

UCS Content Models

March 31, 2010



Contents

Chapter 1	Topic content models	5
	About topic-based writing	5
	About topics in general	6
	About core topic types	6
	About concept topics	8
	About task topics	9
	About reference topics	10
	About process topics	11
	About process topic tables	13
	Title guidelines for topic types	15
Chapter 2	Chapter content models	19
	About chapter content models	19
	Introducing <i>product name</i> chapter content model	20
	Planning for installation chapter content model	20
	Installing <i>product name</i> chapter content model	21
	Key task chapter content model	22
Chapter 3	Deliverable content models	25
	About deliverable content models	25
	Administration guide content model	26
	Client guide content model	27
	Deployment guide content model	27
	Getting started content model	28
	Hardware content models	28
	Implementation guide content model	29
	Installation guide content model	29
	Migration guide content model	30
	Readme content model	31
	Reference guide content model	31
	Release notes content model	32
	Third-party license agreement content model	34
	Troubleshooting guide content model	34
	User guide content model	35

Index 37

Topic content models

This chapter includes the following topics:

- [About topic-based writing](#)
- [About topics in general](#)
- [About core topic types](#)
- [About concept topics](#)
- [About task topics](#)
- [About reference topics](#)
- [About process topics](#)
- [About process topic tables](#)
- [Title guidelines for topic types](#)

About topic-based writing

Topic-based writing is a structured and modular approach to content development where topics are the basic units. It is a way to develop focused and concise units of information that anticipate and target specific user goals. The topic-based writing approach is designed to support all output formats, from PDF with more linear book-like outlines to Help or Web content. Topics can be used as stand-alone units or as building blocks that can be grouped into larger units such as chapters, books, or parts. Topics are connected to other topics by cross-references to provide context and navigation.

See [“About topics in general”](#) on page 6.

See [“About core topic types”](#) on page 6.

About topics in general

A topic is the basic building block in the topic-based writing approach. It is the basic unit of all the deliverables that InfoDev builds and the smallest unit of reuse. A topic corresponds to a section in Vasont. It consists of a title and the content that follows the title.

Create the content of the topic with the following principles of effective technical communication and learning theory:

- Users and readers process information in small, manageable units.
- Writers need to create and label manageable units of content called topics.
- Every topic needs to focus on only one idea.
- All information in a topic must be relevant to the idea.

A topic has the following characteristics:

- Fully answers a single question that is related to a user goal
- Treated as a unit, often seen by users on a single screen
- Can be accessed individually, often by multiple methods
- Has a recognizable, predictable structure
- Cross-references to related information in other topics

See [“About topic-based writing”](#) on page 5.

See [“About concept topics”](#) on page 8.

See [“About task topics”](#) on page 9.

See [“About reference topics”](#) on page 10.

See [“About process topics”](#) on page 11.

About core topic types

Symantec InfoDev uses information typing to distinguish between conceptual, task-focused, and reference types of content. This kind of content architecture is based on industry standards and is designed to assist users in finding exactly the information they are looking for. The following core topics support the different types of information:

Concept	A concept topic provides conceptual background information that users need to perform tasks successfully.
---------	---

What do I need to know about the master server?

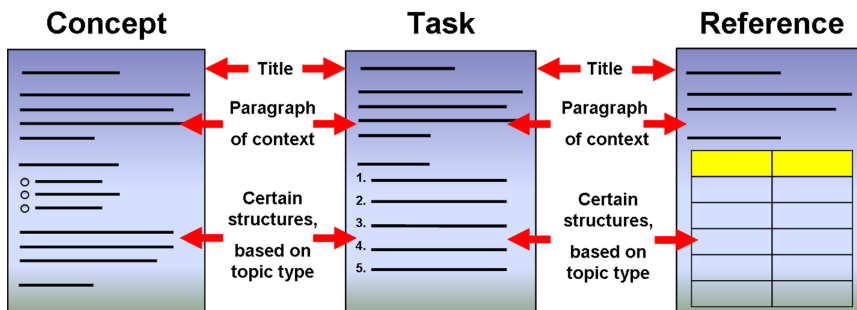
- Task** A task topic provides the step-by-step information that users need to perform tasks successfully.

Installing or configuring the master server
- Reference** A reference topic provides the data or reference information that users need to perform task successfully.

Parameters, logs, error messages for the master server

The core topic types have recognizable structures. For example, every topic has a heading as a necessary component. Every topic should have a paragraph of body text that provides an introduction for the topic and some context for the remaining information. Every topic's information type will be identifiable by certain XML elements. For example, conceptual information belongs in paragraphs and bullet lists. Task information always belongs in a procedure, which has numbered steps. Reference information is collected in a table so it is easy to scan.

Figure 1-1 Recognizable structures of core topics types



Without even being able to read the text of a topic, it is possible to determine what core topic type it is, based only on the topic's structure. In other words, the elements in a topic model indicate what kind of information is contained in the topic.

- See [“About topic-based writing”](#) on page 5.
- See [“About concept topics”](#) on page 8.
- See [“About task topics”](#) on page 9.
- See [“About reference topics”](#) on page 10.

About concept topics

Concept topics describe ideas, things, and processes. New users read concept topics and experienced users skip them. They are more likely to be read offline, in print, while the user works on something other than a task within the product. Concept topics support task topics and create context for them along with reference, and process topics.

Putting conceptual information in separate topics allows users to access it directly, or through a cross-reference from a task topic or a process topic. This practice also helps the content developer maintain the information more easily.

Concept topics answer the following types of questions:

- What is important to know about a particular thing?
- What does that thing do?
- How does that thing work?
- What is the difference between two things?
- How are they related?
- What phases make up a particular process?
- What do you have to know to perform the process?
- How can I reach my goal?
- What do I need to do and when?

Include the following in concept topics:

- Narrative descriptions
- Definitions
- Overview information and explanations
- Illustrations, flowcharts, decision trees
- Cross-references to related topics

Do not include the following in concept topics:

- Lengthy reference information
- Any screen shots (usually)
- Procedures

See [“About core topic types”](#) on page 6.

See [“About task topics”](#) on page 9.

See [“About reference topics”](#) on page 10.

See [“About process topics”](#) on page 11.

About task topics

Task topics are the essence of user documentation. Task topics are supported by concept, reference, and process topics.

Task topics describe when, why, and how to do something with step-by-step instructions. New users consult task topics to learn how to do something. Experienced users reference task topics to remind themselves how to do something or to make sure they don't forget a step.

Task topics are most likely to be used while user is trying to accomplish task. You structure task topics so that they are easy to skim. They are used, not read.

Task topics answer the following types of questions:

- How do I do something?
- When do I do particular steps?
- Why do I do something in a particular order?

Include the following in task topics:

- Some Contextual information
- A procedure
- Minimal screenshots
- Tables
- Lists
- Cross-references to related topics, including the process topic, if task is part of a process.
Note: When the cross-reference does not relate to the body text, a cross-reference to a related topic or set of topics can be placed after a procedure.
- If the task topic includes multiple procedures, include a bulleted list or table that describes when and why the procedures must be performed. Include cross-references to each procedure.
- Match the procedure title to the title of the topic.

Do not include the following in task topics:

- Lengthy narrative description
- Lengthy reference information

Putting procedural information in a separate topic serves users and content developers in the following ways:

- Users can locate and access it directly or by cross-reference from another topic
- Users can focus on procedural details to complete the task quickly
- Content developers can maintain it more easily

See “[About core topic types](#)” on page 6.

See “[About concept topics](#)” on page 8.

See “[About reference topics](#)” on page 10.

See “[About process topics](#)” on page 11.

About reference topics

Reference topics provide information that is rarely used or too detailed to remember. Reference topics assist users while they perform a task and a lack of information blocks them from completing the task successfully. Information in reference topics is often tabular and ordered numerically or alphabetically. Advanced users often consult reference topics to quickly look up details. Advanced users know what to do with the information, they just need a reminder or some specific detail.

Reference topics can answer the following types of example questions:

- What are the database table definitions?
- What are the error messages?
- What are the log file locations?
- What are the specific requirements for fields in a UI?

Include the following in reference topics:

- Introductory text to provide context
- Tables
- Lists
- Cross-references to related topics

Do not include the following in reference topics:

- Lengthy narrative descriptions
- Any screenshots
- Any procedures

Structuring reference information into a separate topic helps users and content developers in the following ways:

- Users can locate and access reference information directly or by cross-reference, most likely from a task topic

- Content developers can maintain reference information more easily as the information is frequently created by engineers or generated directly from program code

See [“About core topic types”](#) on page 6.

See [“About concept topics”](#) on page 8.

See [“About task topics”](#) on page 9.

About process topics

A process is a high level user task made up of a series of actions, events, stages, or phases. A process takes place over time and has an identifiable result. A process usually involves one or more persons or things.

The process topic is a high-level description of the process that users need to follow to achieve their overall goal. It serves as a navigational tool that outlines phases in a solution and ties together tasks and concepts. It is not a task or a procedure. It helps readers understand why they would do certain tasks and the sequence in which to do the tasks. At Symantec, process topics are used for everything from installation, configuration, and customization to performing software deliveries and managing virtual applications.

See [“About core topic types”](#) on page 6.

The process topic model answers the following types of user questions:

- How can I reach my goal?
- What do I need to do and when?
- Why is this process the recommended process?
- How does this process relate to or effect other processes?
- What does the final outcome look like?
- What are the consequences of not following this process?

Processes have the following types:

Linear	Describes the tasks or events over a specific, sequential time period
Branching	Describes cause-and-effect relationships or the different courses of action
Cyclical	Describes the stages in a repetitive cycle

A process topic always includes a table that lists the stages of the process, an imperative statement that describes the task done at the stage, and a description

of what happens in each phase. The description cell also includes cross-references to specific tasks that a user must perform to complete the process. Also, the table can include cross-references to concept topics that describe things the user must understand to proceed from phase to phase. For long, complex processes, the table should also include a column that numbers the steps or phases.

See “[About process topic tables](#)” on page 13.

A process topic may have flowcharts or diagrams that show the progression through a system or to achieve a solution. A table always accompanies a flowchart or diagram in a process topic because it contains explanations and cross-references to related topics. A table can stand alone without a flowchart. Most linear processes are clear enough without a graphic. Complex branching and some cyclical processes can benefit from a graphic.

A process topic usually falls at the beginning of a chapter. It serves as a navigational hub that connects the many tasks that make up a larger process. It guides the user through various tasks and gives them context for important decisions.

Table 1-1 Components of a process topic

Elements	Purpose	Description/Cross-reference
Title	The title answers a user question such as: How do I...?	The title begins with a gerund and discusses tasks at a high level.
Paragraph(s)	The first paragraph provides context for the process.	The introductory paragraph reflects the context for all the tasks, concepts, and operations that make up the process. It describes the high-level user goal, planning considerations and recommendations for success.
Cross-reference (stand alone)	A cross-reference after the first paragraph ties the process to larger concepts related to it.	Users might need to understand a high-level concept. Cross-reference to it. Cross-reference to the first topic in Key Task Chapter or other relevant topic. Users might need to understand a related process. Cross-reference another process.

Table 1-1 Components of a process topic (*continued*)

Elements	Purpose	Description/Cross-reference
Figure (optional)	A flowchart, decision tree, or swim-lane representation can contain a visual summary of the phases in process.	Linear processes rarely benefit from graphics. Use graphics for cyclical, branching, and operational processes. All graphics must have a supporting table.
Table	The table names the phases of the process and provides descriptions of each phase. Number the phases in a separate column if there are more than four. The description column includes a brief summary of the action and cross-references to related task and concept topics.	For a high-level or a long linear process, use a table with gridlines, a header row, and three columns. Number the steps or phases in the first column. For a lower level or shorter process, use a table with gridlines, a header row, and two columns. Do not number the steps or phases. See “About process topic tables” on page 13.
Other elements as needed	You may need additional supporting information for the process. Use additional elements as necessary.	You can add more body text or bullet lists of considerations, for example. Do not include any procedures.

See [“About concept topics”](#) on page 8.

See [“About task topics”](#) on page 9.

See [“About reference topics”](#) on page 10.

About process topic tables

The process topic table serves as a navigational tool that outlines phases in a solution and cross-references users to related task and concept topics. Topics that are part of a process always include a cross-reference back to the process topic so the user can follow the steps or phases in order. Cross-references from related topics to the process topic also inform the user that a particular topic is part of a process, in case the user does not know.

In a process topic, you may use either the table or the informaltable element to create a table. If the process you are documenting contains 2-3 phases or steps,

use the informaltable element; it is not necessary to number the steps or phases. If the process is longer than 4 steps, use a complex table and number the steps or phases in the first column. Use the word "step" or "phase" before each Arabic number.

Figure 1-2 shows components of a process table for a high-level or a long process.

Figure 1-2 Components of a process topic table

Table X-X **Table title**

Step	Action	Description
Step X or Phase X	A brief explanation of the action required at the step or phase.	A brief explanation of the step or phase. A cross reference to the topic containing the specific information about the step or phase.

Figure 1-3 Example of a process topic table

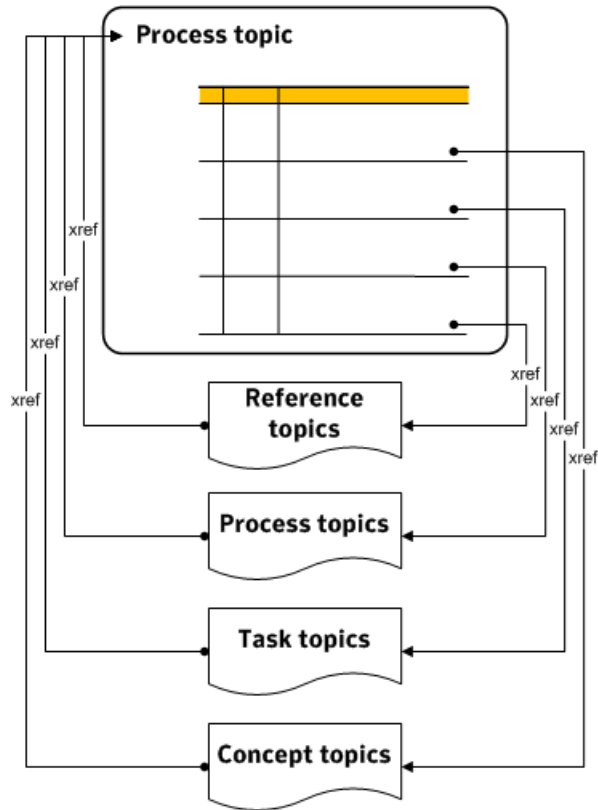
Table 2-1 summarizes the installation steps for Symantec Endpoint Protection Small Business Edition.

Table 5-2 **Installation planning**

Step	Action	Description
Step 1	Plan network architecture	Identify the computers on which you want to install Symantec Endpoint Protection Small Business Edition. See "Network architecture considerations" on page 23.
⋮		
Step 4	Prepare computers for installation	Uninstall other virus protection software from your computers. See "Preparing your computers for installation" on page 28.
Step 5	Identify installation settings	Identify the user names, passwords, email addresses, and other installation settings. Have the information during the installation. See "About the Symantec Protection Center installation settings" on page 57. See "About the client installation settings" on page 32.
Step 6	Install server	Install Symantec Protection Center. See "Installing Symantec Protection Center" on page 38.
⋮		

Figure 1-4 shows how to adopt a one-to-many/many-to-one paradigm to eliminate the need for cross-references between every related topic. In the process topic table, cross-reference to each step or phase topic. In each step or phase topic, cross-reference back to the process topic.

Figure 1-4 How to cross-reference a process topic table



See [“About process topics”](#) on page 11.

Title guidelines for topic types

Each topic type has specific and individual functions. They also have unique and recognizable structures. Topic titles guide the users to the kind of information they need. Each type of topic requires a different kind of title.

Table 1-2 Summary of topic types, title guidelines, and functions

Topic type	What to know about the title	Topic function
<p>Concept</p>	<p>Titles of concept topics can begin with the word "About." They cannot begin with a gerund.</p> <p>The following are examples of titles for concept topics:</p> <ul style="list-style-type: none"> ■ About Symantec LiveState Delivery ■ About the Software Catalog ■ About centralized management 	<p>The following characteristics define the function of concept topics:</p> <ul style="list-style-type: none"> ■ Describes ideas, things, and user goals ■ Most often used by new users ■ Skipped by experienced users ■ More likely to be read offline, in print, while user is not working or trying to complete task
<p>Task</p>	<p>Titles of task topics must begin with a gerund (-ing). The title of a task topic should mirror the title of the procedure inside of the task topic.</p> <p>The following are examples of titles for task topics:</p> <ul style="list-style-type: none"> ■ Installing client type software on a master server ■ Creating a Managed Delivery policy ■ Configuring security settings 	<p>The following characteristics define the function of task topics:</p> <ul style="list-style-type: none"> ■ Describes when, why, and how to do something ■ Used by new users to learn how to do something ■ Referenced by experienced users to as a reminder about how to do something or to make sure they don't forget a step ■ Mostly used while a user is trying to accomplish task ■ Easy to skim; they are used, not read
<p>Reference</p>	<p>Titles of reference topics begin with a noun. They should not begin with a gerund, nor the word "About."</p> <p>The following are examples of titles for reference topics:</p> <ul style="list-style-type: none"> ■ Event mapping ■ Error logs ■ Schema packages 	<p>The following characteristics define the function of reference topics:</p> <ul style="list-style-type: none"> ■ Provides information that is either too detailed to remember or information that is rarely used ■ Often tabular in format ■ Often used by more advanced users who know what to do with the information, but just need a quick detail ■ Most likely to be referenced while the user is performing a task. A lack of information often blocks them from completing the task successfully

Table 1-2 Summary of topic types, title guidelines, and functions (*continued*)

Topic type	What to know about the title	Topic function
Process	<p>Titles of process topics begin with a gerund, like task topics, because proces topics answer the user question, "How do I?" Process topics titles should not begin with the word "About."</p> <p>The following are examples of titles for process topics:</p> <ul style="list-style-type: none"> ■ Performing advanced software deliveries ■ Populating the software catalog ■ Redirecting a restore to a different client 	<p>The following characteristics define the function of process topics:</p> <ul style="list-style-type: none"> ■ Contains high-level description of series of tasks user must follow to achieve their goal ■ Outlines phases, ties together tasks, orders tasks, and provides context and navigation to the user ■ Includes flowcharts, tables, and itemized lists ■ Includes cross-references to task and reference topics

See [“About concept topics”](#) on page 8.

See [“About task topics”](#) on page 9.

See [“About reference topics”](#) on page 10.

Chapter content models

This chapter includes the following topics:

- [About chapter content models](#)
- [Introducing product name chapter content model](#)
- [Planning for installation chapter content model](#)
- [Installing product name chapter content model](#)
- [Key task chapter content model](#)

About chapter content models

A chapter contains a set of related topics (a topic cluster). A chapter may contain several smaller topic clusters, if the clusters cover related subjects.

When you create a chapter in a deliverable, consider whether the chapter contains all necessary topics to document the subject matter, and whether the chapter structure makes the information easy to find and use.

If a chapter contains many topics, you may want to break the topics out into separate chapters. If a deliverable contains many short chapters, you may want to combine some chapters or group multiple chapters into parts. In some cases, you may decide to create a separate deliverable for some of the information (for example, a separate Migration Guide).

See [“About deliverable content models”](#) on page 25.

In our topic-based publishing paradigm, deliverable-based criteria such as the chapter length in pages are no longer relevant. Topics are the building blocks, and a chapter is judged by the completeness of the information included in the chapter.

See [“About topic-based writing”](#) on page 5.

Several of the most common types of chapters are defined in chapter content models.

See [“Introducing *product name* chapter content model”](#) on page 20.

See [“Installing *product name* chapter content model”](#) on page 21.

See [“Planning for installation chapter content model”](#) on page 20.

See [“Key task chapter content model”](#) on page 22.

Introducing *product name* chapter content model

The Introducing *product name* chapter provides a general description of the product. It describes at a high level what users can do with the product and how it works. This type of chapter is designed to inform rather than instruct and contains concept topics rather than task topics. Because it serves as an introduction to the product, this chapter is included as the first chapter in many types of deliverables.

See [“About deliverable content models”](#) on page 25.

An Introducing chapter contains the following sections in the order listed:

- About *product name*
- What’s new in *product name* (optional)
- Components of *product name* (if applicable)
- How *product name* works
- What you can do with *product name*
 - Lists key tasks and processes with cross-references to the appropriate task topics
- Where to get more information about *product name*

You can clone a template of the Introducing *product name* chapter from the Templates collection in Vasont.

See [“About chapter content models”](#) on page 19.

Planning for installation chapter content model

Planning for installation is an optional chapter content model. You use this chapter when you need to provide multiple topics that inform users about the considerations and decisions that need to be made before installation or deployment. This chapter can be included in an installation guide, an implementation guide, or a deployment guide.

The information in this chapter instructs users on how to do certain planning activities. It does not instruct users to do anything procedural with the software itself; instead, it includes cross-references to task topics in other chapters when applicable.

A Planning for installation chapter contains the following sections in the order listed:

- Planning your network architecture
- Planning for *process* (1-n)
- System requirements

Depending on the nature of the product, additional topics that are specific to installation planning may also be included.

See [“Installation guide content model”](#) on page 29.

See [“Implementation guide content model”](#) on page 29.

See [“Deployment guide content model”](#) on page 27.

See [“About chapter content models”](#) on page 19.

Installing *product name* chapter content model

The Installing *product name* chapter model informs users about what has to happen before, during, and after installation. It tells users what to expect during installation and how to install all product components. This chapter may inform users of the most important tasks to perform immediately after installation, but it does not describe the product or what you can do with it.

If applicable, the Installing *product name* chapter also instructs users on how to do the following:

- Update previous versions of the product
- Configure the product
- Roll out installation to other computers

The Installing *product name* chapter should also instruct users on how to uninstall the product.

An Installing *product name* chapter contains the following sections in the order listed:

- Before you install
- System requirements
- Accessing *product name* (optional)

For licensing information, if applicable

- Installing the *product name* product (1-n)

May require several topics, each of which describes how to install a separate component

- Post-installation tasks
- Migrating to *X* (optional)
- Uninstalling *product name*

You can clone a template of the Installing *product name* chapter from the Templates collection in Vasont.

See [“About chapter content models”](#) on page 19.

Key task chapter content model

A key task is a major task or process that users must perform to use Symantec software or hardware. The Key task chapter contains all necessary topics to inform users about the task or process, why it is important, and when and how to perform it. If applicable, it should include any requirements or decision options the user should know about. It should describe all of the tasks involved in the major task or process.

Examples of key task chapters include:

- Updating *product name*
- Configuring *product name*
- Configuring scanner settings
- Filtering spam
- Managing alerts
- Creating and viewing reports

Depending on the the complexity of the key task or the number of variables, a Key task chapter may include multiple concept topics and task topics. A process topic may be required if multiple tasks must be completed. A Key task chapter may or may not require reference topics, depending on the nature of the task.

See [“About core topic types”](#) on page 6.

See [“About process topics”](#) on page 11.

A Key task chapter contains the following types of sections, the order of which may vary based on the content:

- [1-n] concept topics
- [1-n] process topics (optional)
- [1-n] task topics that support completion of the key task
- [1-n] reference topics (optional)

See [“About chapter content models”](#) on page 19.

Deliverable content models

This chapter includes the following topics:

- [About deliverable content models](#)
- [Administration guide content model](#)
- [Client guide content model](#)
- [Deployment guide content model](#)
- [Getting started content model](#)
- [Hardware content models](#)
- [Implementation guide content model](#)
- [Installation guide content model](#)
- [Migration guide content model](#)
- [Readme content model](#)
- [Reference guide content model](#)
- [Release notes content model](#)
- [Third-party license agreement content model](#)
- [Troubleshooting guide content model](#)
- [User guide content model](#)

About deliverable content models

Deliverable content models are guidelines rather than templates. Product team needs may vary, which dictates what is included in your final deliverables.

The InfoDev managers own the deliverable content models. As a need arises for a new or modified content model, an InfoDev manager should bring the issue to the InfoDev Managers' meeting. At the meeting, a workgroup can be identified to propose the new model.

When a proposal is agreed on, the symTOP architecture team must review the model to ensure that it is DTD-compliant.

Symantec InfoDev has developed deliverable content models that provide the following benefits:

- Symantec customers know what to expect from each type of guide.
- The guides present consistent content which can be repurposed in cross-product documentation sets or a multi-product Help system running on a console.

See [“Administration guide content model”](#) on page 26.

See [“Client guide content model”](#) on page 27.

See [“Deployment guide content model”](#) on page 27.

See [“Getting started content model”](#) on page 28.

See [“Implementation guide content model”](#) on page 29.

See [“Installation guide content model”](#) on page 29.

See [“Migration guide content model”](#) on page 30.

See [“Readme content model”](#) on page 31.

See [“Release notes content model”](#) on page 32.

See [“User guide content model”](#) on page 35.

Administration guide content model

An administration guide provides users with enough information to configure the product, optimize performance, perform key tasks, and maintain the product. An administration guide is always provided along with an installation guide. If you do not have enough information for separate guides, use an implementation guide instead.

See [“Implementation guide content model”](#) on page 29.

An administration guide has the following deliverable structure:

- Front matter (copyright, license, warranty, service, and support) pulled from Boilerplate Collection
- Introducing *product name* chapter (optional)

See “[Introducing *product name* chapter content model](#)” on page 20.

- Key task chapter(s)
See “[Key task chapter content model](#)” on page 22.
- Glossary (optional)
- Index

See “[About deliverable content models](#)” on page 25.

Client guide content model

A client guide provides the minimum amount of information needed to install or use a managed product (client on a client/server network). Information in a client guide is typically included in an administration guide. Use a client guide when end users need to perform installation or key tasks.

A client guide has the following structure:

- Front matter (copyright, license, warranty, service, and support) pulled from Boilerplate Collection
- Installing *product name* chapter (optional)
This chapter is only necessary if the client is not automatically installed.
See “[Installing *product name* chapter content model](#)” on page 21.
- Key tasks chapter(s)
See “[Key task chapter content model](#)” on page 22.
- Index

See “[About deliverable content models](#)” on page 25.

Deployment guide content model

A deployment guide introduces and provides support for deployment issues that are found in the field (for example, sizing, tuning, and setup).

A deployment guide meets the following criteria:

- Is specific to a customer scenario
- Is for an internal sales engineer (SE) audience or the customer equivalent
- Contains information on planning only (no maintenance info)

A suggested deployment guide structure is as follows:

- Introducing deployment issues

- About product factors
- Necessary equipment
- Developing a deployment plan
- About integrating and deploying a complete solution

See “[About deliverable content models](#)” on page 25.

Getting started content model

A getting started deliverable provides users with enough information to install the product or to prepare to install the product. If this information is to be used in a portfolio, consult with Branding about length and other content considerations.

See the following document for Branding guidelines:

http://infodev.corp.symantec.com/infodev/infowiki/New%20Enterprise%20pkging%20guidelines_03_06.pdf

A getting started deliverable has the following structure:

- Front matter (copyright, license, warranty, service and support) pulled from Boilerplate Collection
- Introducing *product name* chapter
See “[Introducing *product name* chapter content model](#)” on page 20.
Depending on what your team feels is necessary and length considerations, you may include only some of the information in this chapter model.
- Installing *product name* chapter
See “[Installing *product name* chapter content model](#)” on page 21.
Depending on what your team feels is necessary and length considerations, you may include only some of the information in this chapter model.

See “[About deliverable content models](#)” on page 25.

Hardware content models

If you are documenting a product that is shipped with hardware, safety information may be required by law. In this case, the information should be provided by the hardware manufacturer. If you are instructed differently, discuss your needs with Karen Goldsmith.

See “[About deliverable content models](#)” on page 25.

Implementation guide content model

An implementation guide is designed for users who are responsible for installing and implementing a product or solution.

It provides users with enough information to do the following:

- Plan for implementation.
- Implement and deploy a complete solution for network security or for a component of network security.
- Optimize performance, perform key tasks, and maintain and upgrade the product or solution.

An implementation guide contains the same content as an installation guide and an administration guide. Use an implementation guide when you do not have enough content for separate guides.

See [“Installation guide content model”](#) on page 29.

See [“Administration guide content model”](#) on page 26.

An implementation guide has the following deliverable structure:

- Front matter (copyright, license, warranty, service and support) pulled from Boilerplate Collection
- Introducing *product name* chapter
See [“Introducing *product name* chapter content model”](#) on page 20.
- Planning for installation chapter (optional)
See [“Planning for installation chapter content model”](#) on page 20.
- Installing *product name* chapter
See [“Installing *product name* chapter content model”](#) on page 21.
- Migrating or upgrading chapter (optional)
- Key task chapters
See [“Key task chapter content model”](#) on page 22.
- Glossary (optional)
- Index

See [“About deliverable content models”](#) on page 25.

Installation guide content model

An installation guide is designed for users who are responsible for planning an installation and installing a product or solution. An installation guide is always

provided in conjunction with an administration guide. If you do not have enough information for separate guides, use an implementation guide instead.

See “[Implementation guide content model](#)” on page 29.

An installation guide has the following deliverable structure:

- Front matter (copyright, license, warranty, service and support) pulled from Boilerplate Collection
- Introducing *product name* chapter
See “[Introducing *product name* chapter content model](#)” on page 20.
- Planning for installation chapter(s) (optional)
See “[Planning for installation chapter content model](#)” on page 20.
- Installing *product name* chapter(s)
See “[Installing *product name* chapter content model](#)” on page 21.
- Migrating or upgrading chapter(s) (optional)
- Glossary (optional)
- Index

See “[About deliverable content models](#)” on page 25.

Migration guide content model

A migration guide is designed to inform customers how to migrate data and applications. For example, you might migrate from one platform to another, or from one technology stack to another. The migration process can also involve a product upgrade or an additional purchase that integrates with the base product.

Note: Upgrade information from one version of the product to another belongs with installation material.

See “[Installation guide content model](#)” on page 29.

See “[Implementation guide content model](#)” on page 29.

A suggested structure for a migration guide is as follows:

- Front matter (copyright, license, warranty, service and support) pulled from Boilerplate Collection
- Introducing migration issues
- Migrating from *x* to *y*

- Testing the migration
- Upgrading *x*
- Index

See [“About deliverable content models”](#) on page 25.

Readme content model

A readme has the following purposes:

- Contains last-minute issues that are found after the final build
- Contains all known issues that could cause difficulty in installing, configuring, or using the product out of the box

A readme can accompany full releases, point releases, maintenance updates, and hot fixes. A readme can be specific to a component of a product or to the product as a whole. Best practice for readmes is to use HTML format. PDF format is also acceptable. Text readmes are problematic because plain text can't reproduce structures such as tables.

A readme has specific boilerplate frontmatter that is available in the Boilerplate Collection.

See [“Release notes content model”](#) on page 32.

See [“About deliverable content models”](#) on page 25.

Reference guide content model

A reference guide provides detailed information for a specific audience. Use a reference guide for content with a limited audience or to separate some specialized content that may not be localized.

Examples of reference guide content include the following:

- Man pages
- Database table definitions

The title of a reference guide includes the subject between the product name and the guide type: *ProductName Subject Reference Guide* (for example, *Symantec NetBackup Command Line Interface Reference Guide*).

A reference guide has the following deliverable structure:

- Front matter (copyright, license, warranty, service, and support) pulled from Boilerplate Collection

- Introducing *product name* chapter
See “[Introducing product name chapter content model](#)” on page 20.
- Chapters or appendixes of primarily reference topics (optional)
- Appendixes of man page content (refentry elements), if a man page reference (optional)
- Index

The following are examples of how specific types of reference guides can be structured:

Man page reference guide	A guide titled <i>ProductName Command Line Interface Reference Guide</i> : <ul style="list-style-type: none">■ Front matter■ Introducing <i>product name</i> chapter■ Appendixes of man page content■ Index
Database table definitions reference guide	A guide titled <i>ProductName Database Table Definitions Reference Guide</i> : <ul style="list-style-type: none">■ Front matter■ Introducing <i>product name</i> chapter■ Chapters with groups of related database table definitions■ Index

This model is scalable. If you only have a few man pages or reference topics, you can include them in an appendix in another guide.

For man page references, some teams present man pages in a single appendix alphabetically. Other teams with large numbers of man pages group related man pages in separate appendixes. Note that to be compliant with DocBook, only refentries (man pages) or sections are allowed in an appendix, not both.

See “[About deliverable content models](#)” on page 25.

Release notes content model

Release notes inform customers about new features, changes between one version and another, and important issues at the time of ship. Release notes supplement existing documentation and can accompany full releases, point releases, maintenance updates, or hotfixes.

In general, release notes should not repeat content that is already included in the existing documentation. Instead, they should address any specific changes. For

example, there is no need to repeat the complete installation information that is also provided in a getting started or in an installation guide. On the other hand, if one or more steps in a procedure have changed, it would be useful to include the complete procedure for the convenience of the readers.

Release notes do not document or provide workarounds for bugs if that information is also provided in a readme file. We recommend not to include bug numbers unless you have specific PM requirements.

Note: Some teams provide technical release notes for internal release only that do contain bug numbers and detailed workarounds. The audience for these is Tech Support.

The specific content and structure of release notes may vary from team to team and should be based on their product requirements.

A suggested structure for a release notes deliverable is as follows:

- Front matter (copyright, license, warranty, service and support) pulled from Boilerplate Collection
- About *product name*
- What's new
- Release notes (1-n) such as the following:
 - Changes in product compatibility, supported platforms
 - Changes in pre-installation information such as system requirements
 - Changes in instructions for installing, uninstalling, upgrading, or configuring the product
 - Installation instructions for patches
 - Resolved issues (1-n)
 - Known issues and workarounds (1-n), including description and mitigation information as needed
 - Where to get more information

See [“Readme content model”](#) on page 31.

See [“Getting started content model”](#) on page 28.

See [“About deliverable content models”](#) on page 25.

Third-party license agreement content model

A third-party license agreement document contains the full text of license agreements for any third-party software that is distributed with the Symantec product. For example, a product that provides a Web interface may use the Apache open-source Web server software. If so, the legal department requires us to provide the full text of the applicable Apache license agreement.

The requirement for a third-party license agreement document should be specified in the InfoDev plan for a product release. The PgM is typically responsible for providing the legal department with the list of third-party software that the product uses. The legal department then supplies the license agreement text.

The required third-party license agreements must always be delivered as a separate, standalone PDF. The third-party license agreements cannot be in the form of a readme file or an appendix to an existing installation, administration, or other product guide. This requirement provides consistency and helps ensure that localization does not inadvertently localize third-party license agreements that are included in other product documentation. (The separate third-party license agreement document is not sent out for localization. The legal department provides localized versions of the third-party license agreements directly to localization.) This requirement also ensures that Symantec does not incur printing costs by including the third-party license agreements in a document that is printed.

Note: In Vasont, the Boilerplate collection includes a third-party license agreements book template that you can clone.

A third-party license agreement deliverable has the following structure:

- Front matter (copyright, license, warranty, service and support) pulled from Boilerplate Collection
- Chapter containing a separate section for each third-party license agreement

See [“About deliverable content models”](#) on page 25.

Troubleshooting guide content model

Most products do not require a separate troubleshooting guide. Troubleshooting information may be included in release notes, readmes, or even in one or two topics within a key task chapter or in an appendix. If you need a troubleshooting guide, it should list known product error messages or common issues and include a description of the problem and the solution.

A troubleshooting guide has the following structure:

- Front matter (copyright, license, warranty, service and support) pulled from Boilerplate Collection
- One or more chapters on error or status messages (1-n)
- One or more chapters on common issues and solutions

See [“About deliverable content models”](#) on page 25.

User guide content model

A user guide provides a complete guide to using a non-managed product. It may include installation information.

A user guide has the following structure:

- Front matter (copyright, license, warranty, service and support) pulled from Boilerplate Collection
- Introducing chapter
See [“Introducing *product name* chapter content model”](#) on page 20.
- Key task chapter(s)
See [“Key task chapter content model”](#) on page 22.
- Index

See [“About deliverable content models”](#) on page 25.

Index

A

administration guide content model 26

C

chapters

- content models for 19
- installation chapter model 21
- installation planning model 20
- introducing model 20
- key task model 22

client guide content model 27

concept topics 8

content models

- administration guide 26
- chapter 19
- client guide 27
- deliverable 25
- deployment guide 27
- getting started guide 28
- hardware (safety information) 28
- implementation guide 29
- installation guide 29
- migration guide 30
- readme 31
- reference guide 31
- release notes 32
- third-party license agreement 34
- topic 6
- troubleshooting guide 34
- user guide 35

D

deliverables

- content models for 25

deployment guide content model 27

G

getting started content model 28

I

implementation guide content model 29

installation guide content model 29

installing chapter content model 21

introducing chapter content model 20

K

key processes

- about 22

key task chapter content model 22

M

migration guide content model 30

P

planning chapter content model 20

process topics

- about 11
- tables in 13

R

readme content model 31

reference guide content model 31

reference topics

- about 10

release notes content model 32

T

task topics

- about 9

third-party license agreements

- content model 34

titles

- by topic type 15

topic-based writing

- about 5
- concept topics 8
- process topic tables 13
- process topics 11

- topic-based writing (*continued*)
 - reference topics 10
 - task topics 9
 - title guidelines 15
- topics
 - about 6
 - core types 6
- troubleshooting guide content model 34

U

- user guide content model 35