MICROSOFT SQL SERVER USER GUIDE pdf

1: SQL Server on Windows and Linux | Microsoft

This manual documents EMS SQL Management Studio for SQL Server No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the.

The channels that are actually available for a sensor depend on the performance counters that you choose during setup. Channel Overview User Connections Number of user connections. Because each user connection consumes some memory, configuring overly high numbers of user connections could affect throughput. Set user connections to the maximum expected number of concurrent users. Logins Total number of logins started per second. Logouts Total number of logout operations started per second. Full Scans Number of unrestricted full scans per second. These can be either base-table or full-index scans. Page Splits Number of page splits per second that occur as the result of overflowing index pages. Table Lock Escalations Number of times locks on a table were escalated. Buffer Cache Hit Ratio Percentage of pages found in the buffer cache without having to read from disk. The ratio is the total number of cache hits divided by the total number of cache lookups since an instance of SQL Server was started. After a long period of time, the ratio moves very little. Because reading from the cache is much less expensive than reading from disk, you want this ratio to be high. Generally, you can increase the buffer cache hit ratio by increasing the amount of memory available to SQL Server. Database Pages Number of pages in the buffer pool with database content. Stolen Pages Number of pages used for miscellaneous server purposes including procedure cache. Page Life Expectancy Number of seconds a page will stay in the buffer pool without references. Connection Memory KB Total amount of dynamic memory the server is using for maintaining connections. Optimizer Memory KB Total amount of dynamic memory the server is using for query optimization. Total Server Memory KB Total amount of dynamic memory in kilobytes that the server is using currently. Lock Requests Number of new locks and lock conversions per second requested from the lock manager. Deadlocks Number of lock requests per second that resulted in a deadlock. Average Wait Time Average amount of wait time in milliseconds for each lock request that resulted in a wait. High batch requests mean good throughput. Indicates the number of times the compile code path is entered. Includes compiles due to recompiles. After SQL Server user activity is stable, this value reaches a steady state. Counts the number of times recompiles are triggered. In general, you want the recompiles to be low. Requires credentials for Windows systems to be defined for the device you want to use the sensor on. WoW64 Windows bit on Windows bit must be installed on target systems that run Windows Server Try to stay below WMI sensors per probe. Above this number, please consider using multiple Remote Probes for load balancing. If you want to use this sensor type, please add it to a remote probe device. Add Sensor The Add Sensor dialog appears when you manually add a new sensor to a device. It only shows the setting fields that are required for creating the sensor. Therefore, you will not see all setting fields in this dialog. The settings you choose will be valid for all sensors that you create when you finish this dialog. Add check marks in front of the respective lines to select the desired items. PRTG creates one sensor for each selection. You can also use the check box in the table head to select and deselect all items. Display name and service name are provided as returned by the SQL Server. Usually, you can keep the default value. Every sensor that PRTG creates for the server instances monitors the performance counters you select here. Read general performance counters. This shows the number of user connections, and the number of logins and logouts per second. Read access method counters. This shows the number of full scans, page splits, and table lock escalations per second. Read buffer manager counters. This shows the buffer cache hit ratio in percent, and the number of database pages and stolen pages. Read memory manager counters. This shows the connection memory, optimizer memory, total server memory, target server memory, and SQL cache memory in kb. This shows the number of lock requests and deadlocks per second , and the average wait time. Depending on your selection, PRTG creates a sensor with the specified channels. To monitor more than one of the listed groups of performance counters, please add the sensor several times for the instance s. Sensor Settings On the details page of a sensor, click the Settings tab to change its settings. See the Device Settings for details. For some sensor types, you can define the monitoring target explicitly in the sensor settings. Please see below for details on available settings. By default, PRTG shows this name in the device tree , as well as in alarms , logs , notifications , reports , maps , libraries , and tickets. Parent Tags Shows Tags that this sensor inherits from its parent device, group, and probe. This setting is shown for your information only and cannot be changed here. Tags Enter one or more Tags , separated by spaces or commas. You can use tags to group sensors and use tagâ€”filtered views later on. Tags are not case sensitive. We recommend that you use the default value. You can add additional tags to the sensor if you like. Other tags are automatically inherited from objects further up in the device tree. These are visible above as Parent Tags. Priority Select a priority for the sensor. This setting determines where the sensor is placed in sensor lists. Top priority is at the top of a list. Choose from one star low priority to five stars top priority. Once a sensor is created, you cannot change this value. It is shown for reference purposes only. If you need to change this, please add the sensor anew. Name Shows the name of the server instance that this sensor monitors. Choose WMI class automatically. We recommend this setting. Manually enter a WMI class name. Select this option if your server instance returns an error code in automatic mode. This setting is intended for experienced users only. Enter the WMI class name that the sensor uses for monitoring your server instance. Do not store the sensor result. Write sensor result to disk Filename: Result of Sensor [ID]. Store the last result received from the sensor to the Logs Sensors directory in the PRTG data folder on the probe system the sensor is running on on the Master node if in a cluster. This is for debugging purposes. PRTG overwrites these files with each scanning interval. For more information on how to find the folder used for storage, see section Data Storage. Sensor Display Primary Channel Select a channel from the list to define it as the primary channel. The available options depend on what channels are available for this sensor. Graph Type Define how different channels will be shown for this sensor. Show channels independently default: Show an own graph for each channel. Stack channels on top of each other: Stack channels on top of each other to create a multi-channel graph. This will generate an easy-to-read graph that visualizes the different components of your total traffic.

2: SQL Server Guides | Microsoft Docs

SQL Server is a central part of the Microsoft data platform. SQL Server is an industry leader in operational database management systems (ODBMS). This documentation helps you install, configure, and use SQL Server on both Windows and Linux.

This Cmdlet Reference contains the help files for these cmdlets. The topics in this guide include information about the cmdlets and their associated parameters, and provide examples about how to use the cmdlets. TechNet Library E-book publication date: This guide introduces core features and functionality, with technical advice and under-the-hood insights from a Microsoft MVP and members of the System Center team at Microsoft. System Center E-book publication date: January pages Microsoft System Center: February pages Microsoft System Center: System Center can be used to transform enterprise IT from a device-based infrastructure and deployment strategy to a service-based user-centric consumption model based on private cloud computing. Windows Azure on the other hand is a subscription-based public cloud platform that enables the development, deployment, and management of cloud solutions. App Controller is the glue that unifies these two platforms by providing a single interface that enables administrators to perform complex operations without overwhelming them with the underlying technical complexities involved. This book serves as an introduction to implementing and managing the hybrid computing solutions using App Controller. It describes the basic concepts, processes, and operations involved in connecting, consuming, and managing resources that are deployed both on and off premises. Each chapter provides a concise, self-contained walkthrough for a specific aspect of managing private, public, and hybrid clouds using App Controller. November pages Microsoft System Center: We want you to get the most out of using Configuration Manager in your environment regardless of whether the task at hand is querying the Configuration Manager database for system information, creating and customizing reports, or deploying operating system images to client machines. October pages Microsoft System Center: Designing Orchestrator Runbooks David Ziembicki, Aaron Cushner, Andreas Rynes, Mitch Tulloch Guide We believe that orchestration and automation are becoming increasingly important in IT organizations of all sizes and across all infrastructure types ranging from on-premises to cloud-based. Orchestration and automation can help reduce the cost of IT while improving consistency and quality of IT service delivery. Like any powerful technology. Our objective with this book is to provide a framework for runbook design and IT process automation to help you get the most out of System Center Orchestrator and to help you utilize Orchestrator in concert with the rest of the System Center for an enterprise-wide and systematic approach to process automation. September pages Microsoft System Center: March 94 pages Microsoft System Center: Written by experts on the Microsoft System Center team and with Microsoft MVP Mitch Tulloch as series editor, this title delivers concise guidance, from-the-field insights, and best practices for optimizing and maintaining your Service Manager environment. December 96 pages Microsoft System Center: While most of you who are Configuration Manager administrators are fairly comfortable with the product and can perform common management tasks, many of you still have pain points when it comes to certain aspects of how the product works. This book is our attempt to address some of these gaps and pain points. November pages Technical Documentation for System Center - Virtual Machine Manager VMM Information Experience Team Guide Virtual Machine Manager VMM is a management solution for the virtualized datacenter, enabling you to configure and manage your virtualization host, networking, and storage resources in order to create and deploy virtual machines and services to private clouds that you have created.

3: Importing and Exporting SQL Server Databases - Amazon Relational Database Service

SQL Server licensing makes choosing the right edition simple and economical. Unlike other major vendors, there's no having to pay for expensive add-ons to run your most demanding applicationsâ€"because every feature and capability is already built in. Cloud-optimized licensing with the ability to.

Ascending or descending order on the columns in the index Full-table versus filtered for nonclustered indexes You can also customize the initial storage characteristics of the index to optimize its performance or maintenance by setting an option such as FILLFACTOR. Also, you can determine the index storage location by using filegroups or partition schemes to optimize performance. Index Placement on Filegroups or Partitions Schemes As you develop your index design strategy, you should consider the placement of the indexes on the filegroups associated with the database. Careful selection of the filegroup or partition scheme can improve query performance. By default, indexes are stored in the same filegroup as the base table on which the index is created. A nonpartitioned clustered index and the base table always reside in the same filegroup. However, you can do the following: Create nonclustered indexes on a filegroup other than the filegroup of the base table or clustered index. Partition clustered and nonclustered indexes to span multiple filegroups. By creating the nonclustered index on a different filegroup, you can achieve performance gains if the filegroups are using different physical drives with their own controllers. Data and index information can then be read in parallel by the multiple disk heads. This creates no performance gain. Because you cannot predict what type of access will occur and when it will occur, it could be a better decision to spread your tables and indexes across all filegroups. This would guarantee that all disks are being accessed because all data and indexes are spread evenly across all disks, regardless of which way the data is accessed. This is also a simpler approach for system administrators. Partitions Across Multiple Filegroups You can also consider partitioning clustered and nonclustered indexes across multiple filegroups. Partitioned indexes are partitioned horizontally, or by row, based on a partition function. The partition function defines how each row is mapped to a set of partitions based on the values of certain columns, called partitioning columns. A partition scheme specifies the mapping of the partitions to a set of filegroups. Partitioning an index can provide the following benefits: Provide scalable systems that make large indexes more manageable. OLTP systems, for example, can implement partition-aware applications that deal with large indexes. Make queries run faster and more efficiently. When queries access several partitions of an index, the query optimizer can process individual partitions at the same time and exclude partitions that are not affected by the query. For more information, see Partitioned Tables and Indexes. Index Sort Order Design Guidelines When defining indexes, you should consider whether the data for the index key column should be stored in ascending or descending order. Ascending is the default and maintains compatibility with earlier versions of SQL Server. Specifying the order in which key values are stored in an index is useful when queries referencing the table have ORDER BY clauses that specify different directions for the key column or columns in that index. In these cases, the index can remove the need for a SORT operator in the query plan; therefore, this makes the query more efficient. For example, the buyers in the Adventure Works Cycles purchasing department have to evaluate the quality of products they purchase from vendors. The buyers are most interested in finding products sent by these vendors with a high rejection rate. As shown in the following query, retrieving the data to meet this criteria requires the RejectedQty column in the Purchasing. PurchaseOrderDetail table to be sorted in descending order large to small and the ProductID column to be sorted in ascending order small to large. The Database Engine can move equally efficiently in either direction. Sort order can be specified only for key columns. Clustered Index Design Guidelines Clustered indexes sort and store the data rows in the table based on their key values. There can only be one clustered index per table, because the data rows themselves can only be sorted in one order. With few exceptions, every table should have a clustered index defined on the column, or columns, that offer the following: Can be used for frequently used queries. Provide a high degree of uniqueness. By default, this index is clustered; however, you can specify a nonclustered index when you create the constraint. Can be used in range queries. When it is required, the Database Engine automatically adds a uniqueifier value to a row to make each key unique. This column and its values are used internally and cannot be seen or accessed by users. Each page in an index B-tree is called an index node. The top node of the B-tree is called the root node. The bottom nodes in the index are called the leaf nodes. Any index levels between the root and the leaf nodes are collectively known as intermediate levels. In a clustered index, the leaf nodes contain the data pages of the underlying table. The root and intermediate level nodes contain index pages holding index rows. Each index row contains a key value and a pointer to either an intermediate level page in the B-tree, or a data row in the leaf level of the index. The pages in each level of the index are linked in a doubly-linked list. Clustered indexes have one row in sys. By default, a clustered index has a single partition. When a clustered index has multiple partitions, each partition has a B-tree structure that contains the data for that specific partition. For example, if a clustered index has four partitions, there are four B-tree structures; one in each partition. Depending on the data types in the clustered index, each clustered index structure will have one or more allocation units in which to store and manage the data for a specific partition. The pages in the data chain and the rows in them are ordered on the value of the clustered index key. All inserts are made at the point where the key value in the inserted row fits in the ordering sequence among existing rows. This illustration shows the structure of a clustered index in a single partition. Query Considerations Before you create clustered indexes, understand how your data will be accessed. Consider using a clustered index for queries that do the following: After the row with the first value is found by using the clustered index, rows with subsequent indexed values are guaranteed to be physically adjacent. For example, if a query retrieves records between a range of sales order numbers, a clustered index on the column SalesOrderNumber can quickly locate the row that contains the starting sales order number, and then retrieve all successive rows in the table until the last sales order number is reached. Return large result sets. Use JOIN clauses; typically these are foreign key columns. This improves query performance. Column Considerations Generally, you should define the clustered index key with as few columns as possible. Consider columns that have one or more of the following attributes: Are unique or contain many distinct values For example, an employee ID uniquely identifies employees. Alternatively, a clustered index could be created on LastName, FirstName, MiddleName because employee records are frequently grouped and queried in this way, and the combination of these columns would still provide a high degree of difference. Are accessed sequentially For example, a product ID uniquely identifies products in the Production. Product table in the AdventureWorks database. This is because the rows would be stored in sorted order on that key column. Used frequently to sort the data retrieved from a table. It can be a good idea to cluster, that is physically sort, the table on that column to save the cost of a sort operation every time the column is queried. Clustered indexes are not a good choice for the following attributes: Columns that undergo frequent changes This causes in the whole row to move, because the Database Engine must keep the data values of a row in physical order. This is an important consideration in high-volume transaction processing systems in which data is typically volatile. Wide keys Wide keys are a composite of several columns or several large-size columns. The key values from the clustered index are used by all nonclustered indexes as lookup keys. Any nonclustered indexes defined on the same table will be significantly larger because the nonclustered index entries contain the clustering key and also the key columns defined for that nonclustered index. Nonclustered Index Design Guidelines A nonclustered index contains the index key values and row locators that point to the storage location of the table data. You can create multiple nonclustered indexes on a table or indexed view. Generally, nonclustered indexes should be designed to improve the performance of frequently used queries that are not covered by the clustered index. Similar to the way you use an index in a book, the query optimizer searches for a data value by searching the nonclustered index to find the location of the data value in the table and then retrieves the data directly from that location. This makes nonclustered indexes the optimal choice for exact match queries because the index contains entries describing the exact location in the table of the data values being searched for in the queries. For example, to query the HumanResources. The query optimizer can quickly find all entries in the index that match the specified ManagerID. Each index entry points to the exact page and row in the table, or clustered index, in which the corresponding data can be found. After the query optimizer finds all entries in the index, it can go directly to the exact page and row to retrieve the data. Nonclustered Index Architecture Nonclustered indexes have the same B-tree structure as clustered indexes, except for the following significant differences: The data rows of the underlying table are not sorted and stored in order based on their nonclustered keys. The leaf layer of a nonclustered index is made up of index pages instead of data pages. The row locators in nonclustered index rows are either a pointer to a row or are a clustered index key for a row, as described in the following: If the table is a heap, which means it does not have a clustered index, the row locator is a pointer to the row. The pointer is built from the file identifier ID , page number, and number of the row on the page. If the table has a clustered index, or the index is on an indexed view, the row locator is the clustered index key for the row. Nonclustered indexes have one row in sys.

4: SQL Server Documentation | Microsoft Docs

MS SQL Server - Overview. This chapter introduces SQL Server, discusses its usage, advantages, versions, and components. What is SQL Server? It is a software, developed by Microsoft, which is implemented from the specification of RDBMS.

You can import and export SQL Server databases in a single, easily portable file. The following diagram shows the supported scenarios. There are many additional advantages to using native backup and restore. You can do the following: Migrate databases to Amazon RDS. Import and export data. Migrate schemas, stored procedures, triggers and other database code. Backup and restore single databases, instead of entire DB instances. Create copies of databases for testing, training, and demonstrations. Store and transfer backup files into and out of Amazon RDS through Amazon S3, giving you an added layer of protection for disaster recovery. The following are some limitations to using native backup and restore: If you restore a backup file from one time zone to a different time zone, you must audit your queries and applications for the effects of the time zone change. We recommend that you use native backup and restore to migrate your database to Amazon RDS if your database can be offline while the backup file is created, copied, and restored. Native backup and restore is not intended to replace the data recovery capabilities of the cross-region snapshot copy feature. We recommend that you use snapshot copy to copy your database snapshot to another region for cross-region disaster recovery in Amazon RDS. For more information, see Copying a Snapshot. An Amazon S3 bucket to store your backup files. If you already have an Amazon S3 bucket, you can use that. If you want to create a new bucket manually, see Creating a Bucket. If you already have an IAM role, you can use that. If you want to create a new IAM role manually, or attach trust and permissions policies to an existing IAM role, take the approach discussed in the next section. When you create an IAM role, you attach trust and permissions policies. For the native backup and restore feature, use trust and permissions policies similar to the examples following. In the first example following, we use the service principle name rds. The format for your ARN is arn: There are stored procedures for backing up your database, restoring your database, canceling tasks that are in progress, and tracking the status of the backup and restore tasks. For instructions on how to call each stored procedure, see the following subsections:

5: WMI Microsoft SQL Server Sensor | PRTG Network Monitor User Manual

IBMTivoli CompositeApplication Manager for Microsoft Applications:Microsoft SQL ServerAgent Version User's Guide SC

Shared memory for local connections and troubleshooting purpose. Named pipes for connections which are in LAN connectivity. It contains Query parser, Query optimizer and Query executor. SQL OS provides various operating system services, such as memory management deals with buffer pool, log buffer and deadlock detection using the blocking and locking structure. Apart from this, it also writes the log records from log buffer to physical file. Writing of Dirty pages from buffer cache to data file is also known as Hardening of dirty pages. It is a dedicated process and runs automatically by SQL Server at specific intervals. SQL Server runs checkpoint process for each database individually. This also runs in the background but to meet a user-specified target recovery time for the specific database where the option has been configured. Manual checkpoint runs for your current database only. Issued on specific operations such as Shutdown initiates a checkpoint operation on all databases except when shutdown is not clean shutdown with nowait. While taking backup of the database. Checkpoint also takes place when the recovery model of the DB is bulk-logged and a minimally logged operation is performed. This happens when SQL server comes under memory pressure. As far as I am aware, this is controlled by an internal process and there is no setting for it. SQL server constantly monitors memory usage to assess resource contention or availability ; its job is to make sure that there is a certain amount of free space available at all times. As part of this process, when it notices any such resource contention, it triggers Lazy Writer to free up some pages in memory by writing out dirty pages to disk. If Lazy Writer is always active, it could indicate memory bottleneck. Memory Architecture Following are some of the salient features of memory architecture. SQL Server "User address space" is broken into two regions: MemToLeave and Buffer Pool. The buffer management component consists of two mechanisms: The buffer pool is further divided into multiple sections. The most important ones being the buffer cache also referred to as data cache and procedure cache. Buffer cache holds the data pages in memory so that frequently accessed data can be retrieved from cache. The alternative would be reading data pages from the disk. Procedure cache keeps the stored procedure and query execution plans to minimize the number of times that query plans have to be generated. This information includes stored procedure and user-defined function parameters, cursor positions and more. No file can be a member of more than one file group. Log files are never part of a file group. Log space is managed separately from data space. Primary file group contains the primary data file and any other files not specifically assigned to another file group. All pages for the system tables are allocated in the primary file group. User-defined file groups are any file groups specified using the file group keyword in create database or alter database statement. One file group in each database operates as the default file group. When SQL Server allocates a page to a table or index for which no file group was specified when they were created, the pages are allocated from default file group. By default, primary file group is the default file group. Files Databases have three types of files - Primary data file, Secondary data file, and Log file. Primary data file is the starting point of the database and points to the other files in the database. Every database has one primary data file. We can give any extension for the primary data file but the recommended extension is. Secondary data file is a file other than the primary data file in that database. Some databases may have multiple secondary data files. Some databases may not have a single secondary data file. Recommended extension for secondary data file is. Log files hold all of the log information used to recover the database. Database must have at least one log file. We can have multiple log files for one database. The recommended extension for log file is. The location of all the files in a database are recorded in both master database and the primary file for the database. Most of the time, the database engine uses the file location from the master database. Logical name is used to refer to the file in all T-SQL statements. There can be up to 32, files in one database. Extents Extents are basic unit in which space is allocated to tables and indexes. An extent is 8 contiguous pages or 64KB. Uniform extents are made up of only single object. Mixed extents are shared by up to eight objects. The size of the page is 8KB. The start of each page is 96 byte header used to store system information such as type of page, amount of free space on the page and object id of the object owning the page. There are 9 types of data pages in SQL Server. Each log record contains the ID of the transaction that it belongs to. Log records for data modifications record either the logical operation performed or they record the before and after images of the modified data. The before image is a copy of the data before the operation is performed; the after image is a copy of the data after the operation has been performed. To roll the logical operation forward, the operation is performed again. To roll the logical operation back, the reverse logical operation is performed. Before and after image logged. To roll the operation forward, the after image is applied. To roll the operation back, the before image is applied. Different types of operations are recorded in the transaction log. Every data modification insert, update, or delete. This includes changes by system stored procedures or data definition language DDL statements to any table, including system tables. Every extent and page allocation or de allocation. Creating or dropping a table or index. Rollback operations are also logged. Each transaction reserves space on the transaction log to make sure that enough log space exists to support a rollback that is caused by either an explicit rollback statement or if an error is encountered. This reserved space is freed when the transaction is completed. The section of the log file from the first log record that must be present for a successful database-wide rollback to the last-written log record is called the active part of the log, or the active log. This is the section of the log required to a full recovery of the database. No part of the active log can ever be truncated. Virtual log files have no fixed size, and there is no fixed number of virtual log files for a physical log file. The Database Engine chooses the size of the virtual log files dynamically while it is creating or extending log files. The Database Engine tries to maintain a small number of virtual files. The size or number of virtual log files cannot be configured or set by administrators. If the log files grow to a large size because of many small increments, they will have many virtual log files. This can slow down database startup and also log backup and restore operations. SQL Server uses a write-ahead log WAL , which guarantees that no data modifications are written to disk before the associated log record is written to disk. This maintains the ACID properties for a transaction. This allows you to connect to and manage your SQL Server from a graphical interface instead of having to use the command line. In order to connect to a remote instance of an SQL Server, you will need this or similar software. It is used by Administrators, Developers, Testers, etc. SQL Server Management Studio will be open up as shown in the following snapshot in either of the above method. For example, you provide your username and password when logging on to Windows or even your e-mail account. This username and password builds up the credentials. Therefore, credentials are simply a username and a password. A login specific to SQL Server. A login mapped to a certificate. A login mapped to asymmetric key. If you need to create your own credentials username and password, you can create a login specific to SQL Server. Login will be created as shown in the following image.

6: Upgrading the Microsoft SQL Server DB Engine - Amazon Relational Database Service

Microsoft SQL Server turns your mission-critical applications into intelligent applications with in-memory performance and advanced analytics built in.

7: SQL Serverâ€”Pricing and Licensing | Microsoft

NetWorker Module for Microsoft for SQL and SharePoint VSS User Guide 11 Revision history The following table presents the revision history of this document.

8: Enterprise Manager Microsoft SQL Server Plug-in User's Guide - Contents

SQL Server on Linux and SQL Server in Docker containers. SQL Server on Linux is Microsoft's most successful SQL Server product ever, with over seven million downloads since its release in October

9: Download and install Azure Data Studio | Microsoft Docs

MICROSOFT SQL SERVER USER GUIDE pdf

Download content for Azure, www.enganchecubano.com, Office, SQL Server, SharePoint Server and other Microsoft technologies in e-book formats. Reference, guide, and step-by-step information are all available. All the e-books are free.

MICROSOFT SQL SERVER USER GUIDE pdf

Festive Finger Foods (Christmas at Home (Barbour)) 24. Embedded field experiences as professional apprenticeships Thomas E. Hodges and Heidi Mills The Poets Pilgrimage to Waterloo Protecting privacy and preventing misuse of the social security number Functions of physical distribution Why a Jewish Rabbi? Blood of eden book 3 The Poverty of Affluence Introduction to exponential functions activity Medieval Yorkshire Towns Food styling and photography for dummies History of racism in america German expressionist prints from the Ruth and Jacob Kainen Collection St. Nectarios of Aegina False Images of the Guru Minion Hunter (Dark Conspiracy Boardgame [BOX SET] Black-beards skull A traveller child Poverty capital microfinance and the making of development Janes Fighting Ships, 1982-1983 The Buzz on Professional Wrestling Issues and prospects for the new international economic order Human factors for civil flight deck design Understanding the Afterlife in This Life Taguchi methods explained practical steps to robust design Tony robbins new book The Wild Side of Pet Ferrets (Perspectives, the Wild Side of Pets) Writing a Lucene query Jacks Baby (Top Author) Black Beauty (Adventure Classics) Pdr Guide to Drug Interactions, Side Effects, Indications,Contraindication S, 1997 (51st ed, 1997) From French Community to Missouri Town Learn in your car russian Feelings the apple tree sheet music Introduction: The quest for the grail Determination of configurations by chemical methods Color and Two-Dimensional Design Tutorial membuat database dengan xampp The sirius mystery Breakthrough prayer jim cymbala