



SUSE Linux Enterprise Server 15 SP7

AutoYaST 指南

AutoYaST 指南

SUSE Linux Enterprise Server 15 SP7

AutoYaST 系统可用于以无人照管的方式批量部署 SUSE Linux Enterprise Server 系统。该系统使用包含安装和配置数据的 AutoYaST 配置文件。本书将引导您完成自动安装的基本步骤，包括准备、安装和配置。

出版日期：2025 年 9 月 18 日

<https://documentation.suse.com> 

版权所有 © 2006–2025 SUSE LLC 和撰稿人。保留所有权利。

根据 GNU 自由文档许可证 (GNU Free Documentation License) 版本 1.2 或（根据您的选择）版本 1.3 中的条款，在此授予您复制、分发和/或修改本文档的权限；本版权声明和许可证附带不可变部分。许可版本 1.2 的副本包含在“GNU Free Documentation License”部分。

有关 SUSE 商标，请参见 <https://www.suse.com/company/legal/> 。所有第三方商标均是其各自所有者的财产。商标符号（®、™ 等）代表 SUSE 及其关联公司的商标。星号 (*) 代表第三方商标。

本指南力求涵盖所有细节，但这不能确保本指南准确无误。SUSE LLC 及其关联公司、作者和译者对于可能出现的错误或由此造成的后果皆不承担责任。

目录

前言 xii

1 可用文档 xii

2 改进文档 xii

3 文档约定 xiii

4 支持 xv

SUSE Linux Enterprise Server 支持声明 xvi • 技术预览 xvi

1 AutoYaST 简介 1

1.1 动机 1

1.2 概述和概念 1

I 了解和创建 AUTOYAST 控制文件 3

2 AutoYaST 控制文件 4

2.1 简介 4

2.2 格式 4

2.3 结构 5

资源和属性 6 • 嵌套式资源 6 • 属性 7

3 创建 AutoYaST 控制文件 9

3.1 收集信息 9

3.2 使用配置管理系统 (CMS) 9

创建新控制文件 10

3.3 手动创建/编辑控制文件 11

3.4 使用 XSLT 通过脚本创建控制文件 12

3.5 检查控制文件 14

基本检查 14 • 运行前脚本 15 • 导入配置文件 15

II AUTOYAST 配置示例 17

4 配置和安装选项 18

4.1 一般选项 18

mode 部分 19 • 配置“安装设置”屏幕 23 • self-update 部分 23 • semi-automatic 部分 25 • 签名处理部分 25 • wait 部分 28 • 忽略 IBM Z 上未使用的设备 29 • general 部分的示例 29

4.2 报告 31

4.3 系统注册和扩展选择 33

扩展 37

4.4 GRUB 2 引导加载程序 39

加载程序类型 40 • 全局 41 • 设备映射 46

4.5 Systemd 引导加载程序 46

加载程序类型 47 • 全局 47

4.6 分区 47

自动分区 48 • 引导式分区 48 • 专家分区 50 • 高级分区功能 66 • 逻辑卷管理器 (LVM) 71 • 软件 RAID 73 • 多路径支持 79 • bcache 配置 80 • 多设备 Btrfs 配置 83 • NFS 配置 85 • tmpfs 配置 86 • 特定于 IBM Z 的配置 86

4.7 iSCSI 发起端概述 88

4.8 以太网光纤通道 (FCoE) 配置 89

4.9 国家/地区设置 91

- 4.10 软件 92
 - 产品选择 92 · 使用 patterns 和 packages 部分选择软件包 93 · 安装其他/自定义的软件包或产品 94 · 内核软件包 100 · 去除自动选择的软件包 101 · 安装推荐的软件包和软件集 101 · 在第 2 个阶段安装软件包 102 · 在第 2 个阶段安装软件集 102 · 在第 2 个阶段联机更新 103
- 4.11 升级 103
- 4.12 服务和目标 105
- 4.13 网络配置 106
 - 配置工作流程 106 · 网络资源 107 · 接口 110 · 指派多个 IP 地址 115 · 网络接口的持久名称 116 · 域名系统 117 · 路由选择 118 · s390 选项 120
- 4.14 代理 121
- 4.15 NIS 客户端和服务端 121
- 4.16 NIS 服务器 122
- 4.17 主机定义 125
- 4.18 Windows 域成员资格 125
- 4.19 Samba 服务器 126
- 4.20 身份验证客户端 128
- 4.21 NFS 客户端和服务端 128
- 4.22 NTP 客户端 130
- 4.23 邮件服务器配置 131
- 4.24 Apache HTTP 服务器配置 133
- 4.25 Squid 服务器 144
- 4.26 FTP 服务器 152

- 4.27 TFTP 服务器 156
- 4.28 首次引导工作流程 157
- 4.29 安全性设置 157
 - 口令设置选项 158 · 引导设置 158 · 登录设置 159 · 新用户设置 (**useradd** 设置) 159 · Linux 安全模块 (LSM) 设置 159 · 使用 OpenSCAP 安全策略 159
- 4.30 Linux 审计框架 (LAF) 161
- 4.31 用户和组 163
 - 用户 163 · 用户默认值 168 · 组 169 · 登录设置 170
- 4.32 自定义用户脚本 171
 - 前脚本 171 · 分区后脚本 172 · Chroot 环境脚本 173 · 后脚本 173 · Init 脚本 173 · 脚本 XML 表示 175 · 脚本示例 178
- 4.33 系统变量 (sysconfig) 180
- 4.34 添加完整配置 181
- 4.35 安装期间要求用户提供值 183
 - 默认值脚本 189 · 脚本 189
- 4.36 内核转储 194
 - 内存预留 196 · 转储保存 197 · 电子邮件通知 199 · Kdump 内核设置 200 · 专家设置 201
- 4.37 DNS 服务器 202
- 4.38 DHCP 服务器 204
- 4.39 防火墙配置 208
 - 一般防火墙配置 208 · 防火墙区域配置 209 · 应用 firewalld 配置文件的安装阶段 210 · 完整示例 211
- 4.40 其他硬件和系统组件 212
 - 打印机 212 · 声音设备 213

- 4.41 导入 SSH 密钥和配置 214
- 4.42 配置管理 215
 - 连接到配置管理服务器 215 • 以独立模式运行 217 • SUSE Multi-Linux Manager Salt 公式支持 218

III 使用动态配置文件管理批量安装 219

5 支持的动态配置文件方法 220

6 规则和类 221

- 6.1 基于规则的自动安装 221
 - 规则文件说明 222 • 自定义规则 225 • 规则的匹配类型 226 • 合并属性 226 • 规则文件结构 227 • 预定义的系统属性 227 • 使用对话框选择规则 229
- 6.2 类别 232
- 6.3 混合规则和类 234
- 6.4 合并规则和类 234

7 ERB 模板 237

- 7.1 什么是 ERB? 237
- 7.2 模板帮助程序 238
 - boot_efi? 238 • disks 238 • network_cards 239 • os_release 240 • hardware 241
- 7.3 运行 ERB 帮助程序 241
- 7.4 呈现 ERB 配置文件 242
- 7.5 调试 ERB 配置文件 243
- 7.6 ERB 与规则和类的对比 243

8 合并 ERB 模板和脚本 246

8.1 在脚本中嵌入 ERB 246

8.2 通过 Ruby 脚本访问 ERB 帮助程序 247

IV 了解自动安装过程 248

9 自动安装过程 249

9.1 简介 249

X11 界面（图形） 249 • 串行控制台 249 • 基于文本的 YaST 安装 249

9.2 选择合适的引导媒体 250

从闪存盘（例如 USB 记忆棒）引导 250 • 从 SUSE Linux Enterprise 安装媒体引导 251 • 通过 PXE 经由网络引导 251

9.3 调用自动安装过程 252

命令行选项 252 • 自动安装单个系统 258 • 将 `linuxrcinfo` 文件与 AutoYaST 控制文件结合使用 259

9.4 系统配置 259

安装后配置和系统配置 260 • 系统自定义 260

V	在安装的系统上使用 AUTOYAST	261
10	在安装的系统中运行 AutoYaST	262
VI	附录	264
A	处理规则	265
B	AutoYaST FAQ — 常见问题	266
C	高级 <code>linuxrc</code> 选项	270
C1	将参数传递给 <code>linuxrc</code>	271
C2	<code>info</code> 文件格式	271
C3	高级网络设置	273
D	SLE 12 与 15 中的 AutoYaST 配置文件之间的差异	275
D1	产品选择	275
D2	软件	275
	使用注册服务器添加模块或扩展	276
	• 使用 SLE-15-SP7-Full-ARCH-GM-media1.iso 映像添加模块或扩展	277
	• 已更名的软件集	278
D3	模块和扩展依赖项的注册	278
D4	分区	279
	GPT 变成 AMD64/Intel 64 上的默认分区类型	279
	• 设置分区号	279
	• 强制设置主分区	279
	• Btrfs: 默认子卷名称	280
	• Btrfs: 禁用子卷	280
	• 不再支持读取现有的 <code>/etc/fstab</code>	281
	• 已去除用于对齐分区的设置	281
	• 使用 <code>type</code> 定义卷组	281

D5 防火墙配置 281

将接口指派给区域 283 • 打开端口 285 • 打开 firewalld 服务 287 • 更多信息 288

D6 NTP 配置 288

D7 在第二阶段需要 AutoYaST 软件包 289

D8 已去除 CA 管理模块 289

D9 升级 289

软件 289 • 注册 289

E GNU licenses 291

前言

1 可用文档

联机文档

可在 <https://documentation.suse.com> 上查看我们的联机文档。您可浏览或下载各种格式的文档。



注意：最新更新

最新的更新通常会在本文档的英文版中提供。

SUSE 知识库

如果您遇到问题，请参考 <https://www.suse.com/support/kb/> 上提供的联机技术信息文档 (TID)。在 SUSE 知识库中搜索根据客户需求提供的已知解决方案。

发行说明

有关发行说明，请参见 <https://www.suse.com/releasesnotes/>。

在您的系统上

如需脱机使用，您也可在系统的 `/usr/share/doc/release-notes` 下找到该发行说明。各软件包的相应文档可在 `/usr/share/doc/packages` 中找到。

许多命令的**手册页**中也对相应命令进行了说明。要查看手册页，请运行 `man` 后跟特定的命令名。如果系统上未安装 `man` 命令，请使用 `sudo zypper install man` 加以安装。

2 改进文档

欢迎您提供针对本文档的反馈及改进建议。您可以通过以下渠道提供反馈：

服务请求和支持

有关产品可用的服务和支持选项，请参见 <https://www.suse.com/support/>。

要创建服务请求，需在 SUSE Customer Center 中注册订阅的 SUSE 产品。请前往 <https://scc.suse.com/support/requests> 并登录，然后单击新建。

Bug 报告

在 <https://bugzilla.suse.com/> 中报告文档问题。

要简化此过程，请点击本文档 HTML 版本中标题旁边的报告问题图标。这样会在 Bugzilla 中预先选择正确的产品和类别，并添加当前章节的链接。然后，您便可以立即开始键入 Bug 报告。

需要一个 Bugzilla 帐户。

贡献

要帮助改进本文档，请点击本文档 HTML 版本中标题旁边的 Edit Source document（编辑源文档）图标。然后您会转到 GitHub 上的源代码，可以在其中提出拉取请求。

需要一个 GitHub 帐户。



注意： Edit source document（编辑源文档）仅适用于英语版本

Edit source document（编辑源文档）图标仅适用于每个文档的英语版本。对于所有其他语言，请改用报告问题图标。

有关用于本文档的文档环境的详细信息，请参见储存库的 README。



邮件

您也可以将有关本文档的错误以及反馈发送至 doc-team@suse.com。请在其中包含文档标题、产品版本和文档发布日期。此外，请包含相关的章节号和标题（或者提供 URL），并提供问题的简要说明。

3 文档约定

本文档中使用了以下通知和排版约定：

- /etc/passwd：目录名称和文件名
- PLACEHOLDER：请将 PLACEHOLDER 替换为实际值
- PATH：环境变量
- ls、--help：命令、选项和参数

- user: 用户或组的名称
- package_name: 软件包的名称
- **Alt**、**Alt + F1**: 按键或组合键。按键以大写字母显示，与键盘上的一样。
- 文件、文件 > 另存为: 菜单项、按钮
- **AMD/Intel** 本段内容仅与 AMD64/Intel 64 体系结构相关。箭头标记文本块的开始位置和结束位置。 
- **IBM Z, POWER** 本段内容仅与 IBM Z 和 POWER 体系结构相关。箭头标记文本块的开始位置和结束位置。 
- Chapter 1, “Example chapter”: 对本指南中其他章节的交叉引用。
- 必须使用 root 特权运行的命令。您还可以在这些命令前加上 sudo 命令，以非特权用户身份来运行它们：

```
# command
> sudo command
```

- 非特权用户也可以运行的命令：

```
> command
```

- 可以通过一行末尾处的反斜线字符 (\) 拆分成两行或多行的命令。反斜线告知外壳命令调用将会在该行末尾后面继续：

```
> echo a b \
c d
```

- 显示命令（前面有一个提示符）和外壳返回的相应输出的代码块：

```
> command
output
```

- 注意事项



警告：警报通知

在继续操作之前，您必须了解的关键性信息。向您指出有关安全问题、潜在数据丢失、硬件损害或物理危害的警告。



重要：重要通知

在继续操作之前，您必须了解的重要信息。



注意：注意通知

额外信息，例如有关软件版本差异的信息。



提示：提示通知

有用信息，例如指导方针或实用性建议。

- 精简通知



额外信息，例如有关软件版本差异的信息。



有用信息，例如指导方针或实用性建议。

4 支持

下面提供了 SUSE Linux Enterprise Server 的支持声明和有关技术预览的一般信息。有关产品生命周期的细节，请参见 <https://www.suse.com/lifecycle>。

如果您有权获享支持，可在 <https://documentation.suse.com/sles-15/html/SLES-all/cha-adm-support.html> 中查找有关如何收集支持票据所需信息的细节。

4.1 SUSE Linux Enterprise Server 支持声明

要获得支持，您需要订阅适当的 SUSE 产品。要查看为您提供的具体支持服务，请前往 <https://www.suse.com/support/> 并选择您的产品。

支持级别的定义如下：

L1

问题判定，该技术支持级别旨在提供兼容性信息、使用支持、持续维护、信息收集，以及使用可用文档进行基本查错。

L2

问题隔离，该技术支持级别旨在分析数据、重现客户问题、隔离问题区域，并针对级别 1 不能解决的问题提供解决方法，或完成准备工作以提交级别 3 处理。

L3

问题解决，该技术支持级别旨在借助工程方法解决级别 2 支持所确定的产品缺陷。

对于签约客户与合作伙伴，SUSE Linux Enterprise Server 包含除以下项目外的其他所有软件包的 L3 支持：

- 技术预览。
- 声音、图形、字体和作品。
- 需要额外客户合同的软件包。
- **Workstation Extension** 模块随附的某些软件包仅享受 L2 支持。
- 名称以 `-devel` 结尾的软件包（包含头文件和类似的开发人员资源）只能与其主软件包一起获得支持。

SUSE 仅支持使用原始软件包，即，未发生更改且未重新编译的软件包。

4.2 技术预览

技术预览是 SUSE 提供的旨在让用户大致体验未来创新的各种软件包、堆栈或功能。随附这些技术预览只是为了提供方便，让您有机会在自己的环境中测试新的技术。非常希望您能提供反馈。如果您测试了技术预览，请联系 SUSE 代表，将您的体验和用例告知他们。您的反馈对于我们的未来开发非常有帮助。

技术预览存在以下限制：

- 技术预览仍处于开发阶段。因此，它们可能在功能上不完整、不稳定，或者**不适合**生产用途。
- 技术预览**不受支持**。
- 技术预览可能仅适用于特定的硬件体系结构。
- 技术预览的细节和功能可能随时会发生变化。因此，可能无法升级到技术预览的后续版本，而只能进行全新安装。
- SUSE 可能会发现某个预览不符合客户或市场需求，或者未遵循企业标准。技术预览可能会随时从产品中删除。SUSE 不承诺未来将提供此类技术的受支持版本。

如需大致了解产品随附的技术预览，请参见 <https://www.suse.com/releasesnotes>  上的发行说明。

1 AutoYaST 简介

1.1 动机

SUSE Linux Enterprise Server 的标准安装通过一个向导工作流程进行。在少量的计算机上安装时，此工作流程非常友好而且高效。但如果在大量计算机上安装，它会涉及到一些重复而且耗时的操作。

为了避免这种情况，您可以通过复制第一个成功安装系统的硬盘来执行批量部署。不过遗憾的是，这又会导致另一个问题：即便计算机之间只存在微小的配置差异，事后也需要分别进行处理。例如，在使用静态 IP 地址时，需要为每台计算机重置这些 IP 地址。

默认情况下，SUSE Linux Enterprise Server 的常规安装是半自动化的。在安装开始时，系统会提示用户选择所需的信息（通常只提示选择语言）。然后，YaST 会根据不同的因素和系统参数生成有关底层系统的建议。通常（尤其是对于新系统），可以参考此类建议来安装系统并得到一个可用的安装。遵循建议执行的步骤是全自动化的。

在无需用户干预或者需要自定义的情况下，可以使用 AutoYaST。YaST 使用 AutoYaST 配置文件来准备系统以进行自定义安装，除非控制安装的文件中有指定，否则 YaST 不会与用户交互。

AutoYaST 不是自动化的 GUI 系统。这意味着通常会跳过许多屏幕 — 例如，您永远看不到语言选择界面。AutoYaST 只会将语言参数传递给子系统，而不会显示任何语言相关的界面。

1.2 概述和概念

使用 AutoYaST 可以同时且快速地轻松安装多个系统。这些系统需要共享相同的环境，以及类似但不必完全相同的硬件。安装由称为“AutoYaST 控制文件”的 XML 配置文件（通常名为 `autoinst.xml`）定义。您可以使用现有的配置资源创建此文件，并针对任何特定环境轻松量身定制。

AutoYaST 是完全集成的，它提供多种选项用于安装和配置系统。与其他自动安装系统相比，其主要优势是能够使用现有模块来配置计算机，并可避免使用通常会在安装结束时执行的自定义脚本。

本文档将指导您完成自动安装的三个步骤：

- 准备：收集有关目标系统的所有相关信息，并将其转换成配置文件中的相应指令。配置文件将传送到目标系统，在目标系统中，其指令将被分析并馈送到 YaST。
- Installation（安装）：YaST 会使用 AutoYaST 配置文件中的数据执行目标系统的安装和基本配置（例如，分区、联网设置、防火墙）。
- 配置后：安装并配置基本系统之后，系统可以运行另一个阶段，来执行只有在目标系统已经运行时才能进行的所有其他配置，例如安装后脚本、第三方模块或某些 YaST 模块。



注意：第二个阶段

常规的 SUSE Linux Enterprise Server 15 SP7 安装在一个阶段内执行。而自动安装过程则分为两个阶段。安装基本系统并对其进行主要的配置之后，它会引导进入第二个阶段，以执行任何安装后配置步骤。

需要安装软件包 `autoyast2` 和 `autoyast2-installation` 才能在安装的系统中正常运行第二个阶段。否则，在引导进入安装的系统之前会显示错误。

第二个阶段仅在绝对必要的情况下运行，可通过 `second_stage` 参数彻底关闭：

```
<general>
  <mode>
    <confirm config:type="boolean">false</confirm>
    <second_stage config:type="boolean">false</second_stage>
  </mode>
</general>
```

I 了解和创建 AutoYaST 控制文件

2 AutoYaST 控制文件 4

3 创建 AutoYaST 控制文件 9

2 AutoYaST 控制文件

2.1 简介

控制文件也称为**配置文件**，是单个系统的配置说明。它由带有属性的资源集构成，包括对复杂结构（例如列表、记录、树和大型嵌入式对象或参照对象）的支持。

2.2 格式

XML 配置格式提供简单易学的一致文件结构，并且在尝试配置新系统时也很容易记住这种结构。

AutoYaST 控制文件使用 XML 来描述系统安装和配置。XML 是一种常用标记，许多用户都很熟悉该语言的概念以及用于处理 XML 文件的工具。如果您要编辑现有控制文件，或者要创建一个控制文件，强烈建议您对该控制文件进行验证。为此，可以使用 [xmllint](#) 或 [jing](#) 等 XML 验证解析器（请参见第 3.3 节“手动创建/编辑控制文件”）。

以下示例演示了一个 XML 格式的控制文件：

例 2.1：AUTOYAST 控制文件（配置文件）

```
<?xml version="1.0"?>
<!DOCTYPE profile>
<profile
  xmlns="http://www.suse.com/1.0/yast2ns"
  xmlns:config="http://www.suse.com/1.0/configns">
  <partitioning config:type="list">
    <drive>
      <device>/dev/sda</device>
      <partitions config:type="list">
        <partition>
          <filesystem config:type="symbol">btrfs</filesystem>
          <size>10G</size>
          <mount>/</mount>
        </partition>
        <partition>
```

```

        <filesystem config:type="symbol">xfs</filesystem>
        <size>120G</size>
        <mount>/data</mount>
    </partition>
</partitions>
</drive>
</partitioning>
<scripts>
    <pre-scripts>
        <script>
            <interpreter>shell</interpreter>
            <filename>start.sh</filename>
            <source>
                <![CDATA[
#!/bin/sh
echo "Starting installation"
exit 0

]]>

            </source>
        </script>
    </pre-scripts>
</scripts>
</profile>

```

2.3 结构

下面是基本控制文件容器的示例，本章稍后将解释其实际内容。

例 2.2：控制文件容器

```

<?xml version="1.0"?>
<!DOCTYPE profile>
<profile
  xmlns="http://www.suse.com/1.0/yast2ns"
  xmlns:config="http://www.suse.com/1.0/configs">
  <!-- RESOURCES -->

```

```
</profile>
```

<profile> 元素（根节点）包含一个或多个相异资源元素。允许的资源元素在纲要文件中指定

2.3.1 资源和属性

资源元素包含多个相异的属性和资源元素、同一资源元素的多个实例，或者为空。资源元素的允许内容在纲要文件中指定。

属性元素为空，或者包含一个文本值。每个资源元素中允许的属性元素和值在纲要文件中指定一个元素可以是其他元素的一个容器（资源），也可以包含一个文本值（属性），但永远不会同时以这两种形式存在。此项限制在纲要文件中指定。包含多个值的配置组件必须以属性值中的嵌入式列表形式或者以嵌套式资源的形式来表示。

经过分析的数据模型中将**不存在**空元素（例如 <foo></foo> 或 <bar/>）。通常会将这种元素解释为需要提供合理的默认值。如果您需要明确为空的字符串，请使用 CDATA 部分：<foo><![CDATA[]]></foo>。

2.3.2 嵌套式资源

嵌套式资源元素允许在任何级别构建配置组件的树形结构。

有两种嵌套式资源：映射和列表。映射（也称为关联性数组、哈希或字典）包含按标记名称标识的混合内容。列表或数组中的所有项具有相同的类型。

例 2.3：嵌套式资源

```
...
<drive>
  <device>/dev/sda</device>
  <partitions config:type="list">
    <partition>
      <size>10G</size>
      <mount></mount>
    </partition>
    <partition>
      <size>1G</size>
```

```
        <mount>/tmp</mount>
    </partition>
</partitions>
</drive>
....
```

在以上示例中，`drive` 资源是由 `device` 属性和 `partitions` 资源构成的映射。`partitions` 资源是包含多个 `partition` 资源实例的列表。每个 `partition` 资源是包含 `size` 和 `mount` 属性的映射。

嵌套式资源的默认类型是映射，不过您可以根据自己的需要指定其类型。必须使用 `config:type="list"` 属性将列表标记为列表类型。



提示：使用更短类型的注释

从 SUSE Linux Enterprise Server 15 SP3 开始，可使用属性 `t` 代替 `config:type` 来指定元素类型。

```
<mode t="boolean">true</mode>
```

2.3.3 属性

使用全局属性 (attribute) 来定义资源和属性 (property) 中的元数据。属性用于定义上下文切换。如前面的章节中所示，它们还可用于命名和键入属性。属性位于独立的名称空间中，因此无需将其视为默认名称空间中的保留字。

`config:type` 属性 (attribute) 用于确定所分析数据模型中的资源或属性 (property) 的类型。对于资源，列表需要 `list` 类型，而映射则是不需要属性的默认类型。有一个例外。如果映射为空，需要将其标记为映射，这样便不会将其与单纯的字符串值混淆。

例 2.4：空映射

```
<general t="map" />
```

对于属性，可以使用 `boolean`、`symbol` 和 `integer`，默认类型为 `string`。

如上所述，除了 map 和 string 值外，其他属性都不是可选属性。之所以属性有时看似可选，是因为纲要的各个部分在使用数据类型方面具有很大的不一致性。在某些位置枚举由符号表示，而在其他位置则需要以字符串表示。一个资源需要 `config:type="integer"`，而另一个资源则是分析字符串属性中的数字。有些资源使用 `config:type="boolean"`，还有些资源则需要 `yes` 甚至 `1`。如有疑问，请查阅纲要文件。

3 创建 AutoYaST 控制文件

3.1 收集信息

要创建控制文件，您需要收集有关所要安装系统的信息。这包括硬件数据和网络信息等。请确保已收集有关您要安装的计算机的以下信息：

- 硬盘类型和大小
- 图形界面和连接的监视器（如果有）
- 网络接口和 MAC 地址（如果已知，例如使用 DHCP 时）

还要验证 autoyast2-installation 和 autoyast2 是否都已安装。

3.2 使用配置管理系统 (CMS)

我们提供了基于 YaST 的配置界面，用于创建一个或多个计算机的控制文件。此系统依赖于现有模块，这些模块通常用于在常规操作模式下配置计算机，例如在安装 SUSE Linux Enterprise Server 后进行配置。

配置管理系统可让您轻松创建控制文件，以及管理可在包含多个客户端的联网环境中使用的配置储存库。

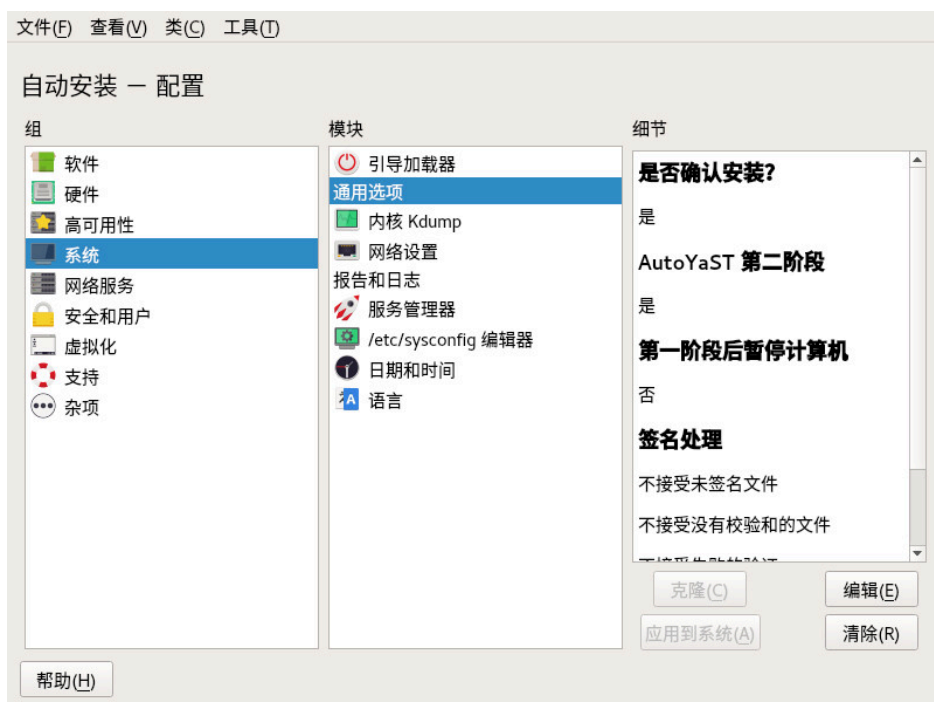


图 3.1：配置系统

3.2.1 创建新控制文件

要创建 AutoYaST 配置文件，最简单的方法是使用现有 SUSE Linux Enterprise Server 系统作为模板。在已安装的系统上，启动 YaST > 杂项 > 自动安装配置。然后从菜单中选择 工具 > 创建参考配置文件。选择您要包含在配置文件中的系统组件。或者，通过启动 YaST > 杂项 > 自动安装克隆系统，或在命令行中运行 `sudo yast clone_system`，创建包含完整系统配置的配置文件。

这两种方法都会创建文件 `/root/autoinst.xml`。克隆的配置文件可用于设置与所克隆系统完全相同的克隆。但通常需要调整该文件，以便安装多台极为相似但不完全相同的计算机。为此，可以使用您偏好的文本编辑器/XML 编辑器调整配置文件。



警告：配置文件中的敏感数据

请注意，配置文件中可能包含敏感信息，例如口令哈希和注册密钥。

请仔细查看导出的配置文件，确保使用限制性文件权限。

控制文件的几乎所有资源都可以使用配置管理系统进行配置，只存在几种例外情况。该系统具有灵活性，某些资源的配置与 YaST 控制中心提供的配置相同。除了现有模块和用户熟悉的模块以外，我们还创建了新的界面用于完成特殊和复杂的配置，例如分区、一般选项和软件的配置。

此外，使用 CMS 还能保证生成的控制文件的有效性，并可直接使用该控制文件来启动自动安装。

确保已安装配置系统（软件包 `autoyast2`）。使用 YaST 控制中心或者以 root 身份使用以下命令来调用 AutoYaST（确保正确设置 `DISPLAY` 变量，以启动图形用户界面而不是基于文本的界面）：

```
/sbin/yast2 autoyast
```

3.3 手动创建/编辑控制文件

如果您要手动编辑控制文件，请确保它采用有效语法。要检查语法，请使用发行套件自带的工具。例如，要校验文件的格式是否正确（采用有效的 XML 结构），请使用 `xmllint` 软件包中提供的实用程序 `libxml2`：

```
xmllint <control file>
```

如果控制文件的格式不正确（例如，某个标记缺少结束部分），`xmllint` 将报告错误。

要验证控制文件，请使用同名软件包中的工具 `jing`。在验证过程中，会检测放错位置或缺少的标记与属性以及错误的属性值。`jing` 软件包随 SUSE Software Development Kit 一起提供。

```
jing /usr/share/YaST2/schema/autoyast/rng/profile.rng <control file>
```

`/usr/share/YaST2/schema/autoyast/rng/profile.rng` 由软件包 `yast2-schema-default` 提供。此文件描述 AutoYaST 配置文件的语法和类。



注意：纲要扩展

AutoYaST 可以通过其他产品和模块进行扩展，但纲要不包含这些扩展的规范。正因如此，如果为 AutoYaST 指定使用某个此类扩展的配置文件，它可能会报告该配置文件无效。

因此，从 SUSE Linux Enterprise Server SP3 开始，AutoYaST 不再验证顶层的未知部分，而是将它们忽略。例如，以下示例中，不会验证 `<sap-inst>`，但会照常验证其余部分。

```
<general>
  <mode>
    <confirm config:type="boolean">true</confirm>
  </mode>
</general>

<sap-inst>
  <!-- this section is not validated -->
</sap-inst>
```

在继续自动安装之前，请修复此类检查发现的所有错误。如果控制文件无效或格式不正确，将无法启动自动安装过程。

您可以使用系统上提供的任何 XML 编辑器，或者支持 XML 的任何文本编辑器（例如 Emacs 和 Vim）。但是，为多台计算机手动创建控制文件并不是最佳做法，应仅将控制文件视为自动安装引擎与配置管理系统 (CMS) 之间的接口。



提示：使用 Emacs 作为 XML 编辑器

内置的 `nxml-mode` 将 Emacs 转变成了一个具有自动标记填写和验证功能的成熟 XML 编辑器。有关如何设置 `nxml-mode` 的说明，请参见 Emacs 帮助。

3.4 使用 XSLT 通过脚本创建控制文件

如果您有一个模板并想要通过脚本或命令行更改一些设置，请使用 `xsltproc` 之类的 XSLT 处理器。例如，如果您有一个 AutoYaST 控制文件，而出于任何原因您想要通过脚本填写主机名则可以使用 XSLT。（如果您经常需要执行此操作，应考虑编写操作脚本。）

首先创建一个 XSL 文件：

例 3.1：通过脚本替换主机名/域的示例文件

```
<?xml version="1.0" encoding="utf-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:y2="http://www.suse.com/1.0/yast2ns"
  xmlns:config="http://www.suse.com/1.0/configns"
  xmlns="http://www.suse.com/1.0/yast2ns"
  version="1.0">
  <xsl:output method="xml" encoding="UTF-8" indent="yes" omit-xml-
declaration="no" cdata-section-elements="source"/>

  <!-- the parameter names -->
  <xsl:param name="hostname"/>
  <xsl:param name="domain"/>

  <xsl:template match="/">
    <xsl:apply-templates select="@*|node()"/>
  </xsl:template>

  <xsl:template match="y2:dns">
    <xsl:copy>
      <!-- where to copy the parameters -->
      <domain><xsl:value-of select="string($domain)"/></domain>
      <hostname><xsl:value-of select="string($hostname)"/></hostname>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>


  <xsl:template match="@*|node()" >
    <xsl:copy>
      <xsl:apply-templates select="@*|node()"/>
    </xsl:copy>
  </xsl:template>

</xsl:stylesheet>
```

此文件需要用户以参数的形式提供主机名和域名。

```
<xsl:param name="hostname"/>
<xsl:param name="domain"/>
```

控制文件的 DNS 部分包含这些参数的副本。这意味着，如果 DNS 部分已包含一个域元素，您将获得另一个域元素，如此会导致冲突。

有关 XSLT 的详细信息，请访问官方网页 www.w3.org/TR/xslt (<https://www.w3.org/TR/xslt>) 

3.5 检查控制文件

根据使用场景，创建 AutoYaST 配置文件可能会非常困难，特别是当您使用规则/类、ERB 模板或前脚本构建动态配置文件时。有关详细信息，请访问 [第 III 部分 “使用动态配置文件管理批量安装”](#)。

从 SUSE Linux Enterprise Server 15 SP3 开始，AutoYaST 会在安装期间验证配置文件，并向用户报告发现的任何问题。尽管我们建议您检查配置文件是否正确，但您也可以通过将 `YAST_SKIP_XML_VALIDATION` 引导参数设为 `1` 来禁用此行为。

此外，为了简化测试和调试过程，AutoYaST 提供了 `check-profile` 命令，用于提取、构建以及根据需要导入配置文件，以检测任何潜在问题。



注意：结果可能不尽相同

尽管此命令使用的是与安装相同的方法，但结果可能会有所不同，具体取决于当前系统和安装媒体之间的差异：YaST 软件包版本、体系结构等



警告：仅使用受信任的配置文件

运行此命令时必须小心，因为安装前脚本和 ERB 代码会以 `root` 用户身份运行。请仅使用您信任的配置文件。

3.5.1 基本检查

使用此命令的最简单的方式就是读取并验证配置文件：

```
> sudo yast2 autoyast check-profile filename=autoinst.xml output=result.xml
```

`result.xml` 文件中包含配置文件的评估结果。请注意，即使您未使用任何高级功能，`autoinst.xml` 和 `result.xml` 的内容也可能不同。原因是 AutoYaST 在处理配置文件时会执行一些清理操作。

`check-profile` 也可以处理远程文件：

```
> sudo yast2 autoyast check-profile filename=http://192.168.1.100/autoinst.xml  
output=result.xml
```

3.5.2 运行前脚本

（可选）AutoYaST 可以运行配置文件中包含的脚本，并会在执行期间报告发现的任何错误。如果您要使用安装前脚本修改配置文件，此功能会非常有用。要启用此功能，需要将 `run-scripts` 选项设为 `true`。

```
> sudo yast2 autoyast check-profile filename=http://192.168.1.100/autoinst.xml  
output=result.xml run-scripts=true
```



警告：脚本以 root 身份运行

启用 `run-scripts` 选项时必须谨慎，因为此类脚本以 root 身份运行，可能会影响当前系统。

3.5.3 导入配置文件

即使配置文件正确有效，将其导入时也可能遇到一些问题。原因是 AutoYaST 在提取、构建和验证配置文件时，不执行任何逻辑检查。

要早发现此类问题，可使用 `check-profile` 命令，它会导入配置文件并报告所检测到的问题。由于此过程需要一段时间，您可以将 `import-all` 选项设为 `false` 来禁用此行为。

```
> sudo yast2 autoyast check-profile filename=http://192.168.1.100/autoinst.xml  
output=result.xml import-all=false
```


导入配置文件属于安全的操作，不会以任何方式更改底层系统。

II AutoYaST 配置示例

4 配置和安装选项 18

4 配置和安装选项

本章包含服务、注册、用户和组管理、升级、分区、配置管理、SSH 密钥管理、防火墙配置及其他安装选项的配置示例。

本章介绍了用于标准用途的控制文件的重要组成部分。要了解其他可用选项，请使用配置管理系统。

请注意，要使某些配置选项有效，需要根据您已配置的软件选项安装其他软件包。如果您选择安装精简系统，则可能会缺少某些软件包，需将这些软件包添加到各个软件包选项中。

YaST 会在 AutoYaST 安装后阶段开始之前的第二个安装阶段安装所需的软件包。但是，如果所需的 YaST 模块在系统中不可用，则会跳过重要的配置步骤。例如，如果未安装 `yast2-security`，则不会配置安全设置。

4.1 一般选项

`general` 部分包含影响安装工作流程的所有设置。此部分的整体结构如下所示：

```
<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns"
  xmlns:config="http://www.suse.com/1.0/configns">
  <general>
    <ask-list>❶
      ...
    </ask-list>
    <cio_ignore>❷
      ...
    </cio_ignore>
    <mode>❸
      ...
    </mode>
    <proposals>❹
      ...
    </proposals>
```

```

<self_update>⑤
...
</self_update>
<self_update_url>
...
</self_update_url>
<semi-automatic config:type="list">⑥
...
</semi-automatic>
<signature-handling>⑦
...
</signature-handling>
<storage>⑧
...
</storage>
<wait>⑨
...
</wait>
</general>
</profile>

```

- ① 第 4.35 节 “安装期间要求用户提供值”
- ② 第 4.1.7 节 “忽略 IBM Z 上未使用的设备”
- ③ 第 4.1.1 节 “mode 部分”
- ④ 第 4.1.2 节 “配置 “安装设置” 屏幕”
- ⑤ 第 4.1.3 节 “self-update 部分”
- ⑥ 第 4.1.4 节 “semi-automatic 部分”
- ⑦ 第 4.1.5 节 “签名处理部分”
- ⑧ 第 4.6 节 “分区”
- ⑨ 第 4.1.6 节 “wait 部分”

4.1.1 mode 部分

Mode 部分配置有关用户确认和重引导的 AutoYaST 行为。允许在 mode 部分中使用以下元素：

activate_systemd_default_target

如果您将此项设置为 false，则不会通过调用 systemd 激活默认的 systemctl isolate 目标。设置此值是可选操作。默认值为 true。

```
<general>
  <mode>
    <activate_systemd_default_target config:type="boolean">
      true
    </activate_systemd_default_target>
  </mode>
  ...
</general>
```

confirm

默认情况下，安装过程会在安装设置屏幕显示时停止。截至此时，尚未对系统进行任何更改，可在此屏幕中更改设置。要继续操作并最终开始安装，用户需要确认设置。将此值设置为 false 会自动接受设置并开始安装。只有设置为 false 才能执行完全无人照管的安装。设置此值是可选操作。默认值为 true。

```
<general>
  <mode>
    <confirm config:type="boolean">true</confirm>
  </mode>
  ...
</general>
```

confirm_base_product_license

如果将此元素设置为 true，则会显示基础产品的 EULA。用户需要接受此许可。否则会取消安装。设置此值是可选操作。默认值为 false。此设置仅适用于基础产品许可。在 add-on 部分使用标志 confirm_license 可确认其他许可（有关细节，请参见第 4.10.3 节“安装其他/自定义的软件包或产品”）。

```
<general>
  <mode>
    <confirm_base_product_license config:type="boolean">
      false
    </confirm_base_product_license>
  </mode>
  ...
</general>
```

```
</mode>
...
</general>
```

final_halt

如果设置为 true，当安装并配置所有组件后（第二个阶段结束时），计算机将会关机。

如果您启用 final_halt，则不需要将 final_reboot 选项设置为 true。

```
<general>
  <mode>
    <final_halt config:type="boolean">false</final_halt>
  </mode>
  ...
</general>
```

final_reboot

如果设置为 true，当安装并配置所有组件后（第二个阶段结束时），计算机将会重引导。

如果您启用 final_reboot，则不需要将 final_halt 选项设置为 true。

```
<general>
  <mode>
    <final_reboot config:type="boolean">true</final_reboot>
  </mode>
  ...
</general>
```

final_restart_services

如果您将此项设置为 false，当安装结束时（第二个阶段结束时，此时已安装并配置好所有组件），将**不**重新启动服务。设置此值是可选操作。默认值为 true。

```
<general>
  <mode>
    <final_restart_services config:type="boolean">
      true
    </final_restart_services>
  </mode>
  ...
</general>
```

halt

完成第一阶段后关闭计算机。已安装所有软件包和引导加载程序，并已运行所有 chroot 脚本。计算机不会重引导进入第二个阶段，而是关闭。如果您再次开机，计算机将会引导，且开始第二个自动安装阶段。设置此值是可选操作。默认值为 false。

```
<general>
  <mode>
    <halt config:type="boolean">false</halt>
  </mode>
  ...
</general>
```

max_systemd_wait

指定 AutoYaST 等待 `systemd` 设置默认目标的最长时间（以秒为单位）。设置此值是可选操作，通常不需要设置。默认值为 30（秒）。

```
<general>
  <mode>
    <max_systemd_wait config:type="integer">30</max_systemd_wait>
  </mode>
  ...
</general>
```

ntp_sync_time_before_installation

指定在开始安装之前用于同步时间的 NTP 服务器。只有设置了此选项才会进行时间同步。请记住，需要建立网络连接并有权访问时间服务器。设置此值是可选操作。默认不会进行时间同步。

```
<general>
  <mode>
    <ntp_sync_time_before_installation>
      &ntpname;
    </ntp_sync_time_before_installation>
  </mode>
  ...
</general>
```

second_stage

常规 SUSE Linux Enterprise Server 安装在一个阶段执行。而自动安装过程则分为两个阶段。安装基本系统后，系统将引导进入第二个阶段，系统配置在此阶段完成。将此选项设置为 `false` 会禁用第二个阶段。设置此值是可选操作。默认值为 `true`。

```
<general>
  <mode>
    <second_stage config:type="boolean">true</second_stage>
  </mode>
  ...
</general>
```

4.1.2 配置“安装设置”屏幕

AutoYaST 允许您配置安装设置屏幕，此屏幕显示安装设置的摘要。在此屏幕上，用户可以更改设置，然后确认设置以开始安装。使用 `proposal` 标记可以控制要在安装屏幕中显示哪些设置（“建议”）。安装媒体上的 `/control.xml` 文件中提供了适用于您产品的有效建议列表。该设置是可选的。默认会显示所有配置选项。

```
<proposals config:type="list">
  <proposal>partitions_proposal</proposal>
  <proposal>timezone_proposal</proposal>
  <proposal>software_proposal</proposal>
</proposals>
```

4.1.3 self-update 部分

在安装期间，YaST 可以自我更新，以解决发布后所发现的安装程序 Bug。有关此功能的更多信息，请参见《部署指南》。

！ 重要：每季度媒体更新：已禁用自我更新

仅当您使用统一安装程序的 GM 映像和软件包 ISO 时，才能使用安装程序自我更新。如果您从作为季度更新发布的 ISO（可以根据名称中的字符串 QU 来识别）进行安装，则安装程序无法自我更新，因为此功能在更新媒体中已禁用。

使用以下标记来配置 YaST 自我更新：

self_update

如果设置为 true 或 false，此选项将启用或禁用 YaST 自我更新功能。设置此值是可选操作。默认值为 true。

```
<general>
  <self_update config:type="boolean">true</self_update>
  ...
</general>
```

或者，您可以在内核命令行上指定引导参数 self_update=1。

self_update_url

在 YaST 自我更新期间要使用的更新储存库的位置。有关详细信息，请参见《部署指南》，第 9 章“安装步骤”，第 9.2.2 节“自定义自我更新储存库”。

！ 重要：仅指定安装程序自我更新储存库

只能为 self_update_url 参数指定安装程序自我更新储存库 URL。不要提供任何其他储存库 URL，例如软件更新储存库 URL。

```
<general>
  <self_update_url>
    http://example.com/updates/$arch
  </self_update_url>
  ...
</general>
```

该 URL 可包含变量 \$arch。该变量将由系统的体系结构（例如 x86_64、s390x 等）替换。

或者，您可以在内核命令行中连同 `self_update=URL` 一起指定引导参数 `self_update=1`。

4.1.4 semi-automatic 部分

AutoYaST 允许在安装期间启动某些 YaST 模块。这使安装计算机的管理员可以手动配置某些安装设置，同时自动完成剩余的安装步骤。在 semi-automatic 部分可以启动以下 YaST 模块：

- 网络设置模块 (`networking`)
- 分区程序 (`partitioning`)
- 注册模块 (`scc`)

以下示例会在安装期间启动所有三个支持的 YaST 模块：

```
<general>
<semi-automatic config:type="list">
  <semi-automatic_entry>networking</semi-automatic_entry>
  <semi-automatic_entry>scc</semi-automatic_entry>
  <semi-automatic_entry>partitioning</semi-automatic_entry>
</semi-automatic>
</general>
```

4.1.5 签名处理部分

默认情况下，AutoYaST 只会从具有已知 GPG 密钥的源安装已签名的软件包。使用此部分可以重写默认设置。



警告：重写签名处理默认设置

安装未签名的软件包、未通过校验和检查的软件包或者来自您不信任的源的软件包会带来重大安全风险。软件包可能已经过修改，并可能会在您的计算机上安装恶意软件。请仅在您确认储存库和软件包可信时，才重写此部分的默认设置。对于在禁用完整性检查的情况下安装的软件所造成的任何问题，SUSE 概不负责。

所有选项的默认值均为 false。如果某个选项设置为 false，而某个软件包或储存库未通过相关测试，则会以静默方式忽略而不予安装。

accept_unsigned_file

如果设置为 true，AutoYaST 将接受未签名的文件（例如内容文件）。

```
<general>
  <signature-handling>
    <accept_unsigned_file config:type="boolean">
      false
    </accept_unsigned_file>
  </signature-handling>
  ...
</general>
```

accept_file_without_checksum

如果设置为 true，AutoYaST 将接受内容文件中没有校验和的文件。

```
<general>
  <signature-handling>
    <accept_file_without_checksum config:type="boolean">
      false
    </accept_file_without_checksum>
  </signature-handling>
  ...
</general>
```

accept_verification_failed

如果设置为 true，即使签名校验失败，AutoYaST 也会接受这些已签名的文件。

```
<general>
  <signature-handling>
    <accept_verification_failed config:type="boolean">
      false
    </accept_verification_failed>
  </signature-handling>
  ...
</general>
```

accept_unknown_gpg_key

如果设置为 true，AutoYaST 将接受安装源的新 GPG 密钥，例如，用来对内容文件签名的密钥。

```
<general>
  <signature-handling>
    <accept_unknown_gpg_key config:type="boolean">
      false
    </accept_unknown_gpg_key>
  </signature-handling>
  ...
</general>
```

accept_non_trusted_gpg_key

将此选项设置为 true 会接受您尚未信任的已知密钥。

```
<general>
  <signature-handling>
    <accept_non_trusted_gpg_key config:type="boolean">
      false
    </accept_non_trusted_gpg_key>
  </signature-handling>
  ...
</general>
```

import_gpg_key

如果设置为 true，AutoYaST 将接受安装源上的新 GPG 密钥并将其导入自身的数据库中。

```
<general>
  <signature-handling>
    <import_gpg_key config:type="boolean">
      false
    </import_gpg_key>
  </signature-handling>
  ...
</general>
```

4.1.6 wait 部分

在第二个安装阶段，将通过运行模块来配置系统，例如网络配置。在 `wait` 部分内，您可以定义在特定模块运行之前和之后要执行的脚本。还可以配置在每个模块运行前后，系统处于非活动状态（“休眠”）的时长。

`pre-modules`

定义启动配置模块之前执行的脚本和休眠时间。以下代码演示了一个将休眠时间设置为 10 秒，并在运行网络配置模块之前执行 `echo` 命令的示例。

```
<general>
  <wait>
    <pre-modules config:type="list">
      <module>
        <name>networking</name>
        <sleep>
          <time config:type="integer">10</time>
          <feedback config:type="boolean">true</feedback>
        </sleep>
        <script>
          <source>echo foo</source>
          <debug config:type="boolean">false</debug>
        </script>
      </module>
    </pre-modules>
    ...
  </wait>
</general>
```

`post-modules`

定义启动配置模块之后执行的脚本和休眠时间。以下代码演示了一个将休眠时间设置为 10 秒，并在运行网络配置模块之后执行 `echo` 命令的示例。

```
<general>
  <wait>
    <post-modules config:type="list">
      <module>
        <name>networking</name>
        <sleep>
```

```

    <time config:type="integer">10</time>
    <feedback config:type="boolean">true</feedback>
  </sleep>
  <script>
    <source>echo foo</source>
    <debug config:type="boolean">false</debug>
  </script>
</module>
</post-modules>
...
</wait>
<general>

```

4.1.7 忽略 IBM Z 上未使用的设备

在 IBM Z 上，您可以通过运行 **`cio_ignore`** 并忽略未使用的硬件设备，来阻止内核查找这些设备。为此，可将同名的 AutoYaST 参数设置为 **`true`**。设置此值是可选操作，仅适用于 IBM Z 硬件上的安装。默认值为 **`true`**。

```

<general>
  <cio_ignore config:type="boolean">true</cio_ignore>
  ...
</general>

```

4.1.8 `general` 部分的示例

本节提供了多个用例。

例 4.1：一般选项

本示例演示 `general` 部分中最常用的选项。`pre-modules` 和 `post-modules` 部分中的脚本只是用于说明概念的虚设脚本。

```

<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns"
  xmlns:config="http://www.suse.com/1.0/configns">

```

```

<general>
  <mode>
    <halt config:type="boolean">false</halt>
    <forceboot config:type="boolean">false</forceboot>
    <final_reboot config:type="boolean">false</final_reboot>
    <final_halt config:type="boolean">false</final_halt>
    <confirm_base_product_license config:type="boolean">
      false
    </confirm_base_product_license>
    <confirm config:type="boolean">true</confirm>
    <second_stage config:type="boolean">true</second_stage>
  </mode>
  <proposals config:type="list">
    <proposal>partitions_proposal</proposal>
  </proposals>
  <self_update config:type="boolean">true</self_update>
  <self_update_url>http://example.com/updates/$arch</self_update_url>
  <signature-handling>
    <accept_unsigned_file config:type="boolean">
      true
    </accept_unsigned_file>
    <accept_file_without_checksum config:type="boolean">
      true
    </accept_file_without_checksum>
    <accept_verification_failed config:type="boolean">
      true
    </accept_verification_failed>
    <accept_unknown_gpg_key config:type="boolean">
      true
    </accept_unknown_gpg_key>
    <import_gpg_key config:type="boolean">true</import_gpg_key>
    <accept_non_trusted_gpg_key config:type="boolean">
      true
    </accept_non_trusted_gpg_key>
  </signature-handling>
  <wait>
    <pre-modules config:type="list">
      <module>

```

```

    <name>networking</name>
    <sleep>
      <time config:type="integer">10</time>
      <feedback config:type="boolean">true</feedback>
    </sleep>
    <script>
      <source>&gt;![CDATA[
echo "Sleeping 10 seconds"
  ]&gt;</source>
      <debug config:type="boolean">>false</debug>
    </script>
  </module>
</pre-modules>
<post-modules config:type="list">
  <module>
    <name>networking</name>
    <sleep>
      <time config:type="integer">10</time>
      <feedback config:type="boolean">true</feedback>
    </sleep>
    <script>
      <source>&gt;![CDATA[
echo "Sleeping 10 seconds"
  ]&gt;</source>
      <debug config:type="boolean">>false</debug>
    </script>
  </module>
</post-modules>
</wait>
</general>
</profile>

```

4.2 报告

report 资源管理安装期间可能显示的三种弹出窗口：

- 消息弹出窗口（通常是非关键的信息性消息）；
- 警告弹出窗口（如果可能出现问题）；
- 错误弹出窗口（如果发生了错误）。

例 4.2：报告行为

```
<report>
  <errors>
    <show config:type="boolean">true</show>
    <timeout config:type="integer">0</timeout>
    <log config:type="boolean">true</log>
  </errors>
  <warnings>
    <show config:type="boolean">true</show>
    <timeout config:type="integer">10</timeout>
    <log config:type="boolean">true</log>
  </warnings>
  <messages>
    <show config:type="boolean">true</show>
    <timeout config:type="integer">10</timeout>
    <log config:type="boolean">true</log>
  </messages>
  <yesno_messages>
    <show config:type="boolean">true</show>
    <timeout config:type="integer">10</timeout>
    <log config:type="boolean">true</log>
  </yesno_messages>
</report>
```

您可以根据自己的经验跳过、记录和显示（结合使用超时）这些消息。建议结合使用超时显示所有 messages。在某些情况下可以跳过警告，但不应将其忽略。

自动安装模式下的默认设置是显示错误且不设置超时，并以 10 秒超时显示所有警告/消息。



警告：关键系统消息

请注意，安装期间显示的消息并非全部由 `report` 资源控制。某些涉及到软件包安装和分区的关键消息将会忽略 `report` 部分中的设置直接显示出来。通常需要单击是或否来回复这些消息。

4.3 系统注册和扩展选择

可以在 `suse_register` 资源中配置于注册服务器中注册系统的操作。以下示例显示在 SUSE Customer Center 中注册系统。如果您的组织提供了自己的注册服务器，则您需要使用 `reg_server*` 属性指定所需的数据。有关细节，请参见下面的列表。

```
<suse_register>
  <do_registration config:type="boolean">true</do_registration>
  <email>tux@example.com</email>
  <reg_code>MY_SECRET_REGCODE</reg_code>
  <install_updates config:type="boolean">true</install_updates>
  <slp_discovery config:type="boolean">false</slp_discovery>
  <--! optionally register some add-ons -->
  <addons config:type="list">
    <addon>
      <name>sle-module-basesystem</name>
      <version>15.7</version>
      <arch>x86_64</arch>
    </addon>
  </addons>
</suse_register>
```

建议至少注册 Basesystem Module，以便能够访问基础系统（Linux 内核、系统库和服务）的更新。

作为全自动注册的替代方法，还可将 AutoYaST 配置为在安装期间启动 YaST 注册模块。这样就可以手动输入注册数据。需要提供以下 XML 代码：

```
<general>
  <semi-automatic config:type="list">
    <semi-automatic_entry>scc</semi-automatic_entry>
```

```
</semi-automatic>
</general>
```



提示：使用安装网络设置

如果您要使用安装时所用的相同网络设置，AutoYaST 需要在启动注册前的第 1 阶段运行网络设置：

```
<networking>
  <setup_before_proposal config:type="boolean">true</
setup_before_proposal>
</networking>
```

suse_register Values

do_registration

布尔

```
<do_registration config:type="boolean">true</do_registration>
```

指定是否应注册系统。如果设置为 false，则会忽略所有其他选项，并且不注册系统。

e-mail

电子邮件地址

```
<email>tux@example.com</email>
```

可选。与注册代码匹配的电子邮件地址。

reg_code

文本

```
<reg_code>SECRET_REGCODE</reg_code>
```

必需。注册代码。

install_updates

布尔

```
<install_updates config:type="boolean">true</install_updates>
```

可选。确定是否应安装来自“更新”频道的更新。默认为不安装这些更新 (false)。

slp_discovery

布尔

```
<slp_discovery config:type="boolean">true</slp_discovery>
```

可选。通过 SLP 搜索注册服务器。默认值为 false。

预期会查找单个服务器。如果找到多个服务器，安装将会失败。如果有多个可用的注册服务器，您需要使用 reg_server 指定一个服务器。

如果 slp_discovery 和 reg_server 都未设置，将在 SUSE Customer Center 中注册系统。

此设置也会影响自我更新功能：如果将其禁用，将不会执行 SLP 搜索。

reg_server

URL

```
<reg_server>https://smt.example.com</reg_server>
```

可选。RMT 服务器 URL。如果 slp_discovery 和 reg_server 都未设置，将在 SUSE Customer Center 中注册系统。

将从 RMT 服务器查询自我更新储存库的 URL。因此，如果未设置 self_update_url，RMT 服务器将会决定下载自我更新的位置。有关此功能的更多信息，请参见《部署指南》。

reg_server_cert_fingerprint_type

SHA1 或 SHA256

```
<reg_server_cert_fingerprint_type>SHA1</reg_server_cert_fingerprint_type>
```

可选。需要通过 reg_server_cert_fingerprint 提供校验和值。建议使用指纹，因为它可以确保 SSL 证书经过校验。当 SSL 通讯由于校验错误而失败时，将自动导入匹配的证书。

reg_server_cert_fingerprint

采用十六进制表示法（不区分大小写）的服务器证书指纹值。

```
<reg_server_cert_fingerprint>01:AB...:EF</reg_server_cert_fingerprint>
```

可选。需要通过 `reg_server_cert_fingerprint_type` 提供指纹类型值。建议使用指纹，因为它可以确保 SSL 证书经过校验。当 SSL 通讯由于校验错误而失败时，将自动导入匹配的证书。

reg_server_cert

URL

```
<reg_server_cert>http://smt.example.com/smt.crt</reg_server_cert>
```

可选。服务器上 SSL 证书的 URL。不建议使用此选项，因为不会校验下载的证书。请改用 `reg_server_cert_fingerprint`。

addons

附加产品列表

指定应添加到安装储存库的注册服务器中的扩展。有关详细信息，请参见第 4.3.1 节“扩展”。



提示：获取服务器证书指纹

要获取与 `reg_server_cert_fingerprint` 项配合使用的服务器证书指纹，请在 SMT 服务器上运行以下命令（如果需要，请编辑 `smt.crt` 文件的默认路径）：

```
openssl x509 -noout -in /srv/www/htdocs/smt.crt -fingerprint -sha256
```

要从 SMT 服务器检索指纹，请使用以下命令：

```
curl --insecure -v https://scc.suse.com/smt.crt 2> /dev/null | openssl \
x509 -noout -fingerprint -sha256
```

请将 `scc.suse.com` 替换为您的服务器。

注：只能在可信网络中使用此命令！在不可信的网络（例如互联网）中，应通过其他方式直接从服务器获取指纹。可通过 SSH、保存的服务器配置和其他源提取指纹。或者，您也可以校验下载的证书是否与服务器上的证书相同。

4.3.1 扩展

SUSE Customer Center 提供多个扩展，例如 `sle-module-development-tools` (Development Tools Module)，在安装期间，可作为附加源包含这些扩展。可通过 `addons` 块中的 `suse_register` 属性添加扩展。



注意：扩展的可用性

扩展的可用性与产品和体系结构有关，并非所有扩展都可在所有体系结构上使用。

某些扩展（例如 `sle-ha`）需要注册代码。根据您的订阅，请使用扩展的专用注册代码，或者重申基础产品的注册代码。

使用 **SUSEConnect --list-extensions**，您可以列出注册系统中所有可用的扩展，以及用于激活和禁用这些扩展的命令。

下面的示例显示了哪些扩展已激活，并标记了需要自己的注册代码的扩展：

```
> sudo SUSEConnect --list-extensions
AVAILABLE EXTENSIONS AND MODULES

Basesystem Module 15 SP 7 x86_64 (Activated)
Deactivate with: SUSEConnect -d -p sle-module-basesystem/15.7/x86_64

Containers Module 15 SP 7 x86_64
Activate with: SUSEConnect -p sle-module-containers/15.7/x86_64

Desktop Applications Module 15 SP 7 x86_64 (Activated)
Deactivate with: SUSEConnect -d -p sle-module-desktop-applications/
15.7/x86_64

SUSE Linux Enterprise Workstation Extension 15 SP 7 x86_64 (BETA)
Activate with: SUSEConnect -p sle-we/15.7/x86_64 -r ADDITIONAL
REGCODE
[...]
```

`-p` 参数（在上面的示例中）显示可在 AutoYaST 配置文件中使用的 `NAME/VERSION/ARCH` 值。

下面的示例显示如何配置扩展列表。相应指令包含在 `suse_register` 块中：

```

<suse_register>
  <do_registration config:type="boolean">true</do_registration>
  <email>tux@example.com</email>
  <reg_code>MY_SECRET_REGCODE</reg_code>
  <install_updates config:type="boolean">true</install_updates>
  <slp_discovery config:type="boolean">false</slp_discovery>

  <!--! optionally register some add-ons -->
<addons config:type="list">
  <addon>
    <!-- Development Tools Module -->
    <!-- Depends on: Desktop Applications Module -->
    <name>sle-module-development-tools</name>
    <version>15.3</version>
    <arch>x86_64</arch>
  </addon>

  <addon>
    <!-- SUSE CaaS Platform (BETA) -->
    <!-- Depends on: Containers Module -->
    <name>caasp</name>
    <version>4.0</version>
    <arch>x86_64</arch>
    <reg_code>REG_CODE_REQUIRED</reg_code>
  </addon>

  <addon>
    <!-- SUSE Enterprise Storage -->
    <!-- Depends on: Server Applications Module -->
    <name>ses</name>
    <version>6</version>
    <arch>x86_64</arch>
    <reg_code>REG_CODE_REQUIRED</reg_code>
  </addon>

  <addon>
    <!-- SUSE Linux Enterprise High Availability Extension -->
    <!-- Depends on: Server Applications Module -->

```

```
<name>sle-ha</name>
<version>15.3</version>
<arch>x86_64</arch>
<reg_code>REG_CODE_REQUIRED</reg_code>
</addon>
</addons>
</suse_register>
```

您也可以在 <https://scc.suse.com/packages> 上查看所有可用的模块和扩展。选择您的产品和体系结构，然后单击“在模块中”表单即可查看所有扩展列表。



注意：扩展的依赖项

从 SLES 15 开始，在注册期间，AutoYaST 会根据扩展的依赖项对扩展进行重新排序。这意味着，AutoYaST 配置文件中的扩展顺序并不重要。

另外，AutoYaST 会自动注册依赖扩展，即使这些扩展在配置文件中缺失也会如此。这意味着，您不需要填写整个扩展列表。

但是，如果依赖扩展需要注册密钥，则必须在配置文件中指定这一点（包括该注册密钥）。否则注册将会失败。

扩展的体系结构和版本不是必要信息。注册工作流程将会评估合适的扩展。

4.4 GRUB 2 引导加载程序

本文档用于 **yast2-bootloader**，并适用于 GRUB 2。对于传统 GRUB 随附的早期产品版本，请参见 `/usr/share/doc/packages/autoyast2/` 中发行套件随附的文档

默认情况下，AutoYaST 会建议使用引导媒体所用的相同引导机制。例如，如果您使用 EFI 引导，则会安装 GRUB 2 for EFI。因此，除非您有特别的要求，否则不能忽略此部分。由于 EFI 引导需要进行特定的分区，我们建议使用自动分区功能（如第 4.6.1 节“自动分区”中所述），这会自动创建所需的分区。

如果您需要调整默认设置，请使用 `<bootloader>` 部分。该部分的总体结构如以下代码段所示：

```
<bootloader>
```



```

<loader_type>
  <!-- boot loader type (grub2 or grub2-efi) -->
</loader_type>
<global>
  <!--
    entries defining the installation settings for GRUB 2 and
    the generic boot code
  -->
</global>
<device_map config:type="list">
  <!-- entries defining the order of devices -->
</device_map>
</bootloader>

```

您不需要提供所有设置，只需定义需要更改的设置。AutoYaST 随后会将默认值与配置文件中指定的值合并。

4.4.1 加载程序类型

此部分定义要使用哪个引导加载程序（UEFI 或 BIOS/传统）。并非所有体系结构都支持引导加载程序的传统和 EFI 这两种变体。最安全的 (default) 选项是让安装程序做出决定。

```
<loader_type>LOADER_TYPE</loader_type>
```

LOADER_TYPE 的可能值为：

- default：安装程序会选择适当的引导加载程序。未定义选项时，这是默认选项。
- grub2：使用传统 BIOS 引导加载程序。
- grub2-efi：使用 EFI 引导加载程序。
- none：引导进程不由安装程序管理和配置。

4.4.2 全局

这是一个可选但又很重要的部分。在此处定义要将 GRUB 2 安装到何处以及引导进程的工作方式。同样，如果您未定义该设置，**yast2-bootloader** 将建议一种配置。通常，AutoYaST 控制文件仅包含此部分，在安装期间 **yast2-bootloader** 会自动添加所有其他部分。除非您有特殊的要求，否则请不要在 XML 文件中指定引导加载程序配置。



提示：休眠

这是一个可选但又很重要的部分。在此处定义要将 GRUB 2 安装到何处以及引导进程的工作方式。同样，如果您未定义配置，**yast2-bootloader** 将建议一种配置。AutoYaST 控制文件通常仅包含此部分，安装期间 **yast2-bootloader** 会自动添加所有其他部分。除非您有特殊的要求，否则请不要在 XML 文件中指定引导加载程序配置。



提示：休眠

如果需要特定的休眠设置，可以使用 `append` 配置中的 `resume` 或 `noresume`。

要禁用休眠（无论安装程序提出什么建议），请在 `append` 部分指定 `noresume` 作为内核参数。

要指定休眠设备，请在设备路径中使用 `resume` 键。为了获得稳定的结果，建议配置您自己的分区并使用带有标签的交换设备：

```
<append>quiet resume=/dev/disk/by-label/my_swap</append>
```

如果您不使用 `resume` 或 `noresume`，或者如果 `resume` 指定的设备在所安装的系统上不存在，则安装程序可能会建议正确的 `resume` 值，或者可能会完全去除休眠参数，具体视安装程序的逻辑而定。

```
<global>
  <activate>true</activate>
  <timeout config:type="integer">10</timeout>
  <terminal>gfxterm</terminal>
  <gfxmode>1280x1024x24</gfxmode>
</global>
```

activate

在引导分区上设置引导标志。如果没有单独的 /boot 分区，引导分区可以是 /。如果引导分区位于逻辑分区上，引导标志将设置为扩展分区。

```
<activate>true</activate>
```

append

在常规和恢复模式的引导项末尾添加的内核参数。

```
<append>nomodeset vga=0x317</append>
```

boot_boot

将 GRUB 2 写入单独的 /boot 分区。如果不存在单独的 /boot 分区，则会将 GRUB 2 写入 /。

```
<boot_boot>false</boot_boot>
```

boot_custom

将 GRUB 2 写入自定义设备。

```
<boot_custom>/dev/sda3</boot_custom>
```

boot_extended

将 GRUB 2 写入扩展分区（如果您想要使用通用引导代码，而 /boot 分区是逻辑分区，则此属性非常重要）。注意：如果引导分区是逻辑分区，则您应该使用 boot_mbr（将 GRUB 2 写入 MBR）而不是 generic_mbr。

```
<boot_extended>false</boot_extended>
```

boot_mbr

按顺序将 GRUB 2 写入第一块磁盘的 MBR（device.map 包含磁盘的顺序）。

```
<boot_mbr>false</boot_mbr>
```

boot_root

将 GRUB 2 写入 / 分区。

```
<boot_root>false</boot_root>
```

cpu_mitigations

用于选择 CPU 缓解措施内核引导命令行参数的默认设置（同时在安全性与性能之间取得平衡）。

可能的值为：

auto

启用您的 CPU 型号所需的全部缓解策略，但不防范跨 CPU 线程的攻击。此设置可能会在一定程度上影响性能，具体视工作负载而定。

nosmt

提供整套可用的安全缓解策略。启用您的 CPU 型号所需的全部缓解策略。此外，它还会禁用同步多线程 (SMT)，以免受到跨多个 CPU 线程发起的边信道攻击。此设置可能会进一步影响性能，具体取决于工作负载。

off

禁用所有缓解策略。CPU 可能会遭到边信道攻击，具体取决于 CPU 型号。此设置不会对性能造成任何影响。

manual

不设置任何缓解措施级别。请使用内核命令行选项手动指定 CPU 缓解措施。

```
<cpu_mitigations>auto</cpu_mitigations>
```

如果未在 AutoYaST 中设置，可通过内核命令行更改相应设置。默认将使用安装媒体上 / control.xml 文件中的（特定于产品的）设置（如果未指定其他设置）。

generic_mbr

将通用引导代码写入 MBR（如果 boot_mbr 设置为 true，将忽略此属性）。

```
<generic_mbr config:type="boolean">false</generic_mbr>
```

gfxmode

GRUB 2 屏幕的图形分辨率（需要将 <terminal> 设置为 gfxterm）。

有效项为 auto、HORIZONTALxVERTICAL 或 HORIZONTALxVERTICAL xCOLOR DEPTH。您可以在运行中系统的 GRUB 2 命令行上，使用 vbeinfo 命令来查看特定系统上的 GRUB 2 支持的屏幕分辨率。

```
<gfxmode>1280x1024x24</gfxmode>
```

os_prober

如果设置为 `true`，则在安装期间会自动搜索已安装的操作系统并为其生成引导项。

```
<os_prober>>false</os_prober>
```

password

如果定义了此元素，将使用口令来保护引导加载程序。除非输入口令，否则系统将不会引导。

它包含三个子元素：`value`、`encrypted` 和 `unrestricted`。

`value` 用于存放口令。口令可以是纯文本，YaST 会对其进行加密，也可以是已使用 **grub-mkpasswd-pbkdf2** 加密的口令。如果您使用已加密口令，请将 `encrypted` 设为 `true`。

如果 `unrestricted` 设为 `false`，用户需要输入由 `value` 子元素定义的口令才能引导或编辑 GRUB 2 菜单项（通过对所选引导菜单项按 **E** 来编辑）。如果其设为 `true`，用户无需输入口令即可引导系统，但需输入口令才能编辑 GRUB 2 菜单项。如果省略该选项，该选项默认设为 `true`。

有关管理引导口令的详细信息，请参见《管理指南》，第 18 章 “引导加载程序 GRUB 2” 使用口令保护引导加载程序。

```
<password><value>my_strong_password</value><encrypted>>false</encrypted><unrestricted>>false</unrestricted></password>
```

suse_btrfs

已过时，不再使用。将自动启用从 Btrfs 快照引导。

serial

当 GRUB 2 终端模式设置为 `serial` 时要执行的命令。

```
<serial>serial --speed=115200 --unit=0 --word=8 --parity=no --stop=1</serials>
```

secure_boot

如果设置为 `false`，则禁用 UEFI 安全引导。仅适用于 `grub2-efi` 引导加载程序。

```
<secure_boot>>false</secure_boot>
```

terminal

指定要使用的 GRUB 2 终端模式。有效项为 `console`、`gfxterm` 和 `serial`。如果设置为 `serial`，则还需要使用 `<serial>` 指定串行命令。

```
<terminal>serial</terminal>
```

timeout

在自动引导默认引导项之前经过的超时（以秒为单位）。

```
<timeout config:type="integer">10</timeout>
```

trusted_boot

如果设置为 `true`，则使用可信 GRUB。可信 GRUB 支持可信平台模块 (TPM)。仅适用于 `grub2` 引导加载程序。

```
<trusted_boot">true</trusted_boot>
```

update_nvram

如果设置为 `true`，则 AutoYaST 会在固件中添加引导加载程序对应的 NVRAM 项。除非您希望保留特定设置或需要解决固件问题，否则此为所需行为。

```
<update_nvram>true</update_nvram>
```

vgamode

将内核参数 `vga=VALUE` 添加到引导项。

```
<vgamode>0x317</vgamode>
```

xen_append

在 Xen Guest 的引导项末尾添加的内核参数。

```
<xen_append>nomodeset vga=0x317</xen_append>
```

xen_kernel_append

在 VM 主机服务器上的 Xen 内核的引导项末尾添加的内核参数。

```
<xen_kernel_append>dom0_mem=768M</xen_kernel_append>
```

4.4.3 设备映射

GRUB 2 在生成其配置文件时，会使用设备 ID 字符串 (UUID) 或文件系统标签来避免 BIOS 驱动器与 Linux 设备之间出现映射问题。GRUB 2 实用程序会即时创建一个临时设备映射，这通常足以满足需要，在单磁盘系统上尤其如此。但是，如果您需要覆盖自动设备映射机制，请在此部分创建自定义映射。

```
<device_map config:type="list">
  <device_map_entry>
    <firmware>hd0</firmware> <!-- order of devices in target map -->
    <linux>/dev/disk/by-id/ata-ST3500418AS_6VM23FX0</linux> <!-- name of device
(disk) -->
  </device_map_entry>
</device_map>
```

4.5 Systemd 引导加载程序

本文档用于 **yast2-bootloader**，并适用于 systemd-boot。

AutoYaST 引导加载程序部分的一般结构如下所示：

```
<bootloader>
  <loader_type>
    systemd-boot
  </loader_type>
  <global>
    <!--
      entries defining the installation settings for systemd-boot and
      the generic boot code
    -->
  </global>
</bootloader>
```

4.5.1 加载程序类型

此部分定义要使用哪个引导加载程序 (systemd-boot)。并非所有体系结构都支持引导加载程序的传统和 EFI 这两种变体。

```
<loader_type>systemd-boot</loader_type>
```

4.5.2 全局

这是一个可选但又很重要的部分。您可在此处定义要将 systemd-boot 安装到何处以及引导进程的工作方式。如果您未定义全局设置，**yast2-bootloader** 将建议一种配置。除非您有特殊的要求，否则请不要在 XML 文件中指定引导加载程序配置。

```
<global>
  <timeout config:type="integer">10</timeout>
  <secure_boot>false</secure_boot>
</global>
```

引导加载程序全局选项

secure_boot

如果设置为 false，则禁用 UEFI 安全引导。

```
<secure_boot>false</secure_boot>
```

timeout

在自动引导默认引导项之前经过的超时（以秒为单位）。

```
<timeout config:type="integer">10</timeout>
```

4.6 分区

在涉及到分区时，我们可以将 AutoYaST 用例分为三个不同的级别：

- 自动分区。用户不关心分区，并信任 AutoYaST 会采取适当的做法。
- 引导式分区。用户想要设置一些基本设置。例如，某个用户想要使用 LVM，但不知道如何配置分区、卷组等。
- 专家分区。用户指定如何布局。但是，他们不需要指定完整的定义，AutoYaST 应为缺少的部分建议合理的默认值。

在一定程度上，这类似于使用常规安装程序。您可以跳过分区屏幕并采用 YaST 的方案、使用引导式建议，或通过专家分区程序定义分区布局。

4.6.1 自动分区

无需任何用户指示，AutoYaST 可提供合理的分区布局。尽管分区布局取决于选择安装的产品，但 AutoYaST 通常建议使用 Btrfs 根文件系统、一个使用 XFS 的独立 `/home`，以及一个交换分区。此外，根据体系结构，它还会添加可能需要引导的任何分区（例如 BIOS GRUB 分区）。但是，这些默认值可能会由于各种因素（例如可用磁盘空间）而发生变化。例如，是否使用独立的 `/home` 取决于可用磁盘空间。

如果您想要影响这些默认值，可以使用第 4.6.2 节“引导式分区”中所述的方法。

4.6.2 引导式分区

虽然 AutoYaST 可以在没有任何用户指示的情况下提供分区布局，但有时设置一些通用参数并让 AutoYaST 来完成其余设置是很有用的。例如，您可能想要使用 LVM 或加密文件系统，但同时又不想要处理细节。这种情况类似于在常规安装中使用引导式建议时的做法。

`storage` 中的 例 4.3 “基于 LVM 的引导式分区”部分指示 AutoYaST 使用 LVM 设置一个分区布局，并删除所有 Windows 分区，不管是否需要这些分区。

例 4.3：基于 LVM 的引导式分区

```
<general>
  <storage>
    <proposal>
      <lvm config:type="boolean">true</lvm>
```

```
<windows_delete_mode config:type="symbol">all</windows_delete_mode>
</proposal>
</storage>
</general>
```

lvm

创建基于 LVM 的建议。默认值为 false。

```
<lvm config:type="boolean">true</lvm>
```

lvm_vg_reuse

在建议中告知安装程序是否应重复使用现有的 LVM。默认值为 true。

```
<lvm_vg_reuse config:type="boolean">>false</lvm_vg_reuse>
```

resize_windows

设置为 true 时，如果需要为安装腾出空间，AutoYaST 将调整 Windows 分区的大小。

```
<resize_windows config:type="boolean">>false</resize_windows>
```

windows_delete_mode

- 如果设置为 none，将不会去除 Windows 分区。
- 如果设置为 ondemand，将在需要时去除 Windows 分区。
- 如果设置为 all，将去除所有 Windows 分区。

```
<windows_delete_mode config:type="symbol">ondemand</windows_delete_mode>
```

linux_delete_mode

- 如果设置为 none，将不会去除 Linux 分区。
- 如果设置为 ondemand，将在需要时去除 Linux 分区。
- 如果设置为 all，将去除所有 Linux 分区。

```
<linux_delete_mode config:type="symbol">ondemand</linux_delete_mode>
```

other_delete_mode

- 如果设置为 `none`，将不会去除其他分区。
- 如果设置为 `ondemand`，将在需要时去除其他分区。
- 如果设置为 `all`，将去除所有其他分区。

```
<other_delete_mode config:type="symbol">ondemand</other_delete_mode>
```

encryption_password

启用使用指定的口令进行加密。默认为禁用加密。

```
<encryption_password>some-secret</encryption_password>
```

4.6.3 专家分区

作为引导式分区的替代方案，AutoYaST 允许通过 `partitioning` 部分描述分区布局。但是，AutoYaST 无需知道每一项细节就能基于一个不太完整的规范构建合理的布局。

`partitioning` 部分是 `drive` 元素的列表。其中的每个部分描述分区布局的一个元素，例如磁盘、LVM 卷组、RAID、多设备 Btrfs 文件系统等。

例 4.4 “创建 `/`、`/home` 和 `swap` 分区” 要求 AutoYaST 使用整个磁盘创建 `/`、`/home` 和 `swap` 分区。请注意，其中缺少了一些信息，例如每个分区应使用的文件系统。但这不是一个问题，因为 AutoYaST 将为缺少的项建议合理的值。

例 4.4：创建 `/`、`/home` 和 `swap` 分区

```
<partitioning config:type="list">
  <drive>
    <use>all</use>
    <partitions config:type="list">
      <partition>
        <mount>/</mount>
        <size>20GiB</size>
      </partition>
      <partition>
        <mount>/home</mount>
```

```

        <size>max</size>
    </partition>
    <partition>
        <mount>swap</mount>
        <size>1GiB</size>
    </partition>
</partitions>
</drive>
<partitioning>

```



提示：建议引导分区

AutoYaST 会检查配置文件中描述的布局是否可引导。如果不可引导，它会添加缺少的分区。因此，如果您不确定需要引导哪些分区，可以依赖 AutoYaST 做出适当的决策。

4.6.3.1 驱动器配置

下面列出的元素必须包含在以下 XML 结构中：

```

<profile>
  <partitioning config:type="list">
    <drive>
      ...
    </drive>
  </partitioning>
</profile>

```

属性、值、描述

device

可选，要配置的设备。如果留空，AutoYaST 将尝试猜测设备。有关如何影响猜测，请参见[提示：跳过设备](#)。

如果设置为 `ask`，AutoYaST 将在安装期间询问用户要使用哪个设备。

可以通过 ID（例如 `/dev/disk/by-id/ata-WDC_WD3200AAKS-75L9`）或 **by-path**（例如 `/dev/disk/by-path/pci-0001:00:03.0-scsi-0:0:0:0`）使用持久设备名称。

```
<device>/dev/sda</device>
```

对于卷组、软件 RAID 或 bcache 设备，已安装系统中的名称可以不同（目的是避免与现有设备相冲突）。

有关处理多路径设备的更多信息，请参见第 4.6.7 节 “多路径支持”。

initialize

可选，默认值为 false。如果设为 true，在 AutoYaST 启动分区计算之前将擦除分区表。

```
<initialize config:type="boolean">true</initialize>
```

partitions

可选，<partition> 项的列表（请参见第 4.6.3.2 节 “分区配置”）。

```
<partitions config:type="list">
  <partition>...</partition>
  ...
</partitions>
```

如果未指定任何分区，AutoYaST 将创建合理的分区布局（请参见第 4.6.3.5 节 “填补缺失”）。

pesize

可选，仅适用于 LVM。对于 LVM 卷组，默认值为 4M。

```
<pesize>8M</pesize>
```

use

建议使用，指定 AutoYaST 将用于对硬盘分区的策略。可选的值：

all，计算新分区时使用整个设备。

linux，仅使用现有的 Linux 分区。

free，仅使用设备上未用的空间，现有分区保留原样。

1,2,3，要使用的分区编号的逗号分隔列表。

type

可选，指定 drive 的类型。默认值为 CT_DISK，表示普通的物理硬盘。所有选项如下：

CT_DISK，表示物理硬盘（默认值）。

CT_LVM，表示 LVM 卷组。

CT_MD，表示软件 RAID 设备。

CT_DMMULTIPATH，表示多路径设备（已弃用，隐式使用 CT_DISK）。

CT_BCACHE，表示软件 bcache 设备。

CT_BTRFS，表示多设备 Btrfs 文件系统。

CT_NFS，表示 NFS。

CT_TMPFS，表示 tmpfs 文件系统。

```
<type config:type="symbol">CT_LVM</type>
```

disklabel

可选。默认情况下，YaST 会确定有意义的值。如果已存在其他类型的分区表，仅当该分区表不包含任何应该保留或重复使用的分区时，才会重新创建给定类型的分区表。要使用磁盘而不创建任何分区，请将此元素设置为 none。所有选项如下：

msdos

gpt

none

```
<disklabel>gpt</disklabel>
```

keep_unknown_lv

可选，默认值为 false。

此值仅对 type=CT_LVM 驱动器有意义。如果您正在重复使用某个逻辑卷组并将此项设置为 true，该组中的所有现有逻辑卷将保留原样，除非在 <partitioning> 部分指定了这些卷。因此，您可保留现有逻辑卷而无需进行指定。

```
<keep_unknown_lv config:type="boolean">false</keep_unknown_lv>
```

enable_snapshots

可选，默认值为 true。

在挂载到 / 的 Btrfs 文件系统上启用快照（不适用于其他文件系统，或未挂载到 / 的 Btrfs 文件系统）。

```
<enable_snapshots config:type="boolean">false</enable_snapshots>
```

quotas

可选，默认值为 `false`。

启用 Btrfs 子卷配额支持。将此元素设为 `true` 会对文件系统启用配额支持。不过，您需要为每个子卷设置限额。请查看第 4.6.3.3 节“Btrfs 子卷”了解更多信息。

```
<quotas config:type="boolean">true</quotas>
```



重要：当心数据丢失

`use` 属性中提供的值确定如何处理现有数据和分区。值 `all` 表示将擦除整个磁盘。如果您需要保留某些包含重要数据的分区，请创建备份并使用 `confirm` 属性。否则，不会出现任何弹出窗口来通知您将要删除分区。



提示：跳过设备

在您未自行指定 `<device>` 项的情况下，可以影响 AutoYaST 的设备猜测。AutoYaST 通常会使用它能够找到的且看上去合理的第一个设备，但您可以将它配置为跳过某些设备，如下所示：

```
<partitioning config:type="list">
  <drive>
    <initialize config:type="boolean">true</initialize>
    <skip_list config:type="list">
      <listentry>
        <!-- skip devices that use the usb-storage driver -->
        <skip_key>driver</skip_key>
        <skip_value>usb-storage</skip_value>
      </listentry>
      <listentry>
        <!-- skip devices that are smaller than 1GB -->
        <skip_key>size_k</skip_key>
        <skip_value>1048576</skip_value>
        <skip_if_less_than config:type="boolean">true</skip_if_less_than>
      </listentry>
      <listentry>
        <!-- skip devices that are larger than 100GB -->
        <skip_key>size_k</skip_key>
```

```

        <skip_value>104857600</skip_value>
        <skip_if_more_than config:type="boolean">true</skip_if_more_than>
    </listentry>
</skip_list>
</drive>
</partitioning>

```

如需所有 `<skip_key>` 可能值列表，请在已安装的系统上运行 **yast2 ayast_probe**。

4.6.3.2 分区配置

下面列出的元素必须包含在以下 XML 结构中：

```

<drive>
  <partitions config:type="list">
    <partition>
      ...
    </partition>
  </partitions>
</drive>

```

create

指定是否必须创建此分区或逻辑卷，或者此分区或逻辑卷是否已存在。如果设置为 `false`，则您还需要设置 `partition_nr`、`lv_name`、`label` 和 `uuid` 中的一个，以告知 AutoYaST 要使用哪个设备。

```
<create config:type="boolean">false</create>
```

crypt_method

可选，将使用以下其中一种方法加密分区：

- `luks1`：常规 LUKS1 加密。
- `luks2`：常规 LUKS2 加密。
- `pervasive_luks2`：普遍卷加密。
- `protected_swap`：使用非永久保护的密钥进行加密。

- secure_swap: 使用非永久的安全密钥进行加密。
- random_swap: 使用非永久的随机密钥进行加密。

```
<crypt_method config:type="symbol">luks1</crypt_method>
```

加密方法选择功能是在 SUSE Linux Enterprise Server 15 SP2 中引入的。要模仿先前版本的行为，请使用 luks1。

如果需要，请查看 crypt_key 元素了解如何指定加密口令。

使用常规 LUKS 加密方法时，可以通过 crypt_pbkdf、crypt_cipher 或 crypt_key_size 自定义加密的多个设置，具体取决于所用的 LUKS 变体。请注意，加密方法及对应的设置可能会显著影响完成安装过程所需的 RAM 容量。以默认参数使用常规 LUKS2 通常意味着系统中需要有数 GB 的 RAM 才能成功加密设备。

crypt_fs

将会加密分区，默认值为 false。此元素已弃用。请改用 crypt_method。

```
<crypt_fs config:type="boolean">true</crypt_fs>
```

crypt_key

如果已将 crypt_method 设置为需要口令的方法（即 luks1、luks2 或 pervasive_luks2），则必须指定此元素。

```
<crypt_key>xxxxxxxx</crypt_key>
```

crypt_cipher

用于 LUKS 加密的加密法。仅当 crypt_method 的值为 luks1 或 luks2 时，才会应用此值。字符串的格式和值均必须与 `--cipher` 命令的 **cryptsetup** 参数一致。可以在 `/proc/crypto` 中找到兼容加密法。

```
<crypt_cipher>aes-xts-plain64</crypt_cipher>
```

crypt_key_size

用于 LUKS 加密的密钥大小（以位为单位）。仅当 crypt_method 的值为 luks1 或 luks2 时，才会应用此值。值必须为 8 的倍数。可能的密钥大小会受限于所用的加密法。

```
<crypt_key_size config:type="integer">256</crypt_key_size>
```

crypt_pbkdf

用于 LUKS2 加密的基于口令的密钥派生函数。仅当 `crypt_method` 为 `luks2` 时，该参数才会起作用。可能的值为 `pbkdf2`、`argon2i` 和 `argon2id`。如果忽略，将使用命令 `cryptsetup` 的默认函数来加密设备。请注意，Argon2 的两种变体都设计为在加密过程中有意占用大量内存。如果使用其中的任何一个函数或忽略此设置（可能会导致使用 Argon2），则意味着需要使用比不加密设置更多的 RAM 才能完成安装过程。

```
<crypt_pbkdf config:type="symbol">argon2id</crypt_pbkdf>
```

crypt_label

所加密设备的 LUKS 标签。仅当 `crypt_method` 为 `luks2` 时，该参数才会起作用。

```
<crypt_label>crypt_home</crypt_label>
```

mount

至少应有一个根分区 (/) 和一个交换分区。

```
<mount>/</mount><mount>swap</mount>
```

fstop

此分区的挂载选项，请参见 `man mount` 了解可用的挂载选项。

```
<fstopt>ro,noatime,user,data=ordered,acl,user_xattr</fstopt>
```

label

分区的标签。格式化设备时很有用（尤其是 `mountby` 参数设置为 `label` 时），并可用于标识已存在的设备（请参见前面所述的 `create`）。请参见 `man e2label` 中的示例。

```
<label>mydata</label>
```

uuid

分区的 UUID。仅可用于标识现有设备（请参见前面所述的 `create`）。不能对新设备强制使用 UUID（请参见 `man uuidgen`。）

```
<uuid>1b4e28ba-2fa1-11d2-883f-b9a761bde3fb</uuid>
```

size

分区的大小，例如 4G、4500M 等。/boot 分区和交换分区可以使用 `auto` 作为大小。然后，AutoYaST 将计算合理的大小。一个分区可以通过设置值 `max` 来使用所有剩余空间。

您也可以按百分比指定大小。即指定 10% 将使用 10% 的硬盘或卷组大小。如果需要，您可以混用 auto、max、size 和百分比。

```
<size>10G</size>
```

从 SUSE Linux Enterprise Server 开始，所有值（包括 auto 和 max）也都可用于调整分区大小。

format

指定 AutoYaST 是否应格式化分区。如果您将 create 设置为 true，则也可能需要将此项设置为 true。

```
<format config:type="boolean">false</format>
```

file system

可选。对于根分区 (/)，默认值为 btrfs；对于数据分区，默认值为 xfs。指定要在此分区上使用的文件系统：

- btrfs
- ext2
- ext3
- ext4
- fat
- xfs
- swap

```
<filesystem config:type="symbol">ext3</filesystem>
```

mkfs_options

可选，指定 mkfs 的选项字符串。请仅在您完全清楚自己所执行的操作时，才使用此元素。（请参见相关 mkfs 手册页以了解您要使用的文件系统。）

```
<mkfs_options>-I 128</mkfs_options>
```

partition_nr

此分区的编号。如果您已设置 `create=false` 或者要使用 LVM，可以通过 `partition_nr` 指定分区。

```
<partition_nr config:type="integer">2</partition_nr>
```

partition_id

`partition_id` 设置分区的 ID。如果您想要为 Linux 分区指定除 131 以外的标识符，或者要为交换分区指定除 130 以外的标识符，请使用 `partition_id` 配置标识符。Linux 分区的默认 ID 为 131，交换分区的默认 ID 为 130。

```
<partition_id config:type="integer">131</partition_id>
```

FAT16 (MS-DOS): 6

NTFS (MS-DOS): 7

FAT32 (MS-DOS): 12

扩展 FAT16 (MS-DOS): 15

DIAG、诊断和固件 (MS-DOS、GPT) : 18

PPC PReP 引导分区 (MS-DOS、GPT) : 65

交换 (MS-DOS、GPT、DASD、隐式) : 130

Linux (MS-DOS、GPT、DASD) : 131

Intel Rapid Start 技术 (MS-DOS、GPT) : 132

LVM (MS-DOS、GPT、DASD) : 142

EFI 系统分区 (MS-DOS、GPT) : 239

MD RAID (MS-DOS、GPT、DASD) : 253

BIOS 引导 (GPT): 257

Windows 基本数据 (GPT): 258

EFI (GPT): 259

Microsoft 预留 (GPT): 261

partition_type

可选。允许的值为 `primary`。使用 `msdos` 分区表时，此元素会将分区类型设置为 `primary`。使用 `gpt` 分区表时会忽略此值，因为在此情况下不存在这种差别。

```
<partition_type>primary</partition_type>
```

mountby

如果不使用分区编号，您可以告知 AutoYaST 按 device、label、uuid、path 或 id (udev 路径和 udev ID) 挂载分区（请参见 /dev/disk/...）。
请参见上述 label 和 uuid 文档。默认值取决于 YaST，通常是 id。

```
<mountby config:type="symbol">label</mountby>
```

subvolumes

要为 Btrfs 类型的文件系统创建的子卷列表。此键仅对 Btrfs 类型的文件系统有意义。（有关详细信息，请参见第 4.6.3.3 节“Btrfs 子卷”。）

如果未为分区描述定义 subvolumes 部分，AutoYaST 将为给定的挂载点创建一组预定义
的子卷。

```
<subvolumes config:type="list">
  <path>tmp</path>
  <path>opt</path>
  <path>srv</path>
  <path>var</path>
  ...
</subvolumes>
```

create_subvolumes

确定是否应创建 Btrfs 子卷。默认设置为 true。如果设置为 false，则不创建子卷。

subvolumes_prefix

设置 Btrfs 子卷前缀名称。如果不需要前缀，则必须将此元素设置为空值：

```
<subvolumes_prefix><![CDATA[]]></subvolumes_prefix>
```

默认设置为 @。

lv_name

如果此分区位于卷组中的某个逻辑卷上，请在此处指定该逻辑卷的名称（请查看驱动器配置中的 type 参数）。

```
<lv_name>opt_lv</lv_name>
```

stripes

用于配置 LVM 分段的整数。指定您要分段到多少个设备（分散数据）。

```
<stripes config:type="integer">2</stripes>
```

stripesize

指定每个块的大小，以 KB 为单位。

```
<stripesize config:type="integer">4</stripesize>
```

lvm_group

如果这是（部分）卷组 (LVM) 使用的物理分区，则您需要在此处指定该卷组的名称。

```
<lvm_group>system</lvm_group>
```

pool

如果 LVM 逻辑卷应是 LVM 精简池，则 pool 必须设置为 true。

```
<pool config:type="boolean">true</pool>
```

used_pool

用作此精简逻辑卷的数据储存的 LVM 精简池名称。如果设置为非空值，则表示该卷是所谓的精简逻辑卷。

```
<used_pool>my_thin_pool</used_pool>
```

raid_name

如果此物理卷是 RAID 阵列的一部分，请指定该 RAID 阵列的名称。

```
<raid_name>/dev/md/0</raid_name>
```

raid_options

指定 RAID 选项。在 partition 级别设置 RAID 选项的功能已弃用。请参见[第 4.6.6 节“软件 RAID”](#)。

bcache_backing_for

如果此设备用作 bcache **后备设备**，请指定 bcache 设备的名称。有关更多详细信息，请参见[第 4.6.8 节“bcache 配置”](#)。

```
<bcache_backing_for>/dev/bcache0</bcache_backing_for>
```

bcache_caching_for

如果此设备用作 **bcache 缓存设备**，请指定 **bcache** 设备的名称。有关更多详细信息，请参见第 4.6.8 节 “**bcache 配置**”。

```
<bcache_caching_for config:type="list"><listentry>/dev/bcache0</listentry></bcache_caching_for>
```

resize

从 SUSE Linux Enterprise Server 15 开始，调整大小功能适用于物理磁盘分区和 LVM 卷

```
<resize config:type="boolean">>false</resize>
```

4.6.3.3 Btrfs 子卷

如第 4.6.3.2 节 “**分区配置**” 中所述，可为每个 Btrfs 文件系统定义一组子卷。最简单的指定方式是使用路径列表：

```
<subvolumes config:type="list">
  <path>usr/local</path>
  <path>tmp</path>
  <path>opt</path>
  <path>srv</path>
  <path>var</path>
</subvolumes>
```

不过，您可以为每个子卷指定更多设置。例如，我们可能需要设置配额，或禁用写入时复制机制。为此，可以扩展下例所示列表中的任何元素：

```
<subvolumes config:type="list">
  <listentry>usr/local</listentry>
  <listentry>
    <path>tmp</path>
    <referenced_limit>1 GiB</referenced_limit>
  </listentry>
  <listentry>opt</listentry>
  <listentry>srv</listentry>
  <listentry>
```

```
<path>var/lib/pgsql</path>
<copy_on_write config:type="boolean">false</copy_on_write>
</listentry>
</subvolumes>
```

path

子卷的挂载点。

```
<path>tmp</tmp>
```

必需。如果未指定 path，AutoYaST 将忽略该子卷。

copy-on-write

是否应对该子卷启用写入时复制选项。

```
<copy-on-write config:type="boolean">false</copy-on-write>
```

可选。默认值为 false。

referenced_limit

为子卷设置配额。

```
<referenced_limit>1 GiB</referenced_limit>
```

可选。默认值为 unlimited。Btrfs 支持两种限制：referenced 和 exclusive。目前仅支持前一种限制。

如果对发行套件使用默认子卷（例如 SUSE Linux Enterprise Server 中的 @），则会自动在所定义子卷的名称前面加上此默认子卷的名称。可以通过在 subvolumes_prefix 中设置第 4.6.3.1 节“驱动器配置”来禁用此行为。

```
<subvolumes_prefix><![CDATA[]]></subvolumes_prefix>
```

4.6.3.4 使用整个磁盘

AutoYaST 允许通过将 disklabel 设置为 none（请参见第 4.6.3.1 节“驱动器配置”）来使用整个磁盘而不创建任何分区。在这种情况下，partition 中第一个 drive 中的配置将应用到整个磁盘。

在以下示例中，我们将使用第二个磁盘 (/dev/sdb) 作为 /home 文件系统。

例 4.5：使用整个磁盘作为文件系统

```
<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <partitions config:type="list">
      <partition>
        <create config:type="boolean">true</create>
        <format config:type="boolean">true</format>
        <mount>/</mount>
        <size>max</size>
      </partition>
    </partitions>
  </drive>
  <drive>
    <device>/dev/sdb</device>
    <disklabel>none</disklabel>
    <partitions config:type="list">
      <partition>
        <format config:type="boolean">true</format>
        <mount>/home</mount>
      </partition>
    </partitions>
  </drive>
```

此外，还可将整个磁盘用作 LVM 物理卷或软件 RAID 成员。有关设置 LVM 或软件 RAID 的更多细节，请参见第 4.6.5 节“逻辑卷管理器 (LVM)”和第 4.6.6 节“软件 RAID”。

出于向后兼容的原因，可以通过将 <partition_nr> 元素设置为 0 来实现相同的效果。但是，从 SUSE Linux Enterprise Server 15 开始，已弃用 <partition_nr> 元素的这种用法。

4.6.3.5 填补缺失

使用专家分区程序方法时，AutoYaST 可以基于一个不太完整的配置文件创建分区方案。以下配置文件演示了如何描述分区布局的一些细节，然后让 AutoYaST 完成余下的操作。

例 4.6：在选定的驱动器上自动分区

下面是单个驱动器系统的示例，该驱动器未预先分区，应根据描述的预定义分区计划进行自动分区。如果您未指定设备，系统将自动检测设备。

```
<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <use>all</use>
  </drive>
</partitioning>
```

有一个更详细的示例演示了如何处理现有分区和多个驱动器。

例 4.7：在多个驱动器上安装

```
<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <use>all</use>
    <partitions config:type="list">
      <partition>
        <mount>/</mount>
        <size>10G</size>
      </partition>
      <partition>
        <mount>swap</mount>
        <size>1G</size>
      </partition>
    </partitions>
  </drive>
  <drive>
    <device>/dev/sdb</device>
    <use>free</use>
    <partitions config:type="list">
      <partition>
        <filesystem config:type="symbol">ext4</filesystem>
        <mount>/data1</mount>
        <size>15G</size>
```

```
</partition>
<partition>
  <filesystem config:type="symbol">xfs</filesystem>
  <mount>/data2</mount>
  <size>auto</size>
</partition>
</partitions>
</drive>
</partitioning>
```

4.6.4 高级分区功能

4.6.4.1 擦除分区表

通常不需要执行此操作，因为 AutoYaST 可以自动地逐个删除分区。但是，您需要使用相应的选项来让 AutoYaST 清除分区表，而不是逐个删除分区。

转到 drive 部分并添加：

```
<initialize config:type="boolean">true</initialize>
```

指定此设置后，AutoYaST 会先删除分区表，然后再开始分析实际分区并计算其分区计划。当然，这也意味着您无法保留任何现有分区。

4.6.4.2 挂载选项

默认情况下，要挂载的文件系统在 /etc/fstab 中按设备名称标识。可以更改此标识，以便通过搜索 UUID 或卷标来查找文件系统。请注意，并非所有文件系统都可按 UUID 或卷标挂载。要指定如何挂载分区，请使用具有 mountby 类型的 symbol 属性。可能的选项为：

- device (默认)
- label
- UUID

如果您选择使用标签挂载新分区，请使用 `label` 属性指定其值。

在 `/etc/fstab` 的第四个字段中添加任何有效挂载选项。用逗号分隔多个选项。可能的 `fstab` 选项：

只读装入 (`ro`)

对文件系统没有写入访问权限。默认值为 `false`。

不更新访问时间 (`noatime`)

读取文件时不更新访问时间。默认值为 `false`。

可由用户挂载 (`user`)

文件系统可由普通用户挂载。默认值为 `false`。

数据日志模式 (`ordered`、`journal`、`writeback`)

`journal`

在将所有数据写入主文件系统之前将其提交到日志。

`ordered`

在将所有数据的元数据提交到日志之前，将所有数据直接写入主文件系统。

`writeback`

不保留数据顺序。

访问控制列表 (`acl`)

在文件系统上启用访问控制列表。

扩展用户属性 (`user_xattr`)

在文件系统上允许扩展用户属性。

例 4.8：挂载选项

```
<partitions config:type="list">
  <partition>
    <filesystem config:type="symbol">ext4</filesystem>
    <format config:type="boolean">true</format>
    <fstopt>ro,noatime,user,data=ordered,acl,user_xattr</fstopt>
    <mount>/local</mount>
```

```
<mountby config:type="symbol">uuid</mountby>
<partition_id config:type="integer">131</partition_id>
<size>10G</size>
</partition>
</partitions>
```



注意：查看支持的文件系统选项

不同的文件系统类型支持不同的选项。请在设置选项之前，请仔细查看相关文档。

4.6.4.3 保留特定的分区

在某些情况下，您应该将分区保留原样，仅格式化特定的目标分区，而不要从头开始创建分区。例如，如果有不同的 Linux 安装共存，或者您安装了另一个操作系统，则您可能不想要擦除这些分区。您可能想要将数据分区保留原样。

此类方案要求对目标系统和硬盘有专门的了解。根据具体的方案，您可能需要知道目标硬盘的确切分区表及其分区 ID、大小和编号。有了这些数据，您便可以告知 AutoYaST 保留特定的分区、格式化其他分区，并根据需要创建新分区。

以下示例将保留分区 1、2、5，并删除分区 6 以创建两个新分区。所有剩余分区只会格式化。

例 4.9：保留分区

```
<partitioning config:type="list">
  <drive>
    <device>/dev/sdc</device>
    <partitions config:type="list">
      <partition>
        <create config:type="boolean">>false</create>
        <format config:type="boolean">>true</format>
        <mount>/</mount>
        <partition_nr config:type="integer">1</partition_nr>
      </partition>
      <partition>
        <create config:type="boolean">>false</create>
        <format config:type="boolean">>false</format>
        <partition_nr config:type="integer">2</partition_nr>
```

```

        <mount>/space</mount>
    </partition>
    <partition>
        <create config:type="boolean">false</create>
        <format config:type="boolean">true</format>
        <filesystem config:type="symbol">swap</filesystem>
        <partition_nr config:type="integer">5</partition_nr>
        <mount>swap</mount>
    </partition>
    <partition>
        <format config:type="boolean">true</format>
        <mount>/space2</mount>
        <size>5G</size>
    </partition>
    <partition>
        <format config:type="boolean">true</format>
        <mount>/space3</mount>
        <size>max</size>
    </partition>
</partitions>
<use>6</use>
</drive>
</partitioning>

```

最后一个示例要求确切地了解应保留的分区的现有分区表和分区编号。但在某些情况下，可能不提供此类数据，尤其是在使用不同硬盘类型和配置的混合硬件环境中。以下方案适用于采用非 Linux 操作系统并可提供指定的区域进行 Linux 安装的系统。

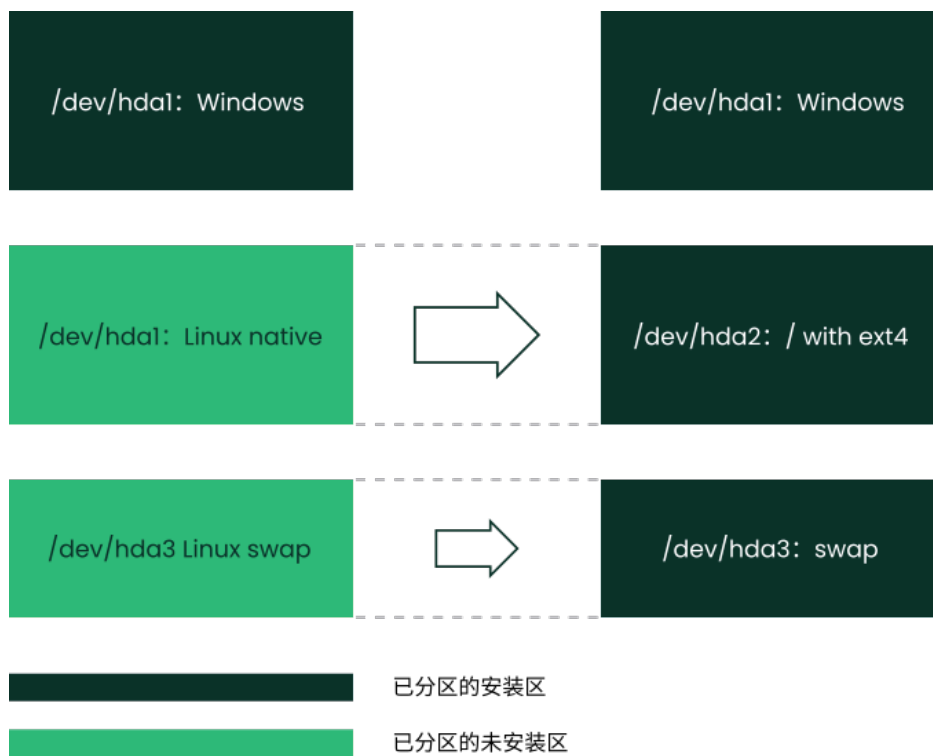


图 4.1：保留分区

在此方案中，如图 4.1 “保留分区”所示，AutoYaST 不会创建新分区，而是在系统上搜索特定的分区类型，并根据控制文件中的分区计划使用这些分区类型。在本例中未指定分区编号，而仅指定了挂载点和分区类型（可以提供其他配置数据，例如文件系统选项、加密和文件系统类型）。

例 4.10：自动检测要保留的分区。

```
<partitioning config:type="list">
  <drive>
    <partitions config:type="list">
      <partition>
        <create config:type="boolean">>false</create>
        <format config:type="boolean">>true</format>
        <mount></mount>
        <partition_id config:type="integer">131</partition_id>
      </partition>
      <partition>
        <create config:type="boolean">>false</create>
        <format config:type="boolean">>true</format>
```

```

    <filesystem config:type="symbol">swap</filesystem>
    <partition_id config:type="integer">130</partition_id>
    <mount>swap</mount>
  </partition>
</partitions>
</drive>
</partitioning>

```



注意：保留加密的设备

当 AutoYaST 探测储存设备时，尚未分析配置文件中的 partitioning 部分。在某些方案中，无法清楚地知道要使用哪个密钥来解锁设备。例如，定义了多个加密密钥时，就可能发生这种情况。为了解决此问题，AutoYaST 会尝试所有已加密设备上定义的所有密钥，直到找到有效的密钥。

4.6.5 逻辑卷管理器 (LVM)

要配置 LVM，请先使用前面所述的常规分区方法创建一个物理卷。

例 4.11：创建 LVM 物理卷

以下示例演示如何在 `partitioning` 资源中准备 LVM。将在类型为 `LVM` 并包含卷组 `system` 的设备 `/dev/sda1` 上创建一个未格式化的分区。此分区将使用驱动器上的所有可用空间。

```

<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <partitions config:type="list">
      <partition>
        <create config:type="boolean">true</create>
        <lvm_group>system</lvm_group>
        <partition_type>primary</partition_type>
        <partition_id config:type="integer">142</partition_id>
        <partition_nr config:type="integer">1</partition_nr>
        <size>max</size>
      </partition>
    </partitions>
  </drive>
</partitioning>

```



```

    </partition>
  </partitions>
  <use>all</use>
</drive>
</partitioning>

```

例 4.12：LVM 逻辑卷

```

<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <partitions config:type="list">
      <partition>
        <lvm_group>system</lvm_group>
        <partition_type>primary</partition_type>
        <size>max</size>
      </partition>
    </partitions>
    <use>all</use>
  </drive>
  <drive>
    <device>/dev/system</device>
    <type config:type="symbol">CT_LVM</type>
    <partitions config:type="list">
      <partition>
        <filesystem config:type="symbol">ext4</filesystem>
        <lv_name>user_lv</lv_name>
        <mount>/usr</mount>
        <size>15G</size>
      </partition>
      <partition>
        <filesystem config:type="symbol">ext4</filesystem>
        <lv_name>opt_lv</lv_name>
        <mount>/opt</mount>
        <size>10G</size>
      </partition>
      <partition>
        <filesystem config:type="symbol">ext4</filesystem>
        <lv_name>var_lv</lv_name>

```

```
<mount>/var</mount>
<size>1G</size>
</partition>
</partitions>
<pesize>4M</pesize>
<use>all</use>
</drive>
</partitioning>
```

对于逻辑卷，可将 `size` 设置为 `max`。当然，只能为一个逻辑卷使用 `max`。无法将一个卷组中的两个逻辑卷都设置为 `max`。

4.6.6 软件 RAID

对软件 RAID 设备的支持在 SUSE Linux Enterprise Server 15 SP2 中已得到显著改进。

如果需要，请参见第 4.6.6.1 节“使用已弃用的语法”来了解有关指定软件 RAID 的旧方法的更多细节，出于向后兼容考虑，这种旧方法仍受支持。

使用 AutoYaST 可以创建和组装软件 RAID 设备。支持的 RAID 级别如下：

RAID 0

此级别可提高磁盘性能。此模式**不**提供冗余。如果某个驱动器崩溃，将无法恢复数据。

RAID 1

此模式提供最佳冗余。在两个或更多个磁盘上可使用此模式。在所有磁盘上保留所有数据的精确副本。只要至少有一个磁盘仍可正常工作，就不会丢失数据。用于此 RAID 类型的分区大小应大致相同。

RAID 5

此模式结合了更多磁盘的管理功能，同时仍可保持一定程度的冗余。在三个或更多个磁盘上可使用此模式。如果有一个磁盘发生故障，所有数据仍将保持不变。如果两个磁盘同时发生故障，所有数据都会丢失。

多路径

此模式允许通过多个控制器访问同一物理设备，以提供冗余来防止控制器卡故障。在至少两个设备上可使用此模式。

类似于 LVM，AutoYaST 配置文件中的软件 RAID 定义由两个不同的部分构成：

- 确定要将哪些磁盘或分区用作 RAID 成员。为此，您需要在此类设备中设置 raid_name 元素。
- 使用专用的 drive 部分定义 RAID 本身。

以下示例演示了一种 RAID10 配置，该配置使用第一个和第二个磁盘中的一个分区作为 RAID 成员：

例 4.13：RAID10 配置

```
<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <partitions config:type="list">
      <partition>
        <mount>/</mount>
        <size>20G</size>
      </partition>
      <partition>
        <raid_name>/dev/md/0</raid_name>
        <size>max</size>
      </partition>
    </partitions>
    <use>all</use>
  </drive>
  <drive>
    <device>/dev/sdb</device>
    <disklabel>none</disklabel>
    <partitions config:type="list">
      <partition>
        <raid_name>/dev/md/0</raid_name>
      </partition>
    </partitions>
    <use>all</use>
  </drive>
  <drive>
    <device>/dev/md/0</device>
```

```

<partitions config:type="list">
  <partition>
    <mount>/home</mount>
    <size>40G</size>
  </partition>
  <partition>
    <mount>/srv</mount>
    <size>10G</size>
  </partition>
</partitions>
<raid_options>
  <chunk_size>4</chunk_size>
  <parity_algorithm>near_2</parity_algorithm>
  <raid_type>raid10</raid_type>
</raid_options>
<use>all</use>
</drive>
</partitioning>

```

如果您不想在软件 RAID 中创建分区，请将 `disklabel` 设置为 `none`，就像对常规磁盘所做那样。在以下示例中，为简洁起见，仅显示了 RAID `drive` 部分：

例 4.14：不包含分区的 RAID10

```

<drive>
  <device>/dev/md/0</device>
  <disklabel>none</disklabel>
  <partitions config:type="list">
    <partition>
      <mount>/home</mount>
      <size>40G</size>
    </partition>
  </partitions>
  <raid_options>
    <chunk_size>4</chunk_size>
    <parity_algorithm>near_2</parity_algorithm>
    <raid_type>raid10</raid_type>
  </raid_options>
  <use>all</use>

```

```
</drive>
```

4.6.6.1 使用已弃用的语法

如果启用了安装程序自我更新功能，则可以将 SUSE Linux Enterprise Server 15 的软件 RAID 分区。但是，该方案在以前的版本中不受支持，因此定义软件 RAID 的方法略有不同。

本节定义了旧式配置的样式，出于向后兼容的原因，仍然支持这种配置。

使用这种已弃用的语法配置 RAID 时请注意以下几点：

- RAID 的设备始终是 `/dev/md`。
- 属性 `partition_nr` 用于确定 MD 设备编号。如果 `partition_nr` 等于 0，则配置 `/dev/md/0`。添加多个 `partition` 部分表示您想要配置多个软件 RAID (`/dev/md/0`、`/dev/md/1` 等)。
- 所有特定于 RAID 的选项都包含在 `raid_options` 资源中。

例 4.15：旧式 RAID10 配置

```
<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <partitions config:type="list">
      <partition>
        <partition_id config:type="integer">253</partition_id>
        <format config:type="boolean">>false</format>
        <raid_name>/dev/md0</raid_name>
        <raid_type>raid1</raid_type>
        <size>4G</size>
      </partition>

      <!-- Insert a configuration for the regular partitions located on
           /dev/sda here (for example / and swap) -->

    </partitions>
    <use>all</use>
  </drive>
```

```

<drive>
  <device>/dev/sdb</device>
  <partitions config:type="list">
    <partition>
      <format config:type="boolean">>false</format>
      <partition_id config:type="integer">253</partition_id>
      <raid_name>/dev/md0</raid_name>
      <size>4gb</size>
    </partition>
  </partitions>
  <use>all</use>
</drive>
<drive>
  <device>/dev/md</device>
  <partitions config:type="list">
    <partition>
      <filesystem config:type="symbol">ext4</filesystem>
      <format config:type="boolean">>true</format>
      <mount>/space</mount>
      <partition_id config:type="integer">131</partition_id>
      <partition_nr config:type="integer">0</partition_nr>
      <raid_options>
        <chunk_size>4</chunk_size>
        <parity_algorithm>near_2</parity_algorithm>
        <raid_type>raid10</raid_type>
      </raid_options>
    </partition>
  </partitions>
  <use>all</use>
</drive>
</partitioning>

```

4.6.6.2 RAID 选项

以下元素必须放在以下 XML 结构中：

```

<partition>
  <raid_options>

```

```
...
</raid_options>
</partition>
```

chunk_size

可以用一个数字和相应单位（例如 32M）或仅使用一个数字表示。如果忽略单位，将使用 KB 作为默认单位。请勿为 RAID1 指定 chunk_size。请注意，raid1 是默认类型。

```
<chunk_size>4</chunk_size>
```

parity_algorithm

可能的值为：

left_asymmetric、left_symmetric、right_asymmetric、right_symmetric、first、last、first_6、left_asymmetric_6、left_symmetric_6、right_asymmetric_6、right_symmetric_6、near_2、offset_2、far_2、near_3、offset_3 和 far_3。

为了与旧版 AutoYaST 向后兼容，系统还可识别以下别名：

parity_first、parity_last、parity_first_6、n2、o2、f2、n3、o3 和 f3。

每个 RAID 可接受的值取决于 RAID 级别（例如 raid5）和 RAID 中的设备数量。如果 RAID0 或 RAID1 不提供任何奇偶校验，请勿为此类设备指定此选项。

```
<parity_algorithm>left_asymmetric</parity_algorithm>
```

raid_type

可能的值为：raid0、raid1、raid5、raid6 和 raid10。

```
<raid_type>raid1</raid_type>
```

默认值为 raid1。

device_order

此列表包含物理设备的顺序：

```
<device_order config:type="list"><device>/dev/sdb2</device><device>/dev/
sda1</device>...</device_order>
```

这是可选元素，默认采用字母顺序。

4.6.7 多路径支持

AutoYaST 可以处理多路径设备。要利用此类设备，您需要按照[例 4.16 “使用多路径设备”](#) 中所示启用多路径支持。或者，可以在内核命令行中使用以下参数：`LIBSTORAGE_MULTIPATH_AUTOSTART=ON`。

与 SUSE Linux Enterprise 12 不同，无需将 drive 部分类型设置为 `CT_DMMULTIPATH`。而应使用 `CT_DISK`，不过出于历史原因，这两个值是等效的。

例 4.16：使用多路径设备

```
<general>
  <storage>
    <start_multipath config:type="boolean">true</start_multipath>
  </storage>
</general>
<partitioning>
  <drive>
    <partitions config:type="list">
      <partition>
        <size>20G</size>
        <mount></mount>
        <filesystem config:type="symbol">ext4</filesystem>
      </partition>
      <partition>
        <size>auto</size>
        <mount>swap</mount>
      </partition>
    </partitions>
    <type config:type="symbol">CT_DISK</type>
    <use>all</use>
  </drive>
</partitioning>
```

如果您要指定设备，可以使用全球标识符 (WWID)、其设备名称（例如 `/dev/dm-0`）、`/dev/disk` 下引用多路径设备的任何其他路径，或者设备的任何路径。

以例 4.17 “列出多路径设备”中的 `multipath` 列表为例，您可以使用 `/dev/mapper/14945540000000000f86756dce9286158be4c6e3567e75ba5`、`/dev/dm-3` 或 `/dev/disk` 下的任何其他相应路径（如例 4.18 “使用 WWID 标识多路径设备”中所示），或设备的任何路径（`/dev/sda` 或 `/dev/sdb`）。

例 4.17：列出多路径设备

```
# multipath -l
14945540000000000f86756dce9286158be4c6e3567e75ba5 dm-3 ATA,VIRTUAL-DISK
size=40G features='0' hwhandler='0' wp=rw
|-+- policy='service-time 0' prio=1 status=active
|  '- 2:0:0:0 sda 8:0 active ready running
`-+- policy='service-time 0' prio=1 status=enabled
   '- 3:0:0:0 sdb 8:16 active ready running
```

例 4.18：使用 WWID 标识多路径设备

```
<drive>
  <partitions config:type="list">
    <device>/dev/mapper/14945540000000000f86756dce9286158be4c6e3567e75ba5</
device>
    <partition>
      <size>20G</size>
      <mount></mount>
      <filesystem config:type="symbol">ext4</filesystem>
    </partition>
  </partitions>
  <type config:type="symbol">CT_DISK</type>
  <use>all</use>
</drive>
```

4.6.8 bcache 配置

`bcache` 是允许使用多个快速驱动器来加速访问一个或多个慢速驱动器的缓存系统。例如，您可以使用一个快速驱动器作为缓存，来改善一个较大（但较慢）驱动器的性能。

有关 SUSE Linux Enterprise Server 上的 bcache 的详细信息，另请参见 <https://www.suse.com/c/combine-the-performance-of-solid-state-drive-with-the-capacity-of-a-hard-drive-with-bcache-and-yast/> 上的博文。

要设置 bcache 设备，AutoYaST 需要一个指定了以下设置的配置文件：

- 要将某个（慢速）块设备设置为**后备设备**，请使用 bcache_backing_for 元素。
- 要将某个（快速）块设备设置为**缓存设备**，请使用 bcache_caching_for 元素。您可以使用同一设备来加速访问多个驱动器。
- 要指定 bcache 设备的布局，请使用 drive 部分，并将 type 元素设置为 CT_BCACHE。bcache 设备的布局可以包含分区。

例 4.19：bcache 定义

```
<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <type config:type="symbol">CT_DISK</type>
    <use>all</use>
    <enable_snapshots config:type="boolean">true</enable_snapshots>
    <partitions config:type="list">
      <partition>
        <filesystem config:type="symbol">btrfs</filesystem>
        <mount>/</mount>
        <create config:type="boolean">true</create>
        <size>max</size>
      </partition>
      <partition>
        <filesystem config:type="symbol">swap</filesystem>
        <mount>swap</mount>
        <create config:type="boolean">true</create>
        <size>2GiB</size>
      </partition>
    </partitions>
  </drive>

  <drive>
    <type config:type="symbol">CT_DISK</type>
```

```

<device>/dev/sdb</device>
<disklabel>msdos</disklabel>
<use>all</use>
<partitions config:type="list">
  <partition>
    <!-- It can serve as caching device for several bcachees -->
    <bcache_caching_for config:type="list">
      <listentry>/dev/bcache0</listentry>
    </bcache_caching_for>
    <size>max</size>
  </partition>
</partitions>
</drive>

<drive>
  <type config:type="symbol">CT_DISK</type>
  <device>/dev/sdc</device>
  <use>all</use>
  <disklabel>msdos</disklabel>
  <partitions config:type="list">
    <partition>
      <!-- It can serve as backing device for one bcache -->
      <bcache_backing_for>/dev/bcache0</bcache_backing_for>
    </partition>
  </partitions>
</drive>

<drive>
  <type config:type="symbol">CT_BCACHE</type>
  <device>/dev/bcache0</device>
  <bcache_options>
    <cache_mode>writethrough</cache_mode>
  </bcache_options>
  <use>all</use>
  <partitions config:type="list">
    <partition>
      <mount>/data</mount>
      <size>20GiB</size>
    </partition>
  </partitions>
</drive>

```

```

    </partition>
    <partition>
      <mount>swap</mount>
      <filesystem config:type="symbol">swap</filesystem>
      <size>1GiB</size>
    </partition>
  </partitions>
</drive>
</partitioning>

```

目前，`bcache_options` 部分中唯一支持的选项是下面所述的 `cache_mode`。

cache_mode

`bcache` 的缓存模式。可能的值为：

- `writethrough`
- `writeback`
- `writearound`
- `none`

```
<cache_mode>writethrough</cache_mode>
```

4.6.9 多设备 Btrfs 配置

Btrfs 支持创建跨越多个储存设备的单个卷，提供与软件 RAID 实现类似的功能，例如 Linux 内核的内置 `mdraid` 子系统。**多设备 Btrfs** 比其他某些 RAID 实现更具优势。例如，可将多设备 Btrfs 卷从一个 RAID 级别动态迁移到另一个 RAID 级别、可按文件设置 RAID 级别等。但是，并非所有这些功能在 SUSE Linux Enterprise Server 15 SP7 中都仍受到完全支持。

使用 AutoYaST 时，可以通过指定 `CT_BTRFS` 类型的驱动器来配置多设备 Btrfs。`device` 属性用作标识每个多设备 Btrfs 的任意名称。

使用 RAID 时，您需要先创建所有块设备（例如分区、LVM 逻辑卷等），然后将它们指派给要在此类块设备上创建的 Btrfs 文件系统。

以下示例演示了一种简单的多设备 Btrfs 配置：

例 4.20：多设备 BTRFS 配置

```
<partitioning config:type="list">
  <drive>
    <device>/dev/sda</device>
    <disklabel>none</disklabel>
    <partitions>
      <partition>
        <btrfs_name>root_fs</btrfs_name>
      </partition>
    </partitions>
    <use>all</use>
  </drive>
  <drive>
    <device>/dev/sdb</device>
    <disklabel>gpt</disklabel>
    <partitions>
      <partition>
        <partition_nr>1</partition_nr>
        <size>4gb</size>
        <filesystem>ext4</filesystem>
        <btrfs_name>root_fs</btrfs_name>
      </partition>
    </partitions>
    <use>all</use>
  </drive>
  <drive>
    <device>root_fs</device>
    <type config:type="symbol">CT_BTRFS</type>
    <partitions>
      <partition config:type="list">
        <mount>/</mount>
      </partition>
    </partitions>
    <btrfs_options>
      <raid_level>raid1</raid_level>
      <metadata_raid_level>raid1</metadata_raid_level>
```

```
</btrfs_options>
</drive>
</partitioning>
```

支持的数据和元数据 RAID 级别为：[default](#)、[single](#)、[dup](#)、[raid0](#)、[raid1](#) 和 [raid10](#)。默认情况下，文件系统元数据会在两个设备中镜像，而数据会在所有设备间分条。如果只存在一个设备，则在这一个设备上复制元数据。

配置多设备 Btrfs 文件系统时请记住以下几点：

- 设备需要指明 [btrfs_name](#) 属性才能包含到多设备 Btrfs 文件系统中。
- 所有 Btrfs 特定的选项都包含在 [CT_BTRFS](#) 驱动器的 [btrfs_options](#) 资源中。

4.6.10 NFS 配置

AutoYaST 允许将 SUSE Linux Enterprise Server 安装到**网络文件系统** (NFS) 共享中。为此，您必须创建 [CT_NFS](#) 类型的驱动器，并提供 NFS 共享名 ([SERVER:PATH](#)) 作为设备名称。与挂载点相关的信息包含在其第一个 `partition` 部分中。请注意，对于 NFS 驱动器，只会考虑第一个分区。

有关安装系统后如何配置 NFS 客户端和服务器的详细信息，请参见[第 4.21 节 “NFS 客户端和服务端”](#)。

例 4.21：NFS 共享定义

```
<partitioning config:type="list">
  <drive>
    <device>192.168.1.1:/exports/root_fs</device>
    <type config:type="symbol">CT_NFS</type>
    <use>all</use>
    <partitions config:type="list">
      <partition>
        <mount>/</mount>
        <fstopt>nocheck</fstopt>
      </partition>
    </partitions>
  </drive>
</partitioning>
```

4.6.11 tmpfs 配置

如果将 `type` 元素设置为 `CT_TMPFS`，AutoYaST 将支持 `tmpfs` 虚拟文件系统的定义。每个 `partition` 部分代表一个 `tmpfs` 文件系统。

例 4.22：tmpfs 定义

```
<partitioning config:type="list">
  <drive>
    <type config:type="symbol">CT_TMPFS</type>
    <partitions config:type="list">
      <partition>
        <mount>/srv</mount>
        <fstopt>size=512M</fstopt>
      </partition>
      <partition>
        <mount>/temp</mount>
      </partition>
    </partitions>
  </drive>
</partitioning>
```

`tmpfs` 设备与普通文件系统（如 Ext4 或 Btrfs）不同。因此，仅有的相关元素是 `mount`（必需）和 `fstopt`。后者用于设置文件系统属性，例如其大小限制、模式等。`tmpfs` 手册页中提供了有关已知选项的更多信息。

4.6.12 特定于 IBM Z 的配置

4.6.12.1 配置 DASD 磁盘

下面列出的元素必须包含在以下 XML 结构中：

```
<dasd>
  <devices config:type="list">
    <listentry>
      ...
```

```
</listentry>
</devices>
</dasd>
```

<profile> 部分中的标记。需在单独的 <listentry> ... </listentry> 部分配置每个磁盘。

DASD 配置

device

DASD 是唯一允许的值。

```
<device>DASD</dev_name>
```

dev_name

指定要在此部分中配置的设备 (dasdN)。

```
<dev_name>/dev/dasda</dev_name>
```

可选，但建议指定。如果留空，AutoYaST 将尝试猜测设备。

channel

用于访问磁盘的通道。

```
<channel>0.0.0150</channel>
```

必需。

diag

启用或禁用 DIAG。可能的值为 true（启用）或 false（禁用）。

```
<diagconfig:type="boolean">true</diag>
```

可选。

4.6.12.2 配置 zFCP 磁盘

以下元素必须放在以下 XML 结构中：

```
<profile>
  <zfc>
    <devices config:type="list">
```



```
<listentry>
    ...
</listentry>
</devices>
</zfcp>
<profile>
```

每个磁盘都需要在单独的 `listentry` 部分进行配置，其中需要提供以下元素：

controller_id

通道编号

```
<controller_id>0.0.fc00</controller_id>
```

wwpn

全球端口号，用于连接 SCSI 设备的目标端口

```
<wwpn>0x500507630300c562</wwpn>
```

fcplun

逻辑单元号

```
<fcplun>0x4010403200000000</fcplun>
```

有关详细信息，请参见相应 IBM 文档：<https://www.ibm.com/docs/en/linux-on-systems?topic=wsd-configuring-devices>。

4.7 iSCSI 发起端概述

使用 `iscsi-client` 资源可将目标计算机配置为 iSCSI 客户端。

例 4.23：iSCSI 客户端

```
<iscsi-client>
  <initiatorname>iqn.2013-02.de.suse:01:e229358d2dea</initiatorname>
  <targets config:type="list">
    <listentry>
      <authmethod>None</authmethod>
      <portal>192.168.1.1:3260</portal>
```

```
<startup>onboot</startup>
<target>iqn.2001-05.com.doe:test</target>
<iface>default</iface>
</listentry>
</targets>
<version>1.0</version>
</iscsi-client>
```

ISCSI 发起端配置设置

initiatorname

InitiatorName 的值取自 /etc/iscsi/initiatorname.iscsi。如果您使用 iBFT，则会从 iBFT 添加此值，您只能在 BIOS 设置中更改此值。

version

YaST 模块的版本。默认值：1.0

targets

目标列表。每个项包含：

authmethod

身份验证方法：None/CHAP

portal

门户地址

startup

值：manual/onboot

target

目标名称

iface

接口名称

4.8 以太网光纤通道 (FCoE) 配置

使用 fcoe_cfg 资源可以配置以太网光纤通道 (FCoE)。

例 4.24：FCOE 配置

```
<fcoe-client>
  <fcoe_cfg>
    <DEBUG>no</DEBUG>
    <USE_SYSLOG>yes</USE_SYSLOG>
  </fcoe_cfg>
  <interfaces config:type="list">
    <listentry>
      <dev_name>eth3</dev_name>
      <mac_addr>01:000:000:000:42:42</mac_addr>
      <device>Gigabit 1313</device>
      <vlan_interface>200</vlan_interface>
      <fcoe_vlan>eth3.200</fcoe_vlan>
      <fcoe_enable>yes</fcoe_enable>
      <dcb_required>yes</dcb_required>
      <auto_vlan>no</auto_vlan>
      <dcb_capable>no</dcb_capable>
      <cfg_device>eth3.200</cfg_device>
    </listentry>
  </interfaces>
  <service_start>
    <fcoe config:type="boolean">true</fcoe>
    <lldpad config:type="boolean">true</lldpad>
  </service_start>
</fcoe-client>
```

fcoe_cfg

值：yes/no

DEBUG 用于启用或禁用 fcoe 服务脚本和 fcoemon 发出的调试消息。

如果设置为 yes，则会将 USE_SYSLOG 消息发送到系统日志。

interfaces

网卡列表，包括 VLAN 和 FCoE 配置的状态。

service_start

值：yes/no

启用或禁用在引导时启动 fcoe 和 lldpad 服务。

启动 `fcoe` 服务意味着启动以太网光纤通道服务守护程序 `fcoemon`，该守护程序用于控制 FCoE 接口并建立与 `lldpad` 守护程序的连接。

`lldpad` 服务提供链接层发现协议代理守护程序 `lldpad`，该守护程序会向 `fcoemon` 告知有关 DCB（数据中心桥接）功能以及接口的配置。

4.9 国家/地区设置

语言、时区和键盘设置。

例 4.25：语言

```
<language>
  <language>en_GB</language>
  <languages>de_DE,en_US</languages>
</language>
```

language

主要语言

languages

逗号分隔的次要语言

在 `/usr/share/YaST2/data/languages` 中可以找到可用语言的列表。

如果主要语言的配置值未知，它将被重置为默认值 `en_US`。

例 4.26：时区

```
<timezone>
  <hwclock>UTC</hwclock>
  <timezone>Europe/Berlin</timezone>
</timezone>
```

hwclock

硬件时钟使用的是本地时间还是 UTC。

值： `localtime/UTC`。

timezone

时区。

在 `/usr/share/YaST2/data/timezone_raw.ycp` 中可以找到可用时区的列表

例 4.27：键盘

```
<keyboard>
  <keymap>german</keymap>
</keyboard>
```

keymap

键盘布局

键映射代码值或键映射别名值均有效。在 `/usr/share/YaST2/lib/y2keyboard/keyboards.rb` 中可以找到可用项的列表。例如，`english-us`，`us`，`english-uk`，`uk`。

4.10 软件

4.10.1 产品选择

从 SUSE Linux Enterprise Server 15 开始，所有产品都是使用单个安装媒体分发的。因此，您需要使用 `product` 标记选择要安装的产品。

`product` 标记的可用值为：

SLES

SUSE Linux Enterprise Server

SLE_HPC

SUSE Linux Enterprise High Performance Computing

SLE_RT

SUSE Linux Enterprise Real Time

SLES_SAP

SUSE Linux Enterprise Server for SAP Applications

SLED

SUSE Linux Enterprise Desktop

SUSE-manager-server

SUSE Multi-Linux Manager Server

SUSE-manager-retail-branch-server

SUSE Multi-Linux Manager for Retail

SUSE-manager-proxy

SUSE Multi-Linux Manager Proxy

例 4.28：明确选择产品

在以下示例中，所选择的产品为 SUSE Linux Enterprise Desktop：

```
<software>
  <products config:type="list">
    <product>SLED</product>
  </products>
</software>
```

在特殊情况下，媒体可能只包含一个产品。如果是这样，则您无需按上面所述明确选择产品。AutoYaST 将自动选择唯一可用的产品。



注意：使用先前版本中的 AutoYaST 文件

如果您正在使用或迁移旧版 SUSE Linux Enterprise Server 中的 AutoYaST 配置文件，请注意一些特别事项。有关细节，请参见 [第 D1 节“产品选择”](#)。

4.10.2 使用 patterns 和 packages 部分选择软件包

软件集或软件包的配置如下所示：

例 4.29：在控制文件中使用 PATTERNS 和 PACKAGES 部分选择软件包

```
<software>
  <patterns config:type="list">
    <pattern>directory_server</pattern>
  </patterns>
```

```
<packages config:type="list">
  <package>apache</package>
  <package>postfix</package>
</packages>
<do_online_update config:type="boolean">true</do_online_update>
</software>
```



注意：软件包和软件集名称

这些值是真实的软件包或软件集名称。如果软件包名称由于升级而发生更改，则您也需要调整这些设置。

可以使用正则表达式指定软件包和软件集名称。在这种情况下，AutoYaST 将选择与表达式匹配的所有软件包或软件集。请注意，必须在此类表达式的两侧添加斜线。在[例 4.30 “使用正则表达式选择软件包”](#)中，将选择名称以 `nginx` 开头的所有软件包（例如，`nginx` 和 `nginx-macros`）。

例 4.30：使用正则表达式选择软件包

```
<software>
  <packages config:type="list">
    <package>/nginx.*</package>
  </packages>
</software>
```

4.10.3 安装其他/自定义的软件包或产品

除了 DVD-ROM 中可供安装的软件包以外，您还可以添加外部软件包（包括自定义的内核）。自定义的内核软件包必须与 SUSE 软件包兼容，并且内核文件必须安装到相同的位置。

与先前版本不同，您无需在控制文件中使用特殊的资源即可安装自定义和外部软件包。不过，您需要重新创建软件包数据库，并使用源储存库中的任何新软件包或新软件包版本更新该数据库。

我们提供了一个脚本用于完成此任务。该脚本会查询储存库中提供的软件包并创建软件包数据库。使用命令 `/usr/bin/create_package_descr`。openSUSE Build Service 的 `inst-source-utils` 软件包中提供了此脚本。创建数据库时，所有语言将重置为英语。

例 4.31：使用附加软件包 `INST-SOURCE-UTILS.RPM` 创建软件包数据库

解包的 DVD 位于 `/usr/local/DVDs/LATEST` 中。

```
> cp /tmp/inst-source-utils-2016.7.26-1.2.noarch.rpm /usr/local/DVDs/
LATEST/suse/noarch
> cd /usr/local/DVDs/LATEST/suse
> create_package_descr -d /usr/local/CDs/LATEST/suse
```

在上面的示例中，`/usr/local/CDs/LATEST/suse` 目录包含与体系结构相关的软件包（例如 `x86_64`）以及与体系结构不相关的软件包（`noarch`）。在其他体系结构上，可能并不相同。

此方法的优势是可以保留包含固定软件包和已更新软件包的最新储存库。此外，此方法可以简化自定义 CD-ROM 的创建。

要添加您自己的模块，例如 SDK (SUSE Software Development Kit)，请将文件 `add_on_products.xml` 添加到安装源的根目录中。

以下示例演示如何将 SDK 模块添加到基础产品储存库。完整的 SDK 储存库将储存在 `/sdk` 目录中。

例 4.32：`add_on_products.xml`

此文件描述基础产品中包含的 SDK 模块。

```
<?xml version="1.0"?>
<add_on_products xmlns="http://www.suse.com/1.0/yast2ns"
  xmlns:config="http://www.suse.com/1.0/configs">
  <product_items config:type="list">
    <product_item>
      <name>SUSE Linux Enterprise Software Development Kit</name>
      <url>relurl:///sdk?alias=SLE_SDK</url>
      <path></path>
      <!-- Users are asked whether to add such a product -->
      <ask_user config:type="boolean">false</ask_user>
      <!-- Defines the default state of pre-selected state in case of
ask_user used. -->
      <selected config:type="boolean">true</selected>
    </product_item>
  </product_items>
</add_on_products>
```


除这种特殊情况外，在安装 AutoYaST 期间，还几乎可以从其他每个位置添加所有其他模块、扩展和附加产品。



注意：注册服务器提供的储存库

如果您要使用注册服务器（SMT、RMT 或 SCC）提供的附加产品，请在 `suse_register` 部分定义这些附加产品。请参见第 4.3.1 节“扩展”。

在安装期间甚至可以添加没有任何产品或模块信息的储存库。这类软件包称为 `other add-ons`。

例 4.33：添加 SDK 扩展和用户定义的储存库

```
<add-on>
  <add_on_products config:type="list">
    <listentry>
      <media_url>cd:///sdk</media_url>
      <product>sle-sdk</product>
      <alias>SLE SDK</alias>
      <product_dir>/</product_dir>
      <priority config:type="integer">20</priority>
      <ask_on_error config:type="boolean">>false</ask_on_error>
      <confirm_license config:type="boolean">>false</confirm_license>
      <name>SUSE Linux Enterprise Software Development Kit</name>
    </listentry>
  </add_on_products>
  <add_on_others config:type="list">
    <listentry>
      <media_url>https://download.opensuse.org/repositories/YaST:/Head/
openSUSE_Leap_15.2/</media_url>
      <alias>yast2_head</alias>
      <priority config:type="integer">30</priority>
      <name>Latest YaST2 packages from OBS</name>
    </listentry>
  </add_on_others>
</add-on>
```

`add_on_others` 和 `add_on_products` 部分支持相同的值：

media_url

产品 URL。可以包含前缀 `cd:///`、`http://`、`ftp://` 等。此项是必需的。

如果您使用 SUSE Linux Enterprise Packages DVD 等多产品媒体，则 URL 路径应指向多产品媒体的根目录。使用 `product_dir` 值选择特定的产品目录（参见下文）。

product

内部产品名（如果附加项是产品）。命令 `zypper products` 显示已安装产品的名称。

alias

储存库别名。由用户定义。

product_dir

可选的子路径。此项只可用于 SUSE Linux Enterprise Packages DVD 等多产品媒体。

priority

设置储存库 libzypp 优先级。1 是最高优先级。数字越大，优先级越低。默认值为 99。

ask_on_error

在给定的位置找不到储存库时，AutoYaST 可以要求用户提供附加产品、模块或扩展，而不是报告超时错误。将 `ask_on_error` 设置为 `true`（默认值为 `false`）。

confirm_license

用户需要确认许可。默认值为 `false`。

name

储存库名称。命令 `zypper lr` 显示已添加的储存库的名称。

要在 AutoYaST 中使用未签名的安装源，请在 AutoYaST 控制文件中使用以下配置关闭检查。



注意：未签名的安装源 — 限制

只能在自动安装过程的第一个阶段禁用签名检查。在第二个阶段，已安装系统的配置将优先于 AutoYaST 配置。

下面列出的元素必须包含在以下 XML 结构中：

```
<general>
  <signature-handling>
```

```
...
</signature-handling>
</general>
```

所有选项的默认值均为 `false`。如果某个选项设置为 `false`，而某个软件包或储存库未通过相关测试，则会以静默方式忽略而不予安装。请注意，将其中的任何选项设置为 `true` 都有可能带来安全风险。使用第三方源提供的软件包或储存库时请勿这样做。

accept_unsigned_file

如果设为 `true`，AutoYaST 将接受未签名的文件，例如内容文件。

```
<accept_unsigned_file config:type="boolean" >true</accept_unsigned_file>
```

accept_file_without_checksum

如果设置为 `true`，AutoYaST 将接受内容文件中没有校验和的文件。

```
<accept_file_without_checksum config:type="boolean" >true</
accept_file_without_checksum>
```

accept_verification_failed

如果设置为 `true`，即使签名校验失败，AutoYaST 也会接受这些已签名的文件。

```
<accept_verification_failed config:type="boolean" >true</
accept_verification_failed>
```

accept_unknown_gpg_key

如果设置为 `true`，AutoYaST 将接受安装源的新 GPG 密钥，例如，用来对内容文件签名的密钥。

```
<accept_unknown_gpg_key config:type="boolean" >true</
accept_unknown_gpg_key>
```

accept_non_trusted_gpg_key

将此选项设置为 `true` 会接受您尚未信任的已知密钥。

```
<accept_non_trusted_gpg_key config:type="boolean" >true</
accept_non_trusted_gpg_key>
```

import_gpg_key

如果设置为 true，AutoYaST 将接受安装源上的新 GPG 密钥并将其导入自身的数据库中。

```
<import_gpg_key config:type="boolean" >true</import_gpg_key>
```

可以单独配置每个附加产品、模块或扩展的签名处理。以下元素必须位于各个附加产品、模块或扩展的 signature-handling 部分之间。所有设置都是可选的。如果未配置，将使用 general 部分中的全局签名处理方式。

accept_unsigned_file

如果设为 true，AutoYaST 将接受未签名的文件，例如此附加产品的内容文件。

```
<accept_unsigned_file config:type="boolean" >true</accept_unsigned_file>
```

accept_file_without_checksum

如果设置为 true，AutoYaST 将接受此附加产品的内容文件中没有校验和的文件。

```
<accept_file_without_checksum config:type="boolean" >true</  
accept_file_without_checksum>
```

accept_verification_failed

如果设置为 true，即使签名校验失败，AutoYaST 也会接受这些已签名的文件。

```
<accept_verification_failed config:type="boolean" >true</  
accept_verification_failed>
```

accept_unknown_gpg_key

如果 all 设置为 true，AutoYaST 将接受安装源上的新 GPG 密钥。

```
<accept_unknown_gpg_key> <all config:type="boolean">true</all> </  
accept_unknown_gpg_key>
```

或者，您可以定义单个密钥：

```
<accept_unknown_gpg_key> <all config:type="boolean">false</all> <keys  
config:type="list"> <keyid>3B3011B76B9D6523</keyid> lt;/keys> </  
accept_unknown_gpg_key>
```

accept_non_trusted_gpg_key

这表示密钥已知，但不受您的信任。可以通过添加以下命令来信任所有密钥：

```
<accept_non_trusted_gpg_key> <all config:type="boolean">true</all> </accept_non_trusted_gpg_key>
```

或者，您可以信任特定的密钥：

```
<accept_non_trusted_gpg_key> <all config:type="boolean">false</all>  
  <keys config:type="list"> <keyid>3B3011B76B9D6523</keyid> </keys> </accept_non_trusted_gpg_key>
```

import_gpg_key

如果 `all` 设置为 `true`，AutoYaST 将接受安装源上的所有新 GPG 密钥并将其导入自身的数据库中。

```
<import_gpg_key> <all config:type="boolean">true</all> </import_gpg_key>
```

只能对特定的密钥执行此操作：

```
<import_gpg_key> <all config:type="boolean">false</all> <keys  
  config:type="list"> <keyid>3B3011B76B9D6523</keyid> </keys> </import_gpg_key>
```

4.10.4 内核软件包

内核软件包不在任何选项内。所需内核是在安装期间确定的。如果将内核软件包添加到任何选项或单个软件包选项，安装基本上都会由于冲突而失败。

要强制安装特定的内核，请使用 `kernel` 属性。下面是强制安装默认内核的示例。即使需要 SMP 或其他内核，也会安装此内核。

例 4.34：在控制文件中选择内核

```
<software>  
  <kernel>kernel-default</kernel>  
  ...
```

```
</software>
```

4.10.5 去除自动选择的软件包

某些软件包是由于存在依赖关系或者可供选择而被自动选择的。

去除这些软件包可能会破坏系统一致性，除非已获得可提供相同服务的替代软件包，否则不建议去除基本软件包。邮件传输代理 (MTA) 软件包就是其中最典型的示例。默认会选择并安装 `postfix`。要使用其他 MTA（例如 `sendmail`），可以使用软件资源中的列表从选定软件包列表中去除 `postfix`。但请注意，SUSE Linux Enterprise Server 并未随附 `sendmail`。以下示例演示了如何执行此操作：

例 4.35： 在控制文件中选择软件包

```
<software>
  <packages config:type="list">
    <package>sendmail</package>
  </packages>
  <remove-packages config:type="list">
    <package>postfix</package>
  </remove-packages>
</software>
```



注意：软件包去除失败

请注意，无法去除某个软件集中的软件包（请参见第 4.10.2 节“使用 `patterns` 和 `packages` 部分选择软件包”）。如果指定去除此类软件包，安装将会失败并出现以下错误消息：

```
The package resolver run failed. Check
your software section in the AutoYaST profile.
```

4.10.6 安装推荐的软件包和软件集

使用 AutoYaST，您可以控制安装哪些**推荐**的软件包和软件集。共有三个选项：

- 安装所有推荐的软件包和软件集
- 仅安装必需的软件包和软件集
- 安装推荐的软件包，忽略推荐的软件集

在配置文件中将 `install_recommended` 标志设为 `true` 会安装所有推荐的软件包和软件集。

如果您需要进行极简安装，只安装**必需**的软件包和软件集，请将该标志设为 `false`。

在配置文件中省略该标志将只安装推荐的软件包，会忽略所有推荐的软件集。请注意，此标志只会影响全新安装，在升级期间将被忽略。



注意：install_recommended 标志仅影响安装过程

请注意，该标志仅会影响安装过程中的软件包解析程序，不会更改 `/etc/zypp/zypp.conf` 中的任何设置。因此，运行中系统上的软件包解析不会受到此 AutoYaST 设置的影响。

```
<software>
  <install_recommended config:type="boolean">false
</install_recommended>
</software>
```

4.10.7 在第 2 个阶段安装软件包

要在第二个阶段重引导后安装软件包，可以使用 `post-packages` 元素：

```
<software>
  <post-packages config:type="list">
    <package>yast2-cim</package>
  </post-packages>
</software>
```

4.10.8 在第 2 个阶段安装软件集

您还可以在第 2 个阶段安装软件集。为此，请使用 `post-patterns` 元素：

```
<software>
  <post-patterns config:type="list">
    <pattern>apparmor</pattern>
  </post-patterns>
</software>
```

4.10.9 在第 2 个阶段联机更新

可以在安装结束时执行联机更新。将布尔值 `do_online_update` 设置为 `true`。当然，仅当您在 `suse-register/customer-center` 部分或者安装后脚本中添加了联机更新储存库时，此设置才起作用。如果已通过 `add-on` 部分在第一个阶段提供了联机更新储存库，则 AutoYaST 应已安装了可用的最新软件包。如果内核更新是通过联机更新完成的，则第二个阶段结束时会触发重引导。

```
<software>
  <do_online_update config:type="boolean">true</do_online_update>
</software>
```

4.11 升级

AutoYaST 还可用于执行系统升级。除升级软件包以外，还支持以下部分：

- `scripts/pre-scripts` 在所有其他操作实际发生之前的早期阶段运行用户脚本。
- `add-on` 定义其他附加产品。
- `language` 设置语言。
- `timezone` 设置时区。
- `keyboard` 设置键盘。
- `software` 安装其他软件/软件集。去除已安装的软件包。
- `suse_register` 运行注册过程。

要控制升级过程，可定义以下部分：

例 4.36：升级和备份

```
<upgrade>
  <stop_on_solver_conflict config:type="boolean">true</
stop_on_solver_conflict>
</upgrade>
<backup>
  <sysconfig config:type="boolean">true</sysconfig>
  <modified config:type="boolean">true</modified>
  <remove_old config:type="boolean">true</remove_old>
</backup>
```

stop_on_solver_conflict

出现软件包依赖性问题时暂停安装。

modified

创建已修改文件的备份。

sysconfig

创建 /etc/sysconfig 目录的备份。

remove_old

从先前的更新中去除备份。

要启动 AutoYaST 升级模式，您需要：

过程 4.1：以脱机升级模式启动 AUTOYAST

1. 将 AutoYaST 配置文件复制到你文件系统上的 /root/autoupg.xml 中。
2. 从安装媒体引导系统。
3. 选择 Upgrade 菜单项。
4. 在命令行上设置 autoupgrade=1。
5. 按 **Enter** 键启动升级过程。

过程 4.2：以联机升级模式启动 AUTOYAST

1. 从安装媒体引导系统。
2. 选择 Upgrade 菜单项。
3. 在命令行上设置 `netsetup=dhcp autoupgrade=1 autoyast=http://192.169.3.1/autoyast.xml`。
此处将通过 DHCP 设置网络。
4. 按 **Enter** 键启动升级过程。

4.12 服务和目标

使用 `services-manager` 资源可以设置默认的 `systemd` 目标，并详细指定要启动或停用哪些系统服务及其启动方式。

`default-target` 属性指定系统要引导进入的默认 `systemd` 目标。有效选项为 `graphical`（图形登录）或 `multi-user`（控制台登录）。

要指定引导时应启动的一组服务，请使用 `enable` 和 `disable` 列表。要启动某个服务，请将其名称添加到 `enable` 列表。要确保该服务在引导时不会启动，请将其添加到 `disable` 列表。

如果某个服务未列为启用或禁用，将使用默认设置。默认设置可以是禁用也可以是启用。

最后，某些服务（例如 `cups`）支持按需激活（套接字激活的服务）。如果您想要利用此类功能，请在 `on_demand` 列表而不是 `enable` 中列出这些服务的名称。

例 4.37：配置服务和目标

```
<services-manager>
  <default_target>multi-user</default_target>
  <services>
    <disable config:type="list">
      <service>libvirt</service>
    </disable>
    <enable config:type="list">
      <service>sshd</service>
    </enable>
    <on_demand config:type="list">
      <service>cups</service>
    </on_demand>
  </services>
</services-manager>
```

```
</on_demand>
</services>
</services-manager>
```

4.13 网络配置

4.13.1 配置工作流程

网络配置主要用于将单台工作站连接到基于以太网的 LAN。通常在 AutoYaST 启动前配置，以从网络位置提取配置文件。一般通过 **linuxrc** 来完成网络配置。



注意：linuxrc 程序

有关 **linuxrc** 的工作原理及其关键字的详细说明，请参见附录 C “高级 **linuxrc** 选项”。

默认情况下，YaST 会将安装期间使用的网络设置复制到最终安装的系统中。此配置会与 AutoYaST 配置文件中定义的配置合并。

AutoYaST 设置的优先级高于任何现有配置文件的优先级。YaST 将基于配置文件中的项编写 **ifcfg-*** 文件，且不去除旧项。如果 DNS 和 routing 部分为空或缺失，YaST 将保留任何先前存在的值。如果该部分存在，则会应用配置文件中的设置。



注意：在安装期间使用 AutoYaST 网络设置

从 SUSE Linux Enterprise Server 15.3 开始，基于配置文件写入配置的步骤发生在第一个阶段结束时。

但如果安装期间需要网络设置，您可以强制 AutoYaST 在进行注册前写入并应用这些设置，只需将 **setup_before_proposal** 选项设为 **true** 即可。如果需要进行网络安装，要求 AutoYaST 在早期阶段设置网络非常有用，但该配置太复杂，无法使用 **linuxrc** 定义（请参见第 9.3.2 节 “自动安装单个系统”）。

```
<setup_before_proposal config:type="boolean">true</setup_before_proposal>
```

如果在安装结束时写入配置，只有当系统重引导后才会应用配置。

网络设置和服务激活选项在 `profile networking` 全局资源中定义。

4.13.2 网络资源

例 4.38：网络配置

```
<networking>
  <dns>
    <dhcp_hostname config:type="boolean">true</dhcp_hostname>
    <hostname>linux-bqua</hostname>
    <nameservers config:type="list">
      <nameserver>192.168.1.116</nameserver>
      <nameserver>192.168.1.117</nameserver>
      <nameserver>192.168.1.118</nameserver>
    </nameservers>
    <resolv_conf_policy>auto</resolv_conf_policy>
    <searchlist config:type="list">
      <search>example.com</search>
      <search>example.net</search>
    </searchlist>
  </dns>
  <interfaces config:type="list">
    <interface>
      <bootproto>dhcp</bootproto>
      <name>eth0</name>
      <startmode>auto</startmode>
    </interface>
  </interfaces>
  <ipv6 config:type="boolean">true</ipv6>
  <keep_install_network config:type="boolean">false</keep_install_network>
  <managed config:type="boolean">false</managed>
  <net-udev config:type="list">
    <rule>
      <name>eth0</name>
```

```

        <rule>ATTR{address}</rule>
        <value>00:30:6E:08:EC:80</value>
    </rule>
</net-udev>
<s390-devices config:type="list">
    <listentry>
        <chanids>0.0.0800:0.0.0801:0.0.0802</chanids>
        <type>qeth</type>
    </listentry>
</s390-devices>
<routing>
    <ipv4_forward config:type="boolean">>false</ipv4_forward>
    <ipv6_forward config:type="boolean">>false</ipv6_forward>
    <routes config:type="list">
        <route>
            <destination>192.168.100.0/24</destination>
            <device>eth1</device>
            <extrapara>scope link src 192.168.100.100 table one</extrapara>
            <gateway>-</gateway>
        </route>
        <route>
            <destination>default</destination>
            <device>eth1</device>
            <gateway>192.168.100.1</gateway>
        </route>
        <route>
            <destination>default</destination>
            <device>lo</device>
            <gateway>192.168.5.1</gateway>
        </route>
    </routes>
</routing>
</networking>

```

如上述示例所示，<networking> 部分可以包含几个子部分：

- interfaces 描述网络接口的配置，包括其 IP 地址、启动方式等。
- dns 指定与 DNS 相关的设置，例如主机名、名称服务器列表等。

- [routing](#) 定义路由规则。
- [s390-devices](#) 包含 z Systems 专用的设备设置。
- [net-udev](#) 枚举用于设置持久名称的 udev 规则。

另外，还有几个元素可用于修改网络配置的应用方式：

backend

选择要使用的网络后端。支持的值为 [wicked](#)、[network_manager](#) 或 [none](#)，最后一个值将禁用网络服务。

```
<backend>network_manager</backend>
```

keep_install_network

如第 4.13.1 节“配置工作流程”中所述，AutoYaST 默认会将来自运行中系统的网络配置与配置文件中定义的相应配置进行合并。如果您只想使用配置文件中的配置，请将此元素设置为 [false](#)。该值默认为 [true](#)。

```
<keep_install_network config:type="boolean">false</keep_install_network>
```

managed

确定是否使用 NetworkManager 取代 Wicked。
已弃用。请改用 [backend](#)。

```
<managed config:type="boolean">true</managed>
```

start_immediately

写入配置后，立即强制 AutoYaST 重新启动网络。

```
<start_immediately config:type="boolean">true</start_immediately>
```

setup_before_proposal

在安装过程中使用配置文件中定义的网络配置。否则，AutoYaST 将依赖于 [linuxrc](#) 设置的配置。

```
<setup_before_proposal config:type="boolean">true</setup_before_proposal>
```

strict_IP_check_timeout

设置网络后，AutoYaST 会检查指派的 IP 地址是否重复。如果重复，它会显示一条警告，警告的超时（以秒为单位）就是通过此元素控制。如果此元素设置为 0，则会停止安装。

```
<strict_IP_check_timeout config:type="integer">5</strict_IP_check_timeout>
```

virt_bridge_proposal

如果选择安装虚拟化软件包（例如，Xen、QEMU 或 KVM），AutoYaST 会配置网桥。您可以通过将此元素设为 false 来禁用此行为。

```
<virt_bridge_proposal config:type="boolean">false</virt_bridge_proposal>
```



提示：IPv6 地址支持

完全支持在 AutoYaST 中使用 IPv6 地址。要禁用 IPv6 地址支持，请设置 `<ipv6 config:type="boolean">false</ipv6>`

4.13.3 接口

interfaces 部分可让用户定义接口的配置，包括接口的启动方式、其 IP 地址、网络等。以下元素必须包含在 `<interfaces>...</interfaces>` 标记中。

bootproto

接口使用的引导协议。可能的值：

- static 表示静态指派的地址。需使用 ipaddr 元素指定 IP。
- dhcp4、dhcp6 或 dhcp，用于通过 DHCP 设置 IP 地址（IPv4、IPv6 或任何地址）。
- dhcp+autoip 用于从 **Zeroconf** 获取 IPv4 配置，或从 DHCP 获取 IPv6 配置。
- autoip 用于从 **Zeroconf** 获取 IPv4 配置。
- ibft 用于通过 iBFT 协议获取 IP 地址。
- none 表示跳过地址设置。此值用于桥接和绑定端口。

必需。

broadcast

广播 IP 地址。

仅适用于 static 引导协议。

device

设备名称。

已弃用。请改用 name。

name

设备名称，例如：eth0。

必需。

lladdr

链路层地址（MAC 地址）。

可选。

ipaddr

指派给接口的 IP 地址。

仅适用于 static 引导协议。它可以包含网络前缀，例如：192.168.1.1/24。

remote_ipaddr

用于点对点连接的远程 IP 地址。

仅适用于 static 引导协议。

netmask

网络掩码，例如：255.255.255.0。

已弃用。请改用 prefixlen，或者在 ipaddr 元素中包含网络前缀。

network

网络 IP 地址。

已弃用。请改为结合 prefixlen 使用 ipaddr。

prefixlen

网络前缀，例如：24。

仅适用于 static 引导协议。

startmode

何时启动接口。可能的值为：

- hotplug（插入设备时。例如，此值可用于 USB 网卡）。
- auto（系统引导时）。onboot 是已弃用的别名。
- ifplugd（设备由 ifplugd 守护程序管理时）。
- manual（应手动启动设备时）。
- nfsroot（需要将设备挂载到根文件系统时，例如，当 / 位于 NFS 卷上时）。
- off（永不启动设备）。

ifplugd_priority

ifplugd 守护程序的优先级。它确定设备的激活顺序。

仅适用于 ifplugd 启动模式。

usercontrol

不再使用的参数。

已弃用。

bonding_slaveX

绑定设备的名称。

对于绑定设备是必需的。X 将替换为从 0 开始的编号，例如 bonding_slave0。每个端口需有一个唯一编号。

bonding_module_opts

绑定设备的选项。

仅适用于 bond 设备。

mtu

接口的最大传送单位。

可选。

ethtool_options

设备激活期间的 Ethtool 选项。

可选。

zone

接口指派到的防火墙区域名称。

可选。

vlan_id

此 VLAN 使用的标识符。

仅适用于 vlan 设备。

etherdevice

VLAN 挂接到的设备。

仅适用于 vlan 设备以及需要它的设备。

bridge

如果接口是网桥，则为 yes。

已弃用。从其他属性推断而来。

bridge_ports

网桥端口的空格分隔列表，例如 eth0 eth1。

仅适用于 bridge 设备以及需要它的设备。

bridge_stp

生成树协议。可能的值为 on（启用时）和 off（禁用时）。

仅适用于 bridge 设备。

bridge_forward_delay

网桥的转发延迟，例如：15。

仅适用于 bridge 设备。有效值为 4 到 30。

aliases

其他 IP 地址。请参见第 4.13.4 节“指派多个 IP 地址”。

例 4.39：绑定接口配置

```
<networking>
  <setup_before_proposal config:type="boolean">false</setup_before_proposal>
  <keep_install_network config:type="boolean">false</keep_install_network>
  <interfaces config:type="list">
```

```

<interface>
  <bonding_master>yes</bonding_master>
  <bonding_module_opts>mode=active-backup miimon=100</bonding_module_opts>
  <bonding_slave0>eth1</bonding_slave0>
  <bonding_slavel>eth2</bonding_slavel>
  <bootproto>static</bootproto>
  <name>bond0</name>
  <ipaddr>192.168.1.61</ipaddr>
  <prefixlen>24</prefixlen>
  <startmode>auto</startmode>
</interface>
<interface>
  <bootproto>none</bootproto>
  <name>eth1</name>
  <startmode>auto</startmode>
</interface>
<interface>
  <bootproto>none</bootproto>
  <name>eth2</name>
  <startmode>auto</startmode>
</interface>
</interfaces>
<net-udev config:type="list">
  <rule>
    <name>eth1</name>
    <rule>ATTR{address}</rule>
    <value>dc:e4:cc:27:94:c7</value>
  </rule>
  <rule>
    <name>eth2</name>
    <rule>ATTR{address}</rule>
    <value>dc:e4:cc:27:94:c8</value>
  </rule>
</net-udev>
</networking>

```

例 4.40：网桥接口配置

```

<interfaces config:type="list">

```

```

<interface>
  <name>br0</name>
  <bootproto>static</bootproto>
  <bridge>yes</bridge>
  <bridge_forwarddelay>0</bridge_forwarddelay>
  <bridge_ports>eth0 eth1</bridge_ports>
  <bridge_stp>off</bridge_stp>
  <ipaddr>192.168.1.100</ipaddr>
  <prefixlen>24</prefixlen>
  <startmode>auto</startmode>
</interface>
<interface>
  <name>eth0</name>
  <bootproto>none</bootproto>
  <startmode>hotplug</startmode>
</interface>
<interface>
  <name>eth1</name>
  <bootproto>none</bootproto>
  <startmode>hotplug</startmode>
</interface>
</interfaces>

```

4.13.4 指派多个 IP 地址

AutoYaST 允许将多个 IP 地址指派给同一个接口。使用包含每个地址的 aliases 项的 aliasX 元素来指定多个地址。

每个项都支持以下元素：

IPADDR

其他 IP 地址。它可以包含网络前缀，例如：192.168.1.1/24。

PREFIXLEN

网络前缀，例如：24。

NETMASK

地址的网络掩码。

已弃用。请改用 PREFIXLEN，或者在 IPADDR 元素中包含网络前缀。

LABEL

地址的标签。



注意：区分大小写的元素

请注意，由于历史原因，aliases 部分中的 IPADDR、PREFIXLEN、LABEL 和 NETMASK 元素区分大小写。

例 4.41：多个 IP 地址

```
<interfaces config:type="list">
  <interface>
    <name>br0</name>
    <bootproto>static</bootproto>
    <ipaddr>192.168.1.100</ipaddr>
    <prefixlen>24</prefixlen>
    <startmode>auto</startmode>
    <aliases>
      <alias0>
        <IPADDR>192.168.1.101</IPADDR>
        <PREFIXLEN>24</PREFIXLEN>
        <LABEL>http</LABEL>
      </alias0>
      <alias1>
        <IPADDR>192.168.2.100</IPADDR>
        <PREFIXLEN>24</PREFIXLEN>
        <LABEL>extra</LABEL>
      </alias1>
    </aliases>
  </interface>
</interfaces>
```

4.13.5 网络接口的持久名称

net - udev 元素允许指定一组可用于向接口指派持久名称的 udev 规则。

name

网络接口名称，例如 eth3。（必需。）

rule

ATTR{address} 用于基于 MAC 的规则，KERNELS 用于基于总线 ID 的规则。（必需。）

value

例如：f0:de:f1:6b:da:69 用于 MAC 规则，0000:00:1c.1 or 0.0.0700 用于总线 ID 规则。（必需。）



提示：处理设备名称冲突

创建不完整的 **udev** 规则集时，所选设备名称可能与现有的设备名称冲突。例如，将网络接口重命名为 eth0 时，可能会与内核自动生成的设备冲突。AutoYaST 会尽力处理这种情况并重命名有冲突的设备。

例 4.42：使用 MAC 地址指派持久名称

```
<net-udev config:type="list">
  <rule>
    <name>eth1</name>
    <rule>ATTR{address}</rule>
    <value>52:54:00:68:54:fb</value>
  </rule>
</net-udev>
```

4.13.6 域名系统

dns 部分用于定义名称服务相关的设置，例如主机名或名称服务器。

主机名

主机名，不包括域名部分。例如：指定 foo 而不是 foo.bar。Linux 内核允许您使用完全完全限定的域名 (FQDN) 取代主机名，YaST 也同样如此。但在 YaST 的 dns 部分中，这并不是正确的用法。解析程序应该会确定 FQDN。（有关 FQDN 解析方式的信息，请参见 man 1 hostname 的“THE FQDN”部分。）

如果未指定主机名且 DHCP 服务器未指派主机名（请参见 [dhcp_hostname](#)），AutoYaST 将生成 [install](#) 作为主机名。

nameservers

名称服务器的列表。示例：

```
<nameservers config:type="list">
  <nameserver>192.168.1.116</nameserver>
  <nameserver>192.168.1.117</nameserver>
</nameservers>
```

searchlist

用于查找主机名的搜索列表。

```
<searchlist config:type="list">
  <search>example.com</search>
</searchlist>
```

可选。

dhcp_hostname

指定是否必须从 DHCP 获取主机名。

```
<dhcp_hostname config:type="boolean">true</dhcp_hostname>
```

4.13.7 路由选择

[routing](#) 表允许指定路由列表，以及 IPv4 和 IPv6 的数据包转发设置。

ipv4_forward

可选：是否必须为 IPv4 启用 IP 转发。

ipv6_forward

可选：是否必须为 IPv6 启用 IP 转发。

routes

可选：路由列表。

下面的设置介绍了定义路由的方式。

destination

必需：路由目标。可以指定地址前缀，例如：`192.168.122.0/24`。

可以使用标头 `default` 来指示路由是与网关位于相同地址族（IPv4 或 IPv6）中的默认网关。

device

必需：与路由关联的接口。

gateway

可选：网关的 IP 地址。

netmask

(不推荐.)目标的网络掩码。

最好是指定前缀作为 `destination` 值的一部分。

extrapara

可选：其他路由选项，例如 `metric`、`mtu` 或 `table`。

例 4.43：网络路由配置

```
<routing>
  <ipv4_forward config:type="boolean">true</ipv4_forward>
  <ipv6_forward config:type="boolean">true</ipv6_forward>
  <routes config:type="list">
    <route>
      <destination>192.168.100.0/24</destination>
      <device>eth1</device>
      <extrapara>scope link src 192.168.100.100 table one</extrapara>
    </route>
    <route>
      <destination>default</destination>
      <device>eth1</device>
      <gateway>192.168.100.1</gateway>
    </route>
    <route>
      <destination>default</destination>
      <device>lo</device>
```



```
<gateway>192.168.5.1</gateway>
</route>
</routes>
</routing>
```

4.13.8 s390 选项

以下元素必须包含在 `<s390-devices>...</s390-devices>` 标记之间。

type

qeth, ctc 或 iucv.

chanids

通道 ID，以冒号（首选）或空格分隔

```
<chanids>0.0.0700:0.0.0701:0.0.0702</chanids>
```

layer2

```
<layer2 config:type="boolean">true</layer2>
```

布尔值；默认值：false

portname

QETH 端口名（从 SLE 12 SP2 开始已弃用）

protocol

可选：CTC/LCS 协议，一个较小的数字（字符串形式）

```
<protocol>1</protocol>
```

路由器

IUCV 路由器/用户

除上述选项外，AutoYaST 还支持在配置文件的其他部分中使用 IBM Z 特定的选项。具体而言，您可以定义逻辑链接地址，即 LLADDR（在以太网中为 MAC 地址）。为此，请在设备定义中使用选项 LLADDR。



提示：VLAN 的 LLADDR

VLAN 设备从底层物理设备继承其 LLADDR。要为 VLAN 设备设置特定的地址，请为底层物理设备设置 LLADDR 选项。

4.14 代理

配置互联网代理（缓存）设置。

使用 `http_proxy`、`https_proxy` 和 `ftp_proxy` 分别配置 HTTP、HTTPS 和 FTP 的代理。需使用 `no_proxy` 指定应可直接访问的地址或名称（空格分隔值）。如果您使用需要授权的代理服务器，请填写 `proxy_user` 和 `proxy_password`。

例 4.44：网络配置：代理

```
<proxy>
  <enabled config:type="boolean">true</enabled>
  <ftp_proxy>http://192.168.1.240:3128</ftp_proxy>
  <http_proxy>http://192.168.1.240:3128</http_proxy>
  <no_proxy>www.example.com .example.org localhost</no_proxy>
  <proxy_password>testpw</proxy_password>
  <proxy_user>testuser</proxy_user>
</proxy>
```



注意

如果强制将网络配置写在建议前面或通过 `linuxrc` 提供代理设置，将会在安装期间写入代理设置。

4.15 NIS 客户端和服务端

使用 `nis` 资源可将目标计算机配置为 NIS 客户端。以下示例演示了使用多个域的详细配置。

例 4.45：网络配置：NIS

```
<nis>
```

```

<nis_broadcast config:type="boolean">true</nis_broadcast>
<nis_broken_server config:type="boolean">true</nis_broken_server>
<nis_domain>test.com</nis_domain>
<nis_local_only config:type="boolean">true</nis_local_only>
<nis_options></nis_options>
<nis_other_domains config:type="list">
  <nis_other_domain>
    <nis_broadcast config:type="boolean">false</nis_broadcast>
    <nis_domain>domain.com</nis_domain>
    <nis_servers config:type="list">
      <nis_server>10.10.0.1</nis_server>
    </nis_servers>
  </nis_other_domain>
</nis_other_domains>
<nis_servers config:type="list">
  <nis_server>192.168.1.1</nis_server>
</nis_servers>
<start_autofs config:type="boolean">true</start_autofs>
<start_nis config:type="boolean">true</start_nis>
</nis>

```

4.16 NIS 服务器

可将目标计算机配置为 NIS 服务器。可以使用 NIS 主服务器和 NIS 工作器服务器以及两者的组合。

例 4.46：NIS 服务器配置

```

<nis_server>
  <domain>mydomain.de</domain>
  <maps_to_serve config:type="list">
    <nis_map>auto.master</nis_map>
    <nis_map>ethers</nis_map>
  </maps_to_serve>
  <merge_passwd config:type="boolean">false</merge_passwd>
  <mingid config:type="integer">0</mingid>
  <minuid config:type="integer">0</minuid>

```

```
<nopush config:type="boolean">false</nopush>
<pwd_chfn config:type="boolean">false</pwd_chfn>
<pwd_chsh config:type="boolean">false</pwd_chsh>
<pwd_srcdir>/etc</pwd_srcdir>
<securenets config:type="list">
  <securenet>
    <netmask>255.0.0.0</netmask>
    <network>127.0.0.0</network>
  </securenet>
</securenets>
<server_type>master</server_type>
<slaves config:type="list"/>
<start_ypbind config:type="boolean">false</start_ypbind>
<start_yppasswdd config:type="boolean">false</start_yppasswdd>
<start_ypxfrd config:type="boolean">false</start_ypxfrd>
</nis_server>
```

属性、值、描述

domain

NIS 域名。

maps_to_serve

服务器可用的映射列表。

值：

auto.master、ethers、group、hosts、netgrp、networks、passwd、protocols、rpc、services、shadow

merge_passwd

选择是否应将您的 passwd 文件与 shadow 文件合并（仅当 shadow 文件存在时才可以这样做）。

值：true/false

mingid

要包含在用户映射中的最小 GID。

minuid

要包含在用户映射中的最小 UID。

nopush

不要将更改推送到工作器服务器。（未发生任何更改时非常有用）。

值：true/false

pwd_chfn

YPPWD_CHFN - 允许更改全名

值：true/false

pwd_chsh

YPPWD_CHSH - 允许更改登录外壳

值：true/false

pwd_srcdir

YPPWD_SRCDIR - passwd 数据的源目录

默认值为: /etc

securenets

允许其查询 NIS 服务器的主机列表

当网络等于主机地址和网络掩码的按位 AND 结果时，允许使用主机地址。

包含网络掩码 255.0.0.0 和网络 127.0.0.0 的项必须存在，才允许从本地主机进行连接。

输入网络掩码 0.0.0.0 和网络 0.0.0.0 会授予对所有主机的访问权限。

server_type

选择是将 NIS 服务器配置为主服务器或工作器，还是不配置 NIS 服务器。

值：master、slave、none

slaves

要配置为 NIS 服务器工作器的主机名列表。

start_ybind

此主机也是 NIS 客户端（仅适用于在本地配置客户端的情况）。

值：true/false

start_yppasswdd

同时启动口令守护程序。

值：true/false

start_ypxfrd

同时启动映射传送守护程序。快速映射分发；将加快向工作器传送映射的过程。

值：true/false

4.17 主机定义

使用 `host` 资源可将更多项添加到 `/etc/hosts` 文件。这不会删除现有的项。有关细节，请参见以下示例。

例 4.47：/ETC/HOSTS

```
<host>
  <hosts config:type="list">
    <hosts_entry>
      <host_address>133.3.0.1</host_address>
      <names config:type="list">
        <name>booking</name>
      </names>
    </hosts_entry>
    <hosts_entry>
      <host_address>133.3.0.5</host_address>
      <names config:type="list">
        <name>test-machine</name>
      </names>
    </hosts_entry>
  </hosts>
</host>
```

4.18 Windows 域成员资格

使用 `samba-client` 资源可以配置工作组、NT 域或 Active Directory 域的成员资格。

例 4.48：SAMBA 客户端配置

```
<samba-client>
  <disable_dhcp_hostname config:type="boolean">true</disable_dhcp_hostname>
```

```
<global>
  <security>domain</security>
  <usershare_allow_guests>No</usershare_allow_guests>
  <usershare_max_shares>100</usershare_max_shares>
  <workgroup>WORKGROUP</workgroup>
</global>
<winbind config:type="boolean">false</winbind>
</samba-client>
```

属性、值、描述

disable_dhcp_hostname

不允许 DHCP 更改主机名。

值: true/false

global/security

身份验证体系的类型（域技术或 Active Directory 服务器 (ADS)）。

值: ADS/domain

global/usershare_allow_guests

允许共享 Guest 访问权限。

值: No/Yes

global/usershare_max_shares

smb.conf 中的共享数上限。

0 表示未启用共享。

global/workgroup

工作组名或域名。

winbind

使用 winbind。

值: true/false

4.19 Samba 服务器

简单 Samba 服务器的配置。

例 4.49：Samba 服务器配置

```
<samba-server>
  <accounts config:type="list"/>
  <backend/>
  <config config:type="list">
    <listentry>
      <name>global</name>
      <parameters>
        <security>domain</security>
        <usershare_allow_guests>No</usershare_allow_guests>
        <usershare_max_shares>100</usershare_max_shares>
        <workgroup>WORKGROUP</workgroup>
      </parameters>
    </listentry>
  </config>
  <service>Disabled</service>
  <trustdom/>
  <version>2.11</version>
</samba-server>
```

属性、值、描述

accounts

Samba 帐户的列表。

backend

可用后端的列表。

值：true/false。

config

在 /etc/samba/smb.conf 中设置其他用户定义参数。

该示例显示了 /etc/samba/smb.conf 的 global 部分中的参数。

service

在引导期间启动 Samba 服务。

值：Enabled/Disabled。

trustdom/

可信域。

两个映射的映射（键：establish，revoke）。每个映射均包含格式为 key: domainname value: password 的项。

version

Samba 版本。

默认值：2.11.

4.20 身份验证客户端

配置文件必须采用 JSON 格式。确认 autoyast2 和 autoyast2-installation 均已安装。使用 YaST 中的自动安装配置模块生成有效的 JSON 配置文件。启动 YaST 并切换到杂项 > 自动安装配置。选择网络服务 > 用户登录管理，单击编辑，然后配置可用设置。完成后单击确定。要保存生成的配置文件，请使用文件 > 保存。



提示：使用 ldaps://

要结和本机 SSL（而不是 TLS）使用 LDAP，请添加 ldaps 资源。

4.21 NFS 客户端和服务

可以使用配置系统将某个系统配置为 NFS 客户端或 NFS 服务器。以下示例演示如何配置 NFS 客户端和服务。

从 SUSE Linux Enterprise Server 15 SP7 开始，NFS 客户端配置的结构发生了变化。引入了一些全局配置选项：enable_nfs4 用于打开/关闭 NFS4 支持，idmapd_domain 用于定义 rpc.idmapd 域名（仅当已启用 NFS4 时才有效）。注意：旧结构与新结构不兼容，因此在较旧版本中创建的包含 NFS 部分的控制文件在较新产品中无效。

有关如何在 NFS 共享上安装 SUSE Linux Enterprise Server 的详细信息，请参见第 4.6.10 节“NFS 配置”。

例 4.50：网络配置：NFS 客户端

```
<nfs>
```

```

<enable_nfs4 config:type="boolean">true</enable_nfs4>
<idmapd_domain>suse.cz</idmapd_domain>
<nfs_entries config:type="list">
  <nfs_entry>
    <mount_point>/home</mount_point>
    <nfs_options>sec=krb5i,intr,rw</nfs_options>
    <server_path>saurus.suse.cz:/home</server_path>
    <vfstype>nfs4</vfstype>
  </nfs_entry>
  <nfs_entry>
    <mount_point>/work</mount_point>
    <nfs_options>defaults</nfs_options>
    <server_path>bivoj.suse.cz:/work</server_path>
    <vfstype>nfs</vfstype>
  </nfs_entry>
  <nfs_entry>
    <mount_point>/mnt</mount_point>
    <nfs_options>defaults</nfs_options>
    <server_path>fallback.suse.cz:/srv/dist</server_path>
    <vfstype>nfs</vfstype>
  </nfs_entry>
</nfs_entries>
</nfs>

```

例 4.51：网络配置：NFS 服务器

```

<nfs_server>
  <nfs_exports config:type="list">
    <nfs_export>
      <allowed config:type="list">
        <allowed_clients>*(ro,root_squash,sync)</allowed_clients>
      </allowed>
      <mountpoint>/home</mountpoint>
    </nfs_export>
    <nfs_export>
      <allowed config:type="list">
        <allowed_clients>*(ro,root_squash,sync)</allowed_clients>
      </allowed>
      <mountpoint>/work</mountpoint>
    </nfs_export>
  </nfs_exports>
</nfs_server>

```

```
</nfs_export>
</nfs_exports>
<start_nfsserver config:type="boolean">true</start_nfsserver>
</nfs_server>
```

4.22 NTP 客户端

！ 重要：NTP 客户端配置文件不兼容

从 SUSE Linux Enterprise Server 15 开始，NTP 客户端配置文件采用了新格式，因此与先前的配置文件**不兼容**。您需要更新在旧版 SUSE Linux Enterprise Server 中使用的 NTP 客户端配置文件，才能与版本 15 和更新版本兼容。

下面是 NTP 客户端配置示例：

例 4.52：网络配置：NTP 客户端

```
<ntp-client>
  <ntp_policy>auto</ntp_policy> ❶
  <ntp_servers config:type="list">
    <ntp_server>
      <address>cz.pool.ntp.org</address> ❷
      <iburst config:type="boolean">false</iburst> ❸
      <offline config:type="boolean">false</offline> ❹
    </ntp_server>
  </ntp_servers>
  <ntp_sync>15</ntp_sync> ❺
</ntp-client>
```

- ❶ `ntp_policy` 采用的值与 `/etc/sysconfig/network/config` 中的 `NETCONFIG_NTP_POLICY` 选项相同。最常用的选项是“static”和“auto”（默认值）。有关详细信息，请参见 `man 8 netconfig`。
- ❷ 时间服务器或时间服务器池的 URL。
- ❸ 启动 `iburst` 后，`chronyd` 会加速特定时间源的初始时间同步。

- ④ 将 `offline` 选项设置为 `true` 时，如果在启动 `chronyd` 时时间服务器不可用，此选项可防止客户端轮询时间服务器。在使用 **chronyc online** 手动启动之前，不会恢复轮询。重引导后此命令将不起作用。将此选项设置为 `false` 可确保客户端始终尝试联系时间服务器，且无需管理员的干预。
- ⑤ 对于 `ntp_sync`，如果正在运行 NTP 守护程序，请输入 “systemd”（默认值）；要通过 `cron` 进行同步，请输入**整数**间隔（以秒为单位）；如果不想自动同步，请输入 “manual”。

以下示例演示了一个 IPv6 配置。可以使用服务器的 IP 地址和/或主机名：

```
<ntp-server>
  <address>2001:418:3ff::1:53</address>
</ntp-server>

<ntp-server>
  <address>2.pool.ntp.org</address>
</ntp-server>
```

4.23 邮件服务器配置

对于客户端的邮件配置，此模块可让您创建详细的邮件配置。该模块包含各种选项。我们建议至少应在进行初始配置时使用该模块。

例 4.53：邮件配置

```
<mail>
  <aliases config:type="list">
    <alias>
      <alias>root</alias>
      <comment></comment>
      <destinations>foo</destinations>
    </alias>
    <alias>
      <alias>test</alias>
      <comment></comment>
      <destinations>foo</destinations>
    </alias>
```

```

</aliases>
<connection_type config:type="symbol">permanent</connection_type>
<fetchmail config:type="list">
  <fetchmail_entry>
    <local_user>foo</local_user>
    <password>bar</password>
    <protocol>POP3</protocol>
    <remote_user>foo</remote_user>
    <server>pop.foo.com</server>
  </fetchmail_entry>
  <fetchmail_entry>
    <local_user>test</local_user>
    <password>bar</password>
    <protocol>IMAP</protocol>
    <remote_user>test</remote_user>
    <server>blah.com</server>
  </fetchmail_entry>
</fetchmail>
<from_header>test.com</from_header>
<listen_remote config:type="boolean">true</listen_remote>
<local_domains config:type="list">
  <domains>test1.com</domains>
</local_domains>
<masquerade_other_domains config:type="list">
  <domain>blah.com</domain>
</masquerade_other_domains>
<masquerade_users config:type="list">
  <masquerade_user>
    <address>joe@test.com</address>
    <comment></comment>
    <user>joeuser</user>
  </masquerade_user>
  <masquerade_user>
    <address>bar@test.com</address>
    <comment></comment>
    <user>foo</user>
  </masquerade_user>
</masquerade_users>

```

```

<mta config:type="symbol">postfix</mta>
<outgoing_mail_server>test.com</outgoing_mail_server>
<postfix_mda config:type="symbol">local</postfix_mda>
<smtp_auth config:type="list">
  <listentry>
    <password>bar</password>
    <server>test.com</server>
    <user>foo</user>
  </listentry>
</smtp_auth>
<use_amavis config:type="boolean">true</use_amavis>
<virtual_users config:type="list">
  <virtual_user>
    <alias>test.com</alias>
    <comment></comment>
    <destinations>foo.com</destinations>
  </virtual_user>
  <virtual_user>
    <alias>geek.com</alias>
    <comment></comment>
    <destinations>bar.com</destinations>
  </virtual_user>
</virtual_users>
</mail>

```

4.24 Apache HTTP 服务器配置

本节介绍 Apache HTTP 服务器的配置。

对于经验不足的用户，我们建议使用 HTTP server YaST 模块 配置 Apache 服务器。完成该操作后，请调用 AutoYaST configuration 模块，选择 HTTP server YaST 模块 并克隆 Apache 设置。可以通过 File 菜单导出这些设置。

例 4.54：HTTP 服务器配置

```

<http-server>
  <Listen config:type="list">
    <listentry>

```

```

    <ADDRESS/>
    <PORT>80</PORT>
  </listentry>
</Listen>
<hosts config:type="list">
  <hosts_entry>
    <KEY>main</KEY>
    <VALUE config:type="list">
      <listentry>
        <KEY>DocumentRoot</KEY>
        <OVERHEAD>
          #
          # Global configuration that will be applicable for all
          # virtual hosts, unless deleted here or overridden elsewhere.
          #
        </OVERHEAD>
        <VALUE>"/srv/www/htdocs"</VALUE>
      </listentry>
      <listentry>
        <KEY>_SECTION</KEY>
        <OVERHEAD>
          #
          # Configure the DocumentRoot
          #
        </OVERHEAD>
        <SECTIONNAME>Directory</SECTIONNAME>
        <SECTIONPARAM>"/srv/www/htdocs"</SECTIONPARAM>
        <VALUE config:type="list">
          <listentry>
            <KEY>Options</KEY>
            <OVERHEAD>
              # Possible values for the Options directive are "None", "All",
              # or any combination of:
              #   Indexes Includes FollowSymLinks SymLinksifOwnerMatch
              #   ExecCGI MultiViews
              #
              # Note that "MultiViews" must be named *explicitly*
              # --- "Options All"
            </OVERHEAD>
          </listentry>
        </VALUE>
      </listentry>
    </VALUE>
  </hosts_entry>
</hosts>

```

```

# does not give it to you.
#
# The Options directive is both complicated and important.
# Please see
# http://httpd.apache.org/docs/2.4/mod/core.html#options
# for more information.
</OVERHEAD>
<VALUE>None</VALUE>
</listentry>
<listentry>
  <KEY>AllowOverride</KEY>
  <OVERHEAD>
    # AllowOverride controls what directives may be placed in
    # .htaccess files. It can be "All", "None", or any combination
    # of the keywords:
    #   Options FileInfo AuthConfig Limit
  </OVERHEAD>
  <VALUE>None</VALUE>
</listentry>
<listentry>
  <KEY>_SECTION</KEY>
  <OVERHEAD>
    # Controls who can get stuff from this server.
  </OVERHEAD>
  <SECTIONNAME>IfModule</SECTIONNAME>
  <SECTIONPARAM>!mod_access_compat.c</SECTIONPARAM>
  <VALUE config:type="list">
    <listentry>
      <KEY>Require</KEY>
      <VALUE>all granted</VALUE>
    </listentry>
  </VALUE>
</listentry>
<listentry>
  <KEY>_SECTION</KEY>
  <SECTIONNAME>IfModule</SECTIONNAME>
  <SECTIONPARAM>mod_access_compat.c</SECTIONPARAM>
  <VALUE config:type="list">

```



```

        <listentry>
            <KEY>Order</KEY>
            <VALUE>allow,deny</VALUE>
        </listentry>
        <listentry>
            <KEY>Allow</KEY>
            <VALUE>from all</VALUE>
        </listentry>
    </VALUE>
</listentry>
</VALUE>
</listentry>
<listentry>
    <KEY>Alias</KEY>
    <OVERHEAD>
        # Aliases: aliases can be added as needed (with no limit).
        # The format is Alias fakename realname
        #
        # Note that if you include a trailing / on fakename then the
        # server will require it to be present in the URL. So "/icons"
        # is not aliased in this example, only "/icons/". If the fakename
        # is slash-terminated, then the realname must also be slash
        # terminated, and if the fakename omits the trailing slash, the
        # realname must also omit it.
        # We include the /icons/ alias for FancyIndexed directory listings.
        # If you do not use FancyIndexing, you may comment this out.
        #
    </OVERHEAD>
    <VALUE>/icons/ "/usr/share/apache2/icons/"</VALUE>
</listentry>
<listentry>
    <KEY>_SECTION</KEY>
    <OVERHEAD>
    </OVERHEAD>
    <SECTIONNAME>Directory</SECTIONNAME>
    <SECTIONPARAM>"/usr/share/apache2/icons"</SECTIONPARAM>
    <VALUE config:type="list">
        <listentry>

```

```

        <KEY>Options</KEY>
        <VALUE>Indexes MultiViews</VALUE>
    </listentry>
    <listentry>
        <KEY>AllowOverride</KEY>
        <VALUE>None</VALUE>
    </listentry>
    <listentry>
        <KEY>_SECTION</KEY>
        <SECTIONNAME>IfModule</SECTIONNAME>
        <SECTIONPARAM>!mod_access_compat.c</SECTIONPARAM>
        <VALUE config:type="list">
            <listentry>
                <KEY>Require</KEY>
                <VALUE>all granted</VALUE>
            </listentry>
        </VALUE>
    </listentry>
    <listentry>
        <KEY>_SECTION</KEY>
        <SECTIONNAME>IfModule</SECTIONNAME>
        <SECTIONPARAM>mod_access_compat.c</SECTIONPARAM>
        <VALUE config:type="list">
            <listentry>
                <KEY>Order</KEY>
                <VALUE>allow,deny</VALUE>
            </listentry>
            <listentry>
                <KEY>Allow</KEY>
                <VALUE>from all</VALUE>
            </listentry>
        </VALUE>
    </listentry>
</VALUE>
</listentry>
<listentry>
    <KEY>ScriptAlias</KEY>
    <OVERHEAD>

```

```

# ScriptAlias: This controls which directories contain server
# scripts. ScriptAliases are essentially the same as Aliases,
# except that documents in the realname directory are treated
# as applications and run by the server when requested rather
# than as documents sent to the client.
# The same rules about trailing "/" apply to ScriptAlias
# directives as to Alias.
#
</OVERHEAD>
<VALUE>/cgi-bin/ "/srv/www/cgi-bin/"</VALUE>
</listentry>
<listentry>
  <KEY>_SECTION</KEY>
  <OVERHEAD>
  # "/srv/www/cgi-bin" should be changed to wherever your
  # ScriptAliased CGI directory exists, if you have that configured.
  #
  </OVERHEAD>
  <SECTIONNAME>Directory</SECTIONNAME>
  <SECTIONPARAM>"/srv/www/cgi-bin"</SECTIONPARAM>
  <VALUE config:type="list">
    <listentry>
      <KEY>AllowOverride</KEY>
      <VALUE>None</VALUE>
    </listentry>
    <listentry>
      <KEY>Options</KEY>
      <VALUE>+ExecCGI -Includes</VALUE>
    </listentry>
    <listentry>
      <KEY>_SECTION</KEY>
      <SECTIONNAME>IfModule</SECTIONNAME>
      <SECTIONPARAM>!mod_access_compat.c</SECTIONPARAM>
      <VALUE config:type="list">
        <listentry>
          <KEY>Require</KEY>
          <VALUE>all granted</VALUE>
        </listentry>

```

```

        </VALUE>
    </listentry>
    <listentry>
        <KEY>_SECTION</KEY>
        <SECTIONNAME>IfModule</SECTIONNAME>
        <SECTIONPARAM>mod_access_compat.c</SECTIONPARAM>
        <VALUE config:type="list">
            <listentry>
                <KEY>Order</KEY>
                <VALUE>allow,deny</VALUE>
            </listentry>
            <listentry>
                <KEY>Allow</KEY>
                <VALUE>from all</VALUE>
            </listentry>
        </VALUE>
    </listentry>
</VALUE>
</listentry>
<listentry>
    <KEY>_SECTION</KEY>
    <OVERHEAD>
    # UserDir: The name of the directory that is appended onto a
    # user's home directory if a ~user request is received.
    # To disable it, simply remove userdir from the list of modules
    # in APACHE_MODULES in /etc/sysconfig/apache2.
    #
    </OVERHEAD>
    <SECTIONNAME>IfModule</SECTIONNAME>
    <SECTIONPARAM>mod_userdir.c</SECTIONPARAM>
    <VALUE config:type="list">
        <listentry>
            <KEY>UserDir</KEY>
            <OVERHEAD>
            # Note that the name of the user directory ("public_html")
            # cannot simply be changed here, since it is a compile time
            # setting. The apache package would need to be rebuilt.
            # You could work around by deleting /usr/sbin/suexec, but

```

```

        # then all scripts from the directories would be executed
        # with the UID of the webserver.
    </OVERHEAD>
    <VALUE>public_html</VALUE>
</listentry>
<listentry>
    <KEY>Include</KEY>
    <OVERHEAD>
        # The actual configuration of the directory is in
        # /etc/apache2/mod_userdir.conf.
    </OVERHEAD>
    <VALUE>/etc/apache2/mod_userdir.conf</VALUE>
</listentry>
</VALUE>
</listentry>
<listentry>
    <KEY>IncludeOptional</KEY>
    <OVERHEAD>
        # Include all *.conf files from /etc/apache2/conf.d/.
        #
        # This is mostly meant as a place for other RPM packages to drop
        # in their configuration snippet.
        #
        #
        # You can comment this out here if you want those bits include
        # only in a certain virtual host, but not here.
    </OVERHEAD>
    <VALUE>/etc/apache2/conf.d/*.conf</VALUE>
</listentry>
<listentry>
    <KEY>IncludeOptional</KEY>
    <OVERHEAD>
        # The manual... if it is installed ('?' means it will not complain)
    </OVERHEAD>
    <VALUE>/etc/apache2/conf.d/apache2-manual?conf</VALUE>
</listentry>
<listentry>
    <KEY>ServerName</KEY>

```

```

        <VALUE>linux-wtyj</VALUE>
    </listentry>
    <listentry>
        <KEY>ServerAdmin</KEY>
        <OVERHEAD>
    </OVERHEAD>
        <VALUE>root@linux-wtyj</VALUE>
    </listentry>
    <listentry>
        <KEY>NameVirtualHost</KEY>
        <VALUE>192.168.43.2</VALUE>
    </listentry>
</VALUE>
</hosts_entry>
<hosts_entry>
    <KEY>192.168.43.2/secondserver.suse.de</KEY>
    <VALUE config:type="list">
        <listentry>
            <KEY>DocumentRoot</KEY>
            <VALUE>/srv/www/htdocs</VALUE>
        </listentry>
        <listentry>
            <KEY>ServerName</KEY>
            <VALUE>secondserver.suse.de</VALUE>
        </listentry>
        <listentry>
            <KEY>ServerAdmin</KEY>
            <VALUE>second_server@suse.de</VALUE>
        </listentry>
        <listentry>
            <KEY>_SECTION</KEY>
            <SECTIONNAME>Directory</SECTIONNAME>
            <SECTIONPARAM>/srv/www/htdocs</SECTIONPARAM>
            <VALUE config:type="list">
                <listentry>
                    <KEY>AllowOverride</KEY>
                    <VALUE>None</VALUE>
                </listentry>
            </listentry>
        </listentry>
    </VALUE>
</hosts_entry>

```

```

        <listentry>
            <KEY>Require</KEY>
            <VALUE>all granted</VALUE>
        </listentry>
    </VALUE>
</listentry>
</VALUE>
</hosts_entry>
</hosts>
<modules config:type="list">
    <module_entry>
        <change>enable</change>
        <name>socache_shmcb</name>
        <userdefined config:type="boolean">true</userdefined>
    </module_entry>
    <module_entry>
        <change>enable</change>
        <name>reqtimeout</name>
        <userdefined config:type="boolean">true</userdefined>
    </module_entry>
    <module_entry>
        <change>enable</change>
        <name>authn_core</name>
        <userdefined config:type="boolean">true</userdefined>
    </module_entry>
    <module_entry>
        <change>enable</change>
        <name>authz_core</name>
        <userdefined config:type="boolean">true</userdefined>
    </module_entry>
</modules>
<service config:type="boolean">true</service>
<version>2.9</version>
</http-server>

```

列表名称、列表元素、描述

Listen

主机 Listen 设置列表

PORT

端口地址

ADDRESS

网络地址。如果此项为空，将采用所有地址。

hosts

主机配置列表

KEY

主机名；<KEY>main</KEY> 定义主要主机，例如 <KEY>192.168.43.2/
secondserver.suse.de</KEY>

VALUE

用于描述主机的不同值的列表。

modules

模块列表。只需描述用户定义的模块。

name

模块名称

userdefined

出于历史原因，此项始终设置为 true。

change

出于历史原因，此项始终设置为 enable。

元素、描述、注释

version

使用的 Apache 服务器版本

仅供参考。默认值 2.9

service

启用 Apache 服务

可选。默认值：false


```

    </options>
    <type>proto</type>
</listentry>
<listentry>
    <name>localhost</name>
    <options config:type="list">
        <option>127.0.0.1/255.255.255.255</option>
    </options>
    <type>src</type>
</listentry>
<listentry>
    <name>to_localhost</name>
    <options config:type="list">
        <option>127.0.0.0/8</option>
    </options>
    <type>dst</type>
</listentry>
<listentry>
    <name>SSL_ports</name>
    <options config:type="list">
        <option>443</option>
    </options>
    <type>port</type>
</listentry>
<listentry>
    <name>Safe_ports</name>
    <options config:type="list">
        <option>80</option>
    </options>
    <type>port</type>
</listentry>
<listentry>
    <name>Safe_ports</name>
    <options config:type="list">
        <option>21</option>
    </options>
    <type>port</type>
</listentry>

```

```

<listentry>
  <name>Safe_ports</name>
  <options config:type="list">
    <option>443</option>
  </options>
  <type>port</type>
</listentry>
<listentry>
  <name>Safe_ports</name>
  <options config:type="list">
    <option>70</option>
  </options>
  <type>port</type>
</listentry>
<listentry>
  <name>Safe_ports</name>
  <options config:type="list">
    <option>210</option>
  </options>
  <type>port</type>
</listentry>
<listentry>
  <name>Safe_ports</name>
  <options config:type="list">
    <option>1025-65535</option>
  </options>
  <type>port</type>
</listentry>
<listentry>
  <name>Safe_ports</name>
  <options config:type="list">
    <option>280</option>
  </options>
  <type>port</type>
</listentry>
<listentry>
  <name>Safe_ports</name>
  <options config:type="list">

```

```

        <option>488</option>
    </options>
    <type>port</type>
</listentry>
<listentry>
    <name>Safe_ports</name>
    <options config:type="list">
        <option>591</option>
    </options>
    <type>port</type>
</listentry>
<listentry>
    <name>Safe_ports</name>
    <options config:type="list">
        <option>777</option>
    </options>
    <type>port</type>
</listentry>
<listentry>
    <name>CONNECT</name>
    <options config:type="list">
        <option>CONNECT</option>
    </options>
    <type>method</type>
</listentry>
</acls>
<http_accesses config:type="list">
    <listentry>
        <acl config:type="list">
            <listentry>manager</listentry>
            <listentry>localhost</listentry>
        </acl>
        <allow config:type="boolean">true</allow>
    </listentry>
    <listentry>
        <acl config:type="list">
            <listentry>manager</listentry>
        </acl>
    </listentry>

```

```

    <allow config:type="boolean">false</allow>
</listentry>
<listentry>
    <acl config:type="list">
        <listentry>!Safe_ports</listentry>
    </acl>
    <allow config:type="boolean">false</allow>
</listentry>
<listentry>
    <acl config:type="list">
        <listentry>CONNECT</listentry>
        <listentry>!SSL_ports</listentry>
    </acl>
    <allow config:type="boolean">false</allow>
</listentry>
<listentry>
    <acl config:type="list">
        <listentry>localhost</listentry>
    </acl>
    <allow config:type="boolean">true</allow>
</listentry>
<listentry>
    <acl config:type="list">
        <listentry>all</listentry>
    </acl>
    <allow config:type="boolean">false</allow>
</listentry>
</http_accesses>
<http_ports config:type="list">
    <listentry>
        <host/>
        <port>3128</port>
        <transparent config:type="boolean">false</transparent>
    </listentry>
</http_ports>
<refresh_patterns config:type="list">
    <listentry>
        <case_sensitive config:type="boolean">true</case_sensitive>

```

```

    <max>10080</max>
    <min>1440</min>
    <percent>20</percent>
    <regexp>^ftp:</regexp>
</listentry>
<listentry>
    <case_sensitive config:type="boolean">true</case_sensitive>
    <max>1440</max>
    <min>1440</min>
    <percent>0</percent>
    <regexp>^gopher:</regexp>
</listentry>
<listentry>
    <case_sensitive config:type="boolean">true</case_sensitive>
    <max>4320</max>
    <min>0</min>
    <percent>20</percent>
    <regexp>.</regexp>
</listentry>
</refresh_patterns>
<service_enabled_on_startup config:type="boolean">true</
service_enabled_on_startup>
<settings>
    <access_log config:type="list">
        <listentry>/var/log/squid/access.log</listentry>
    </access_log>
    <cache_dir config:type="list">
        <listentry>ufs</listentry>
        <listentry>/var/cache/squid</listentry>
        <listentry>100</listentry>
        <listentry>16</listentry>
        <listentry>256</listentry>
    </cache_dir>
    <cache_log config:type="list">
        <listentry>/var/log/squid/cache.log</listentry>
    </cache_log>
    <cache_mem config:type="list">
        <listentry>8</listentry>

```

```

    <listentry>MB</listentry>
</cache_mem>
<cache_mgr config:type="list">
    <listentry>webmaster</listentry>
</cache_mgr>
<cache_replacement_policy config:type="list">
    <listentry>lru</listentry>
</cache_replacement_policy>
<cache_store_log config:type="list">
    <listentry>/var/log/squid/store.log</listentry>
</cache_store_log>
<cache_swap_high config:type="list">
    <listentry>95</listentry>
</cache_swap_high>
<cache_swap_low config:type="list">
    <listentry>90</listentry>
</cache_swap_low>
<client_lifetime config:type="list">
    <listentry>1</listentry>
    <listentry>days</listentry>
</client_lifetime>
<connect_timeout config:type="list">
    <listentry>2</listentry>
    <listentry>minutes</listentry>
</connect_timeout>
<emulate_httpd_log config:type="list">
    <listentry>off</listentry>
</emulate_httpd_log>
<error_directory config:type="list">
    <listentry/>
</error_directory>
<ftp_passive config:type="list">
    <listentry>on</listentry>
</ftp_passive>
<maximum_object_size config:type="list">
    <listentry>4096</listentry>
    <listentry>KB</listentry>
</maximum_object_size>

```

```
<memory_replacement_policy config:type="list">
  <listentry>lru</listentry>
</memory_replacement_policy>
<minimum_object_size config:type="list">
  <listentry>0</listentry>
  <listentry>KB</listentry>
</minimum_object_size>
</settings>
</squid>
```

属性、值、描述

acls

访问控制列表 (ACL) 设置。

每个列表项包含名称、类型和其他选项。使用 YaST Squid 配置模块可以获取可能项的概述。

http_accesses

在访问控制表中，可以拒绝或允许 ACL 组的访问。

如果一个定义中包含更多 ACL 组，则会同时允许或拒绝属于所有 ACL 组成员的访问。

将按此处的列出顺序检查访问控制表。使用第一个匹配项。

http_ports

定义 Squid 将用来侦听客户端 HTTP 请求的所有端口。

Host 可以包含主机名或 IP 地址，也可以留空。

transparent 禁用透明的 PMTU 发现。

refresh_patterns

刷新模式定义 Squid 如何处理缓存中的对象。

将按此处的列出顺序检查刷新模式。使用第一个匹配项。

Min 确定在未明确指定失效时间的情况下，应将已存在多少分钟内的对象视为新对

象。Max 用于设置在未明确指定失效时间的情况下，最多可以将已存在多少分钟内的对象视为新对象。Percent 是对象已存在时间（自上次修改后所经过的时间）的百分比。未指明明确失效时间的对象将被视为新对象。

settings

具有默认值的所有可用常规参数的映射。

使用 YaST Squid 配置模块可以获取有关可能项的概述。

service_enabled_on_startup

引导时启动 Squid 服务。

值：true/false

4.26 FTP 服务器

配置 FTP 互联网服务器设置

例 4.56：FTP 服务器配置：

```
<ftp-server>
  <AnonAuthen>2</AnonAuthen>
  <AnonCreatDirs>N0</AnonCreatDirs>
  <AnonMaxRate>0</AnonMaxRate>
  <AnonReadOnly>N0</AnonReadOnly>
  <AntiWareZ>YES</AntiWareZ>
  <Banner>Welcome message</Banner>
  <CertFile/>
  <ChrootEnable>N0</ChrootEnable>
  <EnableUpload>YES</EnableUpload>
  <FTPUser>ftp</FTPUser>
  <FtpDirAnon>/srv/ftp</FtpDirAnon>
  <FtpDirLocal/>
  <GuestUser/>
  <LocalMaxRate>0</LocalMaxRate>
  <MaxClientsNumber>10</MaxClientsNumber>
  <MaxClientsPerIP>3</MaxClientsPerIP>
  <MaxIdleTime>15</MaxIdleTime>
  <PasMaxPort>40500</PasMaxPort>
  <PasMinPort>40000</PasMinPort>
  <PassiveMode>YES</PassiveMode>
  <SSL>0</SSL>
  <SSLEnable>N0</SSLEnable>
  <SSLv2>N0</SSLv2>
  <SSLv3>N0</SSLv3>
  <StartDaemon>2</StartDaemon>
```

```
<TLS>YES</TLS>
<Umask/>
<UmaskAnon/>
<UmaskLocal/>
<VerboseLogging>NO</VerboseLogging>
<VirtualUser>NO</VirtualUser>
</ftp-server>
```

元素、描述、注释

AnonAuthen

启用/禁用匿名用户和本地用户。

仅限已经过身份验证的用户：1；仅限匿名用户：0；两者均可：2

AnonCreatDirs

匿名用户可以创建目录。

值：YES/NO

AnonReadOnly

匿名用户可以上载。

值：YES/NO

AnonMaxRate

允许匿名客户端使用的最大数据传输速率。

KB/秒

AntiWarez

禁止下载已上载但未经本地管理员验证的文件。

值：YES/NO

Banner

指定包含当某人连接到服务器时所显示文本的文件名。

CertFile

用于 SSL 加密连接的 DSA 证书

此选项指定用于 SSL 加密连接的 DSA 证书的位置。

ChrootEnable

如果启用，则本地用户登录后，默认会将他们放置到其主目录中的 chroot jail 内。

警告：此选项会影响安全性。值：YES/NO

EnableUpload

如果启用，则 FTP 用户可以上载。

要允许匿名用户上载，请启用 AnonReadOnly。值：YES/NO

FTPUser

定义匿名 FTP 用户。

FtpDirAnon

匿名用户的 FTP 目录。

指定匿名 FTP 用户使用的目录。

FtpDirLocal

经身份验证的用户的 FTP 目录。

指定经过身份验证的 FTP 用户使用的目录。

LocalMaxRate

指定允许经身份验证的本地用户使用的最大数据传输速率。

KB/秒

MaxClientsNumber

允许连接的最大客户端数。

MaxClientsPerIP

定义一个 IP 的最大客户端数。

此元素用于限制允许从单个源互联网地址连接的客户端数。

MaxIdleTime

两个 FTP 命令之间远程客户端的最大等待时间（超时）。

分钟

PasMaxPort

用于被动连接答复的端口范围最大值。

PassiveMode 需设置为 YES。

PasMinPort

用于被动连接答复的端口范围最小值。

PassiveMode 需设置为 YES。

PassiveMode

启用被动模式

值：YES/NO

SSL

安全性设置

禁用 TLS/SSL：0；接受 TLS/SSL：1；拒绝不是通过 TLS/SSL 建立的连接：2

SSLEnable

如果启用，则允许 SSL 连接。

值：YES/NO

SSLv2

如果启用，则允许 SSL 版本 2 连接。

值：YES/NO

SSLv3

如果启用，则允许 SSL 版本 3 连接。

值：YES/NO

StartDaemon

如何启动 FTP 守护程序。

手动:0；引导时：1；通过 systemd 套接字：2

TLS

如果启用，则允许 TLS 连接。

值：YES/NO

Umask

文件创建掩码，采用 “(文件的 umask):(目录的 umask)” 格式。

例如，如果您存有怀疑，请将其设置为 177:077。

UmaskAnon

为匿名用户设置的文件创建 umask 的值。

要指定八进制值，请记得添加 “0” 前缀，否则会将该值视为十进制整数。

UmaskLocal

经身份验证的用户的 Umask。

要指定八进制值，请记得添加“0”前缀，否则会将该值视为十进制整数。

VerboseLogging

如果启用，则记录所有 FTP 请求和响应。

值：YES/NO

VirtualUser

使用虚拟用户可以在不影响系统帐户的情况下管理 FTP 帐户。

值：YES/NO



注意：防火墙

需要指定正确的防火墙设置才能让 FTP 服务器正常运行。

4.27 TFTP 服务器

配置 TFTP 互联网服务器设置。

使用此选项可为服务器启用 TFTP（普通文件传送协议）。将使用 systemd 套接字启动该服务器。

注意：TFTP 和 FTP 不同。

例 4.57：TFTP 服务器配置：

```
<tftp-server>
  <start_tftpd config:type="boolean">true</start_tftpd>
  <tftp_directory>/tftpboot</tftp_directory>
</tftp-server>
```

start_tftpd

启用 TFTP 服务器服务。值：true/false。

tftp_directory

引导映像目录：指定提供的文件所在的目录。

值一般为 `/tftpbboot`。如果不存在，将创建该目录。服务器使用该目录作为其根目录（使用 `-s` 选项）。

4.28 首次引导工作流程

在完成安装后运行的 YaST 首次引导实用程序（YaST 初始系统配置）可让您配置全新安装的系统。在安装后首次进行引导时，该实用程序会指导用户完成一系列步骤，让他们能够更轻松地配置系统。YaST 首次引导默认不会运行，需要经过配置后才会运行。

例 4.58：启用首次引导工作流程

```
<firstboot>
  <firstboot_enabled config:type="boolean">true</firstboot_enabled>
</firstboot>
```

4.29 安全性设置

使用此模块的功能可以更改目标系统上的本地安全性设置。本地安全性设置包括引导配置、登录设置、口令设置、用户添加设置和文件权限。

安全性设置自动配置类似于在运行中的系统上可用的安全性模块中的 [Custom Settings](#)。此功能允许您创建自定义的配置。

例 4.59：安全配置

有关以下示例中的设置的含义和可能值，请参见参考信息。

```
<security>
  <console_shutdown>ignore</console_shutdown>
  <displaymanager_remote_access>no</displaymanager_remote_access>
  <fail_delay>3</fail_delay>
  <faillog_enab>yes</faillog_enab>
  <gid_max>60000</gid_max>
  <gid_min>101</gid_min>
  <gdm_shutdown>root</gdm_shutdown>
  <lastlog_enab>yes</lastlog_enab>
  <encryption>md5</encryption>
```

```
<obscure_checks_enab>no</obscure_checks_enab>
<pass_max_days>99999</pass_max_days>
<pass_max_len>8</pass_max_len>
<pass_min_days>1</pass_min_days>
<pass_min_len>6</pass_min_len>
<pass_warn_age>14</pass_warn_age>
<passwd_use_cracklib>yes</passwd_use_cracklib>
<permission_security>secure</permission_security>
<run_updatedb_as>nobody</run_updatedb_as>
<uid_max>60000</uid_max>
<uid_min>500</uid_min>
<selinux_mode>permissive</selinux_mode>
<lsm_select>selinux</lsm_select>
</security>
```

4.29.1 口令设置选项

使用 `<pass_*` 资源可更改各种口令设置，例如口令长度下限、口令失效时间等。

使用 `<encryption>` 资源可激活当前支持的其中一种加密方法。如果未设置，将配置 `sha512`。

您可以使用以下任一加密方法：

- `md5` - 允许使用包含 128 位哈希值的更长口令
- `sha256` 或 `sha512` - 广泛运用的安全哈希算法
- `des` - 由于安全性不足，我们不建议使用这种加密方法

4.29.2 引导设置

使用安全性资源可以更改各项引导设置。

如何解释 `Ctrl - Alt - Del` ？

当在控制台操作的某人按下组合键 `Ctrl - Alt - Del` 时，系统通常会重引导。有时需要忽略此事件，例如，当系统既是工作站又是服务器时。

GDM 的关机行为

配置允许其从 GDM 关闭计算机的用户列表。

4.29.3 登录设置

更改各项登录设置。这些设置主要储存在 /etc/login.defs 文件中。

4.29.4 新用户设置 (**useradd** 设置)

设置允许的最小和最大用户和组 ID。

4.29.5 Linux 安全模块 (LSM) 设置

在 SUSE Linux Enterprise 15 SP4 及更高版本中，安装控制文件中包含一个新选项 <lsm_select>，用于配置安装后默认激活以下哪个主要 Linux 安全模块 (LSM)：AppArmor、SELinux 或者都不激活。

selinux_mode

可选。配置 SELinux 模式。值：permissive、enforcing 和 disabled。

lsm_select

可选。安装期间要选择的主要 Linux 安全模块。值：selinux、apparmor 或 none。

4.29.6 使用 OpenSCAP 安全策略

YaST 允许使用 OpenSCAP 安全策略来强化系统。检查和应用安全策略分两个阶段进行：

- 在安装时，YaST 会检查一部分安全策略规则，尤其是在安装后难以修复的规则，例如用于加密文件系统的规则。如果配置文件中描述的系统不符合其中的任何一个规则，AutoYaST 将报告问题并中止安装。
- 此外，AutoYaST 还会安装并配置 **ssg-apply** 工具。在首次引导期间，可以运行 **ssg-apply** 以扫描系统，并选择性地修正系统以符合选定的策略。

！ 重要：在 SUSE Linux Enterprise 15 SP4 中的可用性

可以通过自我更新或 QU2 媒体在 SUSE Linux Enterprise 15 SP4 GM 中使用此功能。请确保在 `<suse_register>` 部分指定 `<install_updates t="boolean">true</install_updates>`，以便在安装期间启用更新（参见第 4.3 节“系统注册和扩展选择”）。

如果您在无法连接互联网的情况下安装，请将 QU2 媒体中的 `Basesystem` 模块添加到 `<add_on_products>` 部分：

```
<listentry t="map">
  <media_url>relurl://</media_url>
  <product>sle-module-basesystem</product>
  <product_dir>/Module-Basesystem</product_dir>
</listentry>
```

有关详细信息，请参见第 4.10.3 节“安装其他/自定义的软件包或产品”。

`security_policy` 部分会选择一个安全策略并配置 `ssg-apply`。

policy

选择要检查或应用的安全策略。目前，仅支持 国防信息系统局 安全技术实施指南 (DISA STIG)。使用名称 `stig` 来引用此策略。此元素是必需的。

action

指定 `ssg-apply` 在首次引导期间应执行的操作。

- `scan`：在首次引导期间扫描系统。这是默认行为。
- `remediate`：扫描并修正系统以符合选定的策略。
- `none`：在首次引导期间配置但不运行 `ssg-apply`。如果您想要在强化系统之前修改策略，可以使用此选项。

例 4.60：选择 国防信息系统局 安全技术实施指南

以下摘录指示 AutoYaST 检查 DISA STIG 策略，并在首次引导期间修正系统。

```
<security>
```

```
<security_policy>
  <policy>stig</policy>
  <action>remediate</action>
</security_policy>
</security>
```

4.30 Linux 审计框架 (LAF)

此模块允许配置审计守护程序以及为审计子系统添加规则。

例 4.61：LAF 配置

```
<audit-laf>
  <auditd>
    <flush>INCREMENTAL</flush>
    <freq>20</freq>
    <log_file>/var/log/audit/audit.log</log_file>
    <log_format>RAW</log_format>
    <max_log_file>5</max_log_file>
    <max_log_file_action>ROTATE</max_log_file_action>
    <name_format>NONE</name_format>
    <num_logs>4</num_logs>
  </auditd>
  <rules/>
</audit-laf>
```

属性、值、描述

auditd/flush

描述如何将数据写入磁盘。

如果设置为 INCREMENTAL，“频率”参数会告知要在写入多少条记录后向磁盘发出明确刷新命令。NONE 表示不采取特殊措施来刷新数据，DATA 表示将数据部分保持同步，SYNC 表示将数据和元数据完全保持同步。

auditd/freq

此参数告知要在写入多少条记录后向磁盘发出明确刷新命令。

参数 `flush` 需设置为 `INCREMENTAL`。

auditd/log_file

日志文件的完整路径名。

auditd/log_fomat

需要记录多少信息。

设置为 `RAW` 会记录所有数据（完全按照内核发送数据的格式储存数据），设置为 `NOLLOG` 会丢弃所有审计信息，而不是将其写入磁盘（不影响已发送到调度程序的数据）。

auditd/max_log_file

需要记录多少信息。

单位：MB

auditd/num_logs

日志文件数。

`max_log_file_action` 需设置为 `ROTATE`

auditd/max_log_file_action

达到日志容量时所执行的操作。

如果操作设置为 `ROTATE`，“日志文件数”将指定要保留的文件数。如果设置为 `SYSLOG`，审计守护程序会向系统日志写入警告。如果设置为 `SUSPEND`，守护程序将停止向磁盘写入记录。`IGNORE` 表示不执行任何操作，`KEEP_LOGS` 类似于 `ROTATE`，但不会覆盖日志文件。

auditd/name_format

“计算机名称格式”描述如何将计算机名称写入日志文件。

如果设置 `USER`，则会使用用户定义的名称。`NONE` 表示不插入计算机名称。`HOSTNAME` 使用“gethostname”系统调用返回的名称。`FQD` 使用完全限定的域名。

rules

`auditctl` 的规则

您可以手动编辑规则，但我们仅建议由高级用户执行这种操作。有关所有选项的详细信息，请参见 `man auditctl`。

4.31 用户和组

4.31.1 用户

可以在 `<users>` 部分定义用户列表。要能成功登录，请确保已设置 `root` 用户，或者已将 `rootpassword` 指定为 `linuxrc` 选项。

例 4.62：极简用户配置

```
<users config:type="list">
  <user>
    <username>root</username>
    <user_password>password</user_password>
    <encrypted config:type="boolean">false</encrypted>
  </user>
  <user>
    <username>tux</username>
    <user_password>password</user_password>
    <encrypted config:type="boolean">false</encrypted>
  </user>
</users>
```

以下示例演示了一种更复杂的方案。将应用 `/etc/default/useradd` 中的系统范围默认设置，例如外壳，或主目录的父目录。

例 4.63：复杂用户配置

```
<users config:type="list">
  <user>
    <username>root</username>
    <user_password>password</user_password>
    <uid>1001</uid>
    <gid>100</gid>
    <encrypted config:type="boolean">false</encrypted>
    <fullname>Root User</fullname>
```

```

    <authorized_keys config:type="list">
      <listentry> ssh-rsa
      AAAAB3NzaC1yc2EAAAADAQABAAQDKLt1vnW2vTJpBp3VK91rFsBvpY97NljsVLdgUrlPbZ/
      L51FerQQ+djQ/ivDASQj0+567nMGqfYGFA/De1EGMMEoeShza67qjNi14L1HBGgVojaNajMR/
      NI2d1kDyvsgRy7D7FT5UGGUNT0dlcSD3b85zwgHeYLidgcGIoKeRi7HpVD00TyhwUv4sq3ubrPCWARgPe0LdVF9c1C8
      PvMDa96DpxH1VlzJlAIHQsMkMHbsCazPNC0++Kp5ZVERiH root@example.net</listentry>
    </authorized_keys>
  </user>
  <user>
    <username>tux</username>
    <user_password>password</user_password>
    <uid>1002</uid>
    <gid>100</gid>
    <encrypted config:type="boolean">false</encrypted>
    <fullname>Plain User</fullname>
    <home>/Users/plain</home>
    <password_settings>
      <max>120</max>
      <inact>5</inact>
    </password_settings>
  </user>
</users>

```



注意：将重写 `authorized_keys` 文件

如果配置文件在 `authorized_keys` 部分为用户定义了一组 SSH 授权密钥，则会重写现有的 `$HOME/.ssh/authorized_keys` 文件。如果该文件不存在，将使用指定的内容创建该文件。不在 AutoYaST 控制文件中指定相应的部分即可避免重写现有的 `authorized_keys` 文件。



注意：合并 `rootpassword` 和 `root` 用户选项

可以在 `rootpassword` 中指定 **linuxrc**，并为 `root` 用户设置用户部分。如果此部分缺少口令，则会使用 **linuxrc** 中的口令。配置文件中的口令优先于 **linuxrc** 口令。



警告：请勿使用 `root` 以外的名称创建超级用户帐户。

虽然从技术上来说可以创建用户 ID (`uid`) 为 0 但名称不是 `root` 的帐户，但特定应用程序、脚本或第三方产品可能需要名为 `root` 的用户存在才能正常运行。虽然这样的配置始终针对的是个别环境，但供应商更新可能会覆盖必要的调整，因此这变成了一项需要持续执行的任务，而不是一次性设置。在涉及到第三方应用程序的极复杂设置中更是如此。在这些设置中，需要向涉及的每个供应商校验是否支持重命名 `root` 帐户。

由于无法预见重命名 `root` 帐户造成的后果，SUSE 不支持重命名 `root` 帐户。

通常，重命名 `root` 帐户背后的理念是隐藏该帐户或使其不可预测。但是，`/etc/passwd` 要求普通用户具有 644 权限，因此系统的任何用户都能获得用户 ID 0 的登录名。如果要使用更好的方法来保护 `root` 帐户的安全，请参见《安全和强化指南》，第 14 章“用户管理”，第 14.5 节“限制 `root` 登录”和《安全和强化指南》，第 14 章“用户管理”，第 14.5.3 节“限制 SSH 登录”。



注意：指定用户 ID (`uid`)

Linux 系统上的每个用户都有一个数字用户 ID。您可以使用 `uid` 手动在 AutoYaST 控制文件中指定这样的用户 ID，也可以使用 `uid` 让系统自动选择用户 ID。

用户 ID 应在整个系统中唯一。否则，某些应用程序（例如登录管理器 `gdm`）可能不再会如预期般工作。

通过 AutoYaST 控制文件添加用户时，强烈建议不要混合使用用户定义的 ID 和自动提供的 ID。否则无法保证 ID 唯一。为通过 AutoYaST 控制文件添加的所有用户指定 ID，或者让系统为所有用户选择 ID。

属性、值、描述

`username`

文本

```
<username>lukesw</username>
```

必需。应是有效的用户名。如果您不确定，请查看 `man 8 useradd`。

fullname

文本

```
<fullname>Tux Torvalds</fullname>
```

可选。用户的全名。

forename

文本

```
<forname>Tux</forename>
```

可选。用户的名字。

surname

文本

```
<surname>Skywalker</surname>
```

可选。用户的姓氏。

uid

数字

```
<uid>1001</uid>
```

可选。用户 ID。此 ID 应唯一，且必须是非负数。如果未指定，AutoYaST 将自动选择一个用户 ID。有关更多信息，另请参见[注意：指定用户 ID \(uid\)](#)。

gid

数字

```
<gid>100</gid>
```

可选。初始组 ID。此 ID 必须唯一且为非负数。此外，它必须引用现有的组。

home

路径

```
<home>/home/luke</home>
```

可选。用户主目录的绝对路径。默认将使用 /home/username（例如，alice 的主目录将是 /home/alice）。

home_btrfs_subvolume

布尔

```
<home_btrfs_subvolume config:type="boolean">true</home_btrfs_subvolume>
```

可选。在 Btrfs 子卷中生成主目录。默认已禁用。

shell

路径

```
<shell>/usr/bin/zsh</shell>
```

可选。/bin/bash 为默认值。如果您选择其他软件，请确保该软件已安装（将对应的软件包添加到 software 部分）。

user_password

文本

```
<user_password>some-password</user_password>
```

可选。可以写入纯文本（不建议）或加密形式的用户口令。要创建加密的口令，请使用 **mkpasswd**。输入 /etc/shadow（第二列）中写入的口令。要在配置文件中启用或禁用加密的口令，请参见 encrypted 参数。启用加密的口令后，如果您输入感叹号 (!)，该值将会复制到 /etc/shadow 的口令字段中。因此，您的帐户的口令会被锁定，无法在控制台上登录该帐户。

encrypted

布尔

```
<encrypted config:type="boolean">true</encrypted>
```

可选。如果未指定，则视为 false。指示是否加密配置文件中的用户口令。AutoYaST 支持标准加密算法 (请参见 man 3 crypt)。

password_settings

密码设置

```
<password_settings>  
  <expire/>
```



```
<max>60</max>
<warn>7</warn>
</password_settings>
```

可选。可以自定义某些口令设置：expire（采用 YYYY-MM-DD 格式的帐户失效日期）、flag（/etc/shadow 标志）、inact（口令失效多少天后禁用帐户）、max（口令保持有效的最长天数）、min（口令失效后的宽限期天数，在此期间用户仍可更改口令）和 warn（在口令失效之前的多少天开始发出口令更改提醒）。

authorized_keys

授权密钥的列表

```
<authorized_keys config:type="list">
  <listentry>ssh-rsa ...</listentry>
</authorized_keys>
```

要写入到 \$HOME/.ssh/authorized_keys 的授权密钥列表。请参见以下示例。

4.31.2 用户默认值

配置文件可为新用户指定一组默认值，例如口令失效日期、初始组、主目录前缀等。除了将这些设置用作配置文件中定义的用户默认值以外，AutoYaST 还会将这些设置写入到 /etc/default/useradd 或任何其他适当文件中，以供 useradd 读取。

属性、值、描述

group

文本

```
<group>100</group>
```

可选。默认的初始登录组。

home

路径

```
<home>/home</home>
```

可选。用户的主目录前缀。

expire

日期

```
<expire>2017-12-31</expire>
```

可选。默认的口令失效日期，采用 YYYY-MM-DD 格式。

inactive

数字

```
<inactive>3</inactive>
```

可选。在多少天后禁用已失效的帐户。

shell

路径

```
<shell>/usr/bin/fish</shell>
```

默认登录外壳。/bin/bash 为默认值。如果您选择其他软件，请确保该软件已安装（将对应的软件包添加到 software 部分）。

umask

文件创建模式掩码

```
<umask>022</umask>
```

为主目录设置文件创建模式掩码。默认情况下，useradd 将使用 022。有关更多信息，请查看 man 8 useradd 和 man 1 umask。

4.31.3 组

可按以下示例中所示，在 <groups> 中定义组列表。

例 4.64：组配置

```
<groups config:type="list">
  <group>
    <gid>100</gid>
    <groupname>users</groupname>
    <userlist>bob,alice</userlist>
```

```
</group>
</groups>
```

属性、值、描述

groupname

文本

```
<groupname>users</groupname>
```

必需。应是有效的组名。如果您不确定，请查看 [man 8 groupadd](#)。

gid

数字

```
<gid>100</gid>
```

可选。组 ID。此 ID 必须唯一且为非负数。

userlist

用户列表

```
<userlist>bob,alice</userlist>
```

可选。属于该组的用户列表。必须用逗号分隔用户名。

4.31.4 登录设置

可以通过 AutoYaST 配置文件启用两项特殊登录设置：自动登录和无口令登录。默认已禁用这两种登录方式。

例 4.65：启用自动登录和无口令登录

```
<login_settings>
  <autologin_user>vagrant</autologin_user>
  <password_less_login config:type="boolean">true</password_less_login>
</login_settings>
```

属性、值、描述

password_less_login

布尔

```
<password_less_login config:type="boolean">true</password_less_login>
```

可选。启用无口令登录。此设置仅影响图形登录。

autologin_user

文本

```
<autologin_user>alice</autologin_user>
```

可选。为给定的用户启用自动登录。

4.32 自定义用户脚本

通过将脚本添加到自动安装过程，您可以根据自己的需求自定义安装，并实现对不同安装阶段的控制。

在自动安装过程中，可以在不同的时间点执行五种类型的脚本：

除了 init scripts，所有脚本都必须包含在 <scripts> 部分中。

- pre-scripts（初期，在任何其他操作尚未真正发生之前）
- postpartitioning-scripts（在分区并挂载到 /mnt 之后，完成 RPM 安装之前）
- chroot-scripts（在安装软件包之后，首次引导之前）
- post-scripts（在首次引导已安装系统期间，未运行任何服务）

Init scripts（当安装的系统首次引导时、当所有服务都在运行时）不由 YaST 执行，因此具有特殊状态。有关更多信息，请参见第 4.32.5 节“Init 脚本”。

4.32.1 前脚本

在 YaST 对系统进行任何实际更改之前执行（进行分区和软件包安装之前，但在硬件检测之后）。

您可以使用安装前脚本来修改控制文件，并让 AutoYaST 重新读取控制文件。在 /tmp/profile/autoinst.xml 中找到您的控制文件。调整该文件，并将修改后的版本储存在 /tmp/profile/modified.xml 中。完成安装前脚本后，AutoYaST 将读取修改后的文件。

还可以修改安装前脚本中的储存设备。例如，您可以创建新的分区，或更改某些技术（例如多路径）的配置。执行所有前脚本后，AutoYaST 始终会再次检查储存设备。



注意：前脚本和确认

安装前脚本于早期安装阶段执行。这意味着，如果您请求确认安装，则会先执行这些脚本，然后再显示确认屏幕 ([profile/install/general/mode/confirm](#))。



注意：前脚本和 Zypper

要在前脚本中调用 **Zypper**，需要设置环境变量 **ZYPP_LOCKFILE_ROOT="/var/run/autoyast"**，以免与正在运行的 YaST 进程冲突。

必须按如下所示放置 `pre-script` 元素：

```
<scripts>
  <pre-scripts config:type="list">
    <script>
      ...
    </script>
  </pre-scripts>
</scripts>
```

4.32.2 分区后脚本

在 YaST 已完成分区并已写入 `/etc/fstab` 之后执行。空系统已挂载到 `/mnt`。

必须按如下所示放置 `postpartitioning-script` 元素：

```
<scripts>
  <postpartitioning-scripts config:type="list">
    <script>
      ...
    </script>
  </postpartitioning-scripts>
</scripts>
```

4.32.3 Chroot 环境脚本

Chroot 脚本是在首次重引导计算机之前执行的。您可以在安装 chroot 进入已安装系统并配置引导加载程序之前执行 chroot 脚本，也可以在 chroot 进入已安装系统之后（请查看 [chrooted](#) 参数来确定该状态）执行脚本。

必须按如下所示放置 `chroot-scripts` 元素：

```
<scripts>
  <chroot-scripts config:type="list">
    <script>
      ...
    </script>
  </chroot-scripts>
</scripts>
```

4.32.4 后脚本

这些脚本是在 AutoYaST 已完成系统配置并已首次引导系统之后执行的。

必须按如下所示放置 `post-script` 元素：

```
<scripts>
  <post-scripts config:type="list">
    <script>
      ...
    </script>
  </post-scripts>
</scripts>
```

4.32.5 Init 脚本

当 YaST 已完成，于已初始化网络之后的初始引导过程中会执行这些脚本。这些最终脚本是使用 `/usr/lib/YaST2/bin/autoyast-initscripts.sh` 执行的，并且只会执行一次。Init 脚本是使用标记 **init-scripts** 来配置的。

必须按如下所示放置 `init-script` 元素：

```
<init-scripts config:type="list">
  <script>
    ...
  </script>
</init-scripts>
```

Init 脚本与其他脚本类型不同，因为它们不是由 YaST 执行，而是在 YaST 已完成后执行。因此，它们的 XML 表示不同于其他脚本类型。

INIT 脚本 XML 表示

location

定义从中提取脚本的位置。位置可以与配置文件的位置相同（HTTP、FTP、NFS 等）。

```
<location>http://10.10.0.1/myInitScript.sh</location>
```

必须定义 `<location>` 或 `<source>`。

source

脚本本身（源代码），封装在 CDATA 标记中。如果您不想将整个外壳脚本放入 XML 配置文件，请使用位置参数。

```
<source>
<![CDATA[echo "Testing the init script" >/tmp/init_out.txt]]></source>
```

必须定义 `<location>` 或 `<source>`。

filename

脚本的文件名。此文件将储存在 `/tmp` 下的临时目录中

```
<filename>mynitScript5.sh</filename>
```

如果您只有一个 init 脚本，则此元素是可选的。在这种情况下，将使用默认名称 (`init-scripts`)。如果您指定了多个 init 脚本，则必须为每个脚本设置唯一的名称。

rerun

正常情况下，一个脚本只运行一次，即使您使用 `ayast_setup` 多次运行 XML 文件也是如此。将此布尔值设置为 `true` 可更改此默认行为。

```
<rerun config:type="boolean">true</rerun>
```

可选。默认值为 `false`（脚本只运行一次）。

手动将脚本添加到控制文件时，需将脚本包含在 **CDATA** 元素中，以免与文件语法以及控制文件中定义的其他标记产生混淆。

4.32.6 脚本 XML 表示

下面所述的大多数 XML 元素都可用于上述所有脚本类型，但 **init 脚本** 除外，此类脚本的定义可能只包含这些元素的一部分。有关此类脚本的更多信息，请参见第 4.32.5 节“Init 脚本”。

! 重要：弃用的元素

`debug` 元素已弃用，在将来的版本中可能会将其去除。为适应此情况，请在 `interpreter` 中使用解释器特定的调试参数。例如，不要使用 `<interpreter>shell</interpreter>`，而是使用 `<interpreter>/bin/sh -x</interpreter>`，其效果等同于启用 `debug` 标志。

脚本 XML 表示

location

定义从中提取脚本的位置。位置可以与控制文件的位置相同（HTTP、FTP、NFS 等），例如：

```
<location>http://10.10.0.1/myPreScript.sh</location>
```

位置还可以定义为相对 URL，其中的路径相对于包含控制文件的目录。如果使用相对 URL，则 `location` 属性如下所示：

```
<location>relurl://script.sh</location>
```

或者，您可以使用 `repo` URI 方案。脚本位置相对于安装源，定义如下所示：

```
<location>repo:/script.sh</location>
```

必须定义 `location` 或 `source`。

source

脚本本身（源代码），封装在 CDATA 标记中。如果您不想将整个外壳脚本放入 XML 控制文件，请引用位置参数。


```
<source>
<![CDATA[
echo "Testing the pre script" > /tmp/pre-script_out.txt
]]>
</source>
```

必须定义 location 或 source。

interpreter

指定必须为脚本使用的解释器。可以指定给定环境中任何可用的解释器。您可以提供解释器的完整路径，包括参数。debug 标志支持的关键字解释器 “shell”、“perl” 和 “python” 也已弃用。

```
<interpreter>/bin/bash -x</interpreter>
```

可选；默认值为 shell。

file name

脚本的文件名。此文件将储存在 /tmp 下的临时目录中。

```
<filename>myPreScript5.sh</filename>
```

可选；默认值为脚本的类型（在本例中为 pre-scripts）。如果您有多个脚本，应该为每个脚本定义不同的名称。如果未定义 filename，但定义了 location，将会使用该位置路径中的文件名。

feedback

如果此布尔值为 true，则会在弹出窗口中显示脚本的输出和错误消息（STDOUT 和 STDERR）。用户需要通过“确定”按钮确认这些消息。

```
<feedback config:type="boolean">true</feedback>
```

可选；默认值为 false。

feedback_type

此元素可为 message、warning 或 error。在 <report> 部分设置这些弹出窗口的超时。

```
<feedback_type>warning</feedback_type>
```

可选；如果未指定，将使用始终阻止弹出窗口。

debug

如果此元素为 true，将记录外壳脚本的每一行。将在打开警告的情况下运行 Perl 脚本。这仅适用于弃用的关键字 interpreter。对于其他语言，请在 interpreter 值中以参数形式提供解释器的路径，例如 “<interpreter>ruby -w</interpreter>”。

```
<debug config:type="boolean">true</debug>
```

可选；默认值为 true。

notification

将会根据脚本在后台运行的时间长短在弹出窗口中显示此文本。

```
<notification>Please wait while script is running...</notification>
```

可选；如果未配置，则不显示通知弹出窗口。

param-list

可以指定提供给所要调用的脚本的参数。您可以指定多个 param 项。在脚本命令行上使用单个空格字符串串联这些项。如果必须指定任何外壳引用（例如，为了保护嵌入的空格），则需要包含此元素。

```
<param-list config:type="list">
  <param>par1</param>
  <param>par2 par3</param>
  <param>"par4.1 par4.2"</param>
</param-list>
```

可选；如果未配置，则不会将任何参数传递给脚本。

rerun

一个脚本只运行一次。即使您使用 ayast_setup 多次运行 XML 文件，该脚本也只运行一次。将此布尔值设置为 true 可更改此默认行为。

```
<rerun config:type="boolean">true</rerun>
```

可选；默认值为 false，表示脚本只运行一次。

chrooted

在安装期间，新系统将挂载到 `/mnt`。如果此参数设置为 `false`，AutoYaST 将不运行 **chroot**，并且在此阶段不会安装引导加载程序。如果该参数设置为 `true`，则 AutoYaST 会执行 **chroot** 进入 `/mnt` 并安装引导加载程序。结果是，您不再需要使用 `/mnt` 前缀就能更改新安装系统中的任何设置。

```
<chrooted config:type="boolean">true</chrooted>
```

可选；默认值为 `false`。此选项仅适用于 chroot 环境脚本。

4.32.7 脚本示例

例 4.66：脚本配置

```
<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://
www.suse.com/1.0/configs">
  <scripts>
    <chroot-scripts config:type="list">
      <script>
        <chrooted config:type="boolean">true</chrooted>
        <filename>chroot-post.sh</filename>
        <interpreter>shell</interpreter>
        <source><![CDATA[
echo "Testing chroot (chrooted) scripts"
ls
]]>
        </source>
      </script>
      <script>
        <filename>chroot-pre.sh</filename>
        <interpreter>/bin/bash -x</interpreter>
        <source><![CDATA[
echo "Testing chroot scripts"
df
cd /mnt
ls
```

```

]]>
    </source>
</script>
</chroot-scripts>
<post-scripts config:type="list">
    <script>
        <filename>post.sh</filename>
        <interpreter>shell</interpreter>
        <source><![CDATA[
echo "Running Post-install script"
systemctl start portmap
mount -a 192.168.1.1:/local /mnt
cp /mnt/test.sh /tmp
umount /mnt
]]>
        </source>
    </script>
    <script>
        <filename>post.pl</filename>
        <interpreter>perl</interpreter>
        <source><![CDATA[
print "Running Post-install script";
]]>
        </source>
    </script>
</post-scripts>
<pre-scripts config:type="list">
    <script>
        <interpreter>shell</interpreter>
        <location>http://192.168.1.1/profiles/scripts/prescripts.sh</location>
    </script>
    <script>
        <filename>pre.sh</filename>
        <interpreter>shell</interpreter>
        <source><![CDATA[
echo "Running pre-script"
]]>
        </source>

```

```

    </script>
  </pre-scripts>
  <postpartitioning-scripts config:type="list">
    <script>
      <filename>postpart.sh</filename>
      <interpreter>shell</interpreter>
      <debug config:type="boolean">false</debug>
      <feedback config:type="boolean">true</feedback>
      <source><![CDATA[
touch /mnt/testfile
echo Hi
]]>
      </source>
    </script>
  </postpartitioning-scripts>
</scripts>
</profile>

```

完成安装后，可以在 `/var/adm/autoinstall` 目录中找到脚本和输出日志。脚本位于 `scripts` 子目录中，输出日志位于 `log` 目录中。

日志中包含执行脚本时产生的输出，包括标准输出和标准错误输出的组合。

如果脚本以非零退出代码结束，一条包含日志内容的警告将会显示，除非提供了 `feedback` 选项。

4.33 系统变量 (sysconfig)

使用 `sysconfig` 资源可以直接在 `sysconfig` 储存库 (`/etc/sysconfig`) 中定义配置变量。使用 `sysconfig` 变量可以完全按照您的需求微调许多系统组件和环境变量。

以下示例演示如何使用 `sysconfig` 资源设置变量。

例 4.67：SYSCONFIG 配置

```

<sysconfig config:type="list" >
  <sysconfig_entry>
    <sysconfig_key>XNTPD_INITIAL_NTPDATE</sysconfig_key>
    <sysconfig_path>/etc/sysconfig/xntp</sysconfig_path>
  </sysconfig_entry>
</sysconfig>

```

```

    <sysconfig_value>ntp.host.com</sysconfig_value>
  </sysconfig_entry>
  <sysconfig_entry>
    <sysconfig_key>HTTP_PROXY</sysconfig_key>
    <sysconfig_path>/etc/sysconfig/proxy</sysconfig_path>
    <sysconfig_value>proxy.host.com:3128</sysconfig_value>
  </sysconfig_entry>
  <sysconfig_entry>
    <sysconfig_key>FTP_PROXY</sysconfig_key>
    <sysconfig_path>/etc/sysconfig/proxy</sysconfig_path>
    <sysconfig_value>proxy.host.com:3128</sysconfig_value>
  </sysconfig_entry>
</sysconfig>

```

可以提供相对路径和绝对路径。如果未指定绝对路径，则会将其视为 /etc/sysconfig 目录下的某个 sysconfig 文件。

4.34 添加完整配置

对于许多应用程序和服务，您可能都需要将配置文件复制到已安装系统上的相应位置。例如，如果您正在安装 Web 服务器，可能会有一个服务器配置文件 (httpd.conf)。

使用此资源您可以通过指定已安装系统上的最终路径，将该文件嵌入控制文件中。YaST 会将此文件复制到指定的位置。

使用此功能需安装 autoyast2 软件包。如果缺少该软件包，AutoYaST 会自动予以安装。

您可以指定要从中检索文件的 file_location。此位置也可以是网络上的某个位置

（例如 HTTP 服务器）：<file_location>http://my.server.site/issue</file_location>。

您也可以使用 relurl:// 前缀指定本地文件，例如：<file_location>relurl://path/to/file.conf</file_location>。

您可以通过指定以斜线结尾的 file_path 来创建目录。

例 4.68：将文件转储到已安装系统

```
<files config:type="list">
```

```

<file>
  <file_path>/etc/apache2/httpd.conf</file_path>
  <file_contents>

<![CDATA[
some content
]]>

  </file_contents>
</file>
<file>
  <file_path>/mydir/a/b/c/</file_path> <!-- create directory -->
</file>
</files>

```

下面演示了一个更高级的示例。此配置将使用 `file_contents` 中提供的内容创建一个文件，并更改该文件的权限和所有权。将文件复制到系统后，将执行一个脚本。此脚本可用于修改该文件，并使其准备好在客户端环境中使用。

例 4.69：将文件转储到已安装系统

```

<files config:type="list">
  <file>
    <file_path>/etc/someconf.conf</file_path>
    <file_contents>

<![CDATA[
some content
]]>

    </file_contents>
    <file_owner>tux.users</file_owner>
    <file_permissions>444</file_permissions>
    <file_script>
      <interpreter>shell</interpreter>
      <source>

<![CDATA[

```

```
#!/bin/sh

echo "Testing file scripts" >> /etc/someconf.conf
df
cd /mnt
ls
]]>

</source>
</file_script>
</file>
</files>
```

4.35 安装期间要求用户提供值

在安装期间，您可以选择让用户决定控制文件特定部分的值。如果您使用此功能，则安装期间会显示一个弹出窗口请求用户输入控制文件特定部分。如果您想要进行全自动化安装，但应由用户设置本地帐户的口令，则您可以通过在控制文件中使用 `ask` 指令实现此目的。

下面列出的元素必须包含在以下 XML 结构中：

```
<general>
  <ask-list config:type="list">
    <ask>
      ...
    </ask>
  </ask-list>
</general>
```

要求用户提供值：XML 表示

question

您要询问用户的问题。

```
<question>Enter the LDAP server</question>
```

默认值是元素的路径（该路径通常看起来很奇怪，因此我们建议输入问题）。

default

为用户设置预先选择。将使用此值填充文本输入。将为复选框提供 `true` 或 `false`，在选项中会预先选择给定的值。

```
<default>dc=suse,dc=de</default>
```

可选。

help

在问题左侧显示的可选帮助文本。

```
<help>Enter the LDAP server address.</help>
```

可选。

title

在问题上方显示的可选标题。

```
<title>LDAP server</title>
```

可选。

type

要更改的元素类型。可能的值为 `symbol`、`boolean`、`string` 和 `integer`。分区部分中的文件系统为符号，而用户配置中的 `encrypted` 元素为布尔值。在控制文件中查看 `config:type="...."` 属性时可以看到该元素的类型。还可以使用 `static_text` 作为类型。`static_text` 是一段文本，不需要任何用户输入，并可显示帮助文本中不包含的信息。

```
<type>symbol</type>
```

可选。默认值为 `string`。如果 `type` 为 `symbol`，则您还必须提供 `selection` 元素（参见下文）。

password

如果此布尔值设置为 `true`，将弹出口令对话框而不是简单的文本输入。仅当 `true` 为 `string` 时，将此项设置为 `type` 才有意义。

```
<password config:type="boolean">true</password>
```

可选。默认值为 false。

pathlist

path 元素的列表。path 是逗号分隔的元素列表，用于描述您要更改的元素的路径。例如，可以在控制文件的 <networking> 部分中找到网络配置元素。因此，要更改该值，需将路径设置为 networking。

```
<pathlist config:type="list">
  <path>networking,dns,hostname</path>
  <path>...</path>
</pathlist>
```

要更改控制文件中第一个用户的口令，需将路径设置为 users,0,user_password。0 表示配置部分中的第一项。例如，在下面所示的用户列表 <users config:type="list"> 中，它表示 root。1 是第二项，以此类推。

```
<users config:type="list">
  <user>
    <username>root</username>
    <user_password>password to change</user_password>
    <encrypted config:type="boolean">>false</encrypted>
  </user>
  <user>
    <username>tux</username>
    <user_password>password to change</user_password>
    <encrypted config:type="boolean">>false</encrypted>
  </user>
</users>
```

如果 <user> 部分与上面的示例类似，要设置 root 的口令，请按如下所示使用 <pathlist>：

```
<pathlist config:type="list">
  <path>users,0,user_password</path>
</pathlist>
```

此信息是可选的，但您至少应该提供 path 或 file。

file

您可将问题的答案储存在文件中，以供稍后在某个脚本中使用。如果您在 `stage=initial` 阶段提问，并希望在第 2 阶段使用答案，则需要在 `chrooted=false` 的情况下运行的 `chroot` 脚本中复制答案文件。使用以下命令：`cp /tmp/my_answer /mnt/tmp/`。原因是第 1 阶段的 `/tmp` 位于 RAM 磁盘中，在重引导后将会丢失，但安装的系统已挂载到 `/mnt/`。

```
<file>/tmp/answer_hostname</file>
```

此信息是可选的，但您至少应该提供 `path` 或 `file`。

stage

Stage 用于配置弹出问题的安装阶段。您可以将该值设置为 `cont` 或 `initial`。`initial` 表示弹出窗口在安装过程中很早出现，即前脚本运行后不久。`cont` 表示显示问题的对话框出现在第一次重引导后，即系统第一次引导时。您在 `initial` 阶段所回答问题的答案将写入到硬盘上的控制文件。如果您在 `initial` 阶段输入明文口令，则应会了解这一点。当然，在 `cont` 阶段要求使用文件系统没有意义。硬盘在该阶段已经分区，因此问题无效。

```
<stage>cont</stage>
```

可选。默认值为 `initial`。

selection

`selection` 元素包含 `entry` 元素的列表。每个 `entry` 表示可供用户选择的一个可能选项。用户无法在文本框中输入值，但可以从值列表中选择值。

```
<selection config:type="list">
  <entry>
    <value>
      btrfs
    </value>
    <label>
      Btrfs File System
    </label>
  </entry>
  <entry>
```

```
<value>
    ext3
</value>
<label>
    Extended3 File System
</label>
</entry>
</selection>
```

对于 `type=string` 此元素为可选的，对于 `type=boolean` 此元素不可用，对于 `type=symbol` 此元素是必需的。

dialog

您可以在每个对话框询问多个问题。为此，请使用整数指定 `dialog-id`。具有相同 `dialog-id` 的所有问题同属于一个对话框。对话框也按 ID 排序。

```
<dialog config:type="integer">3</dialog>
```

可选。

element

您可以在每个对话框中包含多个问题。为此，需要使用整数指定 `element-id`。对话框中的问题按 ID 排序。

```
<element config:type="integer">1</element>
```

可选（请参见 `dialog`）。

width

您可以增加对话框的默认宽度。如果每个对话框有多个宽度规范，则将使用最大的值。该数字大致相当于字符数。

```
<width config:type="integer">50</width>
```

可选。

height

您可以增加对话框的默认高度。如果每个对话框有多个高度规范，则将使用最大的值。该数字大致相当于行数。

```
<height config:type="integer">15</height>
```

可选。

frametitle

您可以在每个对话框中包含多个问题。对话框中的每个问题都有一个框架，该框架可以具有框架标题（每个问题的小标题）。您可将多个元素放入一个框架中。这些元素需使用相同的框架标题。

```
<frametitle>User data</frametitle>
```

可选；默认不使用框架标题。

script

您可以在回答问题后运行脚本。（有关脚本的详细说明，请参见第 4.35.1 节“默认值脚本”。）

```
<script>...</script>
```

可选；默认为无脚本。

ok_label

您可以更改确定按钮上的标签。以指定对话框标签的最后一个元素为准。

```
<ok_label>Finish</ok_label>
```

可选。

back_label

您可以更改后退按钮上的标签。以指定对话框标签的最后一个元素为准。

```
<back_label>change values</back_label>
```

可选。

timeout

您可在此处指定一个用作超时（以秒为单位）的整数。如果用户在超时之前未回答问题，则会使用默认值回答。当用户触摸或更改对话框中的任一控件时，将关闭超时设置，用户需要单击确定来确认对话框中的操作。

```
<timeout config:type="integer">30</timeout>
```

可选；缺失值将被解释为 0，即关闭超时。

default_value_script

您可以运行脚本来设置问题的默认值（有关默认值脚本的详细说明，请参见第 4.35.1 节“默认值脚本”）。如果您可以 calculate 默认值（特别是与 timeout 选项结合使用时），此功能将十分有用。

```
<default_value_script>...</default_value_script>
```

可选；默认为无脚本。

4.35.1 默认值脚本

您可以运行脚本来设置问题的默认值。如果您可以 calculate 默认值（特别是与 timeout 选项结合使用时），此功能将十分有用。

通过在以下 XML 结构中添加第 4.32.6 节“脚本 XML 表示”中所述的元素来定义脚本。

```
<general>
  <ask-list config:type="list">
    <ask>
      <default_value_script>
        ...
      </default_value_script>
    </ask>
  </ask-list>
</general>
```

echo 到 STDOUT 的任何内容都将用作询问对话框的默认值。如果您的脚本包含非 0 退出代码，则会使用常规 default 元素。使用 **echo -n** 抑制 \n 时请小心，应回显合理的值，对于布尔值不要使用“okay”（请使用“true”）。

4.35.2 脚本

您可以在回答问题后运行脚本。

下面列出的元素必须包含在以下 XML 结构中：

```
<general>
  <ask-list config:type="list">
```

```
<ask>
  <script>
    ...
  </script>
</ask>
</ask-list>
</general>
```

除了第 4.32.6 节“脚本 XML 表示”中列出的元素之外，`<ask>` 元素中的脚本还支持以下选项：

脚本：XML 表示

filename

脚本的文件名。

```
<filename>my_ask_script.sh</filename>
```

默认值为 `ask_script.sh`

environment

一个布尔值，用于将问题答案的值作为环境变量传递给脚本。该变量名为 `VAL`。

```
<environment config:type="boolean">true</environment>
```

可选。默认值为 `false`。

feedback

一个布尔值，用于打开脚本执行的反馈。STDOUT 将显示在弹出窗口中，执行脚本后，必须予以确认。

```
<feedback config:type="boolean">true</feedback>
```

可选，默认值为 `false`。

rerun_on_error

在脚本收到退出代码 0（零）之前，将对话框保持在打开状态。您可以使用此功能来验证用户的输入。该脚本应该列显有意义的错误消息并返回一个非零代码。请注意，您还应将 `feedback` 选项设为 `true`，以使用户可以读取来自脚本的错误消息。可选，默认值为 `false`。

脚本可能会创建 `/tmp/next_dialog` 文件，其中包含要显示的以下对话框的 ID。-1 值会终止该序列。

下面提供了 `ask` 功能的用法示例。

```
<general>
  <ask-list config:type="list">
    <ask>
      <pathlist config:type="list">
        <path>ldap,ldap_server</path>
      </pathlist>
      <stage>cont</stage>
      <help>Choose your server depending on your department</help>
      <selection config:type="list">
        <entry>
          <value>ldap1.mydom.de</value>
          <label>LDAP for development</label>
        </entry>
        <entry>
          <value>ldap2.mydom.de</value>
          <label>LDAP for sales</label>
        </entry>
      </selection>
      <default>ldap2.mydom.de</default>
      <default_value_script>
        <source> <![CDATA[
echo -n "ldap1.mydom.de"
]]>
        </source>
      </default_value_script>
    </ask>
    <ask>
      <pathlist config:type="list">
        <path>networking,dns,hostname</path>
      </pathlist>
      <question>Enter Hostname</question>
      <stage>initial</stage>
      <default>enter your hostname here</default>
    </ask>
```



```

<ask>
  <pathlist config:type="list">
    <path>partitioning,0,partitions,0,filesystem</path>
  </pathlist>
  <question>File System</question>
  <type>symbol</type>
  <selection config:type="list">
    <entry>
      <value config:type="symbol">ext4</value>
      <label>default File System (recommended)</label>
    </entry>
    <entry>
      <value config:type="symbol">ext3</value>
      <label>Fallback File System</label>
    </entry>
  </selection>
</ask>
</ask-list>
</general>

```

以下示例演示了如何在不同的 AutoYaST 控制文件之间进行选择。完成询问对话框后，AutoYaST 将再次读取 modified.xml 文件。这样，您便可以提取全新的控制文件。

```

<general>
  <ask-list config:type="list">
    <ask>
      <selection config:type="list">
        <entry>
          <value>part1.xml</value>
          <label>Simple partitioning</label>
        </entry>
        <entry>
          <value>part2.xml</value>
          <label>encrypted /tmp</label>
        </entry>
        <entry>
          <value>part3.xml</value>
          <label>LVM</label>
        </entry>
      </selection>
    </ask>
  </ask-list>
</general>

```

```

        </selection>
        <title>XML Profile</title>
        <question>Choose a profile</question>
        <stage>initial</stage>
        <default>part1.xml</default>
        <script>
            <filename>fetch.sh</filename>
            <environment config:type="boolean">true</environment>
            <source>
<![CDATA[
wget http://10.10.0.162/$VAL -O /tmp/profile/modified.xml 2>/dev/null
]]>
            </source>
            <debug config:type="boolean">false</debug>
            <feedback config:type="boolean">false</feedback>
        </script>
    </ask>
</ask-list>
</general>

```

您可以使用如下脚本校验问题的答案：

```

<general>
  <ask-list config:type="list">
    <ask>
      <script>
        <filename>my.sh</filename>
        <rerun_on_error config:type="boolean">true</rerun_on_error>
        <environment config:type="boolean">true</environment>
        <source><![CDATA[
if [ "$VAL" = "myhost" ]; then
  echo "Illegal Hostname!";
  exit 1;
fi
exit 0
]]>
        </source>
        <debug config:type="boolean">false</debug>
        <feedback config:type="boolean">true</feedback>

```

```

</script>
<dialog config:type="integer">0</dialog>
<element config:type="integer">0</element>
<pathlist config:type="list">
  <path>networking,dns,hostname</path>
</pathlist>
<question>Enter Hostname</question>
<default>enter your hostname here</default>
</ask>
</ask-list>
</general>

```

4.36 内核转储



注意：可用性

此功能在 AArch64 上或 RAM 小于 1 GB 的系统上不可用。

利用 Kdump，如果整个内核崩溃，系统可以创建崩溃转储文件。崩溃转储文件包含系统崩溃时的内存内容。日后，支持人员或（内核）开发人员可以分析此类核心文件，以找出系统崩溃的原因。Kdump 最适合用于您无法在其中轻松再现这种崩溃情况，但必须解决相应问题的服务器。

但是，Kdump 也有一个弊端。启用 Kdump 时需要为 Kdump 额外预留 64 MB 到 128 MB 的系统 RAM，以备在发生系统崩溃以及需要生成转储时使用。

本节仅介绍如何使用 AutoYaST 设置 Kdump，而涉及 Kdump 的工作原理。有关细节，请参见 `kdump(7)` 手册页。

以下示例演示了 Kdump 一般配置。

例 4.70：KDUMP 配置

```

<kdump>
  <!-- memory reservation -->
  <add_crash_kernel config:type="boolean">true</add_crash_kernel>
  <crash_kernel>256M-:64M</crash_kernel>

```

```

<general>

  <!-- dump target settings -->
  <KDUMP_SAVEDIR>ftp://stravinsky.suse.de/incoming/dumps</KDUMP_SAVEDIR>
  <KDUMP_FREE_DISK_SIZE>64</KDUMP_FREE_DISK_SIZE>
  <KDUMP_KEEP_OLD_DUMPS>5</KDUMP_KEEP_OLD_DUMPS>

  <!-- filtering and compression -->
  <KDUMP_DUMPFORMAT>compressed</KDUMP_DUMPFORMAT>
  <KDUMP_DUMPLEVEL>1</KDUMP_DUMPLEVEL>

  <!-- notification -->
  <KDUMP_NOTIFICATION_TO>tux@example.com</KDUMP_NOTIFICATION_TO>
  <KDUMP_NOTIFICATION_CC>spam@example.com devnull@example.com</
KDUMP_NOTIFICATION_CC>
  <KDUMP_SMTP_SERVER>mail.example.com</KDUMP_SMTP_SERVER>
  <KDUMP_SMTP_USER></KDUMP_SMTP_USER>
  <KDUMP_SMTP_PASSWORD></KDUMP_SMTP_PASSWORD>

  <!-- kdump kernel -->
  <KDUMP_KERNELVER></KDUMP_KERNELVER>
  <KDUMP_COMMANDLINE></KDUMP_COMMANDLINE>
  <KDUMP_COMMANDLINE_APPEND></KDUMP_COMMANDLINE_APPEND>

  <!-- expert settings -->
  <KDUMP_IMMEDIATE_REBOOT>yes</KDUMP_IMMEDIATE_REBOOT>
  <KDUMP_VERBOSE>15</KDUMP_VERBOSE>
  <KEXEC_OPTIONS></KEXEC_OPTIONS>
</general>
</kdump>

```

Kdump 默认处于启用状态。下面的配置展示了如何禁用该功能。

例 4.71：禁用的 KDUMP 配置

```

<kdump>
  <add_crash_kernel config:type="boolean">false</add_crash_kernel>
</kdump>

```

4.36.1 内存预留

第一步是在引导时为 Kdump 预留内存。由于必须在引导过程的早期预留内存，因此，配置是通过名为 `crashkernel` 的内核命令行参数完成的。预留的内存将用于加载第二个内核，如此当第一个内核崩溃时，不需要重引导便可执行第二个内核。这第二个内核具有特殊的 `initrd`，其中包含通过网络保存转储或将转储保存到磁盘、发送通知电子邮件以及最终重引导所需的全部程序。

要为 Kdump 预留内存，请指定 `amount`（例如 `64M` 表示预留 RAM 中的 64 MB 内存）和 `offset`。语法为 `crashkernel=AMOUNT@OFFSET`。内核可以自动检测正确的偏移量（Xen 超级管理程序除外，在其中需要指定 `16M` 作为偏移量）。需要预留的内存量取决于体系结构和主内存。有关为 Kdump 预留的内存容量建议，请参见《系统分析和微调指南》，第 20 章 “Kexec 和 Kdump”，第 20.7.1 节 “Kdump 的手动配置”。

您还可以使用扩展的命令行语法来根据系统 RAM 指定预留的内存量。如果您为多个安装共享一个 AutoYaST 控制文件，或者您经常在一台计算机上去除或安装内存，则这种做法很有用。语法是：

```
BEGIN_RANGE_1-END_RANGE_1:AMOUNT_1,BEGIN_RANGE_2-END_RANGE_2:AMOUNT_2@OFFSET
```

`BEGIN_RANGE_1` 是第一个内存范围的起始（例如：`0M`），`END_RANGE_1` 是第一个内存范围的结束（应假定为 `infinity` 时可为空），依此类推。例如，`256M-2G:64M,2G-:128M` 会在系统的 RAM 介于 256 MB 到 2 GB 之间时预留 64 MB 崩溃内核内存，并在系统的 RAM 为 2 GB 以上时预留 128 MB 崩溃内核内存。

另一方面，可为 `crashkernel` 参数指定多个值。例如，如果您需要预留不同的低内存段和高内存段，请使用诸如 `72M,low` 和 `256M,high` 的值：

例 4.72：使用多个值预留 KDUMP 内存

```
<kdump>
  <!-- memory reservation (high and low) -->
  <add_crash_kernel config:type="boolean">true</add_crash_kernel>
  <crash_kernel config:type="list">
    <listentry>72M,low</listentry>
    <listentry>256M,high</listentry>
  </crash_kernel>
</kdump>
```

下面的列表显示了预留内存所需的设置：

KDUMP 内存预留设置：XML 表示

add_crash_kernel

如果应预留内存并已启用 Kdump，请设置为 true。

```
<add_crash_kernel config:type="boolean">true</add_crash_kernel>
```

必需

crash_kernel

使用上述 crashkernel 命令行语法。

```
<crash_kernel>256M:64M</crash_kernel>
```

还支持值列表。

```
<crash_kernel config:type="list">
  <listentry>72M,low</listentry>
  <listentry>256M,high</listentry>
</crash_kernel>
```

必需

4.36.2 转储保存

本节介绍崩溃转储的储存位置及储存方式。

4.36.2.1 目标

元素 KDUMP_SAVEDIR 指定要将转储保存到的 URL。可能的方法如下：

- 使用 file 可保存到本地磁盘；
- 使用 ftp 可保存到 FTP 服务器（不加密）；
- 使用 sftp 可保存到 SSH2 SFTP 服务器；

- 使用 `nfs` 可保存到 NFS 位置；
- 使用 `cifs` 可将转储保存到 Samba 或 Microsoft Windows 中的 CIFS/SMB 导出。

有关细节，请参见 `kdump(5)` 手册页。两个示例：`file:///var/crash`（符合 FHS 的默认位置）和 `ftp://user:password@host:port/incoming/dumps`。将创建一个名称中包含时戳的子目录，并将转储保存到该子目录。

将转储保存到本地磁盘时，可以使用 `KDUMP_KEEP_OLD_DUMPS` 自动删除旧转储。请将此参数设置为应保留的旧转储数。如果目标分区最终的可用磁盘空间少于 `KDUMP_FREE_DISK_SIZE` 中指定的值，则不会保存转储。

4.36.2.2 过滤和压缩

内核转储未压缩且未过滤。其大小可能与您的系统 RAM 相当。要缩小文件，请在事后压缩转储文件。在打开转储之前需将其解压缩。

要使用页压缩（压缩每个页并允许使用 `crash(8)` 调试工具进行动态解压缩），请将 `KDUMP_DUMPFORMAT` 设置为 `compressed`（默认值）。

您可能不想要保存所有内存页，例如，那些以零填充的页。要过滤转储，请设置 `KDUMP_DUMPLEVEL`。0 会生成完整转储，31 为最小转储。手册页 `kdump(5)` 和 `makedumpfile(8)` 列出了每个值会保存哪些页。

4.36.2.3 摘要

转储目标设置：XML 表示

`KDUMP_SAVEDIR`

一个 URL，指定要将转储和相关文件保存到的目标。

```
<KDUMP_SAVEDIR>file:///var/crash/</KDUMP_SAVEDIR>
```

必需

`KDUMP_FREE_DISK_SIZE`

保存转储后必须保留的可用磁盘空间，以 MB 为单位。如果可用空间不足，将不保存转储。

```
<KDUMP_FREE_DISK_SIZE>64</KDUMP_FREE_DISK_SIZE>
```

可选

KDUMP_KEEP_OLD_DUMPS

当 KDUMP_SAVEDIR 指向本地目录时要保留（不删除）的转储数。如果您不希望自动删除任何转储，请指定 0；如果要删除除当前转储外的其他所有转储，请指定 -1。

```
<KDUMP_KEEP_OLD_DUMPS>4</KDUMP_KEEP_OLD_DUMPS>
```

可选

4.36.3 电子邮件通知

配置发生计算机崩溃以及保存转储时要发出的电子邮件通知。

由于 Kdump 在 initrd 中运行，本地邮件服务器无法发送通知电子邮件。需要指定一台 SMTP 服务器（参见下文）。

需要在 KDUMP_NOTIFICATION_TO 中提供一个（且只能是一个）地址。可以在 KDUMP_NOTIFICATION_CC 中指定更多地址。在这两种情况下只能使用电子邮件地址，而不能使用实名。

指定 KDUMP_SMTP_SERVER 以及（如果服务器需要身份验证）KDUMP_SMTP_USER 和 KDUMP_SMTP_PASSWORD。我们不提供对 TLS/SSL 的支持，但将来可能会添加此项支持。

电子邮件通知设置：XML 表示

KDUMP_NOTIFICATION_TO

应将电子邮件发送到的一个确切电子邮件地址。可以在 KDUMP_NOTIFICATION_CC 中指定其他收件人。

```
<KDUMP_NOTIFICATION_TO  
>tux@example.com</KDUMP_NOTIFICATION_TO>
```

可选（如果为空，则禁用通知）

KDUMP_NOTIFICATION_CC

在通知电子邮件的抄送行中指定的零个、一个或多个收件人。


```
<KDUMP_NOTIFICATION_CC  
>wilber@example.com suzanne@example.com</KDUMP_NOTIFICATION_CC>
```

可选

KDUMP_SMTP_SERVER

用于递送邮件的 SMTP 服务器的主机名。支持 SMTP 身份验证（请参见 KDUMP_SMTP_USER 和 KDUMP_SMTP_PASSWORD），但不支持 TLS/SSL。

```
<KDUMP_SMTP_SERVER>email.suse.de</KDUMP_SMTP_SERVER>
```

可选（如果为空，则禁用通知）

KDUMP_SMTP_USER

与 KDUMP_SMTP_PASSWORD 一起使用的、用于 SMTP 身份验证的用户名。

```
<KDUMP_SMTP_USER>bwalke</KDUMP_SMTP_USER>
```

可选

KDUMP_SMTP_PASSWORD

与 KDUMP_SMTP_USER 一起使用的、用于 SMTP 身份验证的口令。

```
<KDUMP_SMTP_PASSWORD>geheim</KDUMP_SMTP_PASSWORD>
```

可选

4.36.4 Kdump 内核设置

如前所述，会引导一个特殊内核来保存转储。如果您不想使用自动检测机制来找出使用的内核（请参见 `kdump(5)` 手册页，其中介绍了用于查找内核的算法），可以在 KDUMP_KERNELVER 中指定自定义内核的版本。如果您将它设置为 `foo`，则会使用 `/boot/vmlinuz-foo` 或 `/boot/vmlinux-foo`（在包含 `vmlinuz` 文件的平台上会遵循此顺序）中的内核。

您可以指定用于引导 Kdump 内核的命令行。通常会使用引导命令行，减少与 Kdump 不相关的设置（例如 `crashkernel` 参数），添加 Kdump 所需的一些设置（请参见手册页 `kdump(5)`）。要指定其他参数，请使用 KDUMP_COMMANDLINE_APPEND。如果您完全清除自己所执行的操作并想要指定整个命令行，请设置 KDUMP_COMMANDLINE。

内核设置：XML 表示

KDUMP_KERNELVER

用于 Kdump 的内核的版本字符串。将其留空会使用自动检测机制（强烈建议）。

```
<KDUMP_KERNELVER>6.4.0-default</KDUMP_KERNELVER>
```

可选（如果为空则使用自动检测）

KDUMP_COMMANDLINE_APPEND

Kdump 内核的其他命令行参数。

```
<KDUMP_COMMANDLINE_APPEND>console=ttyS0,57600</KDUMP_COMMANDLINE_APPEND>
```

可选

KDUMP_Command Line

重写自动生成的 Kdump 命令行。须谨慎使用该功能。通常

KDUMP_COMMANDLINE_APPEND 应已足够。

```
<KDUMP_COMMANDLINE_APPEND>root=/dev/sda5 nr_cpus=1 irqpoll</KDUMP_COMMANDLINE_APPEND>
```

可选

4.36.5 专家设置

专家设置：XML 表示

KDUMP_IMMEDIATE_REBOOT

如果在保存转储后应自动重引导系统，则为 true，否则为 false。默认设置为自动重引导系统。

```
<KDUMP_IMMEDIATE_REBOOT>true</KDUMP_IMMEDIATE_REBOOT>
```

可选

KDUMP_VERBOSE

用于指定 Kdump 过程详细程度的位掩码。有关细节，请阅读 kdump(5)。

```
<KDUMP_VERBOSE>3</KDUMP_VERBOSE>
```

可选

KEXEC_OPTIONS

加载 Kdump 内核时要传递给 kexec 的其他选项。通常为 空。

```
<KEXEC_OPTIONS>--noio</KEXEC_OPTIONS>
```

可选

4.37 DNS 服务器

可以通过添加 dns-server 资源来配置 Bind DNS 服务器。该资源的三个较直接属性可以使用值 1（启用属性）或 0（禁用属性）。

属性	值	说明
<u>chroot</u>	0/1	必须将 DNS 服务器监禁在 chroot 中。
<u>start_service</u>	0/1	启用 Bind（系统启动时执行）。
<u>use_ldap</u>	0/1	在 LDAP 而不是本机配置文件中储存设置。

例 4.73：基本 DNS 服务器设置

```
<dns-server>
  <chroot>0</chroot>
  <start_service>1</start_service>
  <use_ldap>0</use_ldap>
</dns-server>
```

除了这些基本设置外，还可以使用三个类型列表属性来微调服务配置。

列表	说明
<u>logging</u>	DNS 服务器日志记录选项。

列表	说明
<u>options</u>	Bind 选项，例如要使用的文件和目录、转发器列表，以及其他配置设置。
<u>zones</u>	服务器已知的 DNS 区域列表，包括所有设置、记录和 SOA 记录。

例 4.74：配置 DNS 服务器区域和高级设置

```
<dns-server>
  <logging config:type="list">
    <listentry>
      <key>channel</key>
      <value>log_syslog { syslog; }</value>
    </listentry>
  </logging>
  <options config:type="list">
    <option>
      <key>forwarders</key>
      <value>{ 10.10.0.1; }</value>
    </option>
  </options>
  <zones config:type="list">
    <listentry>
      <is_new>1</is_new>
      <modified>1</modified>
      <options config:type="list"/>
      <records config:type="list">
        <listentry>
          <key>mydom.uwe.</key>
          <type>MX</type>
          <value>0 mail.mydom.uwe.</value>
        </listentry>
        <listentry>
          <key>mydom.uwe.</key>
          <type>NS</type>
          <value>ns.mydom.uwe.</value>
```

```

    </listentry>
  </records>
  <soa>
    <expiry>1w</expiry>
    <mail>root.aaa.aaa.cc.</mail>
    <minimum>1d</minimum>
    <refresh>3h</refresh>
    <retry>1h</retry>
    <serial>2005082300</serial>
    <server>aaa.aaa.cc.</server>
    <zone>@</zone>
  </soa>
  <soa_modified>1</soa_modified>
  <ttd>2d</ttd>
  <type>master</type>
  <update_actions config:type="list">
    <listentry>
      <key>mydom.uwe.</key>
      <operation>add</operation>
      <type>NS</type>
      <value>ns.mydom.uwe.</value>
    </listentry>
  </update_actions>
  <zone>mydom.uwe</zone>
</listentry>
</zones>
</dns-server>

```

4.38 DHCP 服务器

使用 `dhcp-server` 资源可以通过以下六个属性来配置 DHCP 服务器的所有设置。

元素	值	说明
<code>chroot</code>	0/1	值 1 表示必须将 DHCP 服务器监禁在 chroot 中。

元素	值	说明
<u>start_service</u>	0/1	将此元素设置为 1 会启用 DHCP 服务器（即，在系统启动时运行该服务器）。
<u>use_ldap</u>	0/1	如果设置为 1，则会在 LDAP 而不是本机配置文件中储存设置。
<u>other_options</u>	文本	包含启动时要传递给 DHCP 服务器可执行文件的参数的字符串。例如，使用 “-p 1234” 可侦听非标准 1234 端口。有关所有可能选项，请查阅 dhcpd 手册页。如果保留空白，将使用默认值。
<u>allowed_interfaces</u>	列表	DHCP 服务器将在其中操作的网卡的列表。有关确切的格式，请参见以下示例。
<u>settings</u>	列表	用于配置 DHCP 服务器行为的设置的列表。配置是在一个树形结构中定义的，根表示全局选项，其中嵌套子网和主机。 <u>children</u> 、 <u>parent_id</u> 和 <u>parent_type</u> 属性用于表示这种嵌套。有关确切的格式，请参见以下示例。

例 4.75：DHCP-SERVER 部分示例

```
<dhcp-server>
  <allowed_interfaces config:type="list">
    <allowed_interface>eth0</allowed_interface>
```

```

</allowed_interfaces>
<chroot>0</chroot>
<other_options>-p 9000</other_options>
<start_service>1</start_service>
<use_ldap>0</use_ldap>

<settings config:type="list">
  <settings_entry>
    <children config:type="list"/>
    <directives config:type="list">
      <listentry>
        <key>fixed-address</key>
        <type>directive</type>
        <value>192.168.0.10</value>
      </listentry>
      <listentry>
        <key>hardware</key>
        <type>directive</type>
        <value>ethernet d4:00:00:bf:00:00</value>
      </listentry>
    </directives>
    <id>static10</id>
    <options config:type="list"/>
    <parent_id>192.168.0.0 netmask 255.255.255.0</parent_id>
    <parent_type>subnet</parent_type>
    <type>host</type>
  </settings_entry>
  <settings_entry>
    <children config:type="list">
      <child>
        <id>static10</id>
        <type>host</type>
      </child>
    </children>
    <directives config:type="list">
      <listentry>
        <key>range</key>
        <type>directive</type>

```

```

        <value>dynamic-bootp 192.168.0.100 192.168.0.150</value>
    </listentry>
    <listentry>
        <key>default-lease-time</key>
        <type>directive</type>
        <value>14400</value>
    </listentry>
    <listentry>
        <key>max-lease-time</key>
        <type>directive</type>
        <value>86400</value>
    </listentry>
</directives>
<id>192.168.0.0 netmask 255.255.255.0</id>
<options config:type="list"/>
<parent_id/>
<parent_type/>
<type>subnet</type>
</settings_entry>
<settings_entry>
    <children config:type="list">
        <child>
            <id>192.168.0.0 netmask 255.255.255.0</id>
            <type>subnet</type>
        </child>
    </children>
    <directives config:type="list">
        <listentry>
            <key>ddns-update-style</key>
            <type>directive</type>
            <value>none</value>
        </listentry>
        <listentry>
            <key>default-lease-time</key>
            <type>directive</type>
            <value>14400</value>
        </listentry>
    </directives>


```



```
<id/>
<options config:type="list"/>
<parent_id/>
<parent_type/>
<type/>
</settings_entry>
</settings>
</dhcp-server>
```

4.39 防火墙配置

从 SUSE Linux Enterprise Server 15 GA 开始，SuSEfirewall2 已被 firewalld 取代。使用 SuSEfirewall2 属性的配置文件将被转换为 firewalld 配置文件。但是，并非所有配置文件属性都可转换。有关 firewalld 的细节，请参见《安全和强化指南》，第 23 章 “伪装和防火墙”，第 23.4 节 “firewalld”。

 **重要：**与基于 SuSEFirewall2 的配置文件的有限向后兼容性
系统仅部分支持使用基于 SuSEfirewall2 的配置文件，因为许多选项在 firewalld 中无效，并且某些缺失的配置可能会影响网络安全性。

4.39.1 一般防火墙配置

在 firewalld 中，一般配置只会公开少量的属性，大部分配置都是按区域进行。

属性	值	说明
<u>start_firewall</u>	布尔	是否在应用配置后立即启动 <u>firewalld</u> 。
<u>enable_firewall</u>	布尔	是否应在每次系统启动时启动 <u>firewalld</u> 。

属性	值	说明
<code>default_zone</code>	区域名	默认区域适用于未明确指派的任何设置。
<code>log_denied_packets</code>	要记录的丢包类型	启用记录所选类型的丢包。 值: <code>off</code> 、 <code>unicast</code> 、 <code>multicast</code> 、 <code>broad</code>
<code>name</code>	区域标识符	用于标识区域。如果区域未知，将创建一个新区域。
<code>short</code>	区域的简短摘要	简要概述区域的用途。对于已存在的区域，将忽略此属性。如果未指定，则使用名称。
<code>description</code>	区域的说明	描述区域的用途。对于已存在的区域，将忽略此属性。如果未指定，则使用名称。
<code>target</code>	默认操作	定义当没有匹配的规则时要在区域中执行的默认操作。可能的值为 <code>ACCEPT</code> 、 <code>%REJECT%%</code> 、 <code>DROP</code> 和 <code>default</code> 。如果未指定，则使用 <code>default</code> 。有关相应值的细节，请参见 https://firewalld.org/documentation/zone/options.html 。

4.39.2 防火墙区域配置

`firewalld` 的配置基于用于定义连接信任级别、接口或源地址的多个区域的存在性。可通过多种方式优化每个区域的行为，不过，并非所有属性都已公开。

属性	值	说明
<u>interfaces</u>	接口名称列表	指派到此区域的接口名称列表。接口或源只能属于一个区域。
<u>services</u>	服务列表	可在此区域中访问的服务列表。
<u>ports</u>	端口列表	要在指派区域中打开的单个端口或端口范围的列表。
<u>protocols</u>	协议列表	要在指派区域中打开的或者可在其中访问的协议列表。
<u>masquerade</u>	启用掩蔽	将在分配的区域中启用或禁用网络地址转换 (NAT)。

4.39.3 应用 firewalld 配置文件的安装阶段

从 SUSE Linux Enterprise Server 15 SP3 开始，通常在安装的第一个阶段结束时应用 firewalld 配置文件。（要了解有关安装阶段的信息，请参见第 1.2 节“概述和概念”）。但某些情况下是在第二个阶段应用配置文件。下面的列表指出了在第一个或第二个阶段应用 firewalld 配置文件的情况。

- 您运行的 AutoYaST 包含 firewalld 部分，并且您不是通过 SSH 或 VNC 安装 SUSE Linux Enterprise Server 的。在第一个阶段配置防火墙。
- 您运行的 AutoYaST 包含 firewalld 部分，您是通过 SSH 或 VNC 安装 SUSE Linux Enterprise Server 的，并且不需要第二个阶段。在第一个阶段配置防火墙。
- 您运行的 AutoYaST 包含 firewalld 部分，您是通过 SSH 或 VNC 安装 SUSE Linux Enterprise Server 的，并且需要第二个阶段。在第二个阶段配置防火墙。

- 您运行的 AutoYaST 不含 `firewalld` 部分。在第一个阶段根据默认产品建议配置防火墙。
- 您运行的 AutoYaST 包含或不包含 `firewall` 部分，并使用需要连接网络的自定义脚本。在第一个阶段根据配置文件或产品建议配置防火墙，并且必须调整防火墙配置，以便自定义脚本能够根据需要连接网络。

4.39.4 完整示例

下面是 `firewall` 部分的完整示例，其中包括一般属性和区域特定属性。

例 4.76：FIREWALL 部分示例

```
<firewall>
  <enable_firewall config:type="boolean">true</enable_firewall>
  <log_denied_packets>all</log_denied_packets>
  <default_zone>external</default_zone>
  <zones config:type="list">
    <zone>
      <name>public</name>
      <interfaces config:type="list">
        <interface>eth0</interface>
      </interfaces>
      <services config:type="list">
        <service>ssh</service>
        <service>dhcp</service>
        <service>dhcpv6</service>
        <service>samba</service>
        <service>vnc-server</service>
      </services>
      <ports config:type="list">
        <port>21/udp</port>
        <port>22/udp</port>
        <port>80/tcp</port>
        <port>443/tcp</port>
        <port>8080/tcp</port>
      </ports>
    </zone>
  </zones>
</firewall>
```

```
</zone>
<zone>
  <name>dmz</name>
  <interfaces config:type="list">
    <interface>eth1</interface>
  </interfaces>
</zone>
</zones>
</firewall>
```

4.40 其他硬件和系统组件

除了网络身份验证和安全性等核心组件配置外，AutoYaST 还提供多种硬件和系统配置选项，就如同手动安装以及交互式安装的任何系统上默认提供的选项一样。例如，您可以配置打印机、声音设备、电视卡，以及在 YaST 中具有相应模块的任何其他硬件组件。

添加到 YaST 的任何新配置选项将在 AutoYaST 中自动提供。

4.40.1 打印机

AutoYaST 对打印的支持仅限于用于定义如何在客户端上使用 CUPS 进行网络打印的基本设置。AutoYaST 不支持设置本地打印队列。新式打印机通常是通过 USB 连接的。CUPS 通过类似于 `usb://ACME/FunPrinter?serial=1a2b3c` 的型号特定设备 URI 访问 USB 打印机。通常无法提前预测正确的设备 URI，因为此 URI 是在运行时由 CUPS 后端 `usbusb` 确定的。因此，无法使用 AutoYaST 设置本地打印队列。

有关如何在客户端工作站上使用 CUPS 通过网络进行打印的基础知识：

在客户端工作站应用程序中，将打印作业提交给 CUPS 守护程序进程 (`cupsd`)。 `cupsd` 会将打印作业转发给网络中的 CUPS 打印服务器，打印作业将在该服务器中处理。该服务器会将打印机特定的数据发送到打印机设备。

如果网络中只有一台 CUPS 打印服务器，则无需在每个客户端工作站上运行 CUPS 守护程序。更简单的做法是在 `/etc/cups/client.conf` 中指定 CUPS 服务器并直接访问该服务器（只能设置一个 CUPS 服务器项）。在这种情况下，客户端工作站上运行的应用程序会直接将打印作业提交到指定的 CUPS 打印服务器。

例 4.77 “打印机配置” 演示了 `printer` 配置部分。`cupsd_conf_content` 项一字不差地包含了 `cupsd` 配置文件 `/etc/cups/cupsd.conf` 的整个内容。`client_conf_content` 项一字不差地包含了 `/etc/cups/client.conf` 的整个内容。`printer` 部分包含 `cupsd` 配置，但未指定是否应运行 `cupsd`。

例 4.77：打印机配置

```
<printer>
  <client_conf_content>
    <file_contents><![CDATA[
... verbatim content of /etc/cups/client.conf ...
]]></file_contents>
  </client_conf_content>
  <cupsd_conf_content>
    <file_contents><![CDATA[
... verbatim content of /etc/cups/cupsd.conf ...
]]></file_contents>
  </cupsd_conf_content>
</printer>
```



注意：/etc/cups/cups-files.conf

在版本 1.6 中，CUPS 配置文件已分为两个文件：`cupsd.conf` 和 `cups-files.conf`。从 SUSE Linux Enterprise Server 15 SP7 开始，AutoYaST 仅支持修改 `cupsd.conf`，因为对于通常的打印设置，`cups-files.conf` 中的默认设置已够用。

4.40.2 声音设备

下面演示了使用配置系统创建的声音配置示例。

例 4.78：声音配置

```
<sound>
  <autoinstall config:type="boolean">true</autoinstall>
  <modules_conf config:type="list">
```

```

<module_conf>
  <alias>snd-card-0</alias>
  <model>M5451, ALI</model>
  <module>snd-ali5451</module>
  <options>
    <snd_enable>1</snd_enable>
    <snd_index>0</snd_index>
    <snd_pcm_channels>32</snd_pcm_channels>
  </options>
</module_conf>
</modules_conf>
<volume_settings config:type="list">
  <listentry>
    <Master config:type="integer">75</Master>
  </listentry>
</volume_settings>
</sound>

```

4.41 导入 SSH 密钥和配置

YaST 允许从先前的安装导入 SSH 密钥和服务器配置。还可以通过 AutoYaST 配置文件控制此功能的行为。

例 4.79： 从 `/DEV/SDA2` 导入 SSH 密钥和配置

```

<ssh_import>
  <import config:type="boolean">true</import>
  <copy_config config:type="boolean">true</copy_config>
  <device>/dev/sda2</device>
</ssh_import>

```

属性	值	说明
<u>import</u>	true/false	将导入 SSH 密钥。如果设置为 <u>false</u> ，则不会导入任何内容。

属性	值	说明
<code>copy_config</code>	true/false	同时还会导入 SSH 服务器配置。如果 <code>import</code> 设置为 <code>false</code> ，则此设置不起作用。
<code>device</code>	分区	要从中导入密钥和配置的分区。如果未设置，则使用包含最近访问的密钥的分区。

4.42 配置管理

AutoYaST 允许将一部分配置委派给某个**配置管理工具**（例如 Salt）。AutoYaST 负责处理基本系统安装（分区、网络设置等），其余配置任务可委派给其他工具。



注意：仅支持 Salt

虽然本文档中提及了 Puppet，但只有 Salt 受支持。不过，我们欢迎您报告在使用 Puppet 过程中发现的任何问题。

AutoYaST 支持两种不同的方法：

- 使用配置管理服务器。在这种情况下，AutoYaST 将设置一个配置管理工具。此工具会连接到主服务器以获取配置系统的指令。
- 从其他位置（例如 HTTP 服务器，或 USB 记忆棒等闪存盘）获取配置，并以独立模式运行配置管理工具。

4.42.1 连接到配置管理服务器

当配置管理服务器（Salt 中的**主服务器**，以及 Puppet jargon）已准备就绪时，此方法特别有用。在这种情况下，最困难的环节可能是设置适当的身份验证机制。

Salt 和 Puppet 都支持以下身份验证方法：

- 即时手动身份验证。当 AutoYaST 启动客户端时，会生成一个新的身份验证请求。管理员可在服务器上手动接受此请求。AutoYaST 将重试连接。如果同时已接受密钥，则 AutoYaST 会继续安装。
- 使用预置种子密钥。请参见所选配置管理系统的文档来了解如何生成此类密钥。使用 `keys_url` 选项告知 AutoYaST 要在何处查找密钥。

在以下配置示例中，AutoYaST 将启动客户端来生成身份验证请求。它最多尝试连接三次，两次尝试之间等待 15 秒。

例 4.80：使用手动身份验证的客户端/服务器

```
<configuration_management>
  <type>salt</type>
  <master>my-salt-server.example.net</master>
  <auth_attempts config:type="integer">3</auth_attempts>
  <auth_time_out config:type="integer">15</auth_time_out>
</configuration_management>
```

但是，在以下示例中，AutoYaST 将从闪存盘（例如 USB 记忆棒）检索密钥，并使用这些密钥连接到主服务器。

例 4.81：使用预置种子密钥的客户端/服务器

```
<configuration_management>
  <type>salt</type>
  <master>my-salt-server.example.net</master>
  <keys_url>usb:/</keys_url>
</configuration_management>
```

下表汇总了这些方案支持的选项。

属性	值	说明
<code>type</code>	字符串	配置管理名称。目前仅支持 <code>salt</code> 。
<code>master</code>	字符串	配置管理服务器的主机名或 IP 地址。

属性	值	说明
<code>auth_attempts</code>	整数	尝试连接服务器的最大次数。默认值为三次尝试。
<code>auth_time_out</code>	整数	尝试连接服务器的间隔时间（以秒为单位）。默认值为 15 秒。
<code>keys_url</code>	所用密钥的 URL	包含 <code>default.key</code> 和 <code>default.pub</code> 文件的 HTTP 服务器、硬盘、闪存盘或类似设备的路径。配置管理主服务器必须知道此密钥。
<code>enable_services</code>	True/False	安装后在客户端启用配置管理服务。默认值为 <code>true</code> 。

4.42.2 以独立模式运行

对于简单的方案，不必要部署配置管理服务器。可以改为使用**独立**（或**无主服务器**）模式运行的 Salt 或 Puppet。

由于没有服务器，AutoYaST 需要知道从何处获取配置。请将配置置于 TAR 存档中并将其储存在任何位置（例如闪存盘、HTTP/HTTPS 服务器或 NFS/SMB 共享上）。

该 TAR 存档的布局必须与预期相同，即放在 Salt 服务器中的 `/srv` 下。这意味着，您需要将状态放入 `saltsalt` 目录，并将公式放入单独的 `formulas` 目录。

此外，还可以创建一个 `pillar` 目录用于包含 pillar 数据。或者，可以使用 `pillar_url` 选项在单独的 TAR 存档中提供这些数据。

例 4.82：独立模式

```
<configuration_management>
  <type>salt</type>
  <states_url>my-salt-server.example.net</states_url>
```

```
<pillar_url>my-salt-server.example.net</pillar_url>
</configuration_management>
```

属性	值	说明
<u>type</u>	字符串	配置管理名称。目前仅支持 <u>salt</u> 。
<u>states_url</u>	URL	Salt 状态 TAR 存档的位置。此存档可以包含公式和 pillar。文件必须位于 <u>salt</u> 目录。
<u>pillar_url</u>	URL	包含 pillar 的 TAR 存档的位置。
<u>modules_url</u>	URL	Puppet 模块的位置。

4.42.3 SUSE Multi-Linux Manager Salt 公式支持

AutoYaST 以独立模式运行时，可提供 SUSE Multi-Linux Manager Salt 公式支持。如果在状态 TAR 存档中找到了公式，AutoYaST 会显示一个屏幕，让用户选择和配置所要应用的公式。

请记住，此功能会导致 AutoYaST 无法执行无人照管的安装，因为 AutoYaST 在此情况下会等待用户的输入。

III 使用动态配置文件管理批量安装

5 支持的动态配置文件方法 220

6 规则和类 221

7 ERB 模板 237

8 合并 ERB 模板和脚本 246

5 支持的动态配置文件方法

处理多个系统的安装时，使用可针对每个系统自动调整的单个配置文件（或其中一部分）可能会有帮助。在这方面，AutoYaST 提供了三种不同的机制，用于在安装时修改配置文件。

规则和类

规则和类可在安装期间合并多个控制文件，从而提供配置系统的可能性。您可以参见第 6 章“规则和类”了解有关此功能的详细信息。

ERB 模板

AutoYaST 支持 Embedded Ruby (ERB) 模板语法，可在安装期间修改配置文件的内容。第 7 章“ERB 模板”介绍了如何使用 ERB 模板。

安装前脚本

可以使用安装前脚本在安装期间修改配置文件，甚至创建全新的配置文件。第 4.32.1 节“前脚本”介绍了如何利用这些脚本。

安装期间询问用户

作为替代方案，AutoYaST 可在运行时询问用户要在配置文件中使用的值。在此情况下，安装并非完全无人照管，但使用此方法可以设置用户名、口令、IP 地址等。第 4.35 节“安装期间要求用户提供值”一节中提供了有关此功能的详细信息。

6 规则和类

使用规则和类可通过不同的方式自定义计算机集的安装：

- 使用规则可以根据系统的属性配置系统。
- 类代表目标系统组的配置。类可指派到系统。



注意：仅使用 `autoyast` 引导选项

只有引导参数 `autoyast=URL` 支持规则和类。

不支持 `autoyast2=URL`，因为此选项只下载单个 AutoYaST 控制文件。

6.1 基于规则的自动安装

使用规则可以根据系统属性，通过在安装期间合并多个控制文件来配置系统。基于规则的安装由规则文件控制。

例如，使用这种方法可以一次性在两个部门安装系统。假设存在这样的场景：部门 A 的计算机需安装为办公桌面，而部门 B 的计算机需安装为开发人员工作站。您需要创建包含两种不同规则的规则文件。对于每个规则，可以使用不同的系统参数将两项安装区分开来。每个规则还包含每个部门的相应配置文件的链接。

规则文件是一个 XML 文件，包含您想要自动安装的每组系统（或单个系统）的规则。一组规则可以根据一个或多个系统属性来区分系统组。通过所有规则后，每组系统将链接到一个控制文件。规则文件和控制文件都必须位于预定义且可访问的位置。

仅当未使用 `autoyast` 关键字提供特定的控制文件时，才会检索规则文件。例如，如果使用以下命令，将不会评估规则文件：

```
autoyast=http://10.10.0.1/profile/myprofile.xml
autoyast=http://10.10.0.1/profile/rules/rules.xml
```

请改用：

```
autoyast=http://10.10.0.1/profile/
```

此命令会加载 <http://10.10.0.1/profile/rules/rules.xml>（目录名称末尾的斜线非常重要）。

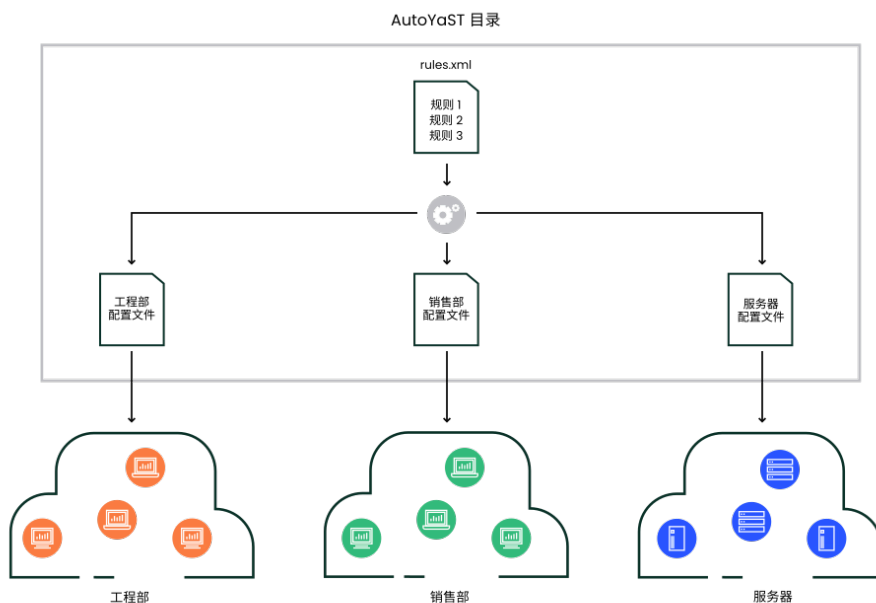


图 6.1：规则

如果应用了多个规则，将使用合并脚本即时生成每个组的最终控制文件。合并进程以规则和最新规则覆盖早期规则中配置数据的顺序为基础。请注意，合并的 XML 文件中前几个部分的名称需遵循字母顺序，这样才能成功合并。

规则文件的用法是可选的。如果未找到规则文件，将通过使用提供的控制文件或根据系统的 MAC 或 IP 地址搜索控制文件的标准方式来继续系统安装。

6.1.1 规则文件说明

例 6.1：简单规则文件

以下简单示例说明如何使用规则文件检索使用已知硬件的客户端的配置。

```
<?xml version="1.0"?>
<!DOCTYPE autoinstall>
<autoinstall xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://
www.suse.com/1.0/configs">
  <rules config:type="list">
```

```

<rule>
  <disksize>
    <match>/dev/sdc 1000</match>
    <match_type>greater</match_type>
  </disksize>
  <result>
    <profile>department_a.xml</profile>
    <continue config:type="boolean">false</continue>
  </result>
</rule>
<rule>
  <disksize>
    <match>/dev/sda 1000</match>
    <match_type>greater</match_type>
  </disksize>
  <result>
    <profile>department_b.xml</profile>
    <continue config:type="boolean">false</continue>
  </result>
</rule>
</rules>
</autoinstall>

```

最后一个示例定义两个规则，并为每个规则提供了不同的控制文件。本例中使用的规则是 `disksize`。分析规则文件后，YaST 会尝试将目标系统与 `rules.xml` 文件中的规则进行匹配。如果目标系统与规则中定义的所有系统属性相匹配，即表示规则匹配。如果系统与规则匹配，则相应的资源将被添加到由 AutoYaST 用来创建最终控制文件的控制文件堆栈。`continue` 属性会告知 AutoYaST 在找到匹配项后其是否应继续处理其他规则。

如果第一个规则不匹配，将检查列表中的下一个规则，直到发现匹配为止。

使用 `disksize` 属性可为使用大小不同硬盘的系统提供不同的配置。第一个规则使用 `match` 属性检查设备 `/dev/sdc` 是否可用，以及其大小是否超过 1 GB。

一个规则必须至少包含一个要匹配的属性。如果您需要检查多个属性（例如内存或体系结构），可按下一个示例所示在规则资源中添加更多属性。

例 6.2：简单规则文件

以下示例说明如何使用规则文件检索使用已知硬件的客户端的配置。


```

<?xml version="1.0"?>
<!DOCTYPE autoinstall>
<autoinstall xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://
www.suse.com/1.0/configns">
  <rules config:type="list">
    <rule>
      <disksize>
        <match>/dev/sdc 1000</match>
        <match_type>greater</match_type>
      </disksize>
      <memsize>
        <match>1000</match>
        <match_type>greater</match_type>
      </memsize>
      <result>
        <profile>department_a.xml</profile>
        <continue config:type="boolean">false</continue>
      </result>
    </rule>
    <rule>
      <disksize>
        <match>/dev/sda 1000</match>
        <match_type>greater</match_type>
      </disksize>
      <memsize>
        <match>256</match>
        <match_type>greater</match_type>
      </memsize>
      <result>
        <profile>department_b.xml</profile>
        <continue config:type="boolean">false</continue>
      </result>
    </rule>
  </rules>
</autoinstall>

```

规则目录必须位于引导时通过 `autoyast` 关键字指定的同一目录中。如果客户端是使用 `autoyast=http://10.10.0.1/profiles/` 引导的，则 AutoYaST 将在 `http://10.10.0.1/profiles/rules/rules.xml` 位置搜索规则文件。

6.1.2 自定义规则

如果 AutoYaST 为规则提供的属性不足以达到您的目的，请使用自定义规则。自定义规则包含一个外壳脚本。可以评估该脚本的输出（将忽略 `STDOUT`、`STDERR`）。

下面是自定义规则的用法示例：

```
<rule>
  <custom1>
    <script>
if grep -i intel /proc/cpuinfo > /dev/null; then
echo -n "intel"
else
echo -n "non_intel"
fi;
    </script>
    <match>*</match>
    <match_type>exact</match_type>
  </custom1>
  <result>
    <profile>@custom1.xml</profile>
    <continue config:type="boolean">true</continue>
  </result>
</rule>
```

此规则中的脚本可针对 `STDOUT` 回显 `intel` 或 `non_intel`（在本例中，`grep` 命令的输出必须定向到 `/dev/null`）。将在两个“@”字符之间填充规则脚本的输出，以确定要提取的控制文件的文件名。AutoYaST 将读取该输出，并提取名为 `intel.xml` 或 `non_intel.xml` 的文件。此文件可以包含用于选择软件的 AutoYaST 配置文件部分；例如，如果您想在 Intel 硬件上（而不是其他硬件上）选择不同的软件。

自定义规则的数量限于 5 个。因此您可以使用 `custom1` 到 `custom5`。

6.1.3 规则的匹配类型

可以使用五种不同的匹配类型：

- exact（默认）
- greater
- lower
- range
- regex（类似于 Bash 中的简单 =~ 运算符）

如果使用 exact，则必须与指定字符串完全匹配。regex 可用于匹配子字符串，例如，ntel 将匹配 Intel、intel 和 intelligent。greater 和 lower 可用于 memsize 或 totaldisk 等。它们只能匹配返回整数值的规则。范围也仅适用于整数值且格式为 value1-value2，例如 512-1024。

6.1.4 合并属性

可以通过逻辑运算符合并多个属性。如果 disksize 大于 1GB 或 memsize 正好为 512MB，则可以让规则匹配。

为此，可以在 rules.xml 文件中使用 operator 元素。可能的运算符包括 and 和 or，and 是默认运算符。示例如下：

```
<rule>
  <disksize>
    <match>/dev/sda 1000</match>
    <match_type>greater</match_type>
  </disksize>
  <memsize>
    <match>256</match>
    <match_type>greater</match_type>
  </memsize>
  <result>
    <profile>machine2.xml</profile>
    <continue config:type="boolean">false</continue>
  </result>
</rule>
```

```
</result>
<operator>or</operator>
</rule>
```

6.1.5 规则文件结构

rules.xml 文件需要：

- 至少包含一个规则；
- 名为 rules.xml；
- 位于配置文件储存库的 rules 目录中；
- 至少包含一个要在规则中匹配的属性。

6.1.6 预定义的系统属性

下表列出了您可以在规则文件中匹配的预定义系统属性。

如果您不确定系统上的某个值，请运行 `/sbin/yast2 ayast_probe ncurses`。可以在显示检测到的值的文本框中滚动浏览。请注意，当另一个需要锁定的 YaST 进程（例如安装程序）正在运行时，此命令将无效。因此，在安装期间无法运行此命令。

表 6.1：系统属性

属性	值	说明
<u>hostaddress</u>	主机的 IP 地址	此属性必须始终完全匹配。
<u>hostname</u>	主机的名称	此属性必须始终完全匹配。
<u>domain</u>	主机的域名	此属性必须始终完全匹配。
<u>installed_product</u>	要安装的产品名称。	此属性必须始终完全匹配。
<u>installed_product_version</u>	要安装的产品版本。	此属性必须始终完全匹配。
<u>network</u>	主机的网络地址	此属性必须始终完全匹配。

属性	值	说明
<u>mac</u>	主机的 MAC 地址	此属性必须始终完全匹配（MAC 地址应采用 <u>0080c8f6484c</u> 格式）。
<u>linux</u>	在系统上安装的 Linux 分区数	此属性可为 0 或更大。
<u>others</u>	在系统上安装的非 Linux 分区数	此属性可为 0 或更大。
<u>xserver</u>	图形适配器所需的 X 服务器	此属性必须始终完全匹配。
<u>memsize</u>	主机上可用的内存，以 MB 为单位	可以使用所有匹配类型。
<u>totaldisk</u>	主机上可用的总磁盘空间，以 MB 为单位	可以使用所有匹配类型。
<u>hostid</u>	IP 地址的十六进制表示形式	必须完全匹配
<u>arch</u>	主机的体系结构	必须完全匹配
<u>karch</u>	主机的内核体系结构（例如 SMP 内核、Xen 内核）	必须完全匹配
<u>disksize</u>	驱动器设备和大小（以 MB 为单位）	可以使用所有匹配类型。
<u>product</u>	SMBIOS 中指定的硬件产品名称	必须完全匹配

属性	值	说明
<u>product_vendor</u>	SMBIOS 中指定的硬件供应商	必须完全匹配
<u>board</u>	SMBIOS 中指定的系统主板名称	必须完全匹配
<u>board_vendor</u>	SMBIOS 中指定的系统主板供应商	必须完全匹配
<u>custom1-5</u>	使用外壳脚本的自定义规则	可以使用所有匹配类型。

6.1.7 使用对话框选择规则

您可以使用包含复选框的弹出对话框来选择您要匹配的规则。

下面列出的元素必须放在 rules.xml 文件中的以下 XML 结构内：

```
<rules config:type="list">
  <rule>
    <dialog>
      ...
    </dialog>
  </rule>
</rules>
```

属性、值、描述

dialog_nr

具有相同 dialog_nr 的所有规则将出现在同一个弹出对话框中。相同的 dialog_nr 可以显示在多个规则中。

```
<dialog_nr config:type="integer">3</dialog_nr>
```

此元素是可选的，缺少的 dialog_nr 的默认值始终为 0。要为所有规则使用一个弹出对话框，无需指定 dialog_nr。

element

指定唯一 ID。即使您有多个对话框，也不能两次使用同一 ID。不支持对对话框 1 和对话框 2 都使用 ID `11`。（此行为与 `ask` 对话框相反，后者可以对多个对话框使用同一 ID。）

```
<element config:type="integer">3</element>
```

可选。如果省略，AutoYaST 将在内部添加自己的 ID。这样，您就无法指定有冲突的规则（参见下文）。

title

弹出对话框的标题

```
<title>Desktop Selection</title>
```

可选

question

在弹出对话框中的复选框后显示的问题。

```
<question>GNOME Desktop</question>
```

可选。如果您在此处未配置文本，将改为显示此规则触发的 XML 文件的名称。

timeout

以秒为单位的超时，在此时间过后，对话框会自动“按下”确定按钮。适用于结合使用规则对话框的无阻碍式安装。

```
<timeout config:type="integer">30</timeout>
```

可选。如果不指定超时，则在用户确认对话框之前，安装进程会一直处于停止状态。

conflicts

与此规则冲突的元素 ID（规则）列表。如果此规则匹配或者已由用户选中，将在弹出对话框中取消选择并禁用所有有冲突的规则。请小心不要造成死锁。

```
<conflicts config:type="list">
  <element config:type="integer">1</element>
  <element config:type="integer">5</element>
</conflicts>
```

```
...  
</conflicts>
```

可选

以下示例演示如何使用包含规则的对话框：

```
<rules config:type="list">  
  <rule>  
    <custom1>  
      <script>  
echo -n 100  
      </script>  
      <match>100</match>  
      <match_type>exact</match_type>  
    </custom1>  
    <result>  
      <profile>rules/gnome.xml</profile>  
      <continue config:type="boolean">>true</continue>  
    </result>  
    <dialog>  
      <element config:type="integer">0</element>  
      <question>GNOME Desktop</question>  
      <title>Desktop Selection</title>  
      <conflicts config:type="list">  
        <element config:type="integer">1</element>  
      </conflicts>  
      <dialog_nr config:type="integer">0</dialog_nr>  
    </dialog>  
  </rule>  
  <rule>  
    <custom1>  
      <script>  
echo -n 100  
      </script>  
      <match>101</match>  
      <match_type>exact</match_type>  
    </custom1>  
    <result>
```



```

    <profile>rules/gnome.xml</profile>
    <continue config:type="boolean">true</continue>
</result>
<dialog>
    <element config:type="integer">1</element>
    <dialog_nr config:type="integer">0</dialog_nr>
    <question>Gnome Desktop</question>
    <conflicts config:type="list">
        <element config:type="integer">0</element>
    </conflicts>
</dialog>
</rule>
<rule>
    <custom1>
        <script>
echo -n 100
        </script>
        <match>100</match>
        <match_type>exact</match_type>
    </custom1>
    <result>
        <profile>rules/all_the_rest.xml</profile>
        <continue config:type="boolean">>false</continue>
    </result>
</rule>
</rules>

```

6.2 类别

类代表目标系统组的配置。与规则不同，类需要在控制文件中进行配置。然后，可将类指派到目标系统。

下面是一个类定义示例：

```

<classes config:type="list">
    <class>
        <class_name>TrainingRoom</class_name>
        <configuration>Software.xml</configuration>
    </class>
</classes>

```

```
</class>
</classes>
```

在上面的示例中，`Software.xml` 文件必须放在 `classes/TrainingRoom/` 子目录中。系统会从 AutoYaST 控制文件和规则所在的相同位置提取该文件。

如果您有多个控制文件，并且这些控制文件共享某些部分，则最好是对公用部分使用类。您也可以使用 XIncludes。

使用配置管理系统可以定义一组类。类定义包括以下变量：

- 名称：类名
- 说明：
- 顺序：该类在迁移堆栈中的顺序（或优先级）

类名	顺序	配置
Department	1	0
Group	2	0
Site	3	0

Department Settings

新建(W) 编辑(E) 删除(D)

帮助(H) 后退(B) 完成(F)

图 6.2：定义类

您可以根据需要创建多个类别，但建议尽量保持小规模类别组从而确保配置系统的简洁。例如，可使用以下一组类：

- site：描述实际位置或站点的类；
- machine：描述计算机类型的类；

- role: 描述计算机功能的类;
- group: 描述某个站点或位置中某个部门或组的类。

在类目录中保存的文件可以采用与常规控制文件相同的语法和格式，只是表示为配置的子集。例如，要为具有特定网络接口的计算机创建新的控制文件，只需使用用于控制网络配置的控制文件资源。如果具有多个网络类型，可将特殊硬件类型所需的网络类型与其他类文件合并，然后创建适合所要安装的系统的新控制文件。

6.3 混合规则和类

可以在自动安装期间混合规则和类别。例如您可以用包含类定义的规则来确定系统。图 A1 “规则检索过程” 中描述了该过程。

检索并合并规则后，将在生成的控制文件中分析并检查类定义。如果定义了类别，那么将从原始储存库中检索类文件并启动新的合并进程。

6.4 合并规则和类

通过类和规则，多个 XML 文件将合并成一个最终的 XML 文件。此合并过程常常让人感到迷惑，因为它的行为与期望大相径庭。首先请务必注意，合并的 XML 文件中前几个部分的名称必须遵循字母顺序，这样才能成功合并。

例如，应该合并以下两个 XML 部分：

```
<partitioning config:type="list">
  <drive>
    <partitions config:type="list">
      <partition>
        <filesystem config:type="symbol">swap</filesystem>
        <format config:type="boolean">true</format>
        <mount>swap</mount>
        <partition_id config:type="integer">130</partition_id>
        <size>2000mb</size>
      </partition>
      <partition>
```

```

    <filesystem config:type="symbol">xfs</filesystem>
    <partition_type>primary</partition_type>
    <size>4Gb</size>
    <mount>/data</mount>
  </partition>
</partitions>
</drive>
</partitioning>

```

```

<partitioning config:type="list">
  <drive>
    <initialize config:type="boolean">>false</initialize>
    <partitions config:type="list">
      <partition>
        <format config:type="boolean">>true</format>
        <filesystem config:type="symbol">xfs</filesystem>
        <mount>/</mount>
        <partition_id config:type="integer">131</partition_id>
        <partition_type>primary</partition_type>
        <size>max</size>
      </partition>
    </partitions>
    <use>all</use>
  </drive>
</partitioning>

```

您可能预期控制文件将包含三个分区。事实并非如此。您最终会得到两个分区，第一个分区是由交换分区和根分区混合构成的。系统将从第二个文件使用这两个分区中配置的设置（例如 `mount` 或 `size`）。仅存在于第一或第二个分区中的设置也将复制到合并后的分区。

在此示例中，您不想要第二个 `drive`。两个驱动器应合并成一个。对于分区，应定义三个独立的分区。使用 `dont_merge` 方法可解决合并问题：

```

<classes config:type="list">
  <class>
    <class_name>swap</class_name>
    <configuration>largeswap.xml</configuration>
    <dont_merge config:type="list">
      <element>partition</element>
    </dont_merge>
  </class>
</classes>

```

```
    </dont_merge>
  </class>
</classes>
```

```
<rule>
  <board_vendor>
    <match>intel</match>
    <match_type>regex</match_type>
  </board_vendor>
  <result>
    <profile>classes/largeswap.xml</profile>
    <continue config:type="boolean">true</continue>
    <dont_merge config:type="list">
      <element>partition</element>
    </dont_merge>
  </result>
  <board_vendor>
    <match>PowerEdge [12]850</match>
    <match_type>regex</match_type>
  </board_vendor>
  <result>
    <profile>classes/smallswap.xml</profile>
    <continue config:type="boolean">true</continue>
    <dont_merge config:type="list">
      <element>partition</element>
    </dont_merge>
  </result>
</rule>
```

7 ERB 模板

ERB 模板用于在 AutoYaST 配置文件内嵌入 Ruby 代码，以在安装期间修改配置文件。利用此方法，您可以通过设置值、添加或跳过某些部分等方式，来检查系统并调整配置文件。

要激活 ERB 处理，配置文件的扩展名必须为 `.erb`（例如 `autoyast.xml.erb`）。之后，它将无法合并规则/类和 ERB 模板。

7.1 什么是 ERB?

ERB 即嵌入式 Ruby (**Embedded Ruby**)。ERB 使用 Ruby 编程语言的功能来生成不同种类的内容。利用 ERB，您可以在配置文件中包含一些 Ruby 代码，以在运行时根据安装系统调整配置文件。

使用 ERB 时，需要将 Ruby 代码包含在 `<%` 和 `%>` 符号之间。使用等号 `=` 可在最终的配置文件中包含命令输出。

例 7.1：使用 ERB 包含文件

```
<bootstrap>
  <% require "open-uri" %>
  <%= URI.open("http://192.168.1.1/profiles/bootstrap-common.xml").read %>
</bootstrap> <!-- this line gets replaced with the content of bootstrap-
common.xml -->
```

可以利用 Ruby 便利性来运行任意命令。如果想获得命令的输出，请将其包含在两个反引号之中。如果想知道命令成功与否，请使用 `system` 函数运行该命令。

例 7.2：使用 RUBY 运行命令

```
<% files = `ls` %> <!-- files contains the output of the command (for instance
"file1\nfile2\nfile3") -->
<% success = system("dmidecode | grep some-model") %> <!-- success contains true
or false -->
```

此外，您还可使用更高级的 Ruby 代码结构（例如条件和循环）。

例 7.3：使用 RUBY 结构

```
<% ip_forward = File.read("/proc/sys/net/ipv4/ip_forward").strip %>
```

```
<% if ip_forward == "1" %>
  <!-- something -->
<% end %>

<% files = `ls /tmp/config/*.xml` %>
<% files.split.each do |file| %>
  <%= file.read %>
<% end %>
```

AutoYaST 提供了一些**帮助程序函数**，用于从底层系统中检索信息，例如 `disks` 或 `network_cards`。您可以在第 7.2 节“模板帮助程序”中查看帮助程序及值列表。

7.2 模板帮助程序

模板帮助程序是 Ruby 方法的集合，可用于配置文件中以检索安装系统的相关信息。

7.2.1 `boot_efi?`

`boot_efi?` 是布尔帮助程序，将返回系统是否使用 EFI 引导的信息。在下面的示例中，配置文件会根据当前引导模式配置引导加载程序。

例 7.4：配置引导加载程序

```
<% if env.boot_efi? %>
  <loader_type>grub2-efi</loader_type>
<% else %>
  <loader_type>grub2</loader_type>
<% end %>
```

7.2.2 `disks`

`disks` 帮助程序可返回检测到的磁盘列表。列表的每个元素包含一些基本信息，例如设备名称或大小。

键	类型	值
<code>:device</code>	字符串	设备内核名称（例如 <code>sda</code> ）。
<code>:model</code>	字符串	磁盘型号
<code>:serial</code>	字符串	序列号
<code>:size</code>	整数	磁盘大小（磁盘扇区数）
<code>:udev_names</code>	字符串数组	磁盘 udev 名称列表。您可以使用其中的任何名称来表示设备。
<code>:vendor</code>	字符串	磁盘供应商名称

以下示例中的配置文件会在最大的磁盘上安装系统。它会按大小对现有磁盘列表排序，然后使用最后一个磁盘。之后，它会使用 `:device` 键作为 `device` 元素的值。

例 7.5：使用最大的磁盘

```
<partitioning t="list">
  <drive>
    <% disk = disks.sort_by { |d| d[:size] }.last %> <!-- find the largest disk -->
    <device><%= disk[:device] %></device> <!-- print the disk device name -->
    <initialize t="boolean">true</initialize>
    <use>all</use>
  </drive>
</partitioning>
```

7.2.3 network_cards

`network_cards` 帮助程序可返回网卡列表，包括网卡名称，状态信息（例如，它们是否已连接）。

键	类型	值
<code>:device</code>	字符串	设备名称（例如 <code>eth0</code> 或 <code>enp3s0</code> ）
<code>:mac</code>	字符串	MAC 地址
<code>:active</code>	布尔	设备是否处于活动状态
<code>:link</code>	布尔	设备是否已连接
<code>:vendor</code>	字符串	磁盘供应商名称

下面的示例会找到最先连接网络的网卡，并将其配置为使用 DHCP。

例 7.6：配置连接的网卡

```
<interfaces t="list">
  <% with_link = network_cards.sort_by { |n| n[:name] }.find { |n| n[:link] } %>
  <% if with_link %>
    <interface>
      <device><%= with_link[:device] %></device>
      <startmode>auto</startmode>
      <bootproto>dhcp</bootproto>
    </interface>
  <% end %>
</interfaces>
```

7.2.4 os_release

`os_release` 帮助程序返回操作系统信息，这些信息包含在 `/etc/os-release` 文件中。

键	类型	值
<code>:id</code>	字符串	发行套件 ID（例如 <code>sles</code> 、 <code>opensuse-tumbleweed</code> ）

键	类型	值
<code>:name</code>	字符串	发行套件名称（例如 <code>SLES</code> 或 <code>openSUSE Tumbleweed</code> ）
<code>:version</code>	字符串	发行套件版本（例如 <code>15.2</code> ）

您可以使用这些信息决定要安装的产品，对所有发行套件（SLE 或 openSUSE 发行套件）使用几乎相同的配置文件。

例 7.7：对不同的发行套件重复使用相同的配置文件

```
<products t="list">
  <% if os_release[:id] == 'sle' %>
    <product>SLES</product>
  <% else %>
    <product>openSUSE</product>
  <% end %>
</products>
```

7.2.5 hardware

`hardware` 帮助程序提供其他硬件信息。它会返回 `hwinfo` 命令返回的所有信息。当通过所述帮助程序获取的信息不足时，您可以使用此帮助程序作为后备手段。下一个示例中使用 `hardware` 帮助程序来过滤 USB 设备。请查看 [第 7.3 节 “运行 ERB 帮助程序”](#) 了解如何检查 `hardware` 帮助程序提供的所有信息。

例 7.8：过滤 USB 设备

```
<% usb_disks = hardware["disk"].select { |d| d["driver"] != "usb-storage" } %>
```

7.3 运行 ERB 帮助程序

您可以使用 Ruby 控制台运行 AutoYaST ERB 帮助程序，并了解它们可提供哪些功能。可以通过 `Y2Autoinstallation::Y2ERB::TemplateEnvironment` 类的实例访问所有 ERB 帮助程序。以 root 身份运行 `irb -ryast -rautoinstall/y2erb` 启动 Ruby 交互式解释器。

例 7.9：运行帮助程序

```
irb > env = Y2Autoinstallation::Y2ERB::TemplateEnvironment.new # the env
variable gives access to the helpers

irb > env.disks
=>
[{:vendor=>"WDC", :device=>"sda", ...},
 {:vendor=>"TOSHIBA", :device=>"sdb", ...},
 ...]

irb > env.hardware.keys
=>
["architecture",
 "bios",
 "bios_video",
 ...]

irb > env.hardware["architecture"]
=>
"x86_64"
```

7.4 呈现 ERB 配置文件

AutoYaST 命令行提供了 `check-profile` 命令，可用于基于 ERB 文件生成配置文件。此命令会要求 AutoYaST 分析、运行 ERB 代码，并生成最终的配置文件。您可以检查呈现的配置文件，确定所有功能均按预期工作。请查看命令帮助了解该命令支持的所有选项：`autoyast check-profile --help`。在下面的示例中，`check-profile` 要求 AutoYaST 下载并分析配置文件，解释 ERB 并运行前脚本。结果将转储到 `result.xml` 文件中。

例 7.10：呈现配置文件

```
> sudo yast2 autoyast check-profile filename=http://192.168.1.100/autoinst.erb
output=result.xml run-scripts=true run-erb=true
```



警告：check-profile 权限

大多数情况下，`check-profile` 都需要 root 权限，因此在以 root 身份运行安装前脚本和 ERB 配置文件时请务必保持谨慎。请仅使用您信任的配置文件。

7.5 调试 ERB 配置文件

在有些情况下，您需要停止 ERB 评估并检查发生了什么情况，为此 YaST 提供了与 `byebug` 调试器的集成。安装 `rubygem(byebug)` 软件包并将 `Y2DEBUGGER` 环境变量设为 1。

例 7.11：准备调试环境

```
> sudo zypper --non-interactive in "rubygem(byebug)"
> sudo Y2DEBUGGER=1 yast2 autoyast check-profile ...
```

在想要停止的位置添加断点就像添加 `<% byebug %>` 一样容易。有关 `byebug` 的详细信息，请参见 <https://github.com/deivid-rodriguez/byebug>。

例 7.12：添加断点

```
<% byebug %>
<% if system("dmidecode | grep some-model") %>
  <!-- do something -->
%<% end %>
```

7.6 ERB 与规则和类的对比

尽管 ERB 和规则/类都能实现动态生成配置文件的功能，但一般而言，ERB 配置文件更容易阅读和理解。一个重要的不同之处是规则和类可以合并配置文件，而 ERB 则不能。请参见第 6 章“规则和类”了解有关合并配置文件的更多信息。另一方面，ERB 具有高级语言 Ruby 的所有优势。我们来看一个使用了两种方式的示例。在下面的示例中，我们要将 `/home` 目录放在 `/dev/sdb` 中（如果该目录存在）。

例 7.13：规则和类

```
<rule>
```

```

    <custom1>
      <script>
if blkid | grep /dev/sdb > /dev/null; then
echo -n "yes"
else
echo -n "no"
fi;
      </script>
      <match>yes</match>
      <match_type>exact</match_type>
    </custom1>
    <result>
      <profile>classes/sdb_home.xml</profile>
      <dont_merge config:type="list">
        <element>partition</element>
      </dont_merge>
    </result>
  </rule>

```

例 7.14：ERB

```

<% home_in_sdb = disks.map { |d| d[:device] }.include?("sdb") %>

<partitioning config:type="list">
  <drive>
    ...
  </drive>
  <% if home_in_sdb %>
  <drive>
    <device>/dev/sdb</device>
    <disklabel>none</disklabel>
    <partitions t="list">
      <partition>
        <format t="boolean">true</format>
        <filesystem t="symbol">xfs</filesystem>
        <mount>/home</mount>
      </partition>
    </partitions>
  </drive>

```

```
<% end %>  
</partitioning>
```

8 合并 ERB 模板和脚本

第 4.32.1 节 “前脚本” 已介绍了如何使用前脚本修改当前配置文件。简而言之，如果该脚本创建了 `/tmp/profile/modified.xml` 文件，则 AutoYaST 会导入该配置文件并忽略最初的那个文件。

这种方法非常灵活，唯一的局限是，您需要依赖于安装媒体中提供的语言和库。

8.1 在脚本中嵌入 ERB

ERB 模板可以与脚本合并，这一点与**规则**不同。AutoYaST 将会在运行脚本前评估您在脚本中包含的任何 ERB 标记。此行为仅适用于在配置文件内定义的脚本，不适用于外部脚本。

下面的示例中的脚本会下载一个名称基于 MAC 地址的配置文件。将该文件保存为 `/tmp/profile/modified.xml` 会使 AutoYaST 下载并使用下载的配置文件。

例 8.1：使用 MAC 地址获取配置文件

```
<scripts>
  <pre-scripts config:type="list">
    <script>
      <interpreter>shell</interpreter>
      <filename>load_profile.sh</filename>
      <% mac = network_cards.first >
      <source><![CDATA[#!/bin/bash
wget -O /tmp/profile/modified.xml http://myserver/<%= network_cards.first[:mac]
%>.xml
]]></source>
    </script>
  </pre-scripts>
</scripts>
```

8.2 通过 Ruby 脚本访问 ERB 帮助程序

可以在 Ruby 脚本中使用 ERB 帮助程序要使用这些帮助程序，您需要 **require yast** 和 **autoinstall/y2erb** 库，并使用 `Y2Autoinstall::Y2ERB::TemplateEnvironment` 类访问它们。

例 8.2：通过 RUBY 脚本访问 ERB 帮助程序

```
<scripts>
  <pre-scripts config:type="list">
    <script>
      <interpreter>/usr/bin/ruby</interpreter>
      <filename>load_profile.rb</filename>
      <source><![CDATA[#!/usr/bin/env ruby
require "yast"
require "autoinstall/y2erb"
helpers = Y2Autoinstallation::Y2ERB::TemplateEnvironment.new
# Now you can use the TemplateEnvironment instance to call the helpers
disk_devices = helpers.disks.map { |d| d[:device] }
File.write("/root/disks.txt", disk_devices.join("\n"))
]]></source>
    </script>
  </pre-scripts>
</scripts>
```


IV 了解自动安装过程

9 自动安装过程 249

9 自动安装过程

9.1 简介

当系统引导进入自动安装并已检索到控制文件后，YaST 将根据控制文件中提供的信息配置系统。所有配置设置都汇总在一个窗口中，默认会显示该窗口。但如果需要全自动化安装，则应停用该窗口。

当 YaST 显示配置摘要时，只表示 YaST 探测到了硬件，并已准备好自动安装系统。此时系统中尚未发生任何更改。如果出现任何错误，您仍可以中止该过程。

无需任何图形适配器或监视器也可自动安装系统。不过，还是建议将监视器挂接到客户端计算机，以便可以监督安装过程并在出错时获得反馈。在图形界面与基于文本的 Ncurses 界面之间进行选择。对于无头客户端，可以使用串行控制台监控系统消息。

9.1.1 X11 界面（图形）

这是自动安装时使用的默认界面无需指定特殊变量即可激活此界面。

9.1.2 串行控制台

通过将关键字 `console`（例如 `console=ttyS0`）添加到内核的命令行，便可开始使用串行控制台安装系统。这会以控制台模式启动 `linuxrc`，然后以串行控制台模式启动 YaST。

9.1.3 基于文本的 YaST 安装

您也可以在命令行上激活此选项。要以文本模式启动 YaST，请在命令行上添加 `textmode=1`。安装小于 64 MB 的客户端或者不应配置 X11 时（尤其是在无头计算机上），建议以文本模式启动 YaST。

9.2 选择合适的引导媒体

可通过不同的方法引导客户端。计算机可从其网络接口卡 (NIC) 引导，以通过 DHCP 或 TFTP 接收引导映像。或者，也可以从闪存盘（例如 USB 记忆棒）或可引导的 DVD-ROM 加载适当的内核和 initrd 映像。

YaST 在启动时会检查引导媒体或 initrd 根目录中的 `autoinst.xml`，如果找到该文件，则切换到自动安装。如果控制文件使用了不同的名称或位于其他位置，请在内核命令行上使用参数 `AutoYaST=URL` 指定其位置。

或者，您也可以将 `autoinst.xml` 放在标记为 `OEMDRV` 的物理或虚拟挂载的设备中。在此情况下，您不需要在内核命令行中指定 `autoinst.xml` 的位置。`autoinst.xml` 必须位于该设备的根目录中。

9.2.1 从闪存盘（例如 USB 记忆棒）引导

出于测试/救援目的，或者在 NIC 没有 PROM 或 PXE 的情况下，您可以构建一张可引导闪存盘来配合 AutoYaST 使用。闪存盘也可以储存控制文件。



提示：将安装媒体映像复制到可卸闪存盘

使用以下命令将安装映像的内容复制到可卸闪存盘。

```
> sudo dd if=IMAGE of=FLASH_DISK bs=4M && sync
```

需将 `IMAGE` 替换为映像文件 `SLE-15-SP7-Online-ARCH-GM-media1.iso` 或 `SLE-15-SP7-Full-ARCH-GM-media1.iso` 的路径。需将 `FLASH_DISK` 替换为闪存设备。要识别设备，请插入该设备并运行：

```
# grep -Ff <(hwinfo --disk --short) <(hwinfo --usb --short)
disk:
/dev/sdc                General USB Flash Disk
```

确保设备的大小足以容纳所需的映像。可使用以下命令检查设备的大小：

```
# fdisk -l /dev/sdc | grep -e "^/dev"
/dev/sdc1 *          2048 31490047 31488000 15G 83 Linux
```

在此示例中，设备的容量为 15 GB。对于 `SLE-15-SP7-Full-ARCH-GM-media1.iso` 使用的命令是：

```
dd if=SLE-15-SP7-Full-ARCH-GM-media1.iso of=/dev/sdc bs=4M && sync
```

在运行 `dd` 命令时，不得挂载该设备。请注意，分区中的所有数据都将被删除！

9.2.2 从 SUSE Linux Enterprise 安装媒体引导

您可将 SUSE Linux Enterprise 安装媒体 (`SLE-15-SP7-Online-ARCH-GM-media1.iso` 或 `SLE-15-SP7-Full-ARCH-GM-media1.iso`) 与其他媒体结合使用。例如，可以通过闪存盘或网络上的指定位置提供控制文件。或者，创建一个包含控制文件的自定义安装媒体。

9.2.3 通过 PXE 经由网络引导

通过 PXE 引导需要提供您网络中的 DHCP 和 TFTP 服务器。然后，计算机无需物理媒体即可引导。有关设置所需基础架构的说明，请参见《部署指南》，第 12 章“远程安装”。

如果您通过 PXE 安装，将无限循环地运行安装。之所以发生此情况，是因为在首次重引导后，计算机将在第二个安装阶段再次执行 PXE 引导并重新启动安装，而不是从硬盘引导。

有多种方法可解决此问题：您可以使用 HTTP 服务器来提供 AutoYaST 控制文件。或者不使用静态控制文件，而是在 Web 服务器上运行一个 CGI 脚本，该脚本可提供控制文件并更改目标主机的 TFTP 服务器配置。这样，下一次就会默认从硬盘完成计算机的 PXE 引导。

另一种方法是使用 AutoYaST 通过控制文件上载目标主机的新 PXE 引导配置。

```
<pxe>
  <pxe_localboot config:type="boolean">true</pxe_localboot>
  <pxelinux-config>
    DEFAULT linux
    LABEL linux
    localboot 0
  </pxelinux-config>
  <tftp-server>192.168.1.115</tftp-server>
  <pxelinux-dir>/pxelinux.cfg</pxelinux-dir>
```

```
<filename>__MAC__</filename>
</pxe>
```

此项会在临近首次重引导前将目标主机的新配置上载到 TFTP 服务器。在大多数安装中，TFTP 守护程序都是以用户 `nobody` 的身份运行的。您需要确保此用户对 `pxelinux.cfg` 目录拥有写入权限。您还可以配置要上载的文件名。如果您使用 “magic” `__MAC__` 文件名，该文件名将是计算机的 MAC 地址，例如 `01-08-00-27-79-49-ee`。如果缺少文件名设置，将使用 IP 地址作为文件名。

要在同一台计算机上执行另一次自动安装，需从 TFTP 服务器中去除该文件。

9.3 调用自动安装过程

9.3.1 命令行选项

添加命令行变量 `autoyast` 会导致 `linuxrc` 以自动模式启动。`linuxrc` 程序会在以下位置搜索应与主控制文件区分开的配置文件：

- 在用于引导系统的初始 RAM 磁盘的根目录中；
- 在引导媒体的根目录中。

`linuxrc` 配置文件支持多个关键字。有关 `linuxrc` 工作原理和其他关键字的详细说明，请参见附录 C “高级 `linuxrc` 选项”。一些较常见的关键字包括：

autoupgrade

启动使用 AutoYaST 的自动升级；请参见第 4.11 节 “升级”。

autoyast

用于自动安装的控制文件的位置；有关细节，请参见 [AutoYaST 控制文件位置](#)。

ifcfg

配置并启动网络。如果要从远程位置提取 AutoYaST，则需要此关键字。有关详细信息，请参见第 C3 节 “高级网络设置”。

insmod

要加载的内核模块

install

安装目录的位置，例如 `install=nfs://192.168.2.1/CDs/`。



注意：禁用 SSL 检查

当您使用 HTTPS 时，系统默认会启用 SSL 检查。如果需要，您可以在 HTTPS URL 中追加 `ssl_verify=no` 来禁用 SL 检查，如下例所示：

```
install=https://192.168.2.1/CDs/?ssl_verify=no
```

如果要传递多个查询选项，请以 & 分隔：

```
install=https://192.168.2.1/CDs/?foo=bar&ssl_verify=no
```

有关详细信息，请参见 `man 8 zypper` 的“FTP/HTTP/HTTPS 目录树”部分。

instmode

安装模式，例如 `http`、`nfs` 等（如果设置了 `install`，则不需要）。

rootpassword

root 用户的口令（如果未在 AutoYaST 配置文件中指定）

server

要连接以获取源目录的服务器 (NFS)

serverdir

NFS 服务器上的目录

y2confirm

即使在控制文件中设置了 `<confirm>no</confirm>`，也会显示确认建议。

这些变量和关键字会将系统启动到可让 YaST 接管主控制文件的位置。目前，源媒体是由系统自动发现的，在某些情况下，无需向 `linuxrc` 发出任何指令，即可启动自动安装过程。

传统 `linuxrc` 配置文件 (info) 的作用是为客户端提供有关安装服务器和源位置的足够信息。

通常不需要此文件，但在不使用 DHCP 和 BOOTP 的特殊网络环境中，或者在需要加载特殊内核模块时，则需要此文件。

您可以使用内核命令行向 **linuxrc** 传递关键字。这可通过多种方式实现。您可以像平时一样，在引导时以交互方式结合使用其他内核参数指定 **linuxrc** 关键字。也可以将内核参数插入到可从网络引导的自定义磁盘映像。还可以配置 DHCP 服务器，结合 Etherboot 或 PXE 来传递内核参数。



注意：使用 **autoyast2** 引导选项而不是 **autoyast**

autoyast2 选项类似于 **autoyast** 选项，但 **linuxrc** 会分析提供的值，并（举例而言）在需要时尝试配置网络。本文档不会介绍此选项。有关 AutoYaST 与 **linuxrc** URI 语法之间差异的信息，请参见 **linuxrc** 附录：附录 C “高级 **linuxrc** 选项”。不支持 AutoYaST 的规则和类。

可按以下列表中所所述的格式使用命令行变量 **autoyast**。

AUTOYAST 控制文件位置

URI 的格式

控制文件位置 URI 的 **autoyast** 语法可能会让人感到迷惑。格式为 SCHEMA://HOST/PATH-TO-FILE。要使用的正斜线数目视情况而异。如果控制文件位于远程位置，则 NFS 服务器的 URI 包含两条斜线，如此示例所示：**autoyast=nfs://SERVER/PATH**。如果控制文件位于本地文件系统，则 URI 将不相同。例如，**autoyast=usb:///profile.xml** 与 **autoyast=usb://localhost/profile.xml** 相同。可以省略本地主机名，但必须保留第三条斜线。**autoyast=usb://profile.xml** 将会失败，因为 **profile.xml** 会被解释为主机名。

无需指定控制文件规范时

对于升级，无需为自动脱机升级指定 **autoyast** 变量，具体请参见过程 4.1 “以脱机升级模式启动 AutoYaST”。

对于全新安装，如果名为 **autoyast** 的文件位于以下三个位置之一，则会启动 **autoinst.xml**：

1. 安装闪存盘（例如 USB 记忆棒）的根目录
2. 安装媒体的根目录
3. 用于引导系统的初始 RAM 磁盘的根目录

autoyast=file:///PATH

当控制文件位于任何本地文件系统（包括 CD 或 USB 驱动器等挂载的外部设备）的顶级目录中时，在相对于源根目录的指定路径（例如 file:///autoinst.xml）中查找控制文件。（这等同于 file://localhost/autoinst.xml。）

autoyast=device://DEVICE/FILENAME

在储存设备上查找控制文件。不要指定设备的完整路径，而是仅指定设备名（例如 device://vda1/autoyast.xml）。您也可以省略指定设备并触发 autoyast 来搜索所有设备，例如 autoyast=device://localhost/autoinst.xml 或 autoyast=device:///autoinst.xml。

autoyast=nfs://SERVER/PATH

在 NFS 服务器上查找控制文件。

autoyast=http://[user:password@]SERVER/PATH

使用 HTTP 协议从 Web 服务器检索控制文件。用户名和口令可选择性指定。

autoyast=https://[user:password@]SERVER/PATH

使用 HTTPS 从 Web 服务器检索控制文件。用户名和口令可选择性指定。

autoyast=tftp://SERVER/PATH

通过 TFTP 检索控制文件。

autoyast=ftp://[user:password@]SERVER/PATH

通过 FTP 检索控制文件。用户名和口令可选择性指定。

autoyast=usb:///PATH

从 USB 设备检索控制文件（autoyast 将搜索所有已连接的 USB 设备）。

autoyast=relurl:///PATH

从安装源检索控制文件：从默认安装源，或者从 install=INSTALLATION_SOURCE_PATH 中定义的安装源。

autoyast=repo:/PATH

从指定路径检索控制文件。路径必须相对于安装源。

autoyast=cifs://SERVER/PATH

在 CIFS 服务器上查找控制文件。

`autoyast=label://LABEL/PATH`

在具有指定标签的设备上搜索控制文件。

可以使用不同类型的基础结构和源媒体来实现多种自动安装方案。最简单的方法是使用 SUSE Linux Enterprise Server (`SLE-15-SP7-Online-ARCH-GM-media1.iso` 或 `SLE-15-SP7-Full-ARCH-GM-media1.iso`)。但是，要启动自动安装过程，应在系统引导时输入自动安装命令行变量，并且控制文件应该可供 YaST 访问。

在脚本上下文中，您可以使用虚拟机的串行控制台，如此便可在文本模式下操作。然后，您可以从 `expect` 脚本或等效脚本传递所需的参数。

以下方案列表解释了控制文件的提供方式：

使用 SUSE Linux Enterprise Server 安装媒体

使用原始安装媒体（需要 `SLE-15-SP7-Online-ARCH-GM-media1.iso` 或 `SLE-15-SP7-Full-ARCH-GM-media1.iso`）时，必须能通过闪存盘（例如 USB 记忆棒）或网络访问控制文件：

闪存盘（例如 USB 记忆棒）： 通过 `autoyast=usb://PATH` 选项访问控制文件。

网络： 通过以下命令访问控制文

件：`autoyast=nfs://...`、`autoyast=ftp://...`、`autoyast=http://...`、`autoyast=https://...`、`autoyast=tftp://...` 或 `autoyast=cifs://...`。需使用 `linuxrc` 中的引导选项定义网络访问。可以通过 DHCP 执行此操作：`netsetup=dhcp`
`autoyast=http://163.122.3.5/autoyast.xml`

使用自定义安装媒体

在这种情况下，您可以直接在安装媒体中包含控制文件。如果将控制文件放入根目录并将其命名为 `autoinst.xml`，则系统会自动找到并在安装时使用它。否则，请使用 `autoyast=file:///PATH` 指定控制文件的路径。

使用网络安装源

这是最重要的选项，因为多个计算机的安装通常是使用 SLP 或 NFS 服务器以及其他网络服务（例如 BOOTP 和 DHCP）完成的。让控制文件可用的最简单方法是将其放入安装源的根目录，并将其命名为 `autoinst.xml`。在这种情况下，系统会自动找到并在安装时使用它。控制文件还可以驻留在以下位置：

闪存盘（例如 USB 记忆棒）： 通过 `autoyast=usb://PATH` 选项访问控制文件。

网络： 通过以下命令访问控制文

件：`autoyast=nfs://...`、`autoyast=ftp://...`、`autoyast=http://...`、`autoyast=https://...`、`autoyast=tftp://..` 或 `autoyast=cifs://...`。



注意：禁用网络和 DHCP

在安装过程中，要在不需要网络或者网络不可用的情况下禁用网络（例如，从 DVD-ROM 自动安装时），请使用 `linuxrc` 选项 `netsetup=0` 禁用网络设置。

在提供了所有 AutoYaST 调用选项的情况下，可通过以下方式指定控制文件的位置：

1. 指定控制文件的确切位置：

```
autoyast=http://192.168.1.1/control-files/client01.xml
```

2. 指定多个控制文件所在的目录：

```
autoyast=http://192.168.1.1/control-files/
```

在本例中，将使用 IP 的十六进制数字表示形式检索相关控制文件，如下所述。

此目录的路径需以 `/` 结尾。

目录中的文件不得带有任何扩展名（例如 `.xml`）。因此文件名只需是 IP 地址或 MAC 地址。

```
> ls -r control-files
C00002 0080C8F6484C default
```

如果只定义了路径前缀变量，YaST 将通过以下方式从指定的位置提取控制文件：

1. 首先，YaST 将使用自身的大写十六制 IP 地址（例如 192.0.2.91 -> C000025B）搜索控制文件。
2. 如果未找到此文件，YaST 将去除一个十六进制数字并重试。此操作将不断重复，直到找到具有正确名称的文件。最终，它会尝试查找使用客户端 MAC 地址作为文件名的文件（mac 应采用以下语法：0080C8F6484C），如果未找到，则查找名为 default（小写）的文件。

例如，对于 192.0.2.91，HTTP 客户端将尝试：

```
C000025B
C000025
C00002
C0000
C000
C000
C00
C00
C0
C
0080C8F6484C
default
```

（按上面所示的顺序）。

要确定客户端 IP 地址的十六进制表示形式，请使用 `syslinux` 软件包中提供的名为 /usr/bin/gethostip 的实用程序。

例 9.1：确定 IP 地址的十六进制代码

```
> /usr/bin/gethostip 10.10.0.1
10.10.0.1 10.10.0.1 0A0A0001
```

9.3.2 自动安装单个系统

如果要在没有网络连接的情况下自动安装系统，最简单的方法是使用原始 SUSE Linux Enterprise Server DVD-ROM 和闪存盘（例如 USB 记忆棒）。您既不需要设置安装服务器，也不需要设置网络环境。

创建控制文件并将其命名为 autoinst.xml。将文件 autoinst.xml 复制到闪存盘。

9.3.3 将 `linuxrcinfo` 文件与 AutoYaST 控制文件结合使用

如果您选择使用 `linuxrc` 文件或者以引导选项的形式向 `info` 传递信息，可以将关键字集成到 AutoYaST 控制文件中。如下例所示添加 `info_file` 部分。此部分包含以冒号分隔的关键字-值对，每行包含一对。

例 9.2：AUTOYAST 控制文件中的 `linuxrc` 选项

```
....  
  <install>  
....  
  <init>  
    <info_file>  
  
install: nfs://192.168.1.1/CDs/full-x86_64  
dud: https://example.com/driver_updates/filename.dud  
upgrade: 1  
textmode: 1  
    </info_file>  
  </init>  
.....  
</install>  
....
```

请注意，`autoyast2` 关键字必须指向同一个文件。如果该文件位于闪存盘（例如 USB 记忆棒）上，则需要使用 `usb://` 选项。如果 `info` 文件储存在初始 RAM 磁盘中，则需要使用 `file:///` 选项。

9.4 系统配置

自动安装期间的系统配置是整个过程中最重要的部分。在前面的章节中可以看到，几乎所有设置都可以在目标系统上自动配置。除了预定义的指令外，您始终可以使用后脚本来更改系统中的其他设置。此外，您也可以更改任何系统变量，如果需要，还可以将完整的配置文件复制到目标系统。

9.4.1 安装后配置和系统配置

在目标系统上安装完最后一个软件包后会直接启动安装后和系统配置，并且在首次引导系统后还会持续进行该配置。

在首次引导系统之前，AutoYaST 会写入安装期间收集的所有数据，并在指定的位置写入引导加载程序。除了这些常规任务外，AutoYaST 还会按照控制文件中的指定执行 chroot 脚本。请注意，这些脚本是在尚未挂载系统的情况下执行的。

如果安装了非默认的内核，则需要进行硬性重引导。也可以在自动安装期间强制执行硬性重引导，而不管安装了哪个内核。使用 `general` 资源的 `reboot` 属性（请参见第 4.1 节“一般选项”）。

9.4.2 系统自定义

大部分系统自定义操作都是在第二个安装阶段完成的。如果您需要完成无法通过 AutoYaST 资源完成的自定义操作，请使用安装后脚本做进一步的修改。

您可以通过编辑控制文件或使用配置系统，在控制文件中定义无限数量的自定义脚本。

V 在安装的系统上使用 AutoYaST

10 在安装的系统中运行 AutoYaST 262

10 在安装的系统中运行 AutoYaST

某些情况下，在运行中的系统上运行 AutoYaST 会很有用。请注意，在这种情况下会忽略 `partitioning` 部分。

以下示例将要安装一个额外的软件包 (`foo`)。要运行此软件，需要添加用户并配置 NTP 客户端。

相应的 AutoYaST 配置文件需要包含软件包安装（第 4.10.7 节 “在第 2 个阶段安装软件包”）部分、用户（第 4.31.1 节 “用户”）部分和 NTP 客户端（第 4.22 节 “NTP 客户端”）部分：

```
<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://
www.suse.com/1.0/configs">
  <ntp-client>
    <peers config:type="list">
      <peer>
        <address>us.pool.ntp.org</address>
        <comment/>
        <options> iburst</options>
        <type>server</type>
      </peer>
    </peers>
    <start_at_boot config:type="boolean">true</start_at_boot>
    <start_in_chroot config:type="boolean">false</start_in_chroot>
    <sync_interval config:type="integer">5</sync_interval>
    <synchronize_time config:type="boolean">false</synchronize_time>
  </ntp-client>
  <software>
    <post-packages config:type="list">
      <package>ntp</package>
      <package>yast2-ntp-client</package>
      <package>foo</package>
    </post-packages>
  </software>
  <users config:type="list">
    <user>
      <encrypted config:type="boolean">false</encrypted>
```

```
<fullname>Foo user</fullname>
<gid>100</gid>
<home>/home/foo</home>
<password_settings>
  <expire/>
  <flag/>
  <inact/>
  <max>99999</max>
  <min>0</min>
  <warn>7</warn>
</password_settings>
<shell>/bin/bash</shell>
<uid>1001</uid>
<user_password>linux</user_password>
<username>foo</username>
</user>
</users>
</profile>
```

将此文件储存为 /tmp/install_foo.xml，并调用以下命令启动 AutoYaST 安装过程：

```
> sudo yast2 ayast_setup setup filename=/tmp/install_foo.xml dopackages="yes"
```

如需详细信息，请运行 **yast2 ayast_setup longhelp**

VI 附录

- A 处理规则 265
- B AutoYaST FAQ — 常见问题 266
- C 高级 **linuxrc** 选项 270
- D SLE 12 与 15 中的 AutoYaST 配置文件之间的差异 275
- E GNU licenses 291

A 处理规则

下图说明了规则的处理方式，以及检索与合并过程。

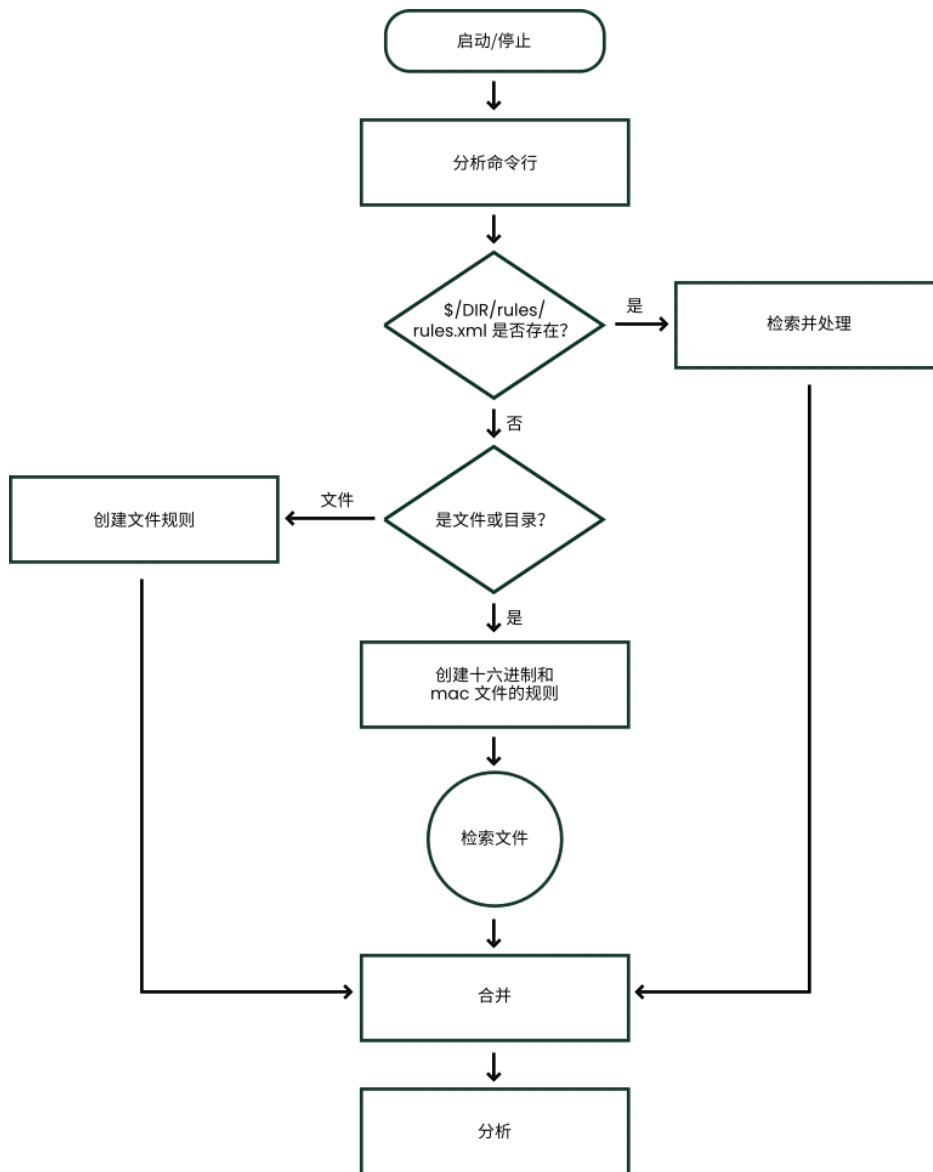


图 A1：规则检索过程

B AutoYaST FAQ — 常见问题

10. 如何调用 AutoYaST 安装？

在所有 SUSE Linux Enterprise Server 版本中，均可通过将 `autoyast=<PATH_TO_PROFILE>` 添加到内核参数列表来调用自动安装功能。例如，添加 `autoyast=http://MYSERVER/MYCONFIG.xml` 将启动自动安装，安装过程中会从 Web 服务器 `myserver` 提取包含 AutoYaST 配置的配置文件。有关更多信息，请参见第 9.3 节 “调用自动安装过程”。

11. 什么是 AutoYaST 配置文件？

此配置文件是指 AutoYaST 配置文件。AutoYaST 配置文件的内容决定了如何配置系统，以及要安装哪些软件包。包括分区、网络设置和软件源等。可在运行中系统上使用 YaST 配置的所有设置几乎也都可以 AutoYaST 配置文件中配置。配置文件格式是 ASCII XML 文件。

12. 如何创建 AutoYaST 配置文件？

要创建 AutoYaST 配置文件，最简单的方法是使用现有 SUSE Linux Enterprise Server 系统作为模板。在已安装的系统上，启动 YaST > 杂项 > 自动安装。接下来，从菜单中选择工具 > 创建参考配置文件。选择您要包含在配置文件中的系统组件。或者，通过在命令行中运行 `sudo yast clone_system` 创建包含完整系统配置的配置文件。

这两种方法都会创建文件 `/root/autoinst.xml`。在命令行中创建的版本可用于设置在其上创建了配置文件的系统的相同克隆。但是，通常需要调整该文件，才能安装多个非常类似但不完全相同的计算机。为此，可以使用您偏好的文本编辑器/XML 编辑器调整配置文件。

13. 如何检查创建的 AutoYaST 配置文件的语法？

检查创建的 AutoYaST 配置文件的最有效方法是使用 `jing` 或 `xmllint`。

有关详细信息，请参见第 3.3 节 “手动创建/编辑控制文件”。

14. 有意义的最小 AutoYaST 配置文件是怎样的？

如果在 AutoYaST 配置文件中未定义某个部分，将使用 YaST 一般安装建议的设置。但是，至少需要指定 `root` 口令才能在安装后登录到计算机。

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://
www.suse.com/1.0/configs">
  <users config:type="list">
    <user>
      <encrypted config:type="boolean">false</encrypted>
      <user_password>linux</user_password>
      <username>root</username>
    </user>
  </users>
</profile>
```

15. 如何在自动检测声卡的情况下执行自动安装？

在配置文件中使用以下 `sound` 部分：

```
<sound>
  <autoinstall config:type="boolean">true</autoinstall>
  <configure_detected config:type="boolean">true</configure_detected>
</sound>
```

16. 我只想从 DVD 安装。要将 AutoYaST 配置文件放在何处？

请将配置文件放在 DVD 的根目录中。使用 `file:///PROFILE.xml` 引用该配置文件。

17. 如何在命令行上测试合并过程？

要合并 `a.xml` 和 `base.xml` 这两个配置文件，请运行以下命令：

```
> /usr/bin/xsltproc --novalid --param replace "'false'" \
--param dontmerge1 "'package'" --param with "'a.xml'" --output out.xml \
/usr/share/autoinstall/xslt/merge.xslt base.xml
```

这两个配置文件中的部分需按字母顺序排列（例如，`<software>` 需列在 `<add-on>` 之后）。如果配置文件是使用 YaST 创建的，则配置文件会自动正确排序。

`dontmerge1` 参数是可选的，用于示范在配置文件中使 `dont_merge` 元素时要怎样做。有关更多信息，请参见第 6.4 节“合并规则和类”。

18. 是否可以从脚本调用 Zypper?

只能从 AutoYaST init 脚本调用 Zypper，因为在安装后脚本阶段，YaST 仍然对 RPM 数据库拥有独占式锁定。

如果您确实想要使用其他脚本类型（例如安装后脚本），则需自行承担破坏锁定的风险：

```
<post-scripts config:type="list">
  <script>
    <filename>yast_clone.sh</filename>
    <interpreter>shell</interpreter>
    <location/>
    <feedback config:type="boolean">false</feedback>
    <source><![CDATA[#!/bin/sh
mv /var/run/zypp.pid /var/run/zypp.sav
zypper in foo
mv /var/run/zypp.sav /var/run/zypp.pid
]]></source>
  </script>
</post-scripts>
```

19. AutoYaST 配置文件中各部分的顺序是否很重要？

顺序实际上并不重要。配置文件中各部分的顺序对 AutoYaST 工作流程没有影响。但是，如果要**合并**不同的配置文件，则需要按字母顺序排列各部分。

20. **linuxrc** 阻止安装并显示 `File not signed`。我需要手动交互。

linuxrc 找到了一个未签名的文件，例如某个驱动程序更新。要使用未签名的文件，可以通过将 `insecure=1`（连同 `autoyast=...` 参数）传递给 **linuxrc** 参数列表来隐藏该消息。

21. 我想从 DVD/USB/HD 安装，而从网络提取 XML 文件。

需要将 `ifcfg` 传递给 **linuxrc**。这样才能设置网络，否则 AutoYaST 无法从远程主机下载配置文件。有关更多信息，请参见第 C3 节“高级网络设置”。

22. 是否可以安装到 NFS 根目录 (/)?

可以，但这种安装方法比其他方法更复杂。设置环境（DHCP、TFTP 等）时必须非常小心。AutoYaST 配置文件必须如下所示：

```

<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://
www.suse.com/1.0/configns">
  <partitioning config:type="list">
    <drive>
      <device>/dev/nfs</device>
      <initialize config:type="boolean">>false</initialize>
      <type config:type="symbol">CT_NFS</type>
      <partitions config:type="list">
        <partition>
          <filesystem config:type="symbol">nfs</filesystem>
          <fstopt>nolock</fstopt>
          <device>10.10.1.53:/tmp/m4</device>
          <mount>/</mount>
        </partition>
      </partitions>
      <use>all</use>
    </drive>
  </partitioning>
</profile>

```

- 23.** 对于本附录中未予解答的问题，我可以在哪里提问？

您可以在 AutoYaST 邮件列表中发布问题。请通过 <https://lists.opensuse.org/opensuse-autoinstall/> 加入我们。

C 高级 **linuxrc** 选项

linuxrc 是在加载内核之后、执行 AutoYaST 或其他阶段之前运行的一个小程序，用于为安装系统做好准备。用户可以通过此程序加载模块、启动安装的系统或救援系统，以及指导完成 YaST 的操作。



注意：AutoYaST 和 **linuxrc** 设置不相同

某些 **linuxrc** 设置恰好与 AutoYaST 在其 `autoyast.xml` 文件中使用的设置同名。但这并不意味着它们采用相同的参数或者以相同的方式工作。例如，AutoYaST 采用 **self_update** 设置。如果此值设置为 `1`，则会读取并遵循另一个设置 **self_update_url**。虽然 **linuxrc** 也具有 **self_update** 设置，但 **linuxrc** 的设置采用值 `0` 或 URL。

请不要将 AutoYaST 参数传递给 **linuxrc**，因为几乎可以肯定这不会产生所需的结果。

如果在计算机上安装了 **linuxrc**，可以在 `/usr/share/doc/packages/linuxrc/` 目录中找到其相关信息。或者，也可以在 <https://en.opensuse.org/SDB:Linuxrc> 上查找其联机文档。



注意：在安装的系统上运行 **linuxrc**

如果您在安装的系统上运行 **linuxrc**，为了不破坏您的安装，它会以略微不同的方式工作。因此，您无法以这种方式测试所有功能。

为了尽可能保留最小的 **linuxrc** 二进制文件，其所有库和其他补充文件已直接链接到主程序二进制文件。这意味着，不需要在初始 RAM 磁盘 `initrd` 中保留任何共享库。

C1 将参数传递给 linuxrc

除非 **linuxrc** 处于手动模式，否则它会在以下位置查找 **info** 文件：首先在闪存盘（例如 USB 记忆棒）上的 **/info** 中查找，如果在该位置未找到，则在 **initrd** 上的 **/info** 中查找。然后，它会分析参数的内核命令行。可以通过设置 **info** 命令行参数来更改 **linuxrc** 读取的 **info** 文件。如果您不希望 **linuxrc** 读取内核命令行（例如，因为您还需要指定 **linuxrc** 可识别的某个内核参数），请使用 **linuxrc=nocmdline**。

linuxrc 始终查找并分析名为 **/linuxrc.config** 的文件。如果需要，请使用此文件更改默认值。一般而言，最好是改用 **info** 文件。请注意，即使是在手动模式下，也会先读取 **/linuxrc.config**，再读取任何 **info** 文件。

C2 info 文件格式

以 **#** 开头的行是注释。有效项的格式如下：

```
key: value
```

请注意，**value** 将扩展到行尾，因此可能包含空格。**key** 的匹配不区分大小写。

您可以使用语法 **key=value**，在内核命令行上使用相同的键-值对。未采用上述格式的行将被忽略。

下表列出了重要的键和示例值。有关 **linuxrc** 参数的完整列表，请参见 <https://en.opensuse.org/SDB:Linuxrc>。

表 C1：高级 **linuxrc** 关键字

关键字：示例值	说明
addswap : 0 3 /dev/ sda5	如果为 0，则永远不请求交换；如果该参数是正数 n ，则激活交换分区；如果该参数是分区名称，则激活此交换分区。
autoyast : ftp:// AUTOYASTFILE	自动安装文件的位置；激活自动安装模式。有关详细信息，请参见 AutoYaST 控制文件位置 。
bootptimeout : 10	BOOTP 请求的超时为 10 秒。

关键字：示例值	说明
<u>bootpwait: 5</u>	在网络激活与启动 bootp 之间休眠 5 秒。
<u>display: color mono alt</u>	设置菜单配色方案。
<u>exec: COMMAND</u>	运行 command 。
<u>forceinsmod: 0 1</u>	运行 <u>-f</u> 命令时使用 insmod （强制）选项。
<u>forcerootimage: 0 1</u>	将安装系统加载到 RAM 磁盘中。
<u>ifcfg: NETWORK_CONFIGURATION</u>	设置并启动网络。有关更多信息，请参见第 C3 节“高级网络设置”。
<u>insmod: MODULE</u>	加载 <u>MODULE</u> 。
<u>install: URL</u>	从 <u>URL</u> 指定的储存库安装。有关 <u>URL</u> 的语法，请参见 https://en.opensuse.org/SDB:Linuxrc#url_descr 。
<u>keytable: de-lat1-nd</u>	要加载的虚拟控制台键盘映射。
<u>language: de_DE</u>	预先选择的安装语言。
<u>loghost: 10.10.0.22</u>	经由 UDP 端口 514 通过 syslog 启用远程日志记录
<u>loghost: @10.10.0.22</u>	经由 TCP 端口 514 通过 syslog 启用远程日志记录
<u>memloadimage: 50000</u>	如果可用内存超过 50000 KB，则将安装系统加载到 RAM 磁盘中。
<u>memlimit: 10000</u>	如果可用内存降到 10000 KB 以下，则请求交换。
<u>memYaST: 20000</u>	如果可用内存低于 20000 KB，则以文本模式运行 YaST。
<u>memYaSTText: 10000</u>	如果可用内存低于 10000 KB，则在启动 YaST 之前请求交换。

关键字：示例值	说明
<code>proxy:</code> <code>http://10.10.0.1:3128</code>	定义 HTTP 代理服务器。有关完整的参数语法，请参见 https://en.opensuse.org/SDB:Linuxrc#p_proxy 。
<code>rescue: 1 nfs://</code> <code>server/dir</code>	加载救援系统；URL 变体明确指定救援映像的位置。
<code>rescueimage: /suse/</code> <code>images/rescue</code>	救援系统映像的位置。
<code>rootimage: /suse/</code> <code>images/root</code>	安装系统映像的位置。
<code>textmode: 1</code>	以文本模式启动 YaST。
<code>usbwait: 4</code>	加载 USB 模块后等待 4 秒。
<code>y2confirm</code>	覆盖控制文件中的确认参数，并请求确认安装建议。

C3 高级网络设置

即使已将 `hostip`、`nameserver` 和 `gateway` 等参数传递给 `linuxrc`，也只会需要在需要时（例如，通过 SSH 或 VNC 安装时）才启动网络。由于 `autoyast` 不是 `linuxrc` 参数（此参数将被 `linuxrc` 忽略，只会传递给 YaST），在为 AutoYaST 配置文件指定远程位置时，系统不会自动启动网络。

因此，需要明确启动网络。使用 `ifcfg` 参数可实现此目的。`ifcfg` 可直接控制 `/etc/sysconfig/network/ifcfg-*` 文件的内容。

DHCP 网络配置

用于配置 DHCP 的一般语法是

```
ifcfg=INTERFACE=DHCP*,OPTION1=VALUE1,OPTION2=VALUE2
```

其中 INTERFACE 是接口名称，例如 eth0 或 eth*（表示所有接口）。DHCP* 可以是 dhcp（IPv4 和 IPv6）、dhcp4 或 dhcp6。

要为 eth0 设置 DHCP，请使用：

```
ifcfg=eth0=dhcp
```

要在所有接口上设置 DHCP，请使用：

```
ifcfg=eth*=dhcp
```

静态网络配置

用于配置静态网络的一般语法是

```
ifcfg=INTERFACE=IP_LIST,GATEWAY_LIST,NAMESERVER_LIST,DOMAINSEARCH_LIST,\
OPTION1=value1,...
```

其中 INTERFACE 是接口名称，例如 eth0。如果使用 eth*，将使用第一个可用设备。其他参数需按给定的顺序替换为相应的值。示例：

```
ifcfg=eth0=192.168.2.100/24,192.168.5.1,192.168.1.116,example.com
```

为某个参数指定多个地址时，请使用空格进行分隔，并以引号括住整个字符串。以下示例使用两个名称服务器，以及一个包含两个域的搜索列表。

```
ifcfg="eth0=192.168.2.100/24,192.168.5.1,192.168.1.116
192.168.1.117,example.com example.net"
```

有关详细信息，请参见 https://en.opensuse.org/SDB:Linuxrc#Network_Configuration。

D SLE 12 与 15 中的 AutoYaST 配置文件之间的差异

SUSE Linux Enterprise Server 15 中的重大更改（例如新的模块概念，或者以 `firewalld` 取代 `SuSEfirewall2`）要求在 AutoYaST 中进行相关更改。如果您想要在 SUSE Linux Enterprise Server 15 中重复使用现有的 SUSE Linux Enterprise Server 12 配置文件，需要按本文所述调整这些配置文件。

D1 产品选择

为了能够与针对早于 SLE15 的产品版本创建的配置文件向后兼容，AutoYaST 实施了一种自动选择产品的启发方法。当配置文件不包含 `product` 元素时，将使用此启发。自动产品选择基于配置文件中的软件包和软件集选择。但是，只要有可能，都应该避免依赖此机制，并将旧配置文件更改为使用明确选择产品的方式。

有关明确选择产品的信息，请参见第 4.10.1 节“产品选择”。

如果自动产品选择失败，将显示错误，并且安装不会继续。

D2 软件

SUSE Linux Enterprise Server 15 SP7 安装媒体仅包含极少的一部分可安装软件包。这组精简的软件包仅提供安装环境，而不包括任何服务器应用程序或高级工具。提供更多软件包的其他软件储存库以“模块”或“扩展”的形式提供。通过以下两种备选方式提供：

- 通过注册服务器（SUSE Customer Center 或本地 SMT/RMT 代理）
- 通过包含所有模块和扩展的 `SLE-15-SP7-Full-ARCH-GM-media1.iso` 映像。使用此媒体不要求在安装期间访问注册服务器。可以通过安装服务器在本地网络上共享此媒体。



注意：维护更新

只需使用注册服务器就会在安装时授予对所有维护更新的访问权限。在不注册的情况下使用 DVD 媒体时无法访问维护更新。

D2.1 使用注册服务器添加模块或扩展

要从注册服务器添加模块或扩展，请对 `addons` 部分中的每个模块/扩展使用 `suse_register` 标记。扩展需要额外的注册代码，可通过 `reg_code` 标记指定该注册代码。

以下 XML 代码会添加 Basesystem 和 Server Applications 模块以及 Live Patching 扩展。

要获取 `name`、`version` 和 `arch` 标记的相应值，请在已安装 SLE 15 SP7 的系统上运行命令 **`SUSEConnect --list-extensions`**。

例 D1：添加模块和扩展（联机）

```
<suse_register>
  <addons config:type="list">
    <addon>
      <name>sle-module-basesystem</name>
      <version>15.7</version>
      <arch>x86_64</arch>
    </addon>
    <addon>
      <name>sle-module-server-applications</name>
      <version>15.7</version>
      <arch>x86_64</arch>
    </addon>
    <addon>
      <name>sle-module-live-patching</name>
      <version>15.7</version>
      <arch>x86_64</arch>
      <reg_code>REGISTRATION_CODE</reg_code>
    </addon>
  </addons>
</suse_register>
```

有关更多信息，请参考第 4.3 节“系统注册和扩展选择”。

D2.2 使用 SLE-15-SP7-Full-ARCH-GM-media1.iso 映像添加模块或扩展

要使用 SLE-15-SP7-Full-ARCH-GM-media1.iso 映像添加模块或扩展，请按以下示例中所示在 add-on 部分中创建相应项。以下 XML 代码会从包含该映像的 USB 闪存盘添加 Basesystem Module：

例 D2：添加模块和扩展（脱机）

```
<add-on>
  <add_on_products config:type="list">
    <listentry>
      <media_url><![CDATA[dvd:///devices=/dev/sda%2C/dev/sdb%2C/dev/sdc%2C/dev/sdd]]></media_url>
      <product_dir>/Module-Basesystem</product_dir>
      <product>sle-module-basesystem</product>
    </listentry>
  </add_on_products>
</add-on>
```



注意：产品名称匹配

product 必须与储存库中包含的内部产品名称相匹配。如果安装时产品名称不匹配，AutoYaST 将报告错误。



提示：在本地服务器上使用安装媒体映像

您可以通过 NFS、FTP 或 HTTP 服务器在本地网络上共享 USB 闪存盘的内容。要执行此操作，请替换 media_url 标记中的 URL，使其指向服务器上媒体的根目录，例如：

```
<media_url>ftp://ftp.example.com/sle_15_sp7_full/</media_url>
```

D2.3 已更名的软件集

软件集自 SUSE Linux Enterprise Server 15 开始也已发生更改。某些软件集已更名；下表中提供了简短摘要。

旧的 SLE12 软件集	新的 SLE 15 软件集
Basis-Devel	devel_basis
gnome-basic	gnome_basic
Minimal	enhanced_base
printing	print_server
SDK-C-C++	devel_basis
SDK-Doc	technical_writing
SDK-YaST	devel_yast

请仔细检查定义的软件集中是否包含全部所需软件包，并相应地调整配置文件。此外，需在已激活的扩展或模块中提供所需的软件集和软件包。

注意

- 上表中的软件集重命名并非一对一替换；某些软件集的内容也发生了更改，某些软件包已移到不同软件集，有的甚至已从 SUSE Linux Enterprise Server 15 中去除。
- 请检查所需的软件包是否仍包含在使用的软件集中，并根据需要使用 `packages` 标记指定其他软件包。
- 以上列表可能不完整，因为某些产品尚未针对 SUSE Linux Enterprise Server 15 发布。

D3 模块和扩展依赖项的注册

从 SUSE Linux Enterprise Server 15 开始，注册期间，AutoYaST 会根据扩展的依赖项自动对扩展重新排序。这意味着，AutoYaST 配置文件中的扩展顺序并不重要。

另外，AutoYaST 会自动添加依赖模块，即使这些模块在配置文件中缺失。这意味着，您无需指定全部所需模块。但是，如果一个扩展依赖于另一个扩展，则需要在配置文件中指定所依赖的扩展（包括注册密钥）。否则注册将会失败。

您可以使用 **SUSEConnect --list-extensions** 命令列出已注册系统中的可用扩展和模块。

D4 分区

先前由 YaST 使用的分区后端 `libstorage` 现已被 `libstorage-ng` 取代，后者可以实现以前所不能实现的新功能。尽管后端有更改，但配置文件的 XML 语法并未更改。不过，SUSE Linux Enterprise Server 15 包含一些常规更改，相关说明请见下文。

D4.1 GPT 变成 AMD64/Intel 64 上的默认分区类型

在 AMD64/Intel 64 系统上，GPT 现在是首选的分区类型。不过，如果您更想保留旧行为，可以通过在配置文件中将 `disklabel` 元素设为 `msdos` 来明确指示。

D4.2 设置分区号

AutoYaST 将不再支持强制分区号，因为在某些情况下，此行为不起作用。此外，GPT 现已成为首选的分区表类型，因此分区号没有太大的意义。

不过，您仍可使用 `partition_nr` 标记来指定要重复使用的分区。有关更多信息，请参考第 4.6.3.2 节“分区配置”。

D4.3 强制设置主分区

仍可将某个分区强制作为 `primary`（仅限在 MS-DOS 分区表中），方法是将 `partition_type` 设为 `primary`。但是，AutoYaST 将忽略任何其他值（例如 `logical`），因为它会自动确定分区类型。

D4.4 Btrfs: 默认子卷名称

新储存层允许用户为每个 Btrfs 文件系统设置不同的默认子卷（或者不设置默认子卷）。如下面的示例所示，可以使用 `subvolumes_prefix` 标记为每个分区指定前缀名称：

例 D3：指定 BTRFS 默认子卷名称

```
<partition>
  <mount>/</mount>
  <filesystem config:type="symbol">btrfs</filesystem>
  <size>max</size>
  <subvolumes_prefix>@</subvolumes_prefix>
</partition>
```

要省略子卷前缀，请设置 `subvolumes_prefix` 标记：

例 D4：禁用 BTRFS 子卷

```
<partition>
  <mount>/</mount>
  <filesystem config:type="symbol">btrfs</filesystem>
  <size>max</size>
  <subvolumes_prefix>@</subvolumes_prefix>
</partition>
```

由于实施新的行为，不再需要旧的 `btrfs_set_default_subvolume_name` 标记，因此不再支持此标记。

D4.5 Btrfs: 禁用子卷

可以通过将 `create_subvolumes` 设为 `false` 来禁用 Btrfs 子卷。要跳过默认的 `@` 子卷，请指定 `subvolumes_prefix`。

```
<partition>
  <create_subvolumes config:type="boolean">>false</create_subvolumes>
  <subvolumes_prefix><![CDATA[]]></subvolumes_prefix>
</partition>]]>
```

D4.6 不再支持读取现有的 `/etc/fstab`

在 SUSE Linux Enterprise Server 15 上，不再支持在尝试确定分区布局时从之前的安装读取现有 `/etc/fstab` 的功能。

D4.7 已去除用于对齐分区的设置

由于柱面的概念已过时，`partition_alignment` 标记没有任何意义，因此不再可用。AutoYaST 始终尝试以最佳方式对齐分区。

D4.8 使用 `type` 定义卷组

已去除 `is_lvm_vg` 元素，以便可将 `type` 设置为 `CT_LVM` 值。有关更多细节，请参见第 4.6.5 节“逻辑卷管理器 (LVM)”。

D5 防火墙配置

在 SUSE Linux Enterprise Server 15 中，`firewalld` 已取代 `SuSEfirewall2` 成为默认防火墙。这两个防火墙的配置具有明显的差异，因此相应的 AutoYaST 配置文件语法已更改。旧配置文件仍有效，但支持的配置极其有限。建议按下面所述更新 SLE 15 的配置文件。如果保留 SLE12 配置文件，我们建议检查最终配置，以避免意外行为或网络安全威胁。

表 D1：SLE 15 中的 AUTOYAST 防火墙配置：向后兼容性

支持（但已弃用）	不支持
<code>FW_CONFIGURATIONS_{DMZ, EXT, INT}</code>	<code>FW_ALLOW_FW_BROADCAST_{DMZ, EXT, INT}</code>
<code>FW_DEV_{DMZ, EXT, INT}</code>	<code>FW_IGNORE_FW_BROADCAST_{DMZ, EXT, INT}</code>
<code>FW_LOG_DROP_ALL</code>	<code>FW_IPSECT_TRUST</code>

支持（但已弃用）	不支持
<u>FW_LOG_DROP_CRIT</u>	<u>FW_LOAD_MODULES</u>
<u>FW_MASQUERADE</u>	<u>FW_LOG_ACCEPT_ALL</u>
<u>FW_SERVICES_{DMZ, INT, EXT}_</u> <u>{TCP, UDP, IP}</u>	<u>FW_LOG_ACCEPT_CRIT</u>
	<u>FW_PROTECT_FROM_INT</u>
	<u>FW_ROUTE</u>
	<u>FW_SERVICES_{DMZ, EXT, INT}_RPC</u>
	<u>FW_SERVICES_ACCEPT_RELATED_{DMZ,</u> <u>EXT, INT}</u>

不再可用的 SuSEfirewall2 配置选项要么在 `firewalld` 中没有等效的对应选项，要么只会在未来的 SUSE Linux Enterprise Server 版本中受支持。某些 `firewalld` 功能尚不受 YaST 和 AutoYaST 的支持 — 您可以在 AutoYaST 配置文件中通过安装后脚本使用这些功能。有关更多信息，请参见第 4.32 节“自定义用户脚本”。



注意：启用和启动防火墙

启用和启动 `firewalld` 的 `systemd` 服务是使用 SLE12 中的相同语法完成的。在 AutoYaST 中，只有这一部分防火墙配置语法未更改：

```
<firewall>
  <enable_firewall config:type="boolean">true</enable_firewall>
  <start_firewall config:type="boolean">true</start_firewall>
  ...
</firewall>
```

以下示例演示如何将已弃用（但仍受支持）的配置文件转换为 SLE 15 语法：

D5.1 将接口指派给区域

SuSEfirewall2 和 `firewalld` 都基于区域，但它们使用不同的预定义规则集和不同的信任级别建立网络连接。

表 D2：SUSEFIREWALL2 和 `firewalld` 区域的映射

<code>firewalld</code> (SLE 15)	SuSEfirewall2 (SLE 12)
dmz	DMZ
external	EXT, <code>FW_MASQUERADE</code> 设为 <code>yes</code>
public	EXT, <code>FW_MASQUERADE</code> 设为 <code>no</code>
internal	INT, <code>FW_PROTECT_FROM_INT</code> 设为 <code>yes</code>
trusted	INT, <code>FW_PROTECT_FROM_INT</code> 设为 <code>no</code>
block	不适用
drop	不适用
home	不适用
work	不适用

在 SuSEfirewall2 中，默认区域是外部区域 (EXT)，但也允许使用特殊关键字 `any` 将未在任何位置列出的所有接口指派给指定区域。

D5.1.1 默认配置

以下两个示例演示了应用于接口 `wlan0`、`eth0`、`eth1` 和 `wlan1` 的默认配置。

例 D5：指派区域：默认配置（已弃用的语法）

```
<firewall>
  <FW_DEV_DMZ>any eth0</FW_DEV_DMZ>
```

```
<FW_DEV_EXT>eth1 wlan0</FW_DEV_EXT>
<FW_DEV_INT>wlan1</FW_DEV_INT>
</firewall>
```

例 D6：指派区域：默认配置（SLE 15 语法）

```
<firewall>
<default_zone>dmz</default_zone>
<zones config:type="list">
  <zone>
    <name>dmz</name>
    <interfaces config:type="list">
      <interface>eth0</interface>
    </interfaces>
  </zone>
  <zone>
    <name>public</name>
    <interfaces config:type="list">
      <interface>eth1</interface>
    </interfaces>
  </zone>
  <zone>
    <name>trusted</name>
    <interfaces config:type="list">
      <interface>wlan1</interface>
    </interfaces>
  </zone>
</zones>
</firewall>
```

D5.1.2 掩蔽和保护内部区域

以下两个示例演示了如何使用掩蔽和受保护内部区域来配置接口 wlan0、eth0、eth1 和 wlan1。

例 D7：掩蔽和保护内部区域（已弃用的语法）

```
<firewall>
```

```

<FW_DEV_DMZ>any eth0</FW_DEV_DMZ>
<FW_DEV_EXT>eth1 wlan0</FW_DEV_EXT>
<FW_DEV_INT>wlan1</FW_DEV_INT>
<FW_MASQUERADE>yes</FW_MASQUERADE>
<FW_PROTECT_FROM_INT>yes</FW_PROTECT_FROM_INT>
</firewall>

```

例 D8：掩蔽和保护内部区域（SLE 15 语法）

```

<firewall>
  <default_zone>dmz</default_zone>
  <zones config:type="list">
    <zone>
      <name>dmz</name>
      <interfaces config:type="list">
        <interface>eth0</interface>
      </interfaces>
    </zone>
    <zone>
      <name>external</name>
      <interfaces config:type="list">
        <interface>eth1</interface>
      </interfaces>
    </zone>
    <zone>
      <name>internal</name>
      <interfaces config:type="list">
        <interface>wlan1</interface>
      </interfaces>
    </zone>
  </zones>
</firewall>

```

D5.2 打开端口

在 SuSEfirewall2 中，使用 FW_SERVICES_{DMZ,EXT,INT}_{TCP,UDP,IP,RPC} 标记来打开不同区域中的端口。

对于 TCP 或 UDP，SuSEfirewall2 支持端口号或端口范围，或者使用相应区域和服务的单个标记指定的 /etc/services 中的服务名称。对于 IP 服务，支持端口号或端口范围，或者可使用以下标记指定的 /etc protocols 中的协议名称：FW_SERVICES_ZONE_IP。

对于 firewalld，在相应区域的 port 部分中，每个端口、端口范围和服务都需要有单独的项。在 protocol 部分中，IP 服务需要有单独的项。

受 SuSEfirewall2 支持的 RPC 服务不再受 firewalld 的支持。

例 D9：打开端口（已弃用的语法）

```
<firewall>
  <FW_SERVICES_DMZ_TCP>ftp ssh 80 5900:5999</FW_SERVICES_DMZ_TCP>
  <FW_SERVICES_EXT_UDP>1723 ipsec-nat-t</FW_SERVICES_EXT_UDP>
  <FW_SERVICES_EXT_IP>esp icmp gre</FW_SERVICES_EXT_IP>
  <FW_MASQUERADE>yes</FW_MASQUERADE>
</firewall>
```

例 D10：打开端口（SLE 15 语法）

```
<firewall>
  <zones config:type="list">
    <zone>
      <name>dmz</name>
      <ports config:type="list">
        <port>ftp/tcp</port>
        <port>ssh/tcp</port>
        <port>80/tcp</port>
        <port>5900-5999/tcp</port>
      </ports>
    </zone>
    <zone>
      <name>external</name>
      <ports config:type="list">
        <port>1723/udp</port>
        <port>ipsec-nat-t/udp</port>
      </ports>
      <protocols config:type="list">
        <protocol>esp</protocol>
      </protocols>
    </zone>
  </zones>
</firewall>
```

```
<protocol>icmp</protocol>
<protocol>gre</protocol>
</protocols>
</zone>
</zones>
</firewall>
```

D5.3 打开 firewalld 服务

为了打开端口和/或协议的组合，SuSEfirewall2 提供了 `FW_CONFIGURATIONS_{EXT, DMZ, INT}` 标记，这些标记与 `firewalld` 中的服务等效。

例 D11：打开服务（已弃用的语法）

```
<firewall>
  <FW_CONFIGURATIONS_EXT>dhcp dhcpv6 samba vnc-server</FW_CONFIGURATIONS_EXT>
  <FW_CONFIGURATIONS_DMZ>ssh</FW_CONFIGURATIONS_DMZ>
</firewall>
```

例 D12：打开服务（SLE 15 语法）

```
<firewall>
  <zones config:type="list">
    <zone>
      <name>dmz</name>
      <services config:type="list">
        <service>ssh</service>
      </services>
    </zone>
    <zone>
      <name>public</name>
      <services config:type="list">
        <service>dhcp</service>
        <service>dhcpv6</service>
        <service>samba</service>
        <service>vnc-server</service>
      </services>
    </zone>
  </zones>
</firewall>
```



```
</zones>
</firewall>
```

在以下两种情况下，可以通过软件包添加服务定义：

- SuSEfirewall2 服务定义：https://en.opensuse.org/SuSEfirewall2/Service_Definitions_Added_via_Packages ↗
- `firewalld` RPM 打包 https://en.opensuse.org/firewalld/RPM_Packaging ↗
`firewalld` 已经为 `/usr/lib/firewalld/services` 中的大多数重要服务提供支持。
在定义新配置之前，请先检查此目录中的现有配置。

D5.4 更多信息

- 适用于 SLE 12 的 SuSEfirewall2/AutoYaST 文档 (<https://documentation.suse.com/sles-12/html/SLES-all/configuration.html#CreateProfile-firewall>) ↗
- 官方 `firewalld` 文档 (<https://www.firewalld.org/documentation/>) ↗

D6 NTP 配置

时间服务器同步守护程序 `ntpd` 已被更新式的守护程序 `chrony` 取代。因此，AutoYaST 中计时守护程序的配置语法已更改。来自 SLE12 的 AutoYaST 配置文件（其中某个部分包含 `ntp:client`）需要更新。

现在，NTP 的配置方式不是包含低级配置选项，而是在默认设置之上应用一组高级选项：

例 D13：NTP 配置（SLE 15 语法）

```
<ntp-client>
  <ntp_policy>auto</ntp_policy>
  <ntp_servers config:type="list">
    <ntp_server>
      <iburst config:type="boolean">false</iburst>
      <address>cz.pool.ntp.org</address>
      <offline config:type="boolean">true</offline>
```

```
</ntp_server>
</ntp_servers>
<ntp_sync>systemd</ntp_sync>
</ntp-client>
```

D7 在第二阶段需要 AutoYaST 软件包

常规安装是在一个阶段中执行的，而通过 AutoYaST 执行的安装通常需要两个阶段。要执行安装的第二阶段，AutoYaST 需要几个额外的软件包，例如 autoyast2-installation 和 autoyast2。如果缺少这些软件包，将显示警告。

D8 已去除 CA 管理模块

用于 CA 管理的模块 (yast2-ca-management) 已从 SUSE Linux Enterprise Server 15 中去除，目前尚无替代模块。如果您要重复使用 SLE12 配置文件，请确保它不包含 ca_mgm 部分。

D9 升级

D9.1 软件

SLE 12 提供两种模式用于评估哪些软件包需要升级。在 SUSE Linux Enterprise Server 15 SP7 中，升级与否始终由依赖关系解析程序决定，相当于使用 **zypper dup**。

这样会使 `software` 部分中的 only_installed_packages 选项过时。

D9.2 注册

升级已注册的系统时，将去除所有旧储存库。这是为了避免新旧储存库之间可能发生的冲突，以及清理已删除产品的储存库。如果您需要保留自定义储存库，请使用 add-on 选项重新添加它们。

例 D14：升级的精简注册配置

```
<suse_register>
  <do_registration config:type="boolean">true</do_registration>
</suse_register>
```

如果注册服务器返回多个可能的迁移目标，AutoYaST 将自动选择第一个目标。目前您无法选择其他迁移目标。

升级未注册的系统或通过省略 suse_register 选项跳过注册升级后，您可能需要手动调整储存库设置。

This appendix contains the GNU Free Documentation License version 1.2.

GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role

of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document

within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.