Module 17: Tracing Access to SQL Server 2012

# Lab 17: Tracing Access to SQL Server 2012

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_17\_PRJ\10775A\_17\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Capture a Trace Using SQL Server Profiler

#### ► Task 1: Create and start a suitable SQL Server Profiler trace

1. In Microsoft SQL Server Management Studio, on the **Tools** menu, click **SQL Server Profiler**.
2. In the Connect to Server window, click **Connect**.
3. In the **Trace name**, type **Proseware Trace**.
4. In the **Use the template** drop-down, select **Tuning**.
5. Check the **Save to file** option.
6. In the Save As window, navigate to the desktop, and click **Save**.
7. Uncheck the **Enable file rollover** option.
8. Change the **Set maximum file size (MB)** option to **500**.
9. Click the **Events Selection** tab. Note the selected events.
10. Click on the **Database Name** column heading to create a filter.
11. Expand the **Like** node and enter **MarketDev** and click **OK**.
12. In the Trace Properties window, click **Run**.
13. From the **Window** menu, uncheck **Auto scroll**.

► **Task 2: Execute the workload**

1. In Solution Explorer, double-click the file **51 – Lab Exercise 1.sql** to open it.
2. Review the **T-SQL** script.
3. From the **Query** menu, select **Query Options**.
4. In the Query Options window, click **Grid**. In the **Grid** options, check **Discard results after execution**.
5. Click **OK** to close the Query Options window.
6. On the **Toolbar** click **Execute**.
7. Wait for the query to complete.



**Note** The query is complete when Query executed successfully appears below the Messages tab. No output messages will be seen.

► **Task 3: Stop the trace**

- In SQL Server Profiler, on the **Toolbar**, click the **Stop Selected Trace** icon, and close **SQL Server Profiler**.

## Exercise 2: Analyze a Trace Using Database Engine Tuning Advisor

► **Task 1: Analyze the captured trace in Database Engine Tuning Advisor**

1. In Microsoft SQL Server Management Studio, from the **Tools** menu, click **Database Engine Tuning Advisor**.
2. In the Connect to Server window, click **Connect**.
3. Maximize the Database Engine Tuning Advisor window.
4. In the **Workload** group box, ensure that **File** is selected, and click the **Browse for a Workload File** button.
5. Browse to the **Desktop** folder, select the **Proseware Trace.trc** file and click **Open**.
6. In the **Database for workload analysis** drop-down, select **MarketDev**.
7. In the **Select databases and tables to tune** list, check the **MarketDev** database.
8. From the **Toolbar**, click the **Start Analysis** button to start the tuning analysis.

► **Task 2: Review the suggested modifications**

1. Observe the tuning progress and when the analysis is complete, note the recommendations. The exact recommendations will vary but it is likely you will see both index and statistics recommendations.
2. Scroll the recommendations output to the right and note the hyperlinks to sample code.
3. Click each of the recommendations in turn and note the suggested index or statistics structures.
4. Close Database Engine Tuning Advisor.

### Challenge Exercise 3: Configure SQL Trace (Only if time permits)

► **Task 1: Create a script that uses SQL Trace procedures to implement the same type of capture as you performed in Exercise 1 but with a different trace name**

1. In Microsoft SQL Server Management Studio, from the **Tools** menu, click **SQL Server Profiler**.
2. In the Connect to Server window, click **Connect**.
3. In the **Trace name**, type **ProsewareTrace2**.
4. In the **Use the template** drop-down, select **Tuning**.
5. Check the **Save to file** option.
6. In the Save As window, navigate to the desktop, and click **Save**.
7. Uncheck the **Enable file rollover** option.
8. Change the **Set maximum file size (MB)** option to **500**.
9. Click the **Events Selection** tab. Note the selected events.
10. Click on the **Database Name** column heading to create a filter.
11. Expand the **Like** node and enter **MarketDev** and click **OK**.
12. In the Trace Properties window, click **Run**.
13. From the **Toolbar**, click the **Stop Selected Trace** icon.
14. From the **File** menu, click **Export**, click **Script Trace Definition**, click **For SQL Server 2005 – SQL11**.
15. Navigate to the **Desktop** and in the **File name** text box, type **ProsewareTrace2** and click **Save**.
16. In the SQL Server Profiler window, click **OK**.
17. Close SQL Server Profiler.

► **Task 2: Test that the script works as expected by using the same workload**

1. In Solution Explorer, from the **File** menu, click **Open**, click **File**.
2. In the Open File window, navigate to the **Desktop**, click **Proseware Trace2.sql** and click **Open**.
3. In the first row that begins with an exec command, change the string **InsertFileNameHere** to **D:\MKTG\ProsewareTrace2**.
4. On the **Toolbar** click **Execute** to start the trace.



**Note** Record the Trace ID value that is returned.

5. In Solution Explorer, double-click the file **51 – Lab Exercise 1.sql** to open it.
6. Review the T-SQL script.
7. From the **Query** menu, select **Query Options**.
8. In the Query Options window, click **Grid**. In the **Grid** options, check **Discard results after execution**.

9. Click **OK** to close the Query Options window.
10. On the **Toolbar** click **Execute**.
11. Wait for the query to complete.



**Note** The query is complete when Query executed successfully appears below the Messages tab. No output messages will be seen.

12. In Solution Explorer, double-click the file **71 – Lab Exercise 3.sql** to open it.
13. Review the T-SQL script and replace the value of the **@TraceID** variable with the value you recorded earlier in this task.
14. On the **Toolbar** click **Execute**.

► **Task 3: Analyze the new captured output and note if the suggested changes are identical to those suggested in Exercise 2**

1. In Microsoft SQL Server Management Studio, from the **Tools** menu, click **Database Engine Tuning Advisor**.
2. In the Connect to Server window, click **Connect**.
3. Maximize the Database Engine Tuning Advisor window.
4. In the **Workload** group box, ensure that **File** is selected, and click the **Browse for a Workload File** button.
5. Browse to the **D:\MKTG** folder, select the **ProsewareTrace2.trc** file and click **Open**.
6. In the **Database for workload analysis** drop-down, select **MarketDev**.
7. In the **Select databases and tables to tune** list, check the **MarketDev** database.
8. From the **Toolbar**, click the **Start Analysis** button to start the tuning analysis.
9. When the analysis is complete, compare the results to the results you saw earlier in Exercise 2.



**Note** The results should be identical.

## Module 18: Monitoring SQL Server 2012

# Lab 18: Monitoring SQL Server 2012

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_18\_PRJ\10775A\_18\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Investigating DMVs

#### ► Task 1: Investigate the use of Dynamic Management Views and Functions

1. In Solution Explorer, double-click the file **51 – Lab Exercise 1.sql** to open it.
2. Follow the instructions in the T-SQL script.

### Exercise 2: Configure Management Data Warehouse

#### ► Task 1: Create a management data warehouse for central collection of performance data

1. In Object Explorer, expand the **Proseware** server, expand **Management**, right-click **Data Collection**, and click **Configure Management Data Warehouse**.
2. In the Configure Management Data Warehouse Wizard window, click **Next**.
3. In the Select Configuration task window, click **Next**.
4. In the Configure Management Data Warehouse Storage window, in the **Server name** textbox ensure **Proseware** has been entered, and click **New**.
5. In the New Database window, in the **Database name** textbox, type **MDW**, and click **OK**.
6. In the Configure Management Data Warehouse Storage window, click **Next**.
7. In the Map Logins and Users window, review the options and click **Next**.
8. In the Complete the Wizard window, click **Finish**.
9. In the Configure Data Collection Wizard Progress window, click **Close**.

### Exercise 3: Configure Instances for Data Collection

#### ► Task 1: Configure data collection on each instance

1. In Object Explorer, expand the **Proseware** server, expand **Management**, right-click **Data Collection**, and click **Configure Management Data Warehouse**.
2. In the Configure Management Data Warehouse Wizard window, click **Next**.
3. In the Select Configuration task window, click **Set up data collection**, and click **Next**.
4. In the Configure Management Data Warehouse Storage window, click the **ellipsis** button beside the **Server name** textbox.
5. In the Connect to Server window, in the **Server name** textbox, type **Proseware**, and click **Connect**.
6. In the **Database name** drop-down list, select **MDW**, and click **Next**.
7. In the Complete the Wizard window, click **Finish**.
8. In the Configure Data Collection Wizard Progress window, click **Close**.
9. In Object Explorer, click **Connect**, then click **Database Engine**. In the Connect to Server window, in the **Server name** textbox type **AdventureWorks**, and click **Connect**.
10. In Object Explorer, expand the **AdventureWorks** server, expand **Management**, right-click **Data Collection**, and click **Configure Management Data Warehouse**.
11. In the Configure Management Data Warehouse Wizard window, click **Next**.
12. In the Select Configuration task window, click **Set up data collection**, and click **Next**.
13. In the Configure Management Data Warehouse Storage window, click the **ellipsis** button beside the **Server name** textbox.
14. In the Connect to Server window, in the **Server name** textbox, type **Proseware**, and click **Connect**.
15. In the **Database name** drop-down list, select **MDW**, and click **Next**.
16. In the Complete the Wizard window, click **Finish**.
17. In the Configure Data Collection Wizard Progress window, click **Close**.

### Challenge Exercise 4: Work with Data Collector Reports (Only if time permits)

#### ► Task 1: Disable data collectors on both instances

1. In Object Explorer, expand the **Proseware** server, expand **Management**, right-click **Data Collection**, and click **Disable Data Collection**. If a confirmation window appears, click **Close**.
2. In Object Explorer, expand the **AdventureWorks** server, expand **Management**, right-click **Data Collection**, and click **Disable Data Collection**. If a confirmation window appears, click **Close**.

► **Task 2: Restore a backup of the MDW database**

1. In Object Explorer, expand the **Proseware** server, expand **Databases**, right-click the **MDW** database and click **Delete**.
2. In the Delete Object window, check the **Close existing connections** checkbox and click **OK**.
3. In Object Explorer, expand the **Proseware** server, right-click **Databases**, and click **Restore Database**.
4. In the Restore Database window, in the **Source options** click **Device**.
5. Click the **ellipsis** beside the **Device** textbox.
6. In the Select backup devices window, click **Add**.
7. Navigate to the file **D:\10775A\_Labs\10775A\_18\_PRJ\10775A\_18\_PRJ\MDW.bak** and click **OK**.
8. In the Select backup devices window, click **OK**.
9. In the **Select a page** pane, click **Options**.
10. In the **Restore options**, check **Overwrite the existing database (WITH REPLACE)**, and click **OK**.
11. In the Microsoft SQL Server Management Studio window, click **OK**.

► **Task 3: Review the available reports**

1. In Object Explorer, expand **Proseware** server, expand **Databases**.
2. Right-click the **MDW** database, click **Reports**, click **Management Data Warehouse**, and click **Management Data Warehouse overview**.
3. Review the **Disk Usage Report** for **MIA-SQL1\MKTG**.
4. Close the report window.
5. Right-click the **MDW** database, click **Reports**, click **Management Data Warehouse**, and click **Management Data Warehouse overview**.
6. Review the **Server Activity Report** for **MIA-SQL1\MKTG**.
7. Close the report window. Right-click the **MDW** database, click **Reports**, click **Management Data Warehouse**, and click **Management Data Warehouse overview**.
8. Review the **Query Statistics Report** for **MIA-SQL1\MKTG**.
9. Close the report window.

**MCT USE ONLY. STUDENT USE PROHIBITED**



## Module 19: Managing Multiple Servers

# Lab 19: Managing Multiple Servers

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the **Open Project** window, open the project **D:\10775A\_Labs\10775A\_19\_PRJ\10775A\_19\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercise 1: Configure CMS and Execute Multi-server Queries

#### ► Task 1: Create a central management server

1. From the **View** menu in **SQL Server Management Studio**, click **Registered Servers**.
2. In the Registered Servers window that opens, expand **Database Engine**.
3. Right-click **Central Management Servers**, and click **Register Central Management Server**.
4. In the New Server Registration window, in the **Server name** textbox, type **Proseware** and click **Save**.

#### ► Task 2: Create a server group within the CMS

1. In SQL Server Management Studio, in **Registered Servers**, expand **Central Management Servers**.
2. Right-click **Proseware** and click **New Server Group**.
3. In the New Server Group Properties window, in the **Group name** textbox, type **Core Servers** and click **OK**.
4. In **Registered Servers**, right-click **Core Servers** and click **New Server Registration**.
5. In the New Server Registration window, in the **Server name** textbox, type **Proseware** and click **Save**.
6. In **Registered Servers**, right-click **Core Servers** and click **New Server Registration**.
7. In the New Server Registration window, in the **Server name** textbox, type **AdventureWorks** and click **Save**.

► **Task 3: Execute a command to find all databases on any core server in full recovery model**

1. In **Registered Servers**, right-click **Core Servers** and click **New Query**.
2. Type the following code into the new query window:

```
SELECT * FROM sys.databases WHERE recovery_model_desc = 'FULL';
```

3. From the **Toolbar**, click **Execute**.



**Note** A list of databases in full recovery model is returned.

## Exercise 2: Deploy a Data-tier Application

► **Task 1: Deploy the data-tier application using SSMS**

1. In Object Explorer, expand the **Proseware** server, expand **Databases**, right-click **Databases**, and click **Deploy Data-tier Application**.
2. In the Introduction window, click **Next**.
3. In the Select Package window, enter or select the path **D:\10775A\_Labs\10775A\_19\_PRJ\10775A\_19\_PRJ\CityCode.dacpac**, and click **Next**.
4. In the Update Configuration window, click **Next**.
5. In the Summary window, click **Next**.
6. In the Deploy DAC window, click **Finish**.
7. In Solution Explorer, double-click the file **61 – Lab Exercise 2.sql** to open it.
8. Review the T-SQL script.
9. On the **Toolbar** click **Execute**.

## Exercise 3: Register and extract a data-tier application

► **Task 1: Register the existing database as a data-tier application**

1. In Object Explorer, expand the **Proseware** server, expand **Databases**, right-click the **Research** database, click **Tasks**, and click **Register as Data-tier Application**.
2. In the Introduction window, click **Next**.
3. In the Set Properties window, click **Next**.
4. In the Validation and Summary window, click **Next**.
5. In the Register DAC window, click **Finish**.

► **Task 2: Extract a dacpac from the database to send to the development team**

1. In Object Explorer, expand the **Proseware** server, expand **Databases**, right-click the **Research** database, click **Tasks**, and click **Extract Data-tier Application**.

2. In the Introduction window, click **Next**.
3. In the Set Properties window, in the **Save to DAC package file** textbox, type **D:\MKTG\Research.dacpac**, and then click **Next**.
4. In the Validation and Summary window, click **Next**.
5. In the Build Package window, click **Finish**.

### Challenge Exercise 4: Upgrade a Data-tier Application (Only if time permits)

#### ► Task 1: Attempt to deploy the v2 upgrade

1. In Object Explorer, expand the **Proseware** server, right-click **Databases** and click **Refresh**.
2. Expand **Databases**, right-click the **CityCode** database, click **Tasks**, and click **Upgrade Data-tier Application**.
3. In the Introduction window, click **Next**.
4. In the Select Package window, enter or select the path **D:\10775A\_Labs\10775A\_19\_PRJ\10775A\_19\_PRJ\CityCode\_v2.dacpac**, and click **Next**.
5. In the Detect Change window, click **Next**.
6. In the Options window, uncheck the **Execute pre-deployment Script**, check **Rollback on failure** and click **Next**.
7. In the Review Upgrade Plan window, check **Proceed despite possible data loss** and click **Next**.
8. In the Summary window, click **Next**.
9. In the Microsoft SQL Server Management Studio window, click **OK**.
10. In the Upgrade DAC window, notice the failures and click **Finish**.
11. In Object Explorer, right-click **Databases**, and click **Refresh**.



**Note** All databases and data-tier applications have been left unchanged.

#### ► Task 2: Deploy the v3 upgrade

1. In Object Explorer, expand **Databases**, right-click the **CityCode** database, click **Tasks**, and click **Upgrade Data-tier Application**.
2. In the Introduction window, click **Next**.
3. In the Select Package window, enter or select the path **D:\10775A\_Labs\10775A\_19\_PRJ\10775A\_19\_PRJ\CityCode\_v3.dacpac**, and click **Next**.
4. In the Detect Change window, click **Next**.
5. In the Options window, check **Rollback on failure** and click **Next**.
6. In the Review Upgrade Plan window, click **Next**.
7. In the Summary window, click **Next**.
8. In the Upgrade DAC window, click **Finish**.
9. In Object Explorer, right-click **Databases**, and click **Refresh**.



**Note** The data-tier application is upgraded.

**MCT USE ONLY. STUDENT USE PROHIBITED**

## Module 20: Troubleshooting Common SQL Server 2012 Administrative Issues

# Lab 20: Troubleshooting Common Issues

### Lab Setup

For this lab, you will use the available virtual machine environment. Before you begin the lab, you must complete the following steps:

1. Revert the virtual machines as per the instructions in D:\10775A\_Labs\Revert.txt.
2. In the virtual machine, click **Start**, click **All Programs**, click **Microsoft SQL Server 2012**, and click **SQL Server Management Studio**.
3. In the Connect to Server window, type **Proseware** in the **Server name** text box.
4. In the **Authentication** drop-down list box, select **Windows Authentication** and click **Connect**.
5. In the **File** menu, click **Open**, and click **Project/Solution**.
6. In the Open Project window, open the project **D:\10775A\_Labs\10775A\_20\_PRJ\10775A\_20\_PRJ.ssmssln**.
7. In Solution Explorer, double-click the query **00-Setup.sql**. When the query window opens, click **Execute** on the toolbar.

### Exercises 1 – 5: Troubleshoot and Resolve SQL Server Administrative Issues

#### Exercise 1 – Issue 1

► **Task 1: Read the supporting documentation for the exercise**

- Read the supporting documentation for the exercise.

► **Task 2: Troubleshoot and resolve the issue**

1. In Object Explorer, expand the **Proseware** server, expand **Security**, and expand **Logins**.
2. Beside the **PromoteApp** user, note the icon with a red down arrow, the login is disabled.
3. Right-click **PromoteApp**, and click **Properties**.
4. In the **Select a page** pane, click **Status**, click **Enabled**, and click **OK**.
5. In Object Explorer, right-click **Logins**, and click **Refresh**. Note that the login is now enabled.

## Exercise 2 – Issue 2

### ► Task 1: Read the supporting documentation for the exercise

- Read the supporting documentation for the exercise.

### ► Task 2: Troubleshoot and resolve the issue

1. In Object Explorer, expand the **Proseware** server, expand **Databases**, and note the status of the **AdminDB** database. The database is in a restoring state, the junior DBA may have performed a tail-log backup.
2. In Object Explorer, right-click the **AdminDB** database, click **Tasks**, click **Restore**, and click **Transaction Log**.
3. In the Restore Transaction Log – AdminDB window, click **From file or tape**, and click **OK**.
4. In the Microsoft SQL Server Management Studio window, click **OK**.

## Exercise 3 – Issue 2

### ► Task 1: Read the supporting documentation for the exercise

- Read the supporting documentation for the exercise.

### ► Task 2: Troubleshoot and resolve the issue

1. In Object Explorer, expand the **Proseware** server, expand **SQL Server Agent**, and expand **Jobs**.
2. Right-click the **Get File List** job, and click **Start Job at Step**.
3. In the Start Jobs – Proseware window, note the failure, and click **Close**.
4. In Object Explorer, right-click the **Get File List** job, and click **View History**.
5. In the Log File Viewer – Proseware window, click on the plus sign in a failing row to expand the job steps.
6. Click on the row for Step ID **1**.
7. In the **Selected row details** pane, scroll down to find the error (Logon failure: unknown user name or bad password). Note when using the proxy for David.Alexander.
8. In the Log File Viewer - Proseware window, click **Close**.
9. In Object Explorer, expand **Security**, expand **Credentials**, right-click **DavidAlexanderCredential**, and click **Properties**.
10. In the Credential Properties - DavidAlexanderCredential window, in the **Password** textbox, and the **Confirm password** textbox, type **Pa\$\$w0rd**, and click **OK**.
11. In Object Explorer, right-click the **Get File List** job, and click **Start Job at Step**.
12. In the Start Jobs - Proseware window, note that the job still fails, and click **Close**.
13. In Object Explorer, right-click the **Get File List** job, and click **View History**.
14. In the Log File Viewer - Proseware window, click on the plus sign in a failing row to expand the job steps.
15. Click on the row for Step ID **1**.

16. In the **Selected row details** pane, scroll down to find the error (Access is denied).
17. In the Log File Viewer - Proseware window, click **Close**. We need to investigate the job step to see why an Access is denied error could be returned.
18. In Object Explorer, right-click the **Get File List** job, and click **Properties**.
19. In the **Select a page** pane, click **Steps**, then click **Edit**.
20. Note that the command being executed requires permission to read the directory entries in the D:\MKTG folder and permission to write to the L:\MKTG folder using the identity of David Alexander.
21. Click **Cancel** and then click **Cancel**.
22. On the basis that write permissions are less likely to be granted than read permissions, we will investigate the write permissions first. In Windows Explorer, right-click folder **L:\MKTG** and click **Properties**.
23. In the MKTG Properties window, click the **Security** tab and note that David Alexander has no access to the folder, nor do any groups that he is likely to be a member of.
24. Click **Edit** then in the Permissions for MKTG window, click **Add**.
25. Ensure that the value of **From this location** is set to **AdventureWorks.msft**, then in the **Enter the object names to select** textbox, type **David.Alexander** and click **OK**.
26. In the Permissions for MKTG window, click the **Allow checkbox in the Modify row** and click **OK**.
27. In the MKTG Properties window, click **OK**.
28. Now we will test if the job runs.
29. In Object Explorer, right-click the **Get File List** job, and click **Start Job at Step**.
30. In the Start Jobs - Proseware window, note that the job still fails, and click **Close**.
31. Now we need to check the read permissions.
32. In Windows Explorer, right-click folder **D:\MKTG** and click **Properties**.
33. In the MKTG Properties window, click the **Security** tab and note that David Alexander has no access to the folder, nor do any groups that he is likely to be a member of.
34. Click **Edit** then in the Permissions for MKTG window, click **Add**.
35. Ensure that the value of **From this location** is set to **AdventureWorks.msft**, then in the **Enter the object names to select** textbox, type **David.Alexander** and click **OK**.
36. In the Permissions for MKTG window, click **OK**.
37. In the MKTG Properties window, click **OK**.
38. Now we will test again whether or not the job runs.
39. In Object Explorer, right-click the **Get File List** job, and click **Start Job at Step**.
40. In the Start Jobs - Proseware window, note that the job now works, and click **Close**.

## Exercise 4 – Issue 2

### ► Task 1: Read the supporting documentation for the exercise

- Read the supporting documentation for the exercise.

### ► Task 2: Troubleshoot and resolve the issue

1. In Object Explorer, expand the **Proseware** server, expand **Databases**, right-click **AdminDB** database, and click **Properties**.
2. In the **Select a page** pane, click **Options**.
3. Review the options related to statistics.
4. Change the **Auto Create Statistics** to **True**.
5. Change the **Auto Update Statistics** to **True**, and click **OK**.



**Note** The following steps cause statistics to be updated immediately rather than waiting for autostats to correct the situation.

6. In Object Explorer, right-click **AdminDB**, and click **New Query**.
7. In the Query window, type the following command:

```
EXEC sp_updatestats;
```

8. On the **Toolbar**, click **Execute**.

## Exercise 5 – Issue 2

### ► Task 1: Read the supporting documentation for the exercise

- Read the supporting documentation for the exercise.

### ► Task 2: Troubleshoot and resolve the issue

1. In Object Explorer, expand the **Proseware** server, expand **Databases**, right-click **CityDetails** database, and click **Properties**.
2. In the **Select a page** pane, click **Options**.
3. Review the options related to statistics.
4. Change the **Auto Close** option to **False** and click **OK**.