Microsoft System Center Reporting Cookbook

Microsoft System Center 2012 is an integrated management platform that helps you to easily and efficiently manage your data centers, client devices, and hybrid cloud IT environments.

This hands-on cookbook will guide you through how to create ready-to-use reports for all the components of System Center. The book starts by showing you how to plan business valued reports, while also discussing the building blocks of your reporting framework. Moving on from the basics, the later recipes demonstrate how you can create System Center Configuration Manager reports using the report builder tool and System Center Operations and Virtual Machine Manager reports with data available from the Operations Manager databases.

The book will then teach you how to build on and enhance the reports previously created by delving into advanced reporting techniques such as creating database reports, based on combined data sources. Finally, you will use Power BI to analyze and visualize System Center data, while also looking into the seamless integration between cloud services and System Center.

Who this book is written for

This book is for IT professionals who are responsible for producing reports using the data from System Center components. Basic knowledge of Microsoft System Center technologies is assumed.

What you will learn from this book

- Understand the goals of reporting
- Plan and organize the reporting environment for sharing report parts and role-based security
- Plan and design System Center reports
- About the System Center report building tools available to you
- Create powerful reports using System Center reporting databases
- Create System Center advanced reports
- Use Power BI to analyze and visualize System Center data

Microsoft System Center Reporting Cookbook

Over 40 practical recipes to help you plan, create, and manage reports efficiently for all components of Microsoft System Center

Foreword by Kathleen Wilson, Architect, Datacenter and Cloud Center of Excellence at Microsoft Corporation

Samuel Erskine
Kurt Van Hoecke
Nasira Ismail
In this package, you will find:

- The authors biography
- A preview chapter from the book, Chapter 1 'Understanding the Goals of Reporting'
- A synopsis of the book’s content
- More information on Microsoft System Center Reporting Cookbook

About the Authors

**Samuel Erskine** (IT Driving Fellow) is an independent IT consultant, who specializes in Service Manager and Configuration Manager. He is the content designer and one of the authors of *Microsoft System Center 2012 Service Manager Cookbook* and *Microsoft System Center 2012 Orchestrator Cookbook*, both by Packt Publishing; a contributing author to *System Center 2012 Configuration Manager Unleashed*; and a coauthor of *System Center 2012 Service Manager Unleashed*, Sams Publishing. With over 18 years of IT experience, he focuses on providing training and service management consultancy services in the United Kingdom and other locations. He also writes blogs at [www.itprocessed.com](http://www.itprocessed.com).

**Dieter Gasser** is an IT consultant and the cofounder of Syliance IT Services, headquartered in Switzerland. He has a strong focus on the delivery and customization of Service Manager.

Dieter has been working in IT for more than 13 years, and has focused on Microsoft technologies. He started his career as an application and database developer, and later became the IT manager of an international manufacturing company.

In 2010, he entered the systems management and automation market. With both his technical and managerial backgrounds, he has focused on Service Manager. Together with his colleagues, he delivers data center management and automation solutions based on Microsoft System Center to customers all across Switzerland.
Kurt Van Hoecke is an MVP and managing consultant at Inovativ Belgium. He focuses on the System Center product suite, including Service Manager, Configuration Manager, and Orchestrator. He is the coauthor of *System Center 2012 Orchestrator Unleashed*, Sams Publishing; *System Center 2012 Service Manager Unleashed*, Sams Publishing; and a contributor to *System Center Service Manager 2010 Unleashed*. He provides consultancy, development services, and blogs at www.scug.be/scsm and www.authoringFriday.com.

Nasira Ismail is an experienced project manager and service delivery consultant. She has result-oriented experience in incident, problem, change, and configuration management processes for medium and large enterprises. She has a strong grasp of the methods used to design processes through to implementation, with technologies (including System Center). She also leads training, mentoring, and coaching sessions to bring about a proactive culture and behavioral change in her organizations.
Microsoft System Center Reporting Cookbook

The System Center product provides you with the best IT systems management and reporting capabilities. The various components of this product continue to provide great value in all areas of management and monitoring. Each of the components has one or more data storing repositories. These data repositories hold valuable strategic information that can assist an organization in transforming its IT business unit into a strategic enabler.

The database (repository) technology of the System Center product is Microsoft SQL Server. There is a vast array of information on the Internet on how to query and report on Microsoft SQL databases. The skill set required for the level of querying and reporting on these databases is normally aligned to Database Administrators (DBAs) and Business Intelligence (BI) specialists. The DBAs and BI specialists use various tools to create business-valued reports based on the data stored in these Microsoft SQL databases.

The challenge DBAs and BI experts face with System Center reporting is a lack of insight into how the data is structured, and more importantly, a lack of deeper understanding of the products responsible for the input data. Conversely, in the majority of cases, the System Center product's Subject Matter Experts (SMEs) do not have the same level of competency as the DBAs and BI experts in creating and enhancing reports.

The aim of this book is to enhance the competency of System Center SMEs in the areas of reporting. There are a number of books on querying Microsoft SQL databases and generating business-value reports. However, those books do not focus on System Center. Also, all the books on System Center's components dedicate a reporting chapter to that specific component of the book's focus.

*Microsoft System Center Reporting Cookbook*, provides a one-stop shop for how to plan, create, and manage reports for all the components of the System Center product.

This book is written in the Packt Publishing style, which provides you with independent, task-oriented steps to achieve specific reporting objectives. We recommend that you read the first three chapters as primers for subsequent chapters. This book may be read in the order of your interest, but wherever relevant, we've referred to the dependent recipes in other chapters.
What This Book Covers

Chapter 1, Understanding the Goals of Reporting, discusses the building blocks of your reporting framework and how to plan for business-valued reports.

Chapter 2, Planning System Center Report Design, covers the internals of System Center reporting database structures, which you must understand in order to create credible and powerful reports. This chapter also includes an introductory recipe to SQL queries.

Chapter 3, Unpacking System Center Report Building Tools, discusses the common report-creating tools available for System Center administrators and Business Intelligence power users.

Chapter 4, Creating Reports for System Center Configuration Manager, provides tasks for System Center administrators, who can create System Center Configuration Manager reports using the Report Builder tool.

Chapter 5, Creating Reports for SCOM and SCVMM, demonstrates how to create System Center Operations and Data Protection Manager reports, with data available from the Operations Manager databases.

Chapter 6, Creating Reports for System Center Data Protection Manager, shows you how to create data-driven reports for System Center Data Protection Manager.

Chapter 7, Creating Reports for System Center Service Manager and Orchestrator, illustrates how to create service management and automation activity reports using data from System Center Service Manager and Orchestrator.

Chapter 8, Creating System Center Advanced Reports, shows you recipes that build on and enhance the reports from previous recipes in this book. These recipes delve into advanced reporting techniques and combined data source reporting.

Chapter 9, Using Power BI to Analyze and Visualize System Center Data, introduces the Power Business Intelligence (Power BI) options available for you from Microsoft. These recipes provide steps for analyzing and visualizing the System Center data using the Microsoft Excel Power BI add-ons, and introduce the cloud Power BI versions available for you.

Appendix, Useful Websites, Chapter Code, and Community Contributions, lists some of the sites that provide ready-made solutions and extensive, real-world, and dynamic content on reporting. Microsoft SQL Server-based reporting, similar to most Microsoft product areas, has an extended solutions partner community. There is an extensive and active support base on the Web. This appendix also provides you with the SQL and XML code referred to in the relevant chapters.
Understanding the Goals of Reporting

In this chapter, we will cover the following recipes:

- Understanding the goals of reporting
- Planning and optimizing dependent data inputs
- Planning report outputs
- Understanding dashboards and their dependencies
- Targeting reports at decision stakeholders
- Documenting Report Designs

Introduction

This chapter and the whole book builds on two guiding principles:

"You can't manage what you don't measure" and "there is no substitute for knowledge".

Reporting allows you to make strategic decisions based on the knowledge acquired from credible data sources. This chapter focuses on what steps you must take to identify and define what business specific objectives an organization aims to achieve with the output of reports. The report categories discussed in the book uses the data from the System Center product.
Understanding the Goals of Reporting

This recipe discusses the drivers of organizational reporting and the general requirements on how to plan for business valued reports.

Getting ready

To prepare for this recipe you need to be ready to make a difference with all the rich data available to you in the area of reporting. This may require a mindset change; be prepared.

How to do it...

The key to successfully identifying what needs to be reported is a clear understanding of what you or the report requestor is trying to measure and why.

Reporting is driven by a number of organizational needs, which may fall into one or more of these sample categories:

- Information to support a business case
- Audit and compliance driven request
- Budget planning and forecasting
- Current operational service level

These categories are examples of the business needs which you must understand. Understanding the business needs of the report increases the value of the report. For example, let us expand on and map the preceding business scenarios to the System Center Product using the following table:

<table>
<thead>
<tr>
<th>Business/organizational objective</th>
<th>Objective details</th>
<th>System Center Product</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information to support a business case</td>
<td>Provide a count of computers out of warranty to justify the request to buy additional computers.</td>
<td>System Center Configuration Manager</td>
</tr>
</tbody>
</table>
| Audit and compliance driven request | ▶ Provide the security compliance state of all windows servers.  
▶ Provide a list of attempted security breaches by month. | ▶ System Center Configuration Manager  
▶ System Center Operations Manager |
In a majority of cases for System Center administrators, the requestor does not provide the business objective. Use the preceding table as an example to guide your understanding of a report request.

**How it works...**

Reporting is a continual life cycle that begins with a request for information and should ultimately satisfy a real need. The typical life cycle of a request is illustrated in the following figure:
The life cycle stages are:

- Report conception
- Report request
- Report creation
- Report enhancement/retirement

The recipe focuses on the report conception stage. This stage is the most important stage of the life cycle. This is due to the fact that a report with a clear business objective will deliver the following:

- **Focused activities**: A report that does have a clear objective will reduce the risk of wasted effort usually associated with unclear requirements.
- **Direct or indirect business benefit**: The reports you create, for example using System Center data, ultimately should benefit the business.

An additional benefit to this stage of report planning is knowing when a report is no longer required. This would reduce the need to manage and support a report that has no value or use.

### Planning and optimizing dependent data inputs

A report is only as good as the accuracy of its data source. A data source is populated and updated by an input channel. This recipe discusses and provides steps for planning for the inputs your report data source(s) depends on.

### Getting ready

Review the *Understanding the goals of reporting* recipe as a primer for this recipe.

### How to do it...

The inputs of reports depend on the type of output you intend to produce and the definition of the accepted fields in the data source. An example is a report that would provide a total count of computers in a System Center Configuration Manager environment. This report will require an input field which stores a numeric value for computers in the database.
Here are the recommended steps you must take to prepare and optimize the data inputs for a report:

1. Identify the data source or sources.
2. Document the source data type properties.
3. Document the process used to populate the data sources (manual or automated process).
4. Agree the authoritative source if there is more than one source for the same data.
5. Identify and document relationship between sources.
6. Document steps 1 to 5.

The following table provides a practical example of the steps for a report on the total count of computers by the Windows operating system. Workgroup computers and computers not in the Active Directory domain are out of scope of this report request.

<table>
<thead>
<tr>
<th>Report input type</th>
<th>Details</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data source</td>
<td>Asset Database</td>
<td>Populated manually by the purchase order team</td>
</tr>
<tr>
<td>Data source</td>
<td>Active Directory</td>
<td>Automatically populated. Orchestrator runbook performs a scheduled clean-up of disabled objects</td>
</tr>
<tr>
<td>Data source</td>
<td>System Center Configuration Manager</td>
<td>Requires an agent and currently not used to manage servers</td>
</tr>
<tr>
<td>Authoritative source</td>
<td>Active Directory</td>
<td>Based on the report scope</td>
</tr>
<tr>
<td>Data source relationship</td>
<td>Microsoft System Center Configuration Manager is configured to discover all systems in the Active directory domain</td>
<td>Alternative source for the report using the All systems collection</td>
</tr>
</tbody>
</table>

Plan to document the specific fields you need from the authoritative data source. For example, use a table similar to the following.

<table>
<thead>
<tr>
<th>Required data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer name</td>
<td>The Fully Qualified domain name of the computer</td>
</tr>
<tr>
<td>Operating system</td>
<td>Friendly operating system name</td>
</tr>
<tr>
<td>Operating system environment</td>
<td>Server or workstation</td>
</tr>
<tr>
<td>Date created in data source</td>
<td>Date the computer joined the domain</td>
</tr>
<tr>
<td>Last logon date</td>
<td>Date the computer last updated the attributes in Active Directory</td>
</tr>
</tbody>
</table>
Understanding the Goals of Reporting

The steps provided discuss an example of identifying input sources and the fields you plan to use in a requested report.

Optimizing Report Inputs

Once the required data for your reports have been identified and documented, you must test for validity and consistency. Data sources which are populated by automated processes tend to be less prone to consistency errors. Conversely, data sources based on manual entry are prone to errors (for example, correct spelling when typing text into forms used to populate the data source). Here are typical recommended practices for improving consistency in manual and automated system populated data sources:

- Automated (for example, agent based):
  1. Implement agent health check and remediation.
  2. Include last agent update information in reports.

- Manual entry:
  1. Avoid free text fields, except description or notes.
  2. Use a list picker.
  3. Implement mandatory constraints on required fields (for example, a request for e-mail address should only accept the right format for e-mail addresses).

How it works...

The reports you create and manage are only as accurate as the original data source. There may be one or more sources available for a report. The process discussed in this recipe provides steps on how to narrow down the list of requirements. The list must include the data source and the specific data fields which contain the data for the proposed report(s). These input fields are populated by manual, automated processes or a combination of both.

The final part of the recipe discussed an example of how to optimize the inputs you select.

These steps will assist in answering one of the typical questions often raised about reports: "Can we trust this information?" The answer, if you have performed these steps will be "Yes, and this is why and how."

See also

- The Planning report outputs recipe
Planning report outputs

The preceding recipe, *Planning and optimizing dependent data inputs*, discussed what you need for a report. This recipe builds on the preceding recipes with a focus on how you plan to view a report (output).

**Getting ready**

Plan to review the *Understanding the goals of reporting* and *Planning and optimizing dependent data inputs* recipes.

**How to do it...**

The type of report output depends on the input you query from the target data source(s). Typically, the output type is defined by the requestor of the report and may be in one or more of these formats:

- List of items (tables)
- Charts (2D, 3D, and formats supported by the reporting program)
- Geographic representation
- Dials and gauges
- A combination of all the listed formats

Here is an example of the steps you must perform to plan and agree the reporting output(s):

1. Request the target format from the initiator of the report.
2. Check the data source supports the requested output.
3. Create a sample dataset from the source.
4. Create a sample output in the requestor's format(s).
5. Agree a final format or combination of formats with the requestor.
Understanding the Goals of Reporting

The steps to plan the output of reports are illustrated in the following figure:

These are the basic minimal steps you must perform to plan for outputs.

**How it works...**

The steps in this recipe are focused on scoping the output of the report. The scope provides you with the following:

- Ensuring the output is defined before working on a large set of data
- Validating that the data source can support the requested output
- Avoids scope creep. The output is agreed and signed off

The objective is to ensure that the request can be satisfied based on what is available and not what is desired. The process also provides an additional benefit of identifying any gaps in data before embarking on the actual report creation.

**There's more...**

When planning report outputs, you may not always have access to the actual source data. The recommend practice is not to work directly with the original source even if this is possible to avoid negatively impacting the source during the planning stage. In either case, there are other options available to you. One of these options is using a spreadsheet program such as Microsoft Excel.
Mock up using Excel

An approach to testing and validating report outputs is the use of Microsoft Excel. You can create a representation of the input source data including the data type (numbers, text, and formula). The data can either be a sample you create yourself or an extract from the original source of the data.

The added benefit is that the spreadsheet can serve as a part of the portfolio of documentation for the report.

See also

- The Understanding dashboards and their dependencies recipe

Understanding dashboards and their dependencies

The Planning report outputs recipe discussed what you need to do to plan for report outputs.

This recipe builds on the preceding recipe highlighting the requirements and dependencies for creating report dashboards. An understanding of dashboards and their dependences is required to ensure that your intended dashboards are fit for purpose and fit for use.

Getting ready

Plan to review the Understanding the goals of reporting, Planning and optimizing dependent inputs, and Planning report outputs recipes.

How to do it...

The term dashboards usually denotes the use of a graphical image to represent data. There are typically two types of dashboards:

- Static
- Interactive

Static dashboards

Static dashboards present an image of the report but does not provide an interactive mechanism. These types of dashboards are similar to charts you create with data in a spreadsheet program (for example, Microsoft Excel).
Interactive dashboards

Interactive dashboards are similar to static dashboards but provide the ability to traverse from the initial dashboard to other dashboards or data tables. This process is commonly described as a drilldown.

In either case the dashboard is a final output of the report creation process. The following figure provides an example of an incident management process dashboard:

The incident management dashboard presented in the preceding figure has the following dashboard sections:

- Incidents Opened Today
- Incidents Closed Today
- Total Open this Month
- Total Closed this Month
- 1st Level Fixed this Month

Ask the right questions to get an understanding on how to create this type of dashboard. We'll using the first section as an example: Incidents Opened Today:

- Is this related to only incidents created today or will it also included incidents reactivated today?
- How can the difference highlighted in the preceding question be validated?
- Is today related to working days or every day of the week?
- What is the smallest and largest display area this dashboard will be viewed on?
- Who is accountable for the data the dashboard depends on?

Build on the example questions using the scope of the specific report request.
**How it works...**

Dashboards share similar attributes to films (movies). A film requires a script, the actors, directors, the set, and the equipment. In addition the film requires a rehearsal and a number of processes which are best described by experts in that field. The final output which can be viewed at the cinema or digital media is the films’ dashboard.

The focus of this recipe is to highlight the fact that dashboards are only as good as the processes and inputs they depend on. Often the dashboard requestor does not have the same background, or in-depth understanding of the data the report creator requires.

Use the questioning technique in this recipe to bridge the gap between the requestor and the creator. The process will ensure that the expectations of the requestor are aligned with the actual effort required to create the dashboard.

An additional factor is the validity and ownership of the information the dashboard will depend on. A dashboard regardless of type, which presents the wrong information will only lead to chaos. Identifying the owner of the information will provide the report creator with an escalation path as well as a policy enforcement option.

**See also**

- Chapter 8, *Creating System Center Advanced Reports*, provides recipes on dashboards using System Center data inputs

**Targeting reports at decision stakeholders**

This recipe discusses the recipients of reports and the objectives the recipients are typically aiming to achieve.

**Getting ready**

The prerequisite to this recipe is a basic understanding of the term "stakeholders" as it relates to reports.

**How to do it...**

The process of targeting stakeholders for reports is based on the following three categories of reporting:

- Proactive reporting
- Reactive reporting
- Highlight reporting
Understanding the Goals of Reporting

Proactive reporting
This category of reporting is based on highlighting potential issues before it impacts the business. The stakeholders you will target for this type are decision makers with influence. Examples of the stakeholders in this category are:

- Business unit directors
- IT security teams
- Chief executives

Proactive reporting may require forward thinking and advance expenditure, so it is critical to win over the right stakeholders.

Reactive reporting
Reactive reporting usually has many stakeholders. Examples of the stakeholders in this category are:

- Data center managers
- IT security teams
- Asset and problem management teams
- Business application owners

Examples of reports in this category are security non-compliance, license non-compliance, capacity issues, and hardware audits.

Highlight reporting
This is the "showing off" type of report and the stakeholders are typically trying to sell their value to the business. *If no one knows then you are not doing it.* The stakeholders for these type of reports are those identified in the first two categories but also includes ad hoc requests.

An approach to targeting reports to stakeholders is to first categorize reports. Allocate primary and secondary recipients/or owners for each category. The following table is an example of such a categorization exercise:

<table>
<thead>
<tr>
<th>Report category</th>
<th>Report(s)</th>
<th>Primary owner</th>
<th>Secondary owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity management</td>
<td>• Annual disk usage</td>
<td>Data Center Manager</td>
<td>Executive Budgetary board</td>
</tr>
<tr>
<td></td>
<td>• Network bandwidth utilization by location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Report category</td>
<td>Report(s)</td>
<td>Primary owner</td>
<td>Secondary owner</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>--------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>Asset management</td>
<td>▶ List of all client operating systems in use</td>
<td>Procurement Manager</td>
<td>Facilities and Clients Services</td>
</tr>
<tr>
<td></td>
<td>▶ Count of users issued out of hours door access</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitoring and availability</td>
<td>The <strong>Line of Business</strong> (LOB) application outages this month</td>
<td>LOB department manager</td>
<td>IT operations director</td>
</tr>
<tr>
<td>Security and policy compliance</td>
<td>List of clients with required security updates older than 60 days</td>
<td>Head of global security</td>
<td>Client Services manager</td>
</tr>
<tr>
<td>Incident and problem management</td>
<td>▶ Number of incidents resolved without escalation</td>
<td>Service Delivery manager</td>
<td>Service Desk team lead</td>
</tr>
<tr>
<td></td>
<td>▶ List of problem records by date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change and release management</td>
<td>Percentage of successful changes completed in the allocated change window</td>
<td>Change Manager</td>
<td>Change advisory board</td>
</tr>
</tbody>
</table>

**How it works...**

The process of targeting reports may be proactive or reactive. The recipe provides you with the process you can follow to ensure the majority of your reports fall in the proactive category. The reactive category is unavoidable but the categorization and ownership of reports is invaluable even in this category.

Plan to adopt and adapt the information provided to your specific requirements and environment.
This recipe is a summation of all the recipes in this chapter. This recipe completes the loop with a discussion and example on documenting a report design.

**Getting ready**

You must plan to review and perform the steps in the preceding recipes in this chapter:

- Understanding goals of reporting
- Planning and optimizing dependent data inputs
- Planning report outputs and their dependencies
- Targeting reports at decision stakeholders

**How to do it...**

Here is the scenario used in this recipe to provide the steps for documenting a report design.

**Scenario:** A service request has been received by the service desk team to create a report which shows how many incidents are raised each month and each quarter.

Here are the steps you can follow to document the requirements of this scenario:

1. Create a document to capture the requirements using your favorite editor.
2. Plan to have the following sections in the document:
   - Report Owner
   - Data source and owner
   - Required output and format
   - Frequency of report
   - Review timeline for the report
3. Follow the steps discussed in the preceding recipes in this chapter. Include the tables in the document or create a spreadsheet where appropriate. The following figure is an illustration of the areas to focus on based on the recipes in this chapter:

![Diagram of reporting process]

**How it works...**

The documentation process approach discussed serves as a guideline and a high-level template. Reporting requirements tend to change as the business or environment evolves. There are parts of the reporting process that tend to be less prone to change (for example, owners of the report and the data sources).

The documentation process should be part of the change management process for the report. Create a baseline document for new reports and continually update the document as the requirements are updated.

One of the sections with high benefit is the review timeline for the document. Set a review date with a reminder for the document.

Documenting the report design saves time and keeps the effort focused.

**See also**

- Appendix A lists additional resources you can use to enhance your knowledge on understanding the goals of reporting
Where to buy this book

You can buy Microsoft System Center Reporting Cookbook from the Packt Publishing website.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

Click here for ordering and shipping details.