



**SUSE Linux Enterprise Server 15 SP2**

# **AutoYaST 指南**

## AutoYaST 指南

### SUSE Linux Enterprise Server 15 SP2

AutoYaST 系统可用于以无人照管的方式批量部署 SUSE Linux Enterprise Server 系统。通过结合使用自定义安装和配置数据与 AutoYaST 控制文件（也称为“配置文件”）来执行 AutoYaST 安装。

出版日期：2024 年 12 月 12 日

<https://documentation.suse.com> 

版权所有 © 2006– 2024 SUSE LLC 和贡献者。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/或修改本文档的许可权限；本版权声明和许可证附带不可变部分。许可证版本 1.2 的副本包含在题为“GNU 自由文档许可证”的部分。

有关 SUSE 商标，请参见 <https://www.suse.com/company/legal/> 。所有其他第三方商标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

<b>1</b>	<b>AutoYaST 简介 1</b>
1.1	动机 1
1.2	概述和概念 1
I	了解和创建 AUTOYAST 控制文件 4
<b>2</b>	<b>AutoYaST 控制文件 5</b>
2.1	简介 5
2.2	格式 5
2.3	结构 7
	资源和属性 7 · 嵌套式资源 7 · 属性 8
<b>3</b>	<b>创建 AutoYaST 控制文件 10</b>
3.1	收集信息 10
3.2	使用配置管理系统 (CMS) 10
	创建新控制文件 11
3.3	手动创建/编辑控制文件 12
3.4	使用 XSLT 通过脚本创建控制文件 13
II	AUTOYAST 配置示例 15
<b>4</b>	<b>配置和安装选项 16</b>
4.1	一般选项 16
	Mode 部分 17 · 配置 ‘安装设置’ 屏幕 21 · 自我更新部 分 21 · Semi-Automatic 部分 23 · 签名处理部分 23 · Wait

- 部分 26 · 将 IBM Z 上未用的设备加入黑名单 27 · 有关 general 部分的示例 27
- 4.2 报告 30
- 4.3 系统注册和扩展选择 31
  - 扩展 35
- 4.4 引导装载程序 37
  - 加载程序类型 37 · 全局 38 · 设备映射 42
- 4.5 分区 43
  - 自动分区 43 · 引导式分区 44 · 专家分区 45 · 高级分区功能 62 · 逻辑卷管理器 (LVM) 66 · 软件 RAID 68 · 多路径支持 75 · bcache 配置 76 · 多设备 Btrfs 配置 79 · NFS 配置 81 · IBM Z 特定的配置 82
- 4.6 iSCSI 发起端概述 83
- 4.7 以太网光纤通道 (FCoE) 配置 84
- 4.8 国家/地区设置 86
- 4.9 软件 88
  - 产品选择 88 · 使用 Patterns 和 Packages 部分选择软件包 89 · 安装其他/自定义的软件包或产品 90 · 内核软件包 97 · 去除自动选择的软件包 98 · 安装建议的软件包/模式 98 · 在第 2 个阶段安装软件包 99 · 在第 2 个阶段安装模式 99 · 在第 2 个阶段联机更新 99
- 4.10 升级 100
- 4.11 服务和目标 101
- 4.12 网络配置 102
  - 接口 106 · 网络接口的持久名称 110 · 域名系统 111 · 路由选择 112 · s390 选项 113 · 代理 113
- 4.13 NIS 客户端和服务 114

- 4.14 NIS 服务器 115
- 4.15 主机定义 117
- 4.16 Windows 域成员资格 118
- 4.17 Samba 服务器 119
- 4.18 身份验证客户端 120
- 4.19 NFS 客户端和服务端 121
- 4.20 NTP 客户端 122
- 4.21 邮件服务器配置 124
- 4.22 Apache HTTP 服务器配置 126
- 4.23 Squid 服务器 136
- 4.24 FTP 服务器 145
- 4.25 TFTP 服务器 149
- 4.26 首次引导工作流程 149
- 4.27 安全性设置 150
  - 口令设置选项 150 • 引导设置 151 • 登录设置 151 • 新用户设置 (useradd 设置) 151
- 4.28 Linux 审计框架 (LAF) 151
- 4.29 用户和组 154
  - 用户 154 • 用户默认值 159 • 组 161 • 登录设置 162
- 4.30 自定义用户脚本 163
  - 安装前脚本 163 • 分区后脚本 164 • Chroot 环境脚本 165 • 安装后脚本 165 • Init 脚本 165 • 脚本 XML 表示形式 167 • 脚本示例 170
- 4.31 系统变量 (Sysconfig) 173

- 4.32 添加完整配置 174
- 4.33 安装期间请求用户提供值 175
  - 默认值脚本 182 · 脚本 183
- 4.34 内核转储 189
  - 内存预留 190 · 转储保存 192 · 电子邮件通知 194 · Kdump 内核设置 196 · 专家设置 197
- 4.35 DNS 服务器 198
- 4.36 DHCP 服务器 200
- 4.37 防火墙配置 204
  - 一般防火墙配置 204 · 防火墙区域配置 205 · 完整示例 206
- 4.38 其他硬件和系统组件 207
  - 打印机 207 · 声音设备 208
- 4.39 导入 SSH 密钥和配置 209
- 4.40 配置管理 210
  - 连接到配置管理服务器 211 · 以独立模式运行 212 · SUSE Manager Salt 公式支持 213

### III 使用规则和类管理批量安装 214

## 5 规则和类 215

- 5.1 基于规则的自动安装 215
  - 规则文件说明 216 · 自定义规则 219 · 规则的匹配类型 220 · 合并属性 220 · 规则文件结构 221 · 预定义的系统属性 221 · 使用对话框选择规则 223
- 5.2 类 227
- 5.3 混合规则和类 228
- 5.4 合并规则和类 229

## IV 了解自动安装过程 232

### 6 自动安装过程 233

#### 6.1 简介 233

X11 界面（图形） 233 • 串行控制台 233 • 基于文本的 YaST 安装 233

#### 6.2 选择正确的引导媒体 234

从闪存盘（例如 USB 记忆棒）引导 234 • 从 SUSE Linux Enterprise 安装媒体引导 235 • 通过 PXE 经由网络引导 235

#### 6.3 调用自动安装过程 236

命令行选项 236 • 自动安装单个系统 242 • 将 `linuxrc info` 文件与 AutoYaST 控制文件结合使用 242

#### 6.4 系统配置 243

安装后和系统配置 243 • 系统自定义 244

## V 在已安装系统上使用 AUTOYAST 245

### 7 在安装的系统中运行 AutoYaST 246

## VI 附录 248

### A 处理规则 249

### B AutoYaST FAQ — 常见问题 250

### C 高级 `linuxrc` 选项 254

#### C.1 将参数传递给 `linuxrc` 255

#### C.2 `info` 文件格式 255

#### C.3 高级网络设置 257



## **D SLE 12 与 15 中 AutoYaST 配置文件之间的差异 259**

### **D.1 产品选择 259**

### **D.2 软件 259**

使用注册服务器添加模块或扩展 260 • 使用 SLE-15-SP2-Full-ARCH-GM-media1.iso 映像添加模块或扩展 261 • 已重命名的软件模式 261

### **D.3 模块和扩展依赖项的注册 262**

### **D.4 分区 263**

GPT 已成为 AMD64/Intel 64 上的默认分区类型 263 • 设置分区号 263 • 强制主分区 263 • Btrfs: 默认子卷名称 263 • Btrfs: 禁用子卷 264 • 不再支持读取现有的 /etc/fstab 264 • 已删除用于对齐分区的设置 265

### **D.5 防火墙配置 265**

将接口指派给区域 266 • 打开端口 269 • 打开 firewalld 服务 270 • 更多信息 272

### **D.6 NTP 配置 272**

### **D.7 在第二阶段需要 AutoYaST 软件包 272**

### **D.8 已删除 CA 管理模块 273**

### **D.9 升级 273**

软件 273 • 注册 273

# 1 AutoYaST 简介

## 1.1 动机

SUSE Linux Enterprise Server 的标准安装基于一个向导工作流程。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。

SUSE Linux Enterprise Server 的常规安装默认是半自动化的。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 控制文件来准备系统的自定义安装，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。最初可以使用现有的配置资源（可根据任何特定环境轻松定制）创建该配置文件。

AutoYaST 是完全集成的，它提供多种选项用于安装和配置系统。相比其他自动安装系统，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控制文件的相应指令。控制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。
- 安装：YaST 使用 AutoYaST 控制文件中的数据执行基本系统的安装。
- 配置：安装基本系统后，将在第二个安装阶段执行系统配置。AutoYaST 控制文件中用户定义的安装后脚本也在此阶段执行。



## 注意：第二个阶段

常规 SUSE Linux Enterprise Server 15 SP2 安装在一个阶段内执行。而自动安装过程则分为两个阶段。安装基本系统后，系统将引导进入第二个阶段，系统配置在此阶段完成。需要安装软件包 `autoyast2` 和 `autoyast2-installation` 才能在安装的系统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。

可以使用 `second_stage` 参数关闭第二个阶段：

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

下图说明了完整详细的过程：

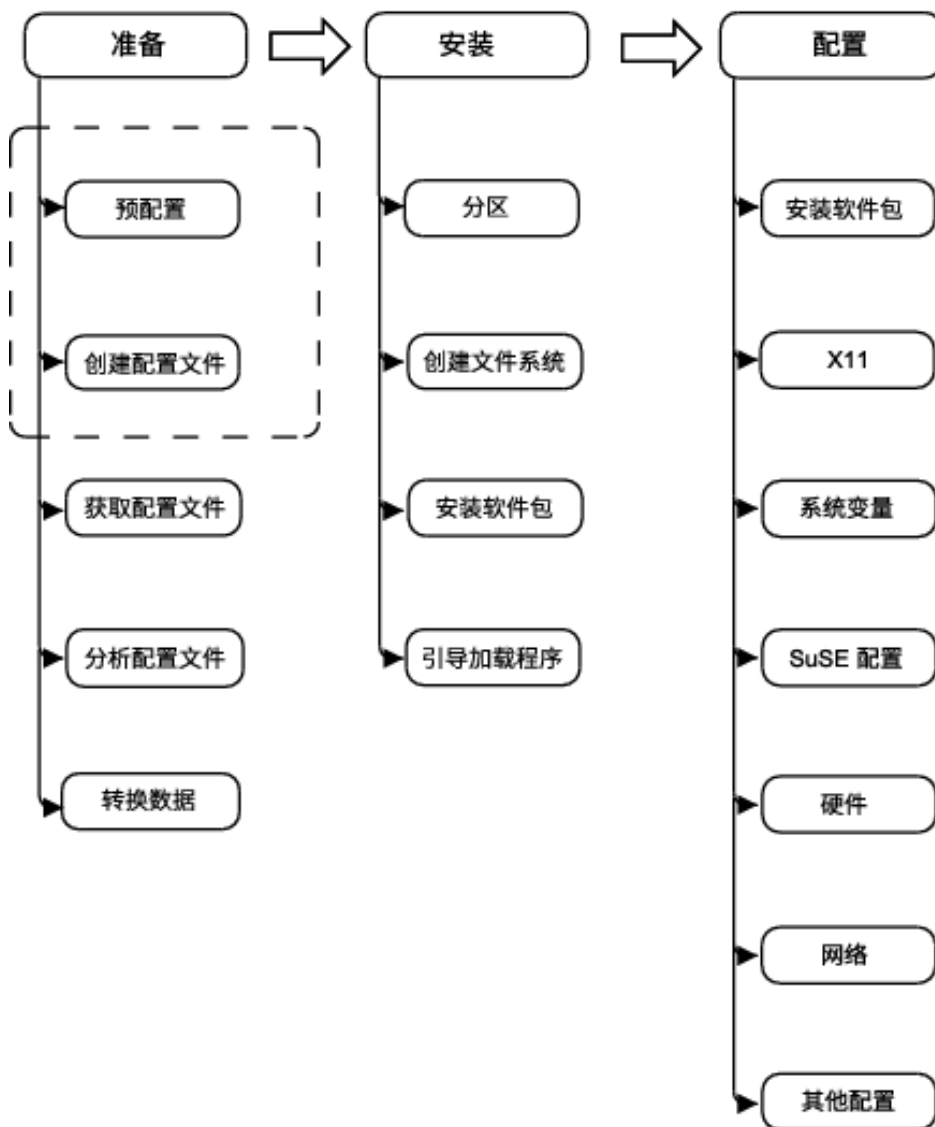


图 1.1：自动安装过程

# I 了解和创建 AutoYaST 控制文件

- 2 AutoYaST 控制文件 **5**
- 3 创建 AutoYaST 控制文件 **10**

## 2 AutoYaST 控制文件

### 2.1 简介

控制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。

#### 重要：来自早于 SLES 12 GA 版本的操作系统的控制文件与 openSUSE 42.0 不兼容

SLES 12 和 openSUSE Leap 42.0 中引入了许多重大更改，例如，切换到 systemd 和 GRUB 2 的功能。这些更改同样需要在 AutoYaST 中进行根本性的更改。因此，您无法使用 SLES 11 上创建的 AutoYaST 控制文件来安装 SUSE Linux Enterprise Server 15 SP2，反之亦然。

### 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。如果您要编辑现有的控制文件，或者要使用编辑器从头开始创建一个控制文件，我们强烈建议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/configs">
  <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/configns">
  <!-- 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。

空元素（例如 <foo></foo> 或 <bar/>）不会在分析的数据模型中存在。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。必须使用 `config:type="list"` 属性将列表标记为列表类型。

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

属性并非可选的。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）

另外，请校验是否安装 [`autoyast2-installation`](#) 和 [`autoyast2`](#)。

### 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。

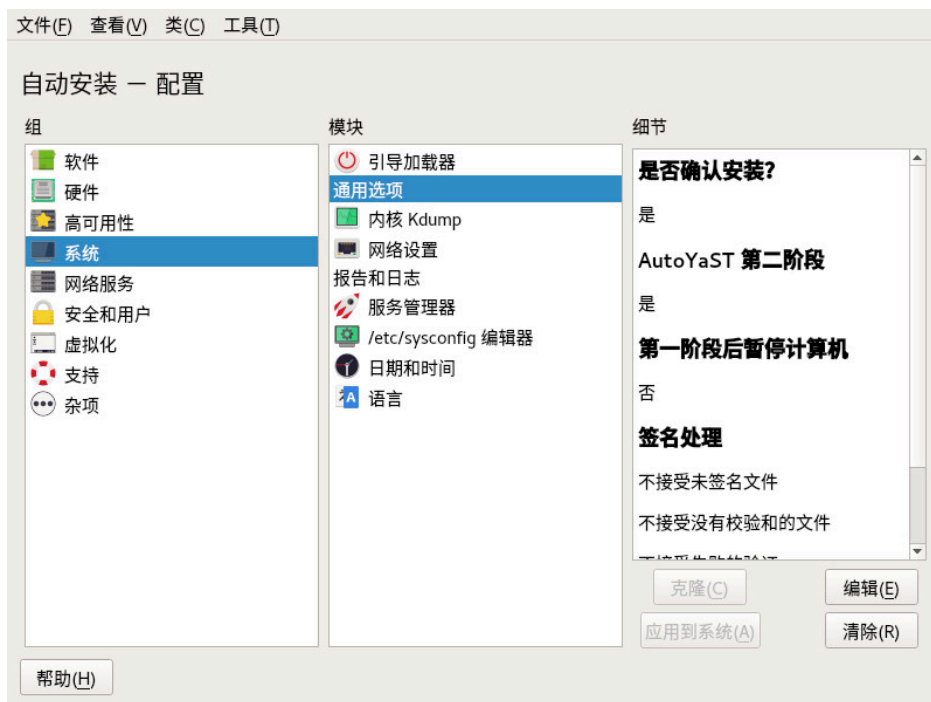


图 3.1：配置系统

### 3.2.1 创建新控制文件

创建 AutoYaST 配置文件的最简单方法是使用现有 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。

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`），然后使用 YaST 控制中心或者以 root 身份使用以下命令来调用该系统（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 `libxml2` 软件包中提供的实用程序 `xmllint`：

```
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。软件包 `jing` 是随 SUSE Software Development Kit 一起提供的。

```
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` 提供。此文件描述 AutoYaST 配置文件的语法和类。

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，为许多计算机手动创建控制文件并不是最好的做法；控制文件仅应看作是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/configs"
  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](http://www.w3.org/TR/xslt) (<http://www.w3.org/TR/xslt>) 

## II AutoYaST 配置示例

### 4 配置和安装选项 16



## 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>①
      ...
    </ask-list>
    <cio_ignore>②
      ...
    </cio_ignore>
    <mode>③
      ...
    </mode>
    <proposals>④
      ...
    </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 4.33 节 “安装期间请求用户提供值”
- ② 第 4.1.7 节 “将 IBM Z 上未用的设备加入黑名单”
- 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 “自我更新部分”
- ⑥ 第 4.1.4 节 “Semi-Automatic 部分”
- ⑦ 第 4.1.5 节 “签名处理部分”
- ⑧ 第 4.5 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。允许将以下元素用于 mode 部分：

## activate\_systemd\_default\_target

如果您将此项设置为 false，则不会通过调用 **systemctl isolate** 激活默认的 systemd 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，会在出现安装设置屏幕时停止安装。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。默认设置为 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 4.9.3 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，则在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，则在安装并配置所有组件后（第二个阶段结束时），计算机将重引导。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，则在安装结束时（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。默认设置为 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>
      ntp.example.com
    </max_systemd_wait>
  </mode>
  ...
</general>
```

## second\_stage

常规 SUSE Linux Enterprise Server 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 标志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 自我更新部分

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：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》。

## ！ 重要：仅提供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上连同 `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。安装计算机的管理员可以通过此便利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，您可以定义在运行特定模块之前和之后要执行的脚本。还可以在每个模块的前面和后面配置系统处于非活动状态（“休眠”）的时间跨度。

### 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 将 IBM Z 上未用的设备加入黑名单

在 IBM Z 上，您可以通过运行 **cio\_ignore** 并将未用的硬件设备加入黑名单，来防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。默认设置为 **false**。

```

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

```

### 4.1.8 有关 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/configs">

```

```

<general>
  <!-- Use cio_ignore on IBM Z only -->
  <cio_ignore config:type="boolean">false</cio_ignore>
  <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>
  <storage>

```

```

    <partition_alignment config:type="symbol">
      align_cylinder
    </partition_alignment>
  </storage>
</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>>![CDATA[
echo "Sleeping 10 seconds"
      ]]></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>>![CDATA[
echo "Sleeping 10 seconds"
      ]]></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答这些消息。

## 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。如果您的组织提供有自己的注册服务器，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>15.2</version>
      <arch>x86_64</arch>
    </addon>
  </addons>
</suse_register>
```

建议至少注册 Basesystem 模块，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>sc</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>
```

元素	说明	注释
<u>do_registration</u>	布尔  <pre>&lt;do_registration   config:type="boolean" &gt;true&lt;/do_registration&gt;</pre>	指定是否应注册系统。如果设置为 <code>false</code> ，则会忽略所有其他选项，并且不注册系统。
<u>e-mail</u>	电子邮件地址  <pre>&lt;email&gt;tux@example.com&lt;/email&gt;</pre>	可选。与注册代码匹配的电子邮件地址。
<u>reg_code</u>	文本  <pre>&lt;reg_code&gt;SECRET_REGCODE&lt;/ reg_code&gt;</pre>	必要。注册代码。
<u>install_updates</u>	布尔  <pre>&lt;install_updates   config:type="boolean" &gt;true&lt;/install_updates&gt;</pre>	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 ( <code>false</code> )。

元素	说明	注释
<u>slp_discovery</u>	布尔 <pre>&lt;slp_discovery   config:type="boolean" &gt;true&lt;/slp_discovery&gt;</pre>	<p>可选。通过 SLP 搜索注册服务器。默认值为 <u>false</u>。</p> <p>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 <u>reg_server</u> 指定一个服务器。</p> <p>如果 <u>slp_discovery</u> 和 <u>reg_server</u> 都未设置，将在 SUSE Customer Center 中注册系统。</p> <p>此设置也会影响自我更新功能：如果禁用此设置，将不执行 SLP 搜索。</p>
<u>reg_server</u>	URL <pre>&lt;reg_server&gt;   https://smt.example.com &lt;/reg_server&gt;</pre>	<p>可选。RMT 服务器 URL。如果 <u>slp_discovery</u> 和 <u>reg_server</u> 都未设置，将在 SUSE Customer Center 中注册系统。</p> <p>将从 RMT 服务器查询自我更新储存库的 URL。因此，如果未设置 <u>self_update_</u></p>

元素	说明	注释
		<a href="#">url</a> ，RMT 服务器将影响自我更新的下载位置。有关此功能的更多信息，请查看部署指南。
<a href="#">reg_server_cert_fingerprint_type</a>	SHA1 或 SHA256  <pre>&lt;reg_server_cert_fingerprint_type&gt;   SHA1 &lt;/ reg_server_cert_fingerprint_type&gt;</pre>	可选。需要通过 <a href="#">reg_server_cert_fingerprint</a> 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。
<a href="#">reg_server_cert_fingerprint</a>	采用十六进制表示法（不区分大小写）的服务器证书指纹值。  <pre>&lt;reg_server_cert_fingerprint&gt;   01:AB...:EF &lt;/reg_server_cert_fingerprint&gt;</pre>	可选。需要通过 <a href="#">reg_server_cert_fingerprint_type</a> 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。
<a href="#">reg_server_cert</a>	URL  <pre>&lt;reg_server_cert&gt;   http://smt.example.com/smt.crt &lt;/reg_server_cert&gt;</pre>	可选。服务器上 SSL 证书的 URL。不建议使用此选项，因为不会校

元素	说明	注释
		验下载的证书。请改用 <code>reg_server_cert_fingerprint</code> 。
<code>addons</code>	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。

注意：只能在可信网络中使用此命令！在不可信的网络（例如因特网）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`（开发工具模块），在安装期间，可作为附加源包含这些扩展。可通过 `suse_register` 块中的 `addons` 属性添加扩展。



## 注意：扩展可用性

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`** 可以列出已注册系统中所有可用的扩展。结果包含如下几行：

```
Install with: SUSEConnect -p sle-module-development-tools/15.2/x86_64
```

`-p` 参数显示可在 AutoYaST 配置文件中使用的 `NAME/VERSION/ARCH` 值，如下所示：

```
<addons config:type="list">
  <addon>
    <!-- Development Tools Module -->
    <name>sle-module-development-tools</name>
    <version>15.2</version>
    <arch>x86_64</arch>
  </addon>
</addons>
```



## 注意：扩展依赖项

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。

有关背景信息和 SLES 的附加产品列表，请参见 <https://github.com/yast/yast-registration/wiki/Available-SCC-Extensions-for-Use-in-Autoyast>。附加产品列表会不时更新。

## 4.4 引导装载程序

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

AutoYaST 引导加载程序部分的一般结构如下所示：

```
<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>
```

### 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：安装程序选择正确的引导加载程序。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，如果您未定义配置，`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。

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

属性	说明
<u>activate</u>	<p>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 <code>/</code>。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。</p> <pre>&lt;activate   config:type="boolean"&gt;true&lt;/ activate&gt;</pre>
<u>append</u>	<p>在常规和恢复模式的引导项末尾添加的内核参数。</p> <pre>&lt;append&gt;nomodeset vga=0x317&lt;/append&gt;</pre>
<u>boot_boot</u>	<p>将 GRUB 2 写入单独的 /boot 分区。如果不存在单独的 /boot 分区，则将 GRUB 2 写入 <code>/</code>。</p> <pre>&lt;boot_boot&gt;&gt;false&lt;/boot_boot&gt;</pre>

属性	说明
<u>boot_custom</u>	<p>将 GRUB 2 写入自定义设备。</p> <pre>&lt;boot_custom&gt;/dev/sda3&lt;/boot_custom&gt;</pre>
<u>boot_extended</u>	<p>将 GRUB 2 写入扩展分区（如果您想要使用通用引导代码，而 <u>/boot</u> 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 <u>boot_mbr</u>（将 GRUB 2 写入 MBR）而不是 <u>generic_mbr</u>。</p> <pre>&lt;boot_extended&gt;&gt;false&lt;/boot_extended&gt;</pre>
<u>boot_mbr</u>	<p>将 GRUB 2 按顺序写入第一个磁盘的 MBR（device.map 包含磁盘顺序）。</p> <pre>&lt;boot_mbr&gt;&gt;false&lt;/boot_mbr&gt;</pre>
<u>boot_root</u>	<p>将 GRUB 2 写入 <u>/</u> 分区。</p> <pre>&lt;boot_root&gt;&gt;false&lt;/boot_root&gt;</pre>
<u>generic_mbr</u>	<p>将通用引导代码写入 MBR（如果 <u>boot_mbr</u> 设置为 <u>true</u>，将忽略此属性）。</p> <pre>&lt;generic_mbr   config:type="boolean"&gt;&gt;false&lt;/ generic_mbr&gt;</pre>
<u>gfxmode</u>	<p>GRUB 2 屏幕的图形分辨率（要求将 <u>&lt;terminal&gt;</u> 设置为 <u>gfxterm</u>）。有效项为 <u>auto</u>、<u>HORIZONTALxVERTICAL</u> 或 <u>HORIZONTALxVERTICALxCOLOR DEPTH</u>。</p>



属性	说明
	<p>您可以在运行中系统的 GRUB 2 命令行上，使用 <b>vbeinfo</b> 命令来查看特定系统上的 GRUB 2 支持的屏幕分辨率。</p> <pre>&lt;gfxmode&gt;1280x1024x24&lt;/gfxmode&gt;</pre>
<u>os_prober</u>	<p>如果设置为 <b>true</b>，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</p> <pre>&lt;os_prober   config:type="boolean"&gt;false&lt;/ os_prober&gt;</pre>
<u>cpu_mitigations</u>	<p>允许选择 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：</p> <p><b>auto</b>： 启用您的 CPU 型号所需的全部缓解措施，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。</p> <p><b>nosmt</b>： 提供整套可用的安全缓解措施。启用您的 CPU 型号所需的全部缓解措施。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。</p> <p><b>off</b>： 禁用所有缓解措施。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。</p>

属性	说明
	<p><u>manual</u>：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。</p> <pre>&lt;cpu_mitigations&gt;auto&lt;/cpu_mitigations&gt;</pre> <p>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 <u>/control.xml</u> 文件中的（特定于产品的）设置（如果未指定其他设置）。</p>
<u>suse_btrfs</u>	已过时，不再使用。将自动启用从 Btrfs 快照引导。
<u>serial</u>	<p>当 GRUB 2 终端模式设置为 <u>serial</u> 时要执行的命令。</p> <pre>&lt;serial&gt;   serial --speed=115200 --unit=0 -- word=8 --parity=no --stop=1 &lt;/serial&gt;</pre>
<u>secure_boot</u>	<p>如果设置为 <u>false</u>，则禁用 UEFI 安全引导。仅适用于 <u>grub2-efi</u> 引导加载程序。</p> <pre>&lt;secure_boot"&gt;false&lt;/secure_boot&gt;</pre>
<u>terminal</u>	<p>指定要使用的 GRUB 2 终端模式，有效项为 <u>console</u>、<u>gfxterm</u> 和 <u>serial</u>。如果设置为 <u>serial</u>，则还需要使用 <u>&lt;serial&gt;</u> 指定串行命令。</p> <pre>&lt;terminal&gt;serial&lt;/terminal&gt;</pre>

属性	说明
<u>timeout</u>	<p>在自动引导默认引导项之前经过的超时（以秒为单位）。</p> <pre>&lt;timeout config:type="integer"&gt;10&lt;/timeout&gt;</pre>
<u>trusted_boot</u>	<p>如果设置为 <u>true</u>，则使用可信 GRUB。可信 GRUB 支持可信平台模块 (TPM)。仅适用于 <u>grub2</u> 引导加载程序。</p> <pre>&lt;trusted_boot"&gt;true&lt;/trusted_boot&gt;</pre>
<u>vgamode</u>	<p>将内核参数 <u>vga=VALUE</u> 添加到引导项。</p> <pre>&lt;vgamode&gt;0x317&lt;/vgamode&gt;</pre>
<u>xen_append</u>	<p>在 Xen Guest 的引导项末尾添加的内核参数。</p> <pre>&lt;xen_append&gt;nomodeset vga=0x317&lt;/xen_append&gt;</pre>
<u>xen_kernel_append</u>	<p>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。</p> <pre>&lt;xen_kernel_append&gt;dom0_mem=768M&lt;/xen_kernel_append&gt;</pre>

### 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 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信任 YaST、使用引导式建议，或通过专家分区程序定义分区布局。

### 4.5.1 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 通常会建议使用 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，是否具有独立的 /home 取决于可用磁盘空间量。

如果您想要影响这些默认值，可以使用第 4.5.2 节“引导式分区”中所述的方法。

## 4.5.2 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。

**例 4.3 “基于 LVM 的引导式分区”** 中的 `storage` 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。默认设置为 `false`。

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

#### 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>
```

## 4.5.3 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 无需知道每个细节就能基于一个不甚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>

```



## 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。

### 4.5.3.1 驱动器配置

下面列出的元素必须放在以下 XML 结构中：

```

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

```

属性	值	说明
<u>device</u>	<p>您要在此部分配置的设备。可以通过 ID 使用持久设备名称（例如 <code>/dev/disk/by-id/ata-WDC_WD3200AAKS-75L9A0_WD-WMAV27368122</code>），或者通过 <code>by-path</code> 使用持久设备名称（例如 <code>/dev/disk/by-path/pci-0001:00:03.0-scsi-0:0:0:0</code>）。</p> <pre>&lt;device&gt;/dev/sda&lt;/device&gt;</pre> <p>对于卷组、软件 RAID 或 <code>bcache</code> 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 4.5.7 节“多路径支持”。</p>	<p>可选。如果留空，AutoYaST 将尝试猜测设备。有关如何影响猜测，请参见<a href="#">提示：跳过设备</a>。</p> <p>如果设置为 <code>ask</code>，AutoYaST 将在安装期间询问用户要使用哪个设备。</p>
<u>initialize</u>	<p>如果设置为 <code>true</code>，在 AutoYaST 启动分区计算之前将擦除分区表。</p> <pre>&lt;initialize   config:type="boolean"&gt;true&lt;/initialize&gt;</pre>	<p>可选。默认设置为 <code>false</code>。</p>
<u>partitions</u>	<p><code>&lt;partition&gt;</code> 项的列表（请参见第 4.5.3.2 节“分区配置”）。</p> <pre>&lt;partitions config:type="list"&gt;   &lt;partition&gt;...&lt;/partition&gt;   ... &lt;/partitions&gt;</pre>	<p>可选。如果未指定任何分区，AutoYaST 将创建合理的分区（请参见第 4.5.3.5 节“填补缺失”）。</p>



属性	值	说明
	<code>&lt;/partitions&gt;</code>	
<u>pesize</u>	<p>此值仅对 LVM 有意义。</p> <pre>&lt;pesize&gt;8M&lt;/pesize&gt;</pre>	可选。对于 LVM 卷组，默认值为 4M。
<u>use</u>	<p>指定 AutoYaST 将用于硬盘分区的策略。</p> <p>选择下列其中一项：</p> <ul style="list-style-type: none"> <li>• <u>all</u>（使用整个设备，同时计算新分区）；</li> <li>• <u>linux</u>（仅使用现有的 Linux 分区）；</li> <li>• <u>free</u>（仅使用设备上未用的空间，其他分区都保留原样）；</li> <li>• 1,2,3（要使用的分区编号的逗号分隔列表）。</li> </ul>	应提供此参数。
<u>type</u>	<p>指定 <u>drive</u> 的类型。</p> <p>选择下列其中一项：</p> <ul style="list-style-type: none"> <li>• <u>CT_DISK</u>（物理硬盘，默认值）；</li> <li>• <u>CT_LVM</u>（LVM 卷组）；</li> <li>• <u>CT_RAID</u>（软件 RAID 设备）；</li> <li>• <u>CT_BCACHE</u>（软件 <u>bcache</u> 设备）。</li> </ul>	可选。默认值为 <u>CT_DISK</u> ，表示普通的物理硬盘。

属性	值	说明
	<pre>&lt;type   config:type="symbol"&gt;CT_LVM&lt;/type&gt;</pre>	
<u>disklabel</u>	<p>描述分区表的类型。</p> <p>选择下列其中一项：</p> <ul style="list-style-type: none"> <li>• <u>msdos</u></li> <li>• <u>gpt</u></li> <li>• <u>none</u></li> </ul> <pre>&lt;disklabel&gt;gpt&lt;/disklabel&gt;</pre>	<p>可选。默认情况下，YaST 会确定有意义的值。如果存在不同类型的分区表，仅当该分区表不包含任何应该保留或重复使用的分区时，才会使用给定的类型重新创建该分区表。要使用磁盘而不创建任何分区，请将此元素设置为 <u>none</u>。</p>
<u>keep_unknown_lv</u>	<p>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 <u>true</u>，该组中的所有现有逻辑卷将保留原样，除非在 &lt;partitioning&gt; 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。</p> <pre>&lt;keep_unknown_lv   config:type="boolean"   &gt;false&lt;/keep_unknown_lv&gt;</pre>	<p>可选。默认设置为 <u>false</u>。</p>
<u>enable_snapshots</u>	<p>在装入到 <u>/</u> 的 Btrfs 文件系统上启用快照（不适用于其他文件系统，或未装入到 <u>/</u> 的 Btrfs 文件系统）。</p>	<p>可选。默认值为 <u>true</u>。</p>

属性	值	说明
	<pre>&lt;enable_snapshots   config:type="boolean"     &gt;false&lt;/ enable_snapshots&gt;</pre>	

## ！ 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`。

4.5.3.2 分区配置

下面列出的元素必须放在以下 XML 结构中：

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

属性	值	说明
<u>create</u>	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。 <div>&lt;create config:type="boolean" &gt;false&lt;/create&gt;</div>	如果设置为 <u>false</u> ，则您还需要设置 <u>partition_nr</u> 、 <u>lv_name</u> 、 <u>label</u> 或 <u>uuid</u> 其中之一，以告知 AutoYaST 要使用哪个设备。
<u>crypt_method</u>	将使用以下方法之一加密分区：	可选。加密方法选择是在 SUSE Linux Enterprise Server 15

属性	值	说明
	<ul style="list-style-type: none"> <li>• <u>luks1</u>：常规 LUKS1 加密。</li> <li>• <u>pervasive_luks2</u>：普遍加密。</li> <li>• <u>protected_swap</u>：使用非永久保护的密钥进行加密。</li> <li>• <u>secure_swap</u>：使用非永久的安全密钥进行加密。</li> <li>• <u>random_swap</u>：使用非永久的随机密钥进行加密。</li> </ul> <pre>&lt;crypt_method config:type="symbol"&gt;luks1&lt;/crypt_method&gt;</pre>	<p>SP2 中引入的。要模仿先前版本的行为，请使用 <u>luks1</u>。</p> <p>如果需要，请查看 <u>crypt_key</u> 元素来确定如何指定加密口令。</p>
<u>crypt_fs</u>	<p>将加密分区。此元素已弃用。请改用 <u>crypt_method</u>。</p> <pre>&lt;crypt_fs config:type="boolean"&gt;true&lt;/crypt_fs&gt;</pre>	默认为 <u>false</u> 。
<u>crypt_key</u>	<p>加密密钥</p> <pre>&lt;crypt_key&gt;xxxxxxxx&lt;/crypt_key&gt;</pre>	如果 <u>crypt_method</u> 已设置为需要口令的方法（例如 <u>luks1</u> 或 <u>pervasive_luks2</u> ），则需要使用此元素。
<u>mount</u>	<p>此分区的安装点。</p> <pre>&lt;mount&gt;/&lt;/mount&gt; &lt;mount&gt;swap&lt;/mount&gt;</pre>	至少应有一个根分区 (/) 和一个交换分区。

属性	值	说明
<u>fstop</u>	此分区的装入选项。  <pre>&lt;fstopt&gt;     ro,noatime,user,data=ordered,acl,user_xattr &lt;/fstopt&gt;</pre>	有关可用装入选项，请参见 <a href="#">man mount</a> 。
<u>label</u>	分区的标签。格式化设备时很有用（尤其是 <a href="#">mountby</a> 参数设置为 <a href="#">label</a> 时），并可用于标识已存在的设备（请参见前面所述的 <a href="#">create</a> ）。  <pre>&lt;label&gt;mydata&lt;/label&gt;</pre>	有关示例，请参见 <a href="#">man e2label</a> 。
<u>uuid</u>	分区的 UUID。仅可用于标识现有设备（请参见前面所述的 <a href="#">create</a> ）。不能对新设备强制使用 UUID。  <pre>&lt;uuid&gt; &gt;1b4e28ba-2fa1-11d2-883f-b9a761bde3fb&lt;/uuid&gt;</pre>	请参见 <a href="#">man uuidgen</a> 。
<u>size</u>	分区的大小，例如 4G、4500M 等。/boot 分区和交换分区可以使用 <a href="#">auto</a> 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 <a href="#">max</a> 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，您可以混合指定 <a href="#">auto</a> 、 <a href="#">max</a> 、 <a href="#">size</a> 和百分比。  <pre>&lt;size&gt;10G&lt;/size&gt;</pre>	从 SUSE Linux Enterprise Server 15 开始，所有值（包括 <a href="#">auto</a> 和 <a href="#">max</a> ）也都可用于调整分区大小。
<u>format</u>	指定 AutoYaST 是否应格式化分区。  <pre>&lt;format config:type="boolean"&gt;false&lt;/format&gt;</pre>	如果您将 <a href="#">create</a> 设置为 <a href="#">true</a> ，则也可能想要将此选项设置为 <a href="#">true</a> 。

属性	值	说明
<u>file</u> <u>system</u>	<p>指定要在此分区上使用的文件系统：</p> <ul style="list-style-type: none"> <li>• <u>btrfs</u></li> <li>• <u>ext2</u></li> <li>• <u>ext3</u></li> <li>• <u>ext4</u></li> <li>• <u>fat</u></li> <li>• <u>xf</u>s</li> <li>• <u>SWAP</u></li> </ul> <pre>&lt;filesystem config:type="symbol" &gt;ext3&lt;/filesystem&gt;</pre>	可选。对于根分区 (/)，默认值为 <u>btrfs</u> ；对于数据分区，默认值为 <u>xf</u> s。
<u>mkfs_</u> <u>options</u>	<p>指定要添加到 mkfs 命令的选项字符串。</p> <pre>&lt;mkfs_options&gt;-I 128&lt;/mkfs_options&gt;</pre>	可选。请仅在您完全清楚自己所执行的操作时，才使用此元素。
<u>partition_</u> <u>nr</u>	<p>此分区的分区编号。如果您已设置 <u>create=false</u> 或者要使用 LVM，则您可以通过 <u>partition_nr</u> 指定分区。您可以强制 AutoYaST 仅通过指派 5 以下的编号来创建主分区。</p> <pre>&lt;partition_nr config:type="integer" &gt;2&lt;/partition_nr&gt;</pre>	通常，编号 1 到 4 是主分区，5 和更高的编号是逻辑分区。

属性	值	说明
<u>partition_id</u>	<p><u>partition_id</u> 用于设置分区的 ID。如果您想要为 Linux 分区指派除 131 以外的标识符，或者要为交换分区指派除 130 以外的标识符，请使用 <u>partition_id</u> 进行配置。</p> <pre>&lt;partition_id config:type="integer"&gt;131&lt;/partition_id&gt;</pre> <p>可能的值为：</p> <p>交换分区： <u>130</u></p> <p>Linux： <u>131</u></p> <p>LVM： <u>142</u></p> <p>MD RAID： <u>253</u></p> <p>EFI 分区： <u>259</u></p>	Linux 分区的默认值为 <u>131</u> ，交换分区的默认值为 <u>130</u> 。
<u>partition_type</u>	<p>使用 <u>msdos</u> 分区表时，此元素用于设置分区的类型。值可以是 <u>primary</u> 或 <u>logical</u>。使用 <u>gpt</u> 分区表时会忽略此值，因为在此情况下不存在这种差别。</p> <pre>&lt;partition_type&gt;primary&lt;/partition_type&gt;</pre>	可选。允许的值 <u>primary</u> （默认值）和 <u>logical</u> 。
<u>mountby</u>	<p>如果不使用分区编号，您可以告知 AutoYaST 按 <u>device</u>、<u>label</u>、<u>uuid</u>、<u>path</u> 或 <u>id</u>（udev 路径和 udev ID）装入分区（请参见 <u>/dev/disk/...</u>）。</p> <pre>&lt;mountby config:type="symbol"&gt;label&lt;/mountby&gt;</pre>	请参见上文有关 <u>label</u> 和 <u>uuid</u> 的说明。默认值取决于 YaST，通常是 <u>id</u> 。



属性	值	说明
<u>subvolumes</u>	<p>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。有关更多信息，请参见第 4.5.3.3 节 “Btrfs 子卷”。</p> <pre>&lt;subvolumes config:type="list"&gt;   &lt;path&gt;tmp&lt;/path&gt;   &lt;path&gt;opt&lt;/path&gt;   &lt;path&gt;srv&lt;/path&gt;   &lt;path&gt;var/crash&lt;/path&gt;   &lt;path&gt;var/lock&lt;/path&gt;   &lt;path&gt;var/run&lt;/path&gt;   &lt;path&gt;var/tmp&lt;/path&gt;   &lt;path&gt;var/spool&lt;/path&gt;   ... &lt;/subvolumes&gt;</pre>	如果未为分区描述定义 <u>subvolumes</u> 部分，AutoYaST 将为给定的安装点创建一组预定义的子卷。
<u>create_subvolumes</u>	确定是否应创建 Btrfs 子卷。	默认设置为 <u>true</u> 。如果设置为 <u>false</u> ，则不创建子卷。
<u>subvolumes_prefix</u>	<p>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：</p> <pre>&lt;subvolumes_prefix&gt;&lt;![CDATA[]]&gt;&lt;/subvolumes_prefix&gt;</pre>	默认设置为 <u>@</u> 。
<u>lv_name</u>	<p>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 <u>is_lvm_vg</u> 参数）。</p> <pre>&lt;lv_name&gt;opt_lv&lt;/lv_name&gt;</pre>	
<u>stripes</u>	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）。	

属性	值	说明
	<pre>&lt;stripes config:type="integer"&gt;2&lt;/stripes&gt;</pre>	
<u>stripesize</u>	指定每个块的大小，以 KB 为单位。  <pre>&lt;stripesize config:type="integer"&gt;4&lt;/stripesize&gt;</pre>	
<u>lvm_group</u>	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。  <pre>&lt;lvm_group&gt;system&lt;/lvm_group&gt;</pre>	
<u>pool</u>	如果 LVM 逻辑卷应是 LVM 精简池，则 <u>pool</u> 必须设 置为 <u>true</u> 。  <pre>&lt;pool config:type="boolean"&gt;&gt;false&lt;/pool&gt;</pre>	
<u>used_pool</u>	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。  <pre>&lt;used_pool&gt;my_thin_pool&lt;/used_pool&gt;</pre>	
<u>raid_name</u>	如果此物理卷是 RAID 的一部分，请指定该 RAID 的名 称。  <pre>&lt;raid_name&gt;/dev/md/0&lt;/raid_name&gt;</pre>	
<u>raid_</u> <u>options</u>	指定 RAID 选项，具体请参见下文。  <pre>&lt;raid_options&gt;...&lt;/raid_options&gt;</pre>	在 <u>partition</u> 级 别设置 RAID 选项的 功能已弃用。请参 见第 4.5.6 节 “软件 RAID”。

属性	值	说明
bcache_ backing_for	<p>如果此设备用作 <u>bcache</u> 后端设备，请指定 <u>bcache</u> 设备的名称。</p> <pre>&lt;bcache_backing_for&gt;/dev/bcache0&lt;/bcache_backing_for&gt;</pre>	有关更多详细信息，请参见第 4.5.8 节“ <u>bcache 配置</u> ”。
bcache_ caching_for	<p>如果此设备用作 <u>bcache</u> 缓存设备，请指定 <u>bcache</u> 设备的名称。</p> <pre>&lt;bcache_caching_for config:type="list"&gt;   &lt;listentry&gt;/dev/bcache0&lt;/listentry&gt; &lt;/bcache_caching_for&gt;</pre>	有关更多详细信息，请参见第 4.5.8 节“ <u>bcache 配置</u> ”。
<u>resize</u>	<p>如果应调整现有分区的大小，则此布尔值必须为 <u>true</u>。在这种情况下，您需要告知 AutoYaST 重复使用设备（请参见 <u>create</u>）并指定 <u>size</u>。</p> <pre>&lt;resize config:type="boolean"&gt;false&lt;/resize&gt;</pre>	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。

### 4.5.3.3 Btrfs 子卷

如第 4.5.3.2 节“分区配置”中所述，可为每个 Btrfs 文件系统定义一组子卷。这组子卷的最简单形式是一个项列表：

```
<subvolumes config:type="list">
  <path>tmp</path>
  <path>opt</path>
  <path>srv</path>
  <path>var/crash</path>
  <path>var/lock</path>
  <path>var/run</path>
  <path>var/tmp</path>
  <path>var/spool</path>
```

```
</subvolumes>
```

AutoYaST 支持对给定的子卷禁用写入时复制。在这种情况下，应使用稍微复杂一点的语法：

```
<subvolumes config:type="list">
  <listentry>tmp</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>
```

如果对发行套件使用了默认子卷（例如 SUSE Linux Enterprise Server 中的 `@`），则此默认子卷的名称将自动成为此列表中名称的前缀。可以通过设置 `subvolumes_prefix` 来禁用此行为。

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

#### 4.5.3.4 使用整个磁盘

AutoYaST 允许通过将 `disklabel` 设置为 `none`（请参见第 4.5.3.1 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，`drive` 中第一个 `partition` 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 4.5.5 节“逻辑卷管理器 (LVM)”和第 4.5.6 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。

#### 4.5.3.5 填补缺失

使用专家分区程序方法时，AutoYaST 可以基于一个不甚完整的配置文件创建分区计划。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>
```

## 4.5.4 高级分区功能

### 4.5.4.1 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。

### 4.5.4.2 装入选项

默认情况下，要装入的文件系统在 /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 或卷标装入。要指定如何装入分区，请使用具有 symbol 类型的 mountby 属性。可能的选项为：

- device (默认值)
- label
- UUID

如果您选择使用标签装入新分区，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装入选项。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。默认为 false。

#### 不更新访问时间 (noatime)

读取文件时不更新访问时间。默认为 false。

## 可由用户装入 (user)

文件系统可由普通用户装入。默认为 `false`。

## 数据日记模式 (`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记。

### `ordered`

在将所有数据的元数据提交到日记之前，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：装入选项

```
<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：检查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。



#### 4.5.4.3 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。

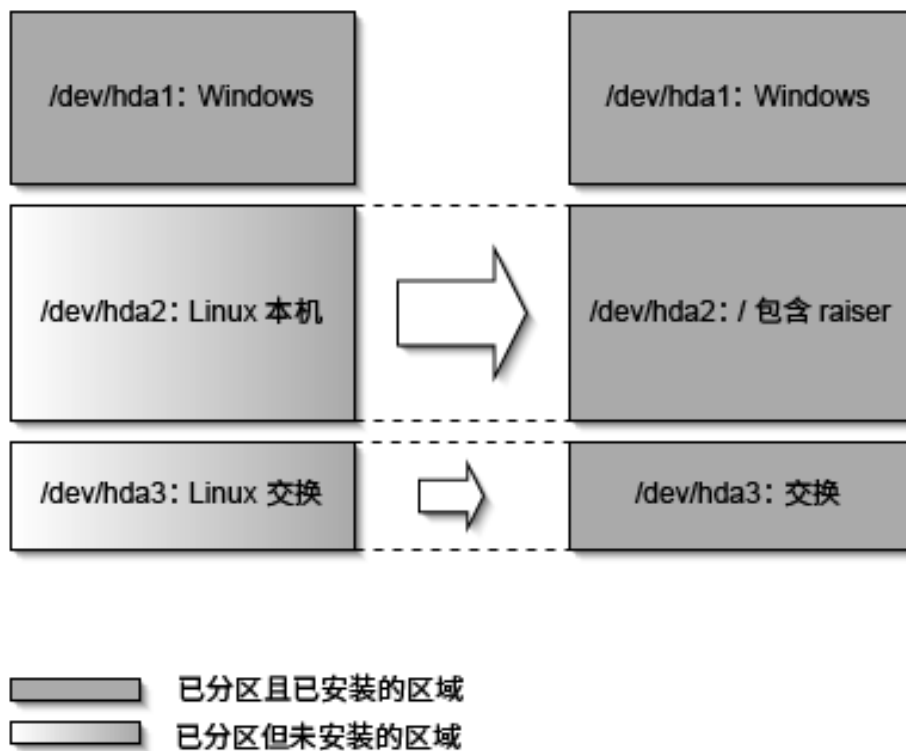


图 4.1：保留分区

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，而仅指定了安装点和分区类型（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。

## 4.5.5 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>
    <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>
    <is_lvm_vg config:type="boolean">true</is_lvm_vg>
```

```

<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`。

## 4.5.6 软件 RAID

对软件 RAID 设备的支持在 SUSE Linux Enterprise Server 15 SP2 中已得到很大改进。

如果需要，请参见第 4.5.6.1 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 RAID1 配置，该配置使用第一和第二个磁盘中的各一个分区作为 RAID 成员：

### 例 4.13：RAID1 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>left_asymmetric</parity_algorithm>
    <raid_type>raid1</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 RAID1

```

<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>left_asymmetric</parity_algorithm>
  <raid_type>raid1</raid_type>
</raid_options>
<use>all</use>
</drive>

```

#### 4.5.6.1 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 RAID1 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>left_asymmetric</parity_algorithm>
            </raid_options>
        </partition>
    </partitions>
    <use>all</use>
</drive>

```

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

#### 4.5.6.2 RAID 选项

以下元素必须放在以下 XML 结构中：

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

属性	值	说明
<u>chunk_size</u>	<pre>&lt;chunk_size&gt;4&lt;/chunk_size&gt;</pre>	
<u>parity_algorithm</u>	<p>可能的值为：</p> <p><u>left_asymmetric</u>、<u>left_symmetric</u>、<u>right_asymmetric</u>、<u>right_symmetric</u>。</p> <p>对于 RAID6 和 RAID10，可使用以下值：</p> <p><u>parity_first</u>、<u>parity_last</u>、<u>left_asymmetric_6</u>、<u>left_symmetric_6</u>、<u>right_asymmetric_6</u>、<u>right</u></p>	

属性	值	说明
	<p><u>symmetric_6</u>、<u>parity_</u> <u>first_6</u>、<u>n2</u>、<u>o2</u>、 <u>f2</u>、<u>n3</u>、<u>o3</u>、<u>f3</u></p> <pre>&lt;parity_algorithm &gt;left_asymmetric&lt;/ parity_algorithm&gt;</pre>	
<u>raid_type</u>	<p>可能的值 为: <u>raid0</u>、<u>raid1</u>、 <u>raid5</u>、<u>raid6</u> 和 <u>raid10</u>。</p> <pre>&lt;raid_type&gt;raid1&lt;/ raid_type&gt;</pre>	默认值为 <u>raid1</u> 。
<u>device_order</u>	<p>此列表包含物理设备的顺序:</p> <pre>&lt;device_order config:type="list"&gt;     &lt;device&gt;/dev/ sdb2&lt;/device&gt;     &lt;device&gt;/dev/ sda1&lt;/device&gt;     ... &lt;/ device_order&gt;</pre>	此元素是可选的，默认为字母顺序。

## 4.5.7 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，可以在内核命令行上使用以下参数：`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` 下表示多路径设备的任何其他路径，或者其任何路径。

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`、`/dev/disk` 下的任何其他对应路径（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>
```

## 4.5.8 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/> 上的博客文章。

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）块设备设置为后端设备，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 bcache -->
    <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表中所述的 `cache_mode`。

属性	值	说明
<code>cache_mode</code>	<code>&lt;cache_mode&gt;writethrough&lt;/cache_mode&gt;</code>	<code>bcache</code> 的缓存模式。可能的值为 <code>writethrough</code> 、 <code>writeback</code> 、 <code>writearound</code> 和 <code>none</code> 。

## 4.5.9 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 SP 2 中仍受到完全支持。

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。

与使用 RAID 时一样，需要先创建所有块设备（例如分区、LVM 逻辑卷等），并将其指派到您要通过此类块设备创建的 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，文件系统元数据会镜像到两个设备，数据在所有设备之间分段。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。

## 4.5.10 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。与安装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 4.5.10 节 “NFS 配置”。

### 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>
```

## 4.5.11 IBM Z 特定的配置

### 4.5.11.1 配置 DASD 磁盘

下面列出的元素必须放在以下 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。

属性	说明	值
<u>device</u>	DASD 是唯一允许的值  <code>&lt;device&gt;DASD&lt;/dev_name&gt;</code>	
<u>dev_name</u>	您要在此部分配置的设备 ( <u>dasdN</u> )。  <code>&lt;dev_name&gt;/dev/dasda&lt;/dev_name&gt;</code>	可选，但建议指定。如果留空，AutoYaST 将尝试猜测设备。
<u>channel</u>	用于访问磁盘的通道。  <code>&lt;channel&gt;0.0.0150&lt;/channel&gt;</code>	必需。
<u>diag</u>	启用或禁用 <u>DIAG</u> 。可能的值为 <u>true</u> (启用) 或 <u>false</u> (禁用)。	可选。

属性	说明	值
	<pre>&lt;diag config:type="boolean"&gt;true&lt;/diag&gt;</pre>	

#### 4.5.11.2 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>
  </zfc>
</profile>
```

需在单独的 `listentry` 部分配置每个磁盘。

属性	值
<u>controller_id</u>	通道编号  <pre>&lt;controller_id &gt;0.0.fc00&lt;/controller_id&gt;</pre>

## 4.6 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 4.22：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>

```

属性	说明
<u>initiatorname</u>	<u>InitiatorName</u> 的值取自 <u>/etc/iscsi/initiatorname.iscsi</u> 。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。
<u>version</u>	YaST 模块的版本。默认值：1.0
<u>targets</u>	目标列表。每个项包含： <u>authmethod</u> 身份验证方法：None/CHAP <u>portal</u> 门户地址 <u>startup</u> 值：manual/onboot <u>target</u> 目标名称 <u>iface</u> 接口名称

## 4.7 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 4.23：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>

```

属性	说明	值
<u>fcoe_cfg</u>	<p><u>DEBUG</u> 用于启用或禁用 fcoe 服务脚本和 fcoemon 发出的调试消息。</p> <p>如果设置为 yes，则会将 <u>USE_SYSLOG</u> 消息发送到系统日志。</p>	yes/no
<u>interfaces</u>	网卡列表，包括 VLAN 和 FCoE 配置的状态。	

属性	说明	值
<u>service_start</u>	<p>启用或禁用在引导时启动 <u>fcoe</u> 和 <u>lldpad</u> 服务。</p> <p>启动 <u>fcoe</u> 服务意味着会启动以太网光纤通道服务守护程序 <u>fcoemon</u>，该守护程序用于控制 FCoE 接口并与 <u>lldpad</u> 守护程序建立连接。</p> <p><u>lldpad</u> 服务提供链接层发现协议代理守护程序 <u>lldpad</u>，该守护程序会向 <u>fcoemon</u> 告知有关 DCB（数据中心桥接）功能以及接口的配置。</p>	yes/no

## 4.8 国家/地区设置

语言、时区和键盘设置。

例 4.24：语言

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

属性	说明	值
<u>language</u>	主要语言	在 <u>/usr/share/YaST2/data/languages</u> 下可以找到可用语言的列表

属性	说明	值
<u>languages</u>	逗号分隔的次要语言	在 <u>/usr/share/YaST2/data/languages</u> 下可以找到可用语言的列表

如果主要语言的配置值未知，它将被重设置为默认值 en\_US。

#### 例 4.25：时区

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

属性	说明	值
<u>hwclock</u>	硬件时钟使用的是本地时间还是 UTC	localtime/UTC
<u>timezone</u>	时区	在 <u>/usr/share/YaST2/data/timezone_raw.ycp</u> 下可以找到可用时区的列表

#### 例 4.26：键盘

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

属性	说明	值
<u>keymap</u>	键盘布局	键映射代码值或键映射别名值均有效。在 <u>/usr/share/YaST2/data/keyboards.rb</u> 中可以



属性	说明	值
		找到可用项的列表。例如 <u>english-us</u> 、 <u>us</u> 、 <u>english-uk</u> 、 <u>uk</u> 。

## 4.9 软件

### 4.9.1 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。

可用于 product 标记的值包括：

#### **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 Manager Server

#### **SUSE-Manager-Retail-Branch-Server**

SUSE Manager for Retail

## SUSE-Manager-Proxy

### SUSE Manager Proxy

#### 例 4.27：显式选择产品

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显式选择产品。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，请注意一些特殊注意事项。有关细节，请参见第 D.1 节“产品选择”。

## 4.9.2 使用 Patterns 和 Packages 部分选择软件包

模式或软件包的配置如下所示：

#### 例 4.28：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：软件包和模式名称

这些值是真实的软件包或模式名称。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。

### 4.9.3 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软件包和外部软件包。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`。可以在 openSUSE Build Service 的 `inst-source-utils` 软件包中找到此命令。创建数据库时，所有语言将重设置为英语。

#### 例 4.29：创建包含额外软件包 INST-SOURCE-UTILS.RPM 的软件包数据库

解压缩的 DVD 位于 `/usr/local/DVDs/LATEST` 中。

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

在以上示例中，目录 `/usr/local/CDs/LATEST/suse` 包含与体系结构相关的（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 储存库将储存在目录 `/sdk` 中。

#### 例 4.30：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/configns">
  <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。

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。这些储存库称为 其他附加项。

#### 例 4.31：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：

属性	值
<u>media_url</u>	<p>产品 URL。可以包含前缀 <u>cd:///</u>、<u>http://</u>、<u>ftp://</u> 等。此项是必需的。</p> <p>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 <u>product_dir</u> 值选择特定的产品目录（参见下文）。</p>
<u>product</u>	<p>内部产品名（如果附加项是产品）。命令 <b><u>zypper products</u></b> 显示已安装产品的名称。</p>
<u>alias</u>	<p>储存库别名。由用户定义。</p>
<u>product_dir</u>	<p>可选的子路径。此项只可用于 SUSE Linux Enterprise Packages DVD 等多产品媒体。</p>
<u>priority</u>	<p>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。</p>

属性	值
<code>ask_on_error</code>	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。请将 <code>ask_on_error</code> 设置为 <code>true</code> （默认值为 <code>false</code> ）。
<code>confirm_license</code>	用户需要确认许可。默认为 <code>false</code> 。
<code>name</code>	储存库名称。命令 <code>zypper lr</code> 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放在以下 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。使用源自第三方的软件包或储存库时请勿这样做。

属性	值
<code>accept_unsigned_file</code>	如果设置为 <code>true</code> ，AutoYaST 将接受未签名的文件（例如内容文件）。

属性	值
	<pre>&lt;accept_unsigned_file   config:type="boolean" &gt;true&lt;/accept_unsigned_file&gt;</pre>
<u>accept_file_without_checksum</u>	<p>如果设置为 <u>true</u>，AutoYaST 将接受内容文件中没有校验和的文件。</p> <pre>&lt;accept_file_without_checksum   config:type="boolean" &gt;true&lt;/accept_file_without_checksum&gt;</pre>
<u>accept_verification_failed</u>	<p>如果设置为 <u>true</u>，即使签名校验失败，AutoYaST 也会接受这些已签名的文件。</p> <pre>&lt;accept_verification_failed   config:type="boolean" &gt;true&lt;/accept_verification_failed&gt;</pre>
<u>accept_unknown_gpg_key</u>	<p>如果设置为 <u>true</u>，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。</p> <pre>&lt;accept_unknown_gpg_key   config:type="boolean" &gt;true&lt;/accept_unknown_gpg_key&gt;</pre>
<u>accept_non_trusted_gpg_key</u>	<p>将此选项设置为 <u>true</u> 会接受您尚未信任的已知密钥。</p> <pre>&lt;accept_non_trusted_gpg_key   config:type="boolean" &gt;true&lt;/accept_non_trusted_gpg_key&gt;</pre>

属性	值
<u>import_gpg_key</u>	<p>如果设置为 <u>true</u>，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。</p> <pre>&lt;import_gpg_key config:type="boolean"&gt;true&lt;/import_gpg_key&gt;</pre>

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。

属性	值
<u>accept_unsigned_file</u>	<p>如果设置为 <u>true</u>，AutoYaST 将接受未签名的文件，例如此附加产品的内容文件。</p> <pre>&lt;accept_unsigned_file   config:type="boolean" &gt;true&lt;/accept_unsigned_file&gt;</pre>
<u>accept_file_without_checksum</u>	<p>如果设置为 <u>true</u>，AutoYaST 将接受此附加产品的内容文件中没有校验和的文件。</p> <pre>&lt;accept_file_without_checksum   config:type="boolean" &gt;true&lt;/accept_file_without_checksum&gt;</pre>
<u>accept_verification_failed</u>	<p>如果设置为 <u>true</u>，即使签名校验失败，AutoYaST 也会接受这些已签名的文件。</p> <pre>&lt;accept_verification_failed   config:type="boolean" &gt;true&lt;/accept_verification_failed&gt;</pre>



属性	值
<u>accept_unknown_gpg_key</u>	<p>如果 <u>all</u> 设置为 <u>true</u>，AutoYaST 将接受安装源上的新 GPG 密钥。</p> <pre>&lt;accept_unknown_gpg_key&gt;   &lt;all config:type="boolean"&gt;true&lt;/all&gt; &lt;/accept_unknown_gpg_key&gt;</pre> <p>否则，您也可以定义单个密钥。</p> <pre>&lt;accept_unknown_gpg_key&gt;   &lt;all config:type="boolean"&gt;false&lt;/all&gt;   &lt;keys config:type="list"&gt;     &lt;keyid&gt;3B3011B76B9D6523&lt;/keyid&gt;   &lt;/keys&gt; &lt;/accept_unknown_gpg_key&gt;</pre>
<u>accept_non_trusted_gpg_key</u>	<p>这表示密钥已知，但不受您的信任。</p> <p>可以通过添加以下命令来信任所有密钥：</p> <pre>&lt;accept_non_trusted_gpg_key&gt;   &lt;all config:type="boolean"&gt;true&lt;/all&gt; &lt;/accept_non_trusted_gpg_key&gt;</pre> <p>或者，您可以信任特定的密钥：</p> <pre>&lt;accept_non_trusted_gpg_key&gt;   &lt;all config:type="boolean"&gt;false&lt;/all&gt;   &lt;keys config:type="list"&gt;     &lt;keyid&gt;3B3011B76B9D6523&lt;/keyid&gt;   &lt;/keys&gt; &lt;/accept_non_trusted_gpg_key&gt;</pre>

属性	值
<code>import_gpg_key</code>	<p>如果 <code>all</code> 设置为 <code>true</code>，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。</p> <pre>&lt;import_gpg_key&gt;   &lt;all config:type="boolean"&gt;true&lt;/all&gt; &lt;/import_gpg_key&gt;</pre> <p>只能对特定的密钥执行此操作：</p> <pre>&lt;import_gpg_key&gt;   &lt;all config:type="boolean"&gt;false&lt;/all&gt;   &lt;keys config:type="list"&gt;     &lt;keyid&gt;3B3011B76B9D6523&lt;/keyid&gt;   &lt;/keys&gt; &lt;/import_gpg_key&gt;</pre>

#### 4.9.4 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 4.32：在控制文件中选择内核

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

## 4.9.5 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掉 `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 4.33：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，无法去除某个模式中的软件包（请参见第 4.9.2 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.
```

## 4.9.6 安装建议的软件包/模式

默认会安装所有建议的软件包/模式。要进行最小安装（仅包括必需的软件包），可以使用标志 `install_recommended` 关闭此行为。请注意，此标志只会影响全新安装，在升级期间将被忽略。

```
<software>
```

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

默认：如果未在配置文件中设置此标志，将安装所有建议的 软件包，但不会安装建议的 模式。

## 4.9.7 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>
```

## 4.9.8 在第 2 个阶段安装模式

您还可以在第 2 个阶段安装模式。为此，请使用 post-patterns 元素：

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

## 4.9.9 在第 2 个阶段联机更新

可以在安装结束时执行联机更新。请将 do\_online\_update 布尔值设置为 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>
```

## 4.10 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/模式。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 4.34：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>
```

元素	说明	注释
<u>stop_on_solver_conflict</u>	出现软件包依赖性问题时暂停安装。	

元素	说明	注释
<code>modified</code>	创建已修改文件的备份。	
<code>sysconfig</code>	创建 <code>/etc/sysconfig</code> 目录的备份。	
<code>remove_old</code>	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 配置文件复制到其文件系统上的 `/root/autoupgrade.xml` 中。
2. 从安装媒体引导系统。
3. 选择 `安装` 菜单项。
4. 在命令行上设置 `autoupgrade=1`。
5. 按 `Enter` 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 `安装` 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` 启动升级过程。

## 4.11 服务和目标

使用 `services-manager` 资源可以设置默认的 `systemd` 目标，并详细指定您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 4.35：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>
```

## 4.12 网络配置

网络配置用于将单个工作站连接到基于以太网的 LAN 或配置拨号连接。它还提供更复杂的配置（多个网卡、路由等）。

如果以下设置设为 `true`，YaST 将保留安装期间（通过 `linuxrc`）创建的网络设置，并/或将其与 AutoYaST 控制文件中的网络设置（如果已定义）合并。



## 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”。

AutoYaST 设置的优先级高于现有的任何配置文件。YaST 将基于控制文件中的项写入 ifcfg-\* 文件，且不去除旧项。如果存在空白或缺少 DNS 和路由部分，YaST 将保留所有现有值。否则将应用控制文件中的设置。

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

在第二个阶段，会先安装附加的软件包，然后再根据配置文件中所述配置网络。**keep\_install\_network** 默认设置为 **true**，以确保在需要通过某个网络安装这些软件包时该网络可供使用。如果在第一个阶段便安装了所有软件包，且在第二个阶段的早期不需要网络，则将 **keep\_install\_network** 设置为 **false** 可避免复制配置。

要自动配置网络设置并激活网络，可使用一个全局资源 **<networking>** 来储存整个网络配置。

### 例 4.36：网络配置

```
<networking>  
  <dns>  
    <dhcp_hostname config:type="boolean">true</dhcp_hostname>  
    <domain>site</domain>  
    <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>
  <device>eth0</device>
  <startmode>auto</startmode>
</interface>
</interfaces>
<ipv6 config:type="boolean">true</ipv6>
<keep_install_network config:type="boolean">false</keep_install_network>
#false means use Wicked, true means use NetworkManager
<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>192.168.2.1</destination>
      <device>eth0</device>
      <extrapara>foo</extrapara>
      <gateway>-</gateway>
      <netmask>-</netmask>
    </route>
    <route>
      <destination>default</destination>
      <device>eth0</device>
      <gateway>192.168.1.1</gateway>
    </route>
  </routes>
</routing>

```

```

        <netmask>-</netmask>
    </route>
    <route>
        <destination>default</destination>
        <device>lo</device>
        <gateway>192.168.5.1</gateway>
        <netmask>-</netmask>
    </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 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` 涉及特定于 s390 的设备设置。
- `net-udev` 枚举用于设置持久名称的 udev 规则。

#### 例 4.37：网桥接口配置

```

<interfaces config:type="list">
  <interface>
    <device>br0</device>
    <bootproto>static</bootproto>
    <bridge>yes</bridge>
    <bridge_forwarddelay>0</bridge_forwarddelay>
    <bridge_ports>eth0 eth1</bridge_ports>
    <bridge_stp>off</bridge_stp>
    <ipaddr>192.168.122.100</ipaddr>
    <netmask>255.255.255.0</netmask>
    <network>192.168.122.0</network>
    <prefixlen>24</prefixlen>
    <startmode>auto</startmode>
  </interface>
  <interface>

```

```

    <device>eth0</device>
    <bootproto>none</bootproto>
    <startmode>hotplug</startmode>
</interface>
<interface>
    <device>eth1</device>
    <bootproto>none</bootproto>
    <startmode>hotplug</startmode>
</interface>
</interfaces>

```



### 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>`

## 4.12.1 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。

元素	说明	注释
<u>bootproto</u>	<p>接口使用的引导协议。可能的值：</p> <ul style="list-style-type: none"> <li>• <u>static</u> 表示静态指派的地址。需使用 <u>ipaddr</u> 元素指定 IP。</li> <li>• <u>dhcp4</u>、<u>dhcp6</u> 或 <u>dhcp</u>，用于通过 DHCP 设置 IP 地址（IPv4、IPv6 或任何地址）。</li> </ul>	必要。

元素	说明	注释
	<ul style="list-style-type: none"> <li>• <u>dhcp+autoip</u> 用于从 Zeroconf 获取 IPv4 配置，或从 DHCP 获取 IPv6 配置。</li> <li>• <u>autoip</u> 用于从 Zeroconf 获取 IPv4 配置。</li> <li>• <u>ibft</u> 用于通过 iBFT 协议获取 IP 地址。</li> <li>• <u>none</u> 表示跳过地址设置。此值用于桥接和绑定从属。</li> </ul>	
<u>broadcast</u>	广播 IP 地址。	仅适用于 <u>static</u> 引导协议。
<u>device</u>	设备名称。	已弃用。请改用 <u>name</u> 。
<u>name</u>	设备名称，例如： <u>eth0</u> 。	必要。
<u>ipaddr</u>	指派给接口的 IP 地址。	仅适用于 <u>static</u> 引导协议。它可以包含网络前缀，例如： <u>192.168.1.1/24</u> 。
<u>remote_ipaddr</u>	用于点对点连接的远程 IP 地址。	仅适用于 <u>static</u> 引导协议。
<u>netmask</u>	网络掩码，例如： <u>255.255.255.0</u> 。	已弃用。请改用 <u>prefixlen</u> ，或者在 <u>ipaddr</u> 元素中包含网络前缀。

元素	说明	注释
<u>network</u>	网络 IP 地址。	已弃用。请改为结合使用 <u>ipaddr</u> 和 <u>prefixlen</u> 。
<u>prefixlen</u>	网络前缀，例如： <u>24</u> 。	仅适用于 <u>static</u> 引导协议。
<u>startmode</u>	<p>何时启动接口。可能的值为：</p> <ul style="list-style-type: none"> <li>• <u>hotplug</u>（插入设备时。例如，此值可用于 USB 网卡）。</li> <li>• <u>auto</u>（系统引导时）。<u>onboot</u> 是已弃用的别名。</li> <li>• <u>ifplugd</u>（设备由 <u>ifplugd</u> 守护程序管理时）。</li> <li>• <u>manual</u>（应手动启动设备时）。</li> <li>• <u>nfsroot</u>（需要将设备装入到根文件系统时，例如，当 <u>/</u> 位于 NFS 卷上时）。</li> <li>• <u>off</u>（永不启动设备）。</li> </ul>	
<u>ifplugd_priority</u>	<u>ifplugd</u> 守护程序的优先级。它确定设备的激活顺序。	仅适用于 <u>ifplugd</u> 启动模式。

元素	说明	注释
<u>usercontrol</u>	不再使用的参数。	已弃用。
<u>bonding_slaveX</u>	绑定设备的名称。	对于绑定设备是必需的。X 将被替换为从 0 开始的编号，例如 <u>bonding_slave0</u> 。每个从属都需要具有一个唯一编号。
<u>bonding_module_opts</u>	绑定设备的选项。	仅适用于 <u>bond</u> 设备。
<u>mtu</u>	接口的最大传送单位。	可选。
<u>ethtool_options</u>	设备激活期间的 Ethtool 选项。	可选。
<u>zone</u>	接口指派到的防火墙区域名称。	可选。
<u>vlan_id</u>	此 VLAN 使用的标识符。	仅适用于 <u>vlan</u> 设备。
<u>etherdevice</u>	VLAN 挂接到的设备。	仅适用于 <u>vlan</u> 设备以及需要它的设备。
<u>bridge</u>	如果接口是网桥，则为 <u>yes</u> 。	已弃用。从其他属性推断而来。
<u>bridge_ports</u>	网桥端口的空格分隔列表，例如 <u>eth0 eth1</u> 。	仅适用于 <u>bridge</u> 设备以及需要它的设备。
<u>bridge_stp</u>	生成树协议。可能的值为 <u>on</u> （启用时）和 <u>off</u> （禁用时）。	仅适用于 <u>bridge</u> 设备。

元素	说明	注释
<u>bridge_forward_delay</u>	网桥的转发延迟，例如： <u>15</u> 。	仅适用于 <u>bridge</u> 设备。有效值为 <u>4</u> 到 <u>30</u> 。

## 4.12.2 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。

元素	说明	注释
<u>name</u>	网络接口名称，例如 <u>eth3</u> 。	必要。
<u>rule</u>	<u>ATTR{address}</u> 针对基于 MAC 的规则， <u>KERNELS</u> 针对基于总线 ID 的规则。	必要。
<u>value</u>	例如： <u>f0:de:f1:6b:da:69</u> 针对 MAC 规则， <u>0000:00:1c.1</u> 或 <u>0.0.0700</u> 针对总线 ID 规则。	必要。



### 提示：处理设备名称的冲突

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 4.38：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>

```

### 4.12.3 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。

元素	说明	注释
<u>hostname</u>	主机名，不包括域名部分。例如： <u>foo</u> （而不是 <u>foo.bar</u> ）。	如果未指定主机名且未从 DHCP 服务器获取主机名（请参见 <u>dhcp_hostname</u> ），AutoYaST 将生成一个随机主机名。
<u>nameservers</u>	名称服务器的列表。示例： <pre>&lt;nameservers   config:type="list"&gt;     &lt;nameserver&gt;192.168.1.116&lt;/nameserver&gt;     &lt;nameserver&gt;192.168.1.117&lt;/nameserver&gt;   &lt;/nameservers&gt;</pre>	
<u>searchlist</u>	用于查找主机名的搜索列表。 <pre>&lt;searchlist   config:type="list"&gt;     &lt;search&gt;example.com&lt;/search&gt;   &lt;/searchlist&gt;</pre>	可选。
<u>dhcp_hostname</u>	指定是否必须从 DHCP 获取主机名。	



元素	说明	注释
	<pre>&lt;dhcp_hostname   config:type="boolean"&gt;true&lt;/ dhcp_hostname&gt;</pre>	

## 4.12.4 路由选择

routing 表允许指定路由列表，以及 IPv4 和 IPv6 的包转发设置。

元素	说明	注释
<u>ipv4_forward</u>	是否必须为 IPv4 启用 IP 转发。	可选。
<u>ipv6_forward</u>	是否必须为 IPv6 启用 IP 转发。	可选。
<u>routes</u>	路由列表。	可选。

下表介绍了如何定义路由。

元素	说明	注释
<u>destination</u>	路由目标。可以指定地址前缀，例如： <u>192.168.122.0/24</u> 。	必要。
<u>device</u>	与路由关联的接口。	必要。
<u>gateway</u>	网关的 IP 地址。	可选。
<u>netmask</u>	目标的网络掩码。	已弃用。最好是指定前缀作为 <u>destination</u> 值的一部分。

## 4.12.5 s390 选项

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

元素	说明	注释
type	qeth、ctc 或 iucv	
chanids	冒号（首选）或空格分隔的通道 ID <pre>&lt;chanids&gt;0.0.0700:0.0.0701:0.0.0702&lt;/chanids&gt;</pre>	
layer2	<pre>&lt;layer2 config:type="boolean"&gt;true&lt;/layer2&gt;</pre>	布尔值；默认值：false
portname	QETH 端口名（从 SLE 12 SP2 开始已弃用）	
protocol	CTC/LCS 协议，一个较小的数字（字符串形式） <pre>&lt;protocol&gt;1&lt;/protocol&gt;</pre>	可选
router	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。

## 4.12.6 代理

配置因特网代理（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）。如果您使用需要身份验证的代理服务器，请填写 `proxy_user` 和 `proxy_password`。

**例 4.39：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>
```

## 4.13 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 4.40：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_by_dhcp config:type="boolean">false</nis_by_dhcp>
  <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>

```

## 4.14 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 从属服务器，以及两者的组合。

### 例 4.41：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_yplib config:type="boolean">false</start_yplib>
  <start_yppasswdd config:type="boolean">false</start_yppasswdd>

```

```
<start_ypxfrd config:type="boolean">false</start_ypxfrd>
</nis_server>
```

属性	值	说明
<u>domain</u>	NIS 域名。	
<u>maps_to_serve</u>	服务器可用的映射列表。	值： auto.master、ethers、group、 hosts、netgrp、networks、 passwd、protocols、rpc、 services、shadow
<u>merge_passwd</u>	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
<u>mingid</u>	要包含在用户映射中的最小 GID。	
<u>minuid</u>	要包含在用户映射中的最小 UID。	
<u>nopush</u>	不要将更改推送到从属服务器。（未发生任何更改时非常有用）。	值：true/false
<u>pwd_chfn</u>	YPPWD_CHFN - 允许更改全名	值：true/false
<u>pwd_chsh</u>	YPPWD_CHSH - 允许更改登录外壳	值：true/false
<u>pwd_srcdir</u>	YPPWD_SRCDIR - passwd 数据的源目录	默认值： <u>/etc</u>

属性	值	说明
<u>securenets</u>	允许其查询 NIS 服务器的主机列表	<p>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。</p> <p>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。</p> <p>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。</p>
<u>server_type</u>	选择是将 NIS 服务器配置为主服务器或从属服务器，还是不配置 NIS 服务器。	值：master、slave、none
<u>slaves</u>	要配置为 NIS 从属服务器的主机名列表。	
<u>start_yplibind</u>	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
<u>start_yppasswdd</u>	同时启动口令守护程序。	值：true/false
<u>start_ypxfrd</u>	同时启动映射传送守护程序。 快速映射分发；将加快向从属服务器传送映射的过程。	值：true/false

## 4.15 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 4.42：/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>
```

## 4.16 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 4.43：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	说明
<code>disable_dhcp_hostname</code>	不允许 DHCP 更改主机名。	值：true/false
<code>global/security</code>	身份验证体系的类型（域技术或 Active Directory 服务器 (ADS)）。	值：ADS/domain
<code>global/usershare_allow_guests</code>	允许共享 Guest 访问权限。	值：No/Yes
<code>global/usershare_max_shares</code>	表示 <code>smb.conf</code> 中的共享数上限。	0 表示未启用共享。
<code>global/workgroup</code>	工作组名或域名。	
<code>winbind</code>	使用 winbind。	值：true/false

## 4.17 Samba 服务器

简单 Samba 服务器的配置。

例 4.44：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	说明
<u>accounts</u>	Samba 帐户的列表。	
<u>backend</u>	可用后端的列表	值：true/false
<u>config</u>	在 <code>/etc/samba/smb.conf</code> 中设置其他用户定义的参数。	该示例显示了 <code>/etc/samba/smb.conf</code> 的 <code>global</code> 部分中的参数。
<u>service</u>	在引导期间启动 Samba 服务。	值：Enabled/Disabled
<u>trustdom/</u>	可信域。	两个映射的映射（键： <u>establish</u> 、 <u>revoke</u> ）。每个映射包含采用 <code>key: domainname</code> <code>value: password</code> 格式的项。
<u>version</u>	Samba 版本。	默认值：2.11

## 4.18 身份验证客户端

配置文件必须采用 JSON 格式。校验 `autoyast2` 和 `autoyast2-installation` 是否都已安装。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://

要结合本机 SSL（而不是 TLS）使用 LDAP，请添加 `ldaps` 资源。

## 4.19 NFS 客户端和服务端

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 客户端和服务端。

从 SUSE Linux Enterprise Server 15 SP2 开始，NFS 客户端配置结构已更改。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 4.5.10 节“NFS 配置”。

### 例 4.45：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 4.46：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>
  <start_nfsserver config:type="boolean">true</start_nfsserver>
</nfs_server>

```

## 4.20 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。需要更新您在旧版 SUSE Linux Enterprise Server 中使用的 NTP 客户端配置文件，然后才能与版本 15 和更新版本兼容。

下面是 NTP 客户端配置示例：

```

<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）。有关更多细节，请参见 **man 8 netconfig**。
- ❷ 时间服务器或时间服务器池的 URL。
- ❸ 启动 `chronyd` 后，`iburst` 会加速特定时间源的初始时间同步。
- ❹ 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。
- ❺ 对于 `ntp_sync`，当正在运行 NTP 守护程序时请输入“systemd”（默认值），要通过 cron 进行同步请输入整数间隔（以秒为单位）或“manual”（不自动同步）。

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：

```

<peer>
  <address>2001:418:3ff::1:53</address>
  <comment/>
  <options/>
  <type>server</type>
</peer>

<peer>
  <address>2.pool.ntp.org</address>

```

```
<comment/>
<options/>
<type>server</type>
</peer>
```

## 4.21 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 4.48：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>

```

## 4.22 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 服务器](#) YaST 模块配置 Apache 服务器。完成该操作后，调用 [AutoYaST 配置](#) 模块，选择 [HTTP 服务器](#) YaST 模块并克隆 Apache 设置。可以通过 [文件](#) 菜单导出这些设置。

### 例 4.49：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"
# 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>
    </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>
</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>
          <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>
</modules>

```

```

</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	说明
Listen		主机 <u>Listen</u> 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>



列表名称	列表元素	说明
	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 <code>true</code> 。
	change	出于历史原因，此项始终设置为 <code>enable</code> 。

元素	说明	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



## 注意：防火墙

要正常运行 Apache 服务器，请确保正确配置了防火墙。

## 4.23 Squid 服务器

Squid 是一个缓存和转发 Web 代理。

### 例 4.50：SQUID 服务器配置

```
<squid>
  <acls config:type="list">
    <listentry>
      <name>QUERY</name>
      <options config:type="list">
```

```

        <option>cgi-bin \?</option>
    </options>
    <type>urlpath_regex</type>
</listentry>
<listentry>
    <name>apache</name>
    <options config:type="list">
        <option>Server</option>
        <option>^Apache</option>
    </options>
    <type>rep_header</type>
</listentry>
<listentry>
    <name>all</name>
    <options config:type="list">
        <option>0.0.0.0/0.0.0.0</option>
    </options>
    <type>src</type>
</listentry>
<listentry>
    <name>manager</name>
    <options config:type="list">
        <option>cache_object</option>
    </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>
        <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	说明
<u>acls</u>	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。
<u>http_accesses</u>	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。
<u>http_ports</u>	定义 Squid 将用来侦听客户端 HTTP 请求的所有端口。	<u>Host</u> 可以包含主机名或 IP 地址，也可以留空。  <u>transparent</u> 禁用透明的 PMTU 发现。
<u>refresh_patterns</u>	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。  <u>Min</u> 确定在未指定显式失效时间的情况下，要将某个对象视为新对象的时间长短（分钟）。 <u>Max</u> 是将未指定显式失效时间的对象视为新对象的时间长短上限。 <u>Percent</u> 是对象新旧程度的百分比（自上次修改以来消逝的时间）。未指定显式失效时间的对象将被视为新对象。
<u>settings</u>	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。

属性	值	说明
<code>service_enabled_on_startup</code>	引导时启动 Squid 服务。	值：true/false

## 4.24 FTP 服务器

配置 FTP 因特网服务器设置

例 4.51：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	说明	注释
<u>AnonAuthen</u>	启用/禁用匿名用户和本地用户。	仅经身份验证的用户：1；仅匿名：0；两者：2
<u>AnonCreatDirs</u>	匿名用户可以创建目录。	值：YES/NO
<u>AnonReadOnly</u>	匿名用户可以上载。	值：YES/NO
<u>AnonMaxRate</u>	允许匿名客户端使用的最大数据传输速率。	KB/秒
<u>AntiWarez</u>	禁止下载已上载但未经本地管理员验证的文件。	值：YES/NO
<u>Banner</u>	指定包含当某人连接到服务器时所显示文本的文件名。	
<u>CertFile</u>	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。
<u>ChrootEnable</u>	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 <u>chroot</u> jail 内。	警告：此选项会影响安全性。 值：YES/NO
<u>EnableUpload</u>	如果启用，则 FTP 用户可以上载。	要允许匿名用户上载，请启用 <u>AnonReadOnly</u> 。值：YES/NO

元素	说明	注释
<u>FTPUser</u>	定义匿名 FTP 用户。	
<u>FtpDirAnon</u>	匿名用户的 FTP 目录。	指定匿名 FTP 用户使用的目录。
<u>FtpDirLocal</u>	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。
<u>LocalMaxRate</u>	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
<u>MaxClientsNumber</u>	允许连接的最大客户端数。	
<u>MaxClientsPerIP</u>	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因特网地址连接的客户端数。
<u>MaxIdleTime</u>	两个 FTP 命令之间远程客户端的最大等待时间（超时）。	分钟
<u>PasMaxPort</u>	用于被动连接答复的端口范围最大值。	<u>PassiveMode</u> 需设置为 YES。
<u>PasMinPort</u>	用于被动连接答复的端口范围最小值。	<u>PassiveMode</u> 需设置为 YES。
<u>PassiveMode</u>	启用被动模式	值：YES/NO
<u>SSL</u>	安全性设置	禁用 SSL/TLS：0；接受 SSL 和 TLS：1；拒绝非 SSL/TLS 连接：2
<u>SSLEnable</u>	如果启用，则允许 SSL 连接。	值：YES/NO

元素	说明	注释
<u>SSLv2</u>	如果启用，则允许 SSL 版本 2 连接。	值：YES/NO
<u>SSLv3</u>	如果启用，则允许 SSL 版本 3 连接。	值：YES/NO
<u>StartDaemon</u>	如何启动 FTP 守护程序。	手动：0；引导时：1；通过 <u>systemd</u> 套接字：2
<u>TLS</u>	如果启用，则允许 TLS 连接。	值：YES/NO
<u>Umask</u>	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 <u>177:077</u> 。
<u>UmaskAnon</u>	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。
<u>UmaskLocal</u>	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。
<u>VerboseLogging</u>	如果启用，则记录所有 FTP 请求和响应。	值：YES/NO
<u>VirtualUser</u>	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。

## 4.25 TFTP 服务器

配置 TFTP 因特网服务器设置。

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 4.52：TFTP 服务器配置：**

```
<tftp-server>
  <start_tftpd config:type="boolean">true</start_tftpd>
  <tftp_directory>/tftpboot</tftp_directory>
</tftp-server>
```

元素	说明	注释
start_tftpd	启用 TFTP 服务器服务。	值：true/false
tftp_directory	引导映像目录：指定所处理文件所在的目录。	值一般为 /tftpboot。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）。

## 4.26 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 4.53：启用首次引导工作流程**

```
<firstboot>
  <firstboot_enabled config:type="boolean">true</firstboot_enabled>
</firstboot>
```

## 4.27 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 自定义设置。此功能允许您创建自定义的配置。

### 例 4.54：安全性配置

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>
</security>
```

### 4.27.1 口令设置选项

更改各项口令设置。这些设置主要储存在 /etc/login.defs 文件中。

使用此资源可激活当前支持的加密方法之一。如果未设置，将配置 [DES](#)。

Linux 默认方法 [DES](#) 适用于所有网络环境，但是存在八个字符的口令长度限制。[MD5](#) 允许更长的口令，因此可提供更高的安全性，但某些网络协议不支持此方法，并且在 NIS 中使用时可能会出现問題。还支持 [Blowfish](#)。

此外，还可将系统设置为检查口令合理性和长度等。

## 4.27.2 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` ？

当控制台端的某人按下 `Ctrl - Alt - Del` 组合键时，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。

## 4.27.3 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。

## 4.27.4 新用户设置 (`useradd` 设置)

设置允许的最小和最大用户和组 ID。

# 4.28 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 4.55：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	说明
<u>auditd/flush</u>	描述如何将数据写入磁盘。	如果设置为 <u>INCREMENTAL</u> ，“频率”参数将告知要在写入多少条记录后向磁盘发出显式刷新命令。 <u>NONE</u> 表示不采取特殊措施来刷新数据， <u>DATA</u> 表示将数据部分保持同步， <u>SYNC</u> 表示将数据和元数据保持完全同步。
<u>auditd/freq</u>	此参数告知要在写入多少条记录后向磁盘发出显式刷新命令。	参数 <u>flush</u> 需设置为 <u>INCREMENTAL</u> 。
<u>auditd/log_file</u>	日志文件的完整路径名。	
<u>auditd/log_fomat</u>	需要记录多少信息。	设置为 <u>RAW</u> 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 <u>NOLOG</u> 会丢弃所有审计信

属性	值	说明
		息，而不是将其写入磁盘（不影响已发送到调度程序的数据）。
<u>auditd/max_log_file</u>	需要记录多少信息。	单位：MB
<u>auditd/num_logs</u>	日志文件数。	<u>max_log_file_action</u> 需设置为 <u>ROTATE</u>
<u>auditd/max_log_file_action</u>	达到日志容量时所执行的操作。	如果操作设置为 <u>ROTATE</u> ，“日志文件数”将指定要保留的文件数。如果设置为 <u>SYSLOG</u> ，审计守护程序会向系统日志写入警告。如果设置为 <u>SUSPEND</u> ，守护程序将停止向磁盘写入记录。 <u>IGNORE</u> 表示不执行任何操作， <u>KEEP_LOGS</u> 类似于 <u>ROTATE</u> ，但不会覆盖日志文件。
<u>auditd/name_format</u>	“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 <u>USER</u> ，则会使用用户定义的名称。 <u>NONE</u> 表示不插入计算机名称。 <u>HOSTNAME</u> 使用“gethostname”系统调用返回的名称。 <u>FQD</u> 使用完全限定的域名。

属性	值	说明
<code>rules</code>	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 <code>man auditctl</code> 。

## 4.29 用户和组

### 4.29.1 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 4.56：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 4.57：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>command="/opt/login.sh" ssh-rsa
    AAAAB3NzaC1yc2EAAAADAQABAAQDKLt1vnW2vTJpBp3VK91rFsBvpY97NljsVLdgUrlPbZ/
    L51FerQQ+djQ/ivDASQj0+567nMGqfYGFA/De1EGMMEoeShza67qjNi14L1HBGgVojaNajMR/
    NI2d1kDyvsgRy7D7FT5UGGUNT0dlcSD3b85zswgHeYLidgcGIoKeRi7HpVD00TyhwUv4sq3ubrPCWAR
    gPe0LdVFfa9clC8PTZdxSeKp4jpNjIHEyREPin2Un1luCIPWr0Yyym7aRJEPopCEqBA9HvfwpbuwBI5F0
    uIWZgSQLfpwW86599fBo/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 `linuxrc` 中指定 `rootpassword`，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。



## 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。您可以在 AutoYaST 控制文件中使用 `uid` 手动指定此类用户 ID，或者不使用 `uid` 而让系统自动选择一个用户 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	说明
<code>username</code>	文本 <pre>&lt;username&gt;lukesw&lt;/username&gt;</pre>	必要。应是有效的用户名。如果您不确定，请查看 <code>man 8 useradd</code> 。
<code>fullname</code>	文本 <pre>&lt;fullname&gt;Tux Torvalds&lt;/fullname&gt;</pre>	可选。用户的全名。

属性	值	说明
<u>forename</u>	文本 <pre>&lt;forename&gt;Tux&lt;/forename&gt;</pre>	可选。用户的名字。
<u>surname</u>	文本 <pre>&lt;surname&gt;Skywalker&lt;/surname&gt;</pre>	可选。用户的姓氏。
<u>uid</u>	数字 <pre>&lt;uid&gt;1001&lt;/uid&gt;</pre>	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 <a href="#">注意：指定用户 ID (uid)</a> 。
<u>gid</u>	数字 <pre>&lt;gid&gt;100&lt;/gid&gt;</pre>	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。
<u>home</u>	路径 <pre>&lt;home&gt;/home/luke&lt;/home&gt;</pre>	可选。用户主目录的绝对路径。默认将使用 <u>/home/username</u> （例如， <u>alice</u> 的主目录将是 <u>/home/alice</u> ）。
<u>home_btrfs_subvolume</u>	布尔 <pre>&lt;home_btrfs_subvolume config:type="boolean"&gt;true&lt;/home_btrfs_subvolume&gt;</pre>	可选。在 Btrfs 子卷中生成主目录。默认已禁用。

属性	值	说明
<u>shell</u>	路径 <pre>&lt;shell&gt;/usr/bin/zsh&lt;/shell&gt;</pre>	可选。 <u>/bin/bash</u> 是默认值。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 <u>software</u> 部分）。
<u>user_password</u>	文本 <pre>&lt;user_password&gt;some-password&lt;/user_password&gt;</pre>	可选。如果您输入感叹号 (!)，将生成随机口令。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 <u>mkpasswd</u> 。输入 <u>/etc/shadow</u> （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 <u>encrypted</u> 参数。
<u>encrypted</u>	布尔 <pre>&lt;encrypted   config:type="boolean"&gt;true&lt;/encrypted&gt;</pre>	可选。如果未指定，则视为 <u>false</u> 。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 <u>man 3 crypt</u> ）。
<u>password_settings</u>	密码设置 <pre>&lt;password_settings&gt;   &lt;expire/&gt;   &lt;max&gt;60&lt;/max&gt;   &lt;warn&gt;7&lt;/warn&gt;</pre>	可选。可以自定义某些口令设置： <u>expire</u> （采用 YYYY-MM-DD 格式的帐户失效日期）、 <u>flag</u> （/

属性	值	说明
	<code>&lt;/password_settings&gt;</code>	<code>etc/shadow</code> 标志)、 <code>inact</code> (口令失效多少天后禁用帐户)、 <code>max</code> (口令保持有效的最长天数)、 <code>min</code> (口令失效后的宽限期天数, 在此期间用户仍可更改口令) 和 <code>warn</code> (在口令失效之前的多少天开始发出口令更改提醒)。
<code>authorized_keys</code>	授权密钥的列表  <code>&lt;authorized_keys   config:type="list"&gt;     &lt;listentry&gt;ssh-rsa ...&lt;/   listentry&gt; &lt;/authorized_keys&gt;</code>	要写入到 <code>\$HOME/.ssh/authorized_keys</code> 的授权密钥列表。请参见以下示例。

## 4.29.2 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` 以供 `useradd` 读取。

属性	值	说明
<code>group</code>	文本  <code>&lt;group&gt;100&lt;/group&gt;</code>	可选。默认的初始登录组。
<code>groups</code>	文本	可选。其他组的列表。



属性	值	说明
	<code>&lt;groups&gt;users&lt;/groups&gt;</code>	
<u>home</u>	路径 <code>&lt;home&gt;/home&lt;/home&gt;</code>	可选。用户的主目录前缀。
<u>expire</u>	日期 <code>&lt;expire&gt;2017-12-31&lt;/expire&gt;</code>	可选。默认的口令失效日期，采用 <u>YYYY-MM-DD</u> 格式。
<u>inactive</u>	数字 <code>&lt;inactive&gt;3&lt;/inactive&gt;</code>	可选。在多少天后禁用已失效的帐户。
<u>no_groups</u>	布尔 <code>&lt;no_groups config:type="boolean"&gt;true&lt;/no_groups&gt;</code>	可选。不使用次要组。
<u>shell</u>	路径 <code>&lt;shell&gt;/usr/bin/fish&lt;/shell&gt;</code>	默认登录外壳。 <u>/bin/bash</u> 是默认值。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 <u>software</u> 部分）。
<u>skel</u>	路径 <code>&lt;skel&gt;/etc/skel&lt;/skel&gt;</code>	可选。添加新用户时要用作主干的文件的位置。可在 <u>man 8 useradd</u> 中找到详细信息。

属性	值	说明
<u>umask</u>	文件创建模式掩码 <pre>&lt;umask&gt;022&lt;/umask&gt;</pre>	为主目录设置文件创建模式掩码。默认情况下， <u>useradd</u> 将使用 <u>022</u> 。有关更多信息，请查看 <u>man 8 useradd</u> 和 <u>man 1 umask</u> 。

### 4.29.3 组

可按以下示例中所示，在 <groups> 中定义组列表。

例 4.58：组配置

```
<groups config:type="list">
  <group>
    <gid>100</gid>
    <groupname>users</groupname>
    <userlist>bob,alice</userlist>
  </group>
</groups>
```

属性	值	说明
<u>groupname</u>	文本 <pre>&lt;groupname&gt;users&lt;/groupname&gt;</pre>	必要。应是有效的组名。如果您不确定，请查看 <u>man 8 groupadd</u> 。
<u>gid</u>	数字 <pre>&lt;gid&gt;100&lt;/gid&gt;</pre>	可选。组 ID。此 ID 必须唯一且为非负数。

属性	值	说明
<u>group_password</u>	文本 <pre>&lt;group_password&gt;password&lt;/group_password&gt;</pre>	可选。可以写入纯文本（不建议）或加密形式的组口令。请检查 <u>encrypted</u> 以选择所需的行为。
<u>encrypted</u>	布尔 <pre>&lt;encrypted config:type="boolean"&gt;true&lt;/encrypted&gt;</pre>	可选。指示是否加密配置文件中的组口令。
<u>userlist</u>	用户列表 <pre>&lt;userlist&gt;bob,alice&lt;/userlist&gt;</pre>	可选。属于该组的用户列表。必须用逗号分隔用户名。

#### 4.29.4 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 4.59：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	说明
<u>password_less_login</u>	布尔	可选。启用无口令登录。此设置仅影响图形登录。

属性	值	说明
	<pre>&lt;password_less_login   config:type="boolean"&gt;true&lt;/ password_less_login&gt;</pre>	
<u>autologin_user</u>	文本 <pre>&lt;autologin_user&gt;alice&lt;/ autologin_user&gt;</pre>	可选。为给定的用户启用自动登录。

## 4.30 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：

所有脚本需位于 `<scripts>` 部分。

- pre-scripts（初期，在任何其他操作尚未真正发生之前）
- postpartitioning-scripts（在分区并装入到 `/mnt` 之后，但在 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（在首次引导已安装的系统期间，所有服务已启动并正在运行）

### 4.30.1 安装前脚本

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。执行所有安装前脚本后，AutoYaST 始终会再次检查储存设备。



### 注意：安装前脚本和确认

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，则会先执行安装前脚本，然后再显示确认屏幕 ([profile/install/general/mode/confirm](#))。



### 注意：安装前脚本和 Zypper

要在安装前脚本中调用 `zypper`，需要设置环境变量 `ZYPP_LOCKFILE_ROOT="/var/run/autoyast"`，以免与运行中的 YaST 进程冲突。

必须按如下所示放置安装前脚本元素：

```
<scripts>
  <pre-scripts config:type="list">
    <script>
      ...
    </script>
  </pre-scripts>
</scripts>
```

## 4.30.2 分区后脚本

在 YaST 已完成分区并已写入 `/etc/fstab` 之后执行。空系统已装入到 `/mnt`。

必须按如下所示放置分区后脚本元素：

```
<scripts>
  <postpartitioning-scripts config:type="list">
    <script>
      ...
    </script>
  </postpartitioning-scripts>
</scripts>
```

### 4.30.3 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 进入已安装的系统并配置引导加载程序之前执行 chroot 脚本，也可以在 chroot 进入已安装的系统之后（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 环境脚本元素：

```
<scripts>
  <chroot-scripts config:type="list">
    <script>
      ...
    </script>
  </chroot-scripts>
</scripts>
```

### 4.30.4 安装后脚本

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安装后脚本元素：

```
<scripts>
  <post-scripts config:type="list">
    <script>
      ...
    </script>
  </post-scripts>
</scripts>
```

### 4.30.5 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` 配置的。

以下元素必须位于 <scripts><init-scripts config:type="list"><script> ... </script></init-scripts>...</scripts> 标记之间

表 4.1：INIT 脚本 XML 表示形式

元素	说明	注释
<u>location</u>	定义从中提取脚本的位置。该位置可与配置文件的位置相同（HTTP、FTP、NFS 等）。  <pre>&lt;location&gt;http://10.10.0.1/myInitScript.sh&lt;/location&gt;</pre>	必须定义 <location> 或 <source>。
<u>source</u>	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。  <pre>&lt;source&gt;&lt;![CDATA[echo "Testing the init script" &gt;/tmp/init_out.txt]]&gt;&lt;/source&gt;</pre>	必须定义 <location> 或 <source>。
<u>filename</u>	脚本的文件名。此文件将储存在 <u>/tmp</u> 下的临时目录中。  <pre>&lt;filename&gt;mynitScript5.sh&lt;/filename&gt;</pre>	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 ( <u>init-scripts</u> )。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。

元素	说明	注释
<u>rerun</u>	<p>正常情况下，一个脚本只运行一次，即使您使用 <u>ayast_setup</u> 多次运行 XML 文件也是如此。将此布尔值设置为 <u>true</u> 可更改此默认行为。</p> <pre>&lt;rerun config:type="boolean"&gt;true&lt;/rerun&gt;</pre>	可选。默认值为 <u>false</u> （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，以避免与文件语法以及控制文件中定义的其他标记产生混淆。

### 4.30.6 脚本 XML 表示形式

可针对上述所有脚本类型使用下面介绍的 XML 元素（chrooted 元素除外，它只能在 chroot 脚本中使用）。

表 4.2：脚本 XML 表示形式

元素	说明	注释
<u>location</u>	<p>定义从中提取脚本的位置。该位置可与控制文件的位置相同（HTTP、FTP、NFS 等）。</p> <pre>&lt;location&gt;http://10.10.0.1/myPreScript.sh&lt;/location&gt;</pre>	必须定义 <u>location</u> 或 <u>source</u> 。
<u>source</u>	<p>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。</p> <pre>&lt;source&gt;&lt;![CDATA[</pre>	必须定义 <u>location</u> 或 <u>source</u> 。



元素	说明	注释
	<pre>echo "Testing the pre script" &gt; /tmp/pre-script_out.txt ]]&gt; &lt;/source&gt;</pre>	
<u>interpreter</u>	<p>指定必须为脚本使用的解释器。支持的选项为 <u>shell</u> 和 <u>perl</u>。</p> <pre>&lt;interpreter&gt;perl&lt;/interpreter&gt;</pre>	可选；默认值为 <u>shell</u> 。
<u>file name</u>	<p>脚本的文件名。此文件将储存在 <u>/tmp</u> 下的临时目录中。</p> <pre>&lt;filename&gt;myPreScript5.sh&lt;/filename&gt;</pre>	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。
<u>feedback</u>	<p>如果此布尔值为 <u>true</u>，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。</p> <pre>&lt;feedback   config:type="boolean"&gt;true&lt;/feedback&gt;</pre>	可选；默认值为 <u>false</u> 。
<u>feedback_type</u>	<p>可以是 <u>message</u>、<u>warning</u> 或 <u>error</u>。在 &lt;report&gt; 部分设置这些弹出窗口的超时。</p> <pre>&lt;feedback_type&gt;warning&lt;/feedback_type&gt;</pre>	可选；如果未指定，将使用始终阻止弹出窗口。

元素	说明	注释
<u>debug</u>	<p>如果此元素为 <code>true</code>，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。</p> <pre>&lt;debug   config:type="boolean"&gt;true&lt;/debug&gt;</pre>	可选；默认值为 <code>true</code> 。
<u>notification</u>	<p>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。</p> <pre>&lt;notification&gt;Please wait   while script is running...&lt;/notification&gt;</pre>	可选；如果未配置，则不显示通知弹出窗口。
<u>param-list</u>	<p>可以指定提供给所要调用的脚本的参数。您可以指定多个 <code>param</code> 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。</p> <pre>&lt;param-list config:type="list"&gt;   &lt;param&gt;par1&lt;/param&gt;   &lt;param&gt;par2 par3&lt;/param&gt;   &lt;param&gt;"par4.1 par4.2"&lt;/param&gt; &lt;/param-list&gt;</pre>	可选；如果未配置，则不会将任何参数传递给脚本。
<u>rerun</u>	<p>一个脚本只运行一次。即使您使用 <code>ayast_setup</code> 多次运行 XML 文件，该脚本也只运行一次。将此布尔值设置为 <code>true</code> 可更改此默认行为。</p>	可选；默认值为 <code>false</code> ，表示脚本只运行一次。

元素	说明	注释
	<pre>&lt;rerun   config:type="boolean"&gt;true&lt;/ rerun&gt;</pre>	
<u>chrooted</u>	<p>在安装期间，新系统将装入到 <u>/mnt</u>。如果此参数设置为 <u>false</u>，AutoYaST 将不运行 <b>chroot</b>，并且在此阶段不会安装引导加载程序。如果该参数设置为 <u>true</u>，则 AutoYaST 会执行 <b>chroot</b> 进入 <u>/mnt</u> 并安装引导加载程序。结果是，您不再需要使用 <u>/mnt</u> 前缀就能更改新安装系统中的任何设置。</p> <pre>&lt;chrooted config:type="boolean" &gt;true&lt;/chrooted&gt;</pre>	<p>可选；默认值为 <u>false</u>。此选项仅适用于 chroot 环境脚本。</p>

## 4.30.7 脚本示例

### 例 4.60：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.sh</filename>
      <interpreter>shell</interpreter>
      <source><![CDATA[
#!/bin/sh
echo "Testing chroot (chrooted) scripts"
```

```

ls
]]>
    </source>
</script>
<script>
    <filename>chroot.sh</filename>
    <interpreter>shell</interpreter>
    <source><![CDATA[
#!/bin/sh
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[
#!/bin/sh

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[
#!/usr/bin/perl
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[
#!/bin/sh
echo "Running pre-install 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。

日志中包含使用以下命令执行外壳脚本时生成的输出：

```
/bin/sh -x SCRIPT_NAME 2&>/var/adm/autoinstall/logs/SCRIPT_NAME.log
```

## 4.31 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 4.61：SYSCONFIG 配置

```
<sysconfig config:type="list" >
  <sysconfig_entry>
    <sysconfig_key>XNTPD_INITIAL_NTPDATE</sysconfig_key>
    <sysconfig_path>/etc/sysconfig/xntp</sysconfig_path>
    <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。

## 4.32 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。

此功能需要安装 `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（例如 HTTP 服务器）上的某个位置：`<file_location>http://my.server.site/issue</file_location>`。

您可以通过指定以斜线结尾的 `file_path` 来创建目录。

### 例 4.62：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 4.63：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>

```

## 4.33 安装期间请求用户提供值

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放在以下 XML 结构中：



```

<general>
  <ask-list config:type="list">
    <ask>
      ...
    </ask>
  </ask-list>
</general>

```

表 4.3：请求用户提供值：XML 表示形式

元素	说明	注释
<u>question</u>	您要询问用户的问题。  <pre>&lt;question&gt;Enter the LDAP server&lt;/question&gt;</pre>	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）。
<u>default</u>	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。  <pre>&lt;default&gt;dc=suse,dc=de&lt;/default&gt;</pre>	可选。
<u>help</u>	在问题左侧显示的可选帮助文本。  <pre>&lt;help&gt;Enter the LDAP server address.&lt;/help&gt;</pre>	可选。
<u>title</u>	在问题上方显示的可选标题。  <pre>&lt;title&gt;LDAP server&lt;/title&gt;</pre>	可选。
<u>type</u>	要更改的元素类型。可能的值为 <u>symbol</u> 、 <u>boolean</u> 、 <u>string</u> 和 <u>integer</u> 。分区部分中的文件系统为符号，而用户配置中的 <u>encrypted</u> 元素为布尔值。如果在控制文件中查看	可选。默认值为 <u>string</u> 。如果 type 为 <u>symbol</u> ，则您还必须提供 selection 元素（参见下文）。

元素	说明	注释
	<p><code>config:type="...."</code> 属性，可以看到该元素的类型。还可以使用 <code>static_text</code> 作为类型。<code>static_text</code> 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。</p> <pre>&lt;type&gt;symbol&lt;/type&gt;</pre>	
<code>password</code>	<p>如果此布尔值设置为 <code>true</code>，将弹出口令对话框而不是简单的文本输入。仅当 <code>type</code> 为 <code>string</code> 时，将此项设置为 <code>true</code> 才有意义。</p> <pre>&lt;password   config:type="boolean"&gt;true&lt;/ password&gt;</pre>	可选。默认设置为 <code>false</code> 。
<code>pathlist</code>	<p><code>path</code> 元素的列表。<code>path</code> 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 <code>&lt;ldap&gt;&lt;ldap_server&gt;</code> 部分中找到 LDAP server 元素。因此，要更改该值，需将路径设置为 <code>ldap,ldap_server</code>。</p> <pre>&lt;pathlist config:type="list"&gt;   &lt;path&gt;networking,dns,hostname&lt;/ path&gt;   &lt;path&gt;...&lt;/path&gt; &lt;/pathlist&gt;</pre>	此信息是可选的，但您至少应该提供 <code>path</code> 或 <code>file</code> 。

元素	说明	注释
	<p>要更改控制文件中第一个用户的口令，需将路径设置为 <code>users,0,user_password</code>。<code>0</code> 表示控制文件中 <code>&lt;users config:type="list"&gt;</code> 用户列表内的第一个用户。<code>1</code> 是第二个用户，依此类推。</p> <pre> &lt;users config:type="list"&gt;   &lt;user&gt;     &lt;username&gt;root&lt;/username&gt;     &lt;user_password&gt;password to change&lt;/user_password&gt;     &lt;encrypted config:type="boolean"&gt;&gt;false&lt;/ encrypted&gt;   &lt;/user&gt;   &lt;user&gt;     &lt;username&gt;tux&lt;/username&gt;     &lt;user_password&gt;password to change&lt;/user_password&gt;     &lt;encrypted config:type="boolean"&gt;&gt;false&lt;/ encrypted&gt;   &lt;/user&gt; &lt;/users&gt; </pre>	
<code>file</code>	<p>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 <code>stage=initial</code> 阶段提问，并希望在第 2 阶段使用答案，则需要在 <code>chrooted=false</code> 的情况下运行的 <code>chroot</code> 脚本中复制答案文件。使用命令：<code>cp /tmp/my_answer /mnt/tmp/</code>。原因是第 1 阶段的 <code>/tmp</code> 位于 RAM 磁盘中，在重引导后将会丢失，但安装的系统已装入到 <code>/mnt/</code>。</p>	<p>此信息是可选的，但您至少应该提供 <code>path</code> 或 <code>file</code>。</p>

元素	说明	注释
	<pre>&lt;file&gt;/tmp/answer_hostname&lt;/file&gt;</pre>	
stage	<p>Stage 用于配置弹出问题的安装阶段。您可将此值设置为 <code>cont</code> 或 <code>initial</code>。 <code>initial</code> 表示在运行安装前脚本后的较短时间内，在安装的初期显示弹出窗口。 <code>cont</code> 表示在第一次引导系统时于完成首次重引导后显示包含问题的对话框。您在 <code>initial</code> 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 <code>initial</code> 阶段输入明文口令，则应会了解这一点。当然，在 <code>cont</code> 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。</p> <pre>&lt;stage&gt;cont&lt;/stage&gt;</pre>	<p>可选。默认值为 <code>initial</code>。</p>
<u>selection</u>	<p><code>selection</code> 元素包含 <code>entry</code> 元素的列表。每个 <code>entry</code> 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。</p> <pre>&lt;selection config:type="list"&gt;   &lt;entry&gt;     &lt;value&gt;       btrfs     &lt;/value&gt;     &lt;label&gt;       Btrfs File System     &lt;/label&gt;   &lt;/entry&gt;   &lt;entry&gt;     &lt;value&gt;       ext3     &lt;/value&gt;   &lt;/entry&gt; &lt;/selection&gt;</pre>	<p>对于 <code>type=string</code> 此元素为可选的，对于 <code>type=boolean</code> 此元素不可用，对于 <code>type=symbol</code> 此元素是必需的。</p>

元素	说明	注释
	<pre>         &lt;/value&gt;         &lt;label&gt;             Extended3 File System         &lt;/label&gt;     &lt;/entry&gt; &lt;/selection&gt; </pre>	
<u>dialog</u>	<p>您可以在每个对话框询问多个问题。为此，请使用整数指定 <code>dialog-id</code>。具有相同 <code>dialog-id</code> 的所有问题同属于一个对话框。对话框也按 ID 排序。</p> <pre> &lt;dialog config:type="integer"&gt;3&lt;/dialog&gt; </pre>	可选。
<u>element</u>	<p>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 <code>element-id</code>。对话框中的问题按 ID 排序。</p> <pre> &lt;element config:type="integer"&gt;1&lt;/element&gt; </pre>	可选（请参见 <code>dialog</code> ）。
<u>width</u>	<p>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。</p> <pre> &lt;width config:type="integer"&gt;50&lt;/width&gt; </pre>	可选。
<u>height</u>	<p>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。</p>	可选。

元素	说明	注释
	<pre>&lt;height config:type="integer"&gt;15&lt;/height&gt;</pre>	
<u>frametitle</u>	<p>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。</p> <pre>&lt;frametitle&gt;User data&lt;/frametitle&gt;</pre>	可选；默认不使用框架标题。
<u>script</u>	<p>您可以在回答问题后运行脚本。有关脚本的详细说明，请参见下表。</p> <pre>&lt;script&gt;...&lt;/script&gt;</pre>	可选；默认为无脚本。
<u>ok_label</u>	<p>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。</p> <pre>&lt;ok_label&gt;Finish&lt;/ok_label&gt;</pre>	可选。
<u>back_label</u>	<p>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。</p> <pre>&lt;back_label&gt;change values&lt;/back_label&gt;</pre>	可选。
<u>timeout</u>	<p>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，并需要通过确定来确认对话框中的操作。</p>	可选；缺失值将被解释为 <u>0</u> ，即关闭超时。

元素	说明	注释
	<pre>&lt;timeout config:type="integer"&gt;30&lt;/timeout&gt;</pre>	
default_value_script	<p>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 4.33.1 节“默认值脚本”）。如果您可以 <u>计算</u> 默认值（特别是与 <u>timeout</u> 选项结合使用时），此功能将十分有用。</p> <pre>&lt;default_value_script&gt;...&lt;/default_value_script&gt;</pre>	可选；默认为无脚本。

### 4.33.1 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 计算 默认值（特别是与 timeout 选项结合使用时），此功能将十分有用。

下面列出的元素必须放在以下 XML 结构中：

```
<general>
  <ask-list config:type="list">
    <ask>
      <default_value_script>
        ...
      </default_value_script>
    </ask>
  </ask-list>
</general>
```

表 4.4：默认值脚本：XML 表示形式

元素	说明	注释
<u>source</u>	<p>脚本的源代码。<u>echo</u> 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，则会使用常规默认元素。使用 <u>echo -n</u> 抑制 \n 时请小心，应回显合理的值而非布尔值 “okay”</p> <pre>&lt;source&gt;...&lt;/source&gt;</pre>	必须指定此值，否则不会执行任何操作。
<u>interpreter</u>	<p>要使用的解释器。</p> <pre>&lt;interpreter&gt;perl&lt;/interpreter&gt;</pre>	<p>默认值为 <u>shell</u>。</p> <p>您还可以设置 <u>/bin/myinterpreter</u> 作为值。</p>

## 4.33.2 脚本

您可以在回答问题后运行脚本。

下面列出的元素必须放在以下 XML 结构中：

```
<general>
  <ask-list config:type="list">
    <ask>
      <script>
        ...
      </script>
    </ask>
  </ask-list>
</general>
```



表 4.5：脚本：XML 表示形式

元素	说明	注释
<u>file name</u>	<p>脚本的文件名。</p> <pre>&lt;filename&gt;my_ask_script.sh&lt;/filename&gt;</pre>	默认值为 ask_script.sh
<u>source</u>	<p>脚本的源代码。与激活的 <u>rerun_on_error</u> 一起使用，可以检查输入的值的健全性。您的脚本可以创建一个包含对话框 ID 的 <u>/tmp/next_dialog</u> 文件，该 ID 指定 AutoYaST 要显示的下一个对话框。值 -1 会终止询问序列。如果未创建该文件，AutoYaST 将按正常顺序运行对话框（仅适用于自 11.0 起的版本）。</p> <pre>&lt;source&gt;...&lt;/source&gt;</pre>	必须指定此值，否则不会执行任何操作。
<u>environment</u>	<p>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 <u>VAL</u>。</p> <pre>&lt;environment config:type="boolean"&gt;true&lt;/environment&gt;</pre>	可选。默认为 <u>false</u> 。
<u>feedback</u>	<p>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。</p>	可选，默认值为 <u>false</u> 。

元素	说明	注释
	<pre>&lt;feedback   config:type="boolean"&gt;true&lt;/feedback&gt;</pre>	
<u>debug</u>	<p>一个布尔值，用于打开脚本执行的调试。</p> <pre>&lt;debug   config:type="boolean"&gt;true&lt;/debug&gt;</pre>	<p>可选，默认值为 <u>true</u>。此值也需要打开 <u>feedback</u>。</p>
<u>rerun_on_error</u>	<p>一个布尔值，在脚本包含退出代码 0（零）之前，它会让对话框一直保持打开状态。因此，您可以在脚本中分析并检查用户提供的答案，并使用 <u>feedback</u> 选项显示错误。</p> <pre>&lt;rerun_on_error   config:type="boolean"&gt;true&lt;/rerun_on_error&gt;</pre>	<p>可选，默认值为 <u>false</u>。此值应与 <u>feedback</u> 选项一起使用。</p>

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>
      <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>tion>
</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>
```

## 4.34 内核转储



### 注意：可用性

此功能不适用于 IBM Z (s390x) 体系结构。

使用 Kdump 时，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 4.64：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_COPY_KERNEL>true</KDUMP_COPY_KERNEL>
    <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_T0>tux@example.com</KDUMP_NOTIFICATION_T0>
<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>

```

### 4.34.1 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。预留的内存将用于装载第二个内核，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 预留的内存量的建议，请参见《系统分析和微调指南》，第 17 章“Kexec 和 Kdump”，第 17.7.1 节“Kdump 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 会在系统具有 256 MB 到 2 GB RAM 时预留 64 MB crashkernel 内存，并在系统具有 2 GB 以上的 RAM 时预留 128 MB crashkernel 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，可使用诸如 72M,low 和 256M,high 的值：

#### 例 4.65：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>
```

下表说明了预留内存所需的设置：

表 4.6：KDUMP 内存预留设置：XML 表示形式

元素	说明	注释
<u>add_crash_kernel</u>	如果应预留内存并已启用 Kdump，请设置为 <u>true</u> 。  <pre>&lt;add_crash_kernel   config:type="boolean"&gt;true&lt;/ add_crash_kernel&gt;</pre>	必要
<u>crash_kernel</u>	使用上述 crashkernel 命令行语法。  <pre>&lt;crash_kernel&gt;256M:64M&lt;/crash_kernel&gt;</pre> 还支持值列表。	必要



元素	说明	注释
	<pre>&lt;crash_kernel config:type="list"&gt;   &lt;listentry&gt;72M,low&lt;/listentry&gt;   &lt;listentry&gt;256M,high&lt;/listentry&gt; &lt;/crash_kernel&gt;</pre>	

## 4.34.2 转储保存

本节介绍崩溃转储的储存位置及储存方式。

### 4.34.2.1 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。

要将整个内核和调试信息（如果已安装）保存到同一个目录，请将 `KDUMP_COPY_KERNEL` 设置为 `true`。您将在一个目录中获得分析转储所需的全部信息（内核模块及其调试信息除外）。

### 4.34.2.2 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。

### 4.34.2.3 摘要

表 4.7：转储目标设置：XML 表示形式

元素	说明	注释
<code>KDUMP_SAVEDIR</code>	一个 URL，指定要将转储和相关文件保存到的目标。  <pre>&lt;KDUMP_SAVEDIR&gt;file:///var/crash/&lt;/KDUMP_SAVEDIR&gt;</pre>	必要
<code>KDUMP_COPY_KERNEL</code>	如果不仅要保存转储到 <code>KDUMP_SAVEDIR</code> ，还要将内核及其调试信息（如果已安装）保存到此目标，请设置为 <code>true</code> 。  <pre>&lt;KDUMP_COPY_KERNEL&gt;&gt;false&lt;/KDUMP_COPY_KERNEL&gt;</pre>	可选
<code>KDUMP_FREE_DISK_SIZE</code>	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。	可选

元素	说明	注释
	<pre>&lt;KDUMP_FREE_DISK_SIZE&gt;64&lt;/KDUMP_FREE_DISK_SIZE&gt;</pre>	
<u>KDUMP_KEEP_OLD_DUMPS</u>	<p>当 <u>KDUMP_SAVEDIR</u> 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。</p> <pre>&lt;KDUMP_KEEP_OLD_DUMPS&gt;4&lt;/KDUMP_KEEP_OLD_DUMPS&gt;</pre>	可选

### 4.34.3 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）。

您需要在 KDUMP\_NOTIFICATION\_TO 中提供一个确切地址。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。

表 4.8：电子邮件通知设置：XML 表示形式

元素	说明	注释
<u>KDUMP_NOTIFICATION_TO</u>	<p>应将电子邮件发送到的一个确切电子邮件地址。可以在 <u>KDUMP_NOTIFICATION_CC</u> 中指定其他收件人。</p> <pre>&lt;KDUMP_NOTIFICATION_TO</pre>	可选（如果为空，则禁用通知）

元素	说明	注释
	<pre>&gt;tux@example.com&lt;/ KDUMP_NOTIFICATION_T0&gt;</pre>	
<u>KDUMP_NOTIFICATION_CC</u>	<p>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。</p> <pre>&lt;KDUMP_NOTIFICATION_CC &gt;wilber@example.com geeko@example.com&lt;/ KDUMP_NOTIFICATION_CC&gt;</pre>	可选
<u>KDUMP_SMTP_SERVER</u>	<p>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 <u>KDUMP_SMTP_USER</u> 和 <u>KDUMP_SMTP_PASSWORD</u>），但不支持 TLS/SSL。</p> <pre>&lt;KDUMP_SMTP_SERVER&gt;email.suse.de&lt;/ KDUMP_SMTP_SERVER&gt;</pre>	可选（如果为空，则禁用通知）
<u>KDUMP_SMTP_USER</u>	<p>与 <u>KDUMP_SMTP_PASSWORD</u> 一起使用的、用于 SMTP 身份验证的用户名。</p> <pre>&lt;KDUMP_SMTP_USER&gt;bwalke&lt;/ KDUMP_SMTP_USER&gt;</pre>	可选
<u>KDUMP_SMTP_PASSWORD</u>	<p>与 <u>KDUMP_SMTP_USER</u> 一起使用的、用于 SMTP 身份验证的口令。</p> <pre>&lt;KDUMP_SMTP_PASSWORD&gt;geheim&lt;/ KDUMP_SMTP_PASSWORD&gt;</pre>	可选

### 4.34.4 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`。

表 4.9：内核设置：XML 表示形式

元素	说明	注释
<code>KDUMP_KERNELVER</code>	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）。 <pre>&lt;KDUMP_KERNELVER&gt;2.6.27-default&lt;/KDUMP_KERNELVER&gt;</pre>	可选（如果为空则使用自动检测）
<code>KDUMP_COMMANDLINE_APPEND</code>	Kdump 内核的其他命令行参数。 <pre>&lt;KDUMP_COMMANDLINE_APPEND&gt;console=ttyS0,57600&lt;/KDUMP_COMMANDLINE_APPEND&gt;</pre>	可选
<code>KDUMP_Command Line</code>	重写自动生成的 Kdump 命令行。须谨慎使用该功能。通常 <code>KDUMP_COMMANDLINE_APPEND</code> 应已足够。 <pre>&lt;KDUMP_COMMANDLINE_APPEND</pre>	可选

元素	说明	注释
	<pre>&gt;root=/dev/sda5 maxcpus=1 irqpoll&lt;/ KDUMP_COMMANDLINE&gt;</pre>	

## 4.34.5 专家设置

表 4.10：专家设置：XML 表示形式

元素	说明	注释
<u>KDUMP_IMMEDIATE_REBOOT</u>	<p>如果在保存转储后应自动重引导系统，则为 <code>true</code>，否则为 <code>false</code>。默认设置为自动重引导系统。</p> <pre>&lt;KDUMP_IMMEDIATE_REBOOT &gt;true&lt;/ KDUMP_IMMEDIATE_REBOOT&gt;</pre>	可选
<u>KDUMP_VERBOSE</u>	<p>用于指定 Kdump 过程详细程度的位掩码。有关细节，请阅读 <code>kdump(5)</code>。</p> <pre>&lt;KDUMP_VERBOSE&gt;3&lt;/ KDUMP_VERBOSE&gt;</pre>	可选
<u>KEXEC_OPTIONS</u>	<p>装载 Kdump 内核时要传递给 <code>kexec</code> 的其他选项。通常为 <code>空</code>。</p> <pre>&lt;KEXEC_OPTIONS&gt;--noio&lt;/ KEXEC_OPTIONS&gt;</pre>	可选

## 4.35 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
<code>chroot</code>	0/1	必须将 DNS 服务器监禁在 <code>chroot</code> 中。
<code>start_service</code>	0/1	启用 Bind（系统启动时执行）。
<code>use_ldap</code>	0/1	在 LDAP 而不是本机配置文件中储存设置。

### 例 4.66：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
<code>logging</code>	DNS 服务器日志记录选项。
<code>options</code>	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。
<code>zones</code>	服务器已知的 DNS 区域列表，包括所有设置、记录和 SOA 记录。

#### 例 4.67：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>
      </soa>
    </listentry>
  </zones>
</dns-server>
```



```

    <server>aaa.aaa.cc.</server>
    <zone>@</zone>
</soa>
<soa_modified>1</soa_modified>
<ttl>2d</ttl>
<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>

```

## 4.36 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。
<code>start_service</code>	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 DHCP 服务器）。
<code>use_ldap</code>	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。

元素	值	说明
<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”可监听非标准 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 4.68：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>
  </directives>
</settings_entry>

```

```

        <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>

```

## 4.37 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《安全指南》，第 19 章“伪装和防火墙”，第 19.4 节“`firewalld`”。

**！ 重要：与基于 SuSEFirewall2 的配置文件的有限向后兼容性**  
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。

### 4.37.1 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完成的。

属性	值	说明
<code>start_firewall</code>	布尔	是否在应用配置后立即启动 <code>firewalld</code> 。
<code>enable_firewall</code>	布尔	是否在每次启动系统时启动 <code>firewalld</code> 。
<code>default_zone</code>	区域名	默认区域适用于未显式指派的任何设置。
<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>broadcast</code> 、 <code>all</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。

属性	值	说明
<u>short</u>	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。
<u>description</u>	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。
<u>target</u>	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 <u>ACCEPT</u> 、 <u>%REJECT%</u> 、 <u>DROP</u> 和 <u>default</u> 。如果未指定，则使用 <u>default</u> 。有关值的细节，请参见 <a href="https://firewalld.org/documentation/zone/options.html">https://firewalld.org/documentation/zone/options.html</a> 。

## 4.37.2 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。

属性	值	说明
<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	将在指派区域中启用或禁用网络地址转换 (NAT)。

### 4.37.3 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 4.69：FIREWALL 部分示例

```
<firewall>
  <enable_firewall>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>
      </ports>
    </zone>
  </zones>
</firewall>
```

```
<port>80/tcp</port>
<port>443/tcp</port>
<port>8080/tcp</port>
</ports>
</zone>
<zone>
  <name>dmz</name>
  <interfaces config:type="list">
    <interface>eth1</interface>
  </interfaces>
</zone>
</zones>
</firewall>
```

## 4.38 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。

### 4.38.1 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 USB 设备 URI，因此 URI 是在运行时由 CUPS 后端 `usb` 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，将打印作业提交到 CUPS 守护程序进程 (`cupsd`)。 `cupsd` 会将打印作业转发到网络中的 CUPS 打印服务器，打印作业将在该服务器中进行处理。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 CPUS 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 4.70 “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 4.70：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 SP2 开始，AutoYaST 仅支持修改 `cupsd.conf`，因为 `cups-files.conf` 中的默认设置便足以满足一般的打印设置。

## 4.38.2 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 4.71：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>
```

## 4.39 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 4.72：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
<code>import</code>	true/false	将导入 SSH 密钥。如果设置为 <code>false</code> ，则不会导入任何内容。
<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。

## 4.40 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），剩余的配置任务可委派给其他工具。



### 注意：只有 Salt 受官方支持

虽然本文档中提及了 Puppet，但只有 Salt 才受官方支持。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。

## 4.40.1 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 4.73：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 4.74：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
<u>type</u>	字符串	配置管理名称。目前只有 <u>salt</u> 受官方支持。
<u>master</u>	字符串	配置管理服务器的主机名或 IP 地址。
<u>auth_attempts</u>	整数	尝试连接服务器的最大次数。默认值为三次尝试。
<u>auth_time_out</u>	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。
<u>keys_url</u>	所用密钥的 URL	包含文件 <u>default.key</u> 和 <u>default.pub</u> 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。
<u>enable_services</u>	True/False	安装后在客户端启用配置管理服务。默认值为 <u>true</u> 。

## 4.40.2 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。可以改用独立（或无主服务器）模式的 Salt 或 Puppet。

由于没有服务器，AutoYaST 需要知道从何处获取配置。请将配置放入 TAR 存档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 Salt 状态放入 salt 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 4.75：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
<code>type</code>	字符串	配置管理名称。目前只有 <code>salt</code> 受官方支持。
<code>states_url</code>	URL	Salt 状态 TAR 存档的位置。此存档可以包含公式和 <code>pillar</code> 。文件必须位于 <code>salt</code> 目录。
<code>pillar_url</code>	URL	包含 <code>pillar</code> 的 TAR 存档的位置。
<code>modules_url</code>	URL	Puppet 模块的位置。

### 4.40.3 SUSE Manager Salt 公式支持

AutoYaST 在以独立模式运行时可以提供对 SUSE Manager Salt 公式的支持。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 使用规则和类管理批量安装

5 规则和类 215

## 5 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。

## 5.1 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/
```



此命令会装载 <http://10.10.0.1/profile/rules/rules.xml>（目录名称末尾的斜线非常重要）。

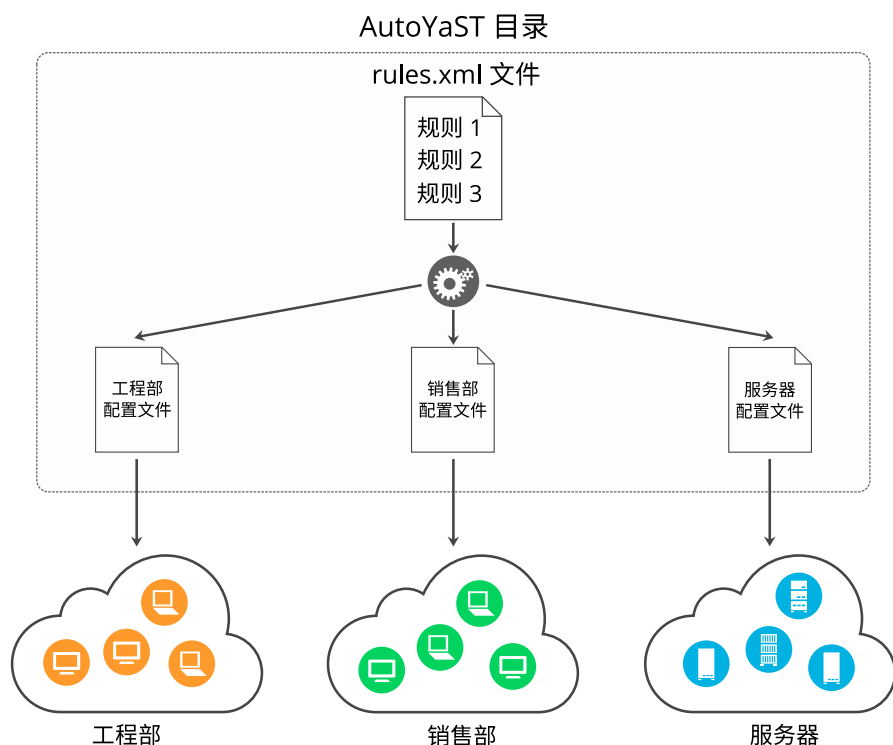


图 5.1：规则

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。

### 5.1.1 规则文件说明

#### 例 5.1：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/configns">
  <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 5.2：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/shda 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。

### 5.1.2 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`。

### 5.1.3 规则的匹配类型

可以使用五种不同的 `match_types`:

- `exact` (默认)
- `greater`
- `lower`
- `range`
- `regex` (类似于 Bash 中的简单 `==` 运算符)

如果使用 `exact`，则字符串必须与指定的值完全匹配。`regex` 可用于匹配子字符串，例如，`ntel` 将匹配 Intel、intel 和 intelligent。`greater` 和 `lower` 可用于匹配 `memsize` 或 `totaldisk` 等值。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`。

### 5.1.4 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，则可让规则匹配。

为此，可以在 `rules.xml` 文件中使用 `operator` 元素。`and` 和 `or` 是可用的运算符，`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>
```

### 5.1.5 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。

### 5.1.6 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。

如果您对系统上的某个值不确定，请运行 `/usr/lib/YaST/bin/y2base 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 5.1：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>host name</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>	要安装的产品的名称。	此属性必须始终完全匹配。
<u>installed_product_</u> <u>version</u>	要安装的产品的版本。	此属性必须始终完全匹配。

属性	值	说明
<u>network</u>	主机的网络地址	此属性必须始终完全匹配。
<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	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
<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

属性	值	说明
<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。

## 5.1.7 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	说明
<u>dialog_nr</u>	具有相同 <u>dialog_nr</u> 的所有规则将出现在同一个弹出对话框中。相同的 <u>dialog_nr</u> 可以显示在多个规则中。  <pre>&lt;dialog_nr   config:type="integer"&gt;3&lt;/ dialog_nr&gt;</pre>	此元素是可选的，缺少的 <u>dialog_nr</u> 的默认值始终为 <u>0</u> 。要为所有规则使用一个弹出对话框，无需指定 <u>dialog_nr</u> 。
<u>element</u>	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	可选。如果留空，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）。



属性	值	说明
	<p>ID <u>1</u> 。（此行为与 <u>ask</u> 对话框相反，后者可以对多个对话框使用同一 ID。）</p> <pre>&lt;element   config:type="integer"&gt;3&lt;/element&gt;</pre>	
<u>title</u>	<p>弹出对话框的标题</p> <pre>&lt;title&gt;Desktop Selection&lt;/title&gt;</pre>	可选
<u>question</u>	<p>在弹出对话框中的复选框后显示的问题。</p> <pre>&lt;question&gt;GNOME Desktop&lt;/question&gt;</pre>	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。
<u>timeout</u>	<p>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。</p> <pre>&lt;timeout   config:type="integer"&gt;30&lt;/timeout&gt;</pre>	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。
<u>conflicts</u>	<p>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。</p>	<u>可选</u>

属性	值	说明
	<pre> &lt;conflicts   config:type="list"&gt;     &lt;element       config:type="integer"&gt;1&lt;/ element&gt;     &lt;element       config:type="integer"&gt;5&lt;/ element&gt;     ...   &lt;/conflicts&gt; </pre>	

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>

```

## 5.2 类

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>
```

在以上示例中，文件 `Software.xml` 必须位于子目录 `classes/TrainingRoom/` 中。将从提取 AutoYaST 控制文件和规则的同一位置提取该文件。

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）



图 5.2：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。

## 5.3 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 A.1 “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。

## 5.4 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>

```

```

<rule>
  <board_vendor>
    <match>ntel</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>
</rule>

```

```
</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>
```



## IV 了解自动安装过程

### 6 自动安装过程 233

## 6 自动安装过程

### 6.1 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视系统消息。

#### 6.1.1 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。

#### 6.1.2 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。

#### 6.1.3 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。

## 6.2 选择正确的引导媒体

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 装载适当的内核和 `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。

### 6.2.1 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，您可以构建一张配合 AutoYaST 使用的可引导闪存盘。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。

```
tux > sudo dd if=IMAGE of=FLASH_DISK bs=4M && sync
```

需将 `IMAGE` 替换为 `SLE-15-SP2-Online-ARCH-GM-media1.iso` 或 `SLE-15-SP2-Full-ARCH-GM-media1.iso` 映像文件的路径。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：

```
root # 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：

```
root # 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-SP2-Full-ARCH-GM-media1.iso` 使用的命令是：

```
dd if=SLE-15-SP2-Full-ARCH-GM-media1.iso of=/dev/sdc1 bs=4M && sync
```

在运行 **dd** 命令时，不得装入该设备。请注意，分区中的所有数据都将被删除！

## 6.2.2 从 SUSE Linux Enterprise 安装媒体引导

您可将 SUSE Linux Enterprise 安装媒体（[SLE-15-SP2-Online-ARCH-GM-media1.iso](#) 或 [SLE-15-SP2-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。

## 6.2.3 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。有关设置所需基础结构的说明，请参见《部署指南》，第 11 章“远程安装”。

如果您通过 PXE 安装，将无限循环地运行安装。发生此情况的原因是：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“魔术” `__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。

## 6.3 调用自动安装过程

### 6.3.1 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：

表 6.1： `linuxrc` 的关键字

关键字	值
<code>autoupgrade</code>	启动使用 AutoYaST 的自动升级；请参见第 4.10 节 “升级”。
<code>autoyast</code>	用于自动安装的控制文件的位置；有关细节，请参见 <a href="#">AutoYaST 控制文件位置</a> 。
<code>ifcfg</code>	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 C.3 节 “高级网络设置”。

关键字	值
<u>insmod</u>	要装载的内核模块
<u>install</u>	安装目录的位置，例如 <u>install=nfs://192.168.2.1/CDs/</u> 。
<u>instmode</u>	安装模式，例如 <u>nfs</u> 、 <u>http</u> 等（如果设置了 <u>install</u> 则不需要）。
<u>rootpassword</u>	root 用户的口令（如果未在 AutoYaST 配置文件中指定）
<u>server</u>	要连接以获取源目录的服务器 (NFS)
<u>serverdir</u>	NFS 服务器上的目录
<u>y2confirm</u>	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，或者在需要装载特殊内核模块时，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` 等同于 `autoyast=usb://localhost/profile.xml`。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 `autoinst.xml` 的文件位于以下三个位置之一，则会启动 `autoyast`：

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 驱动器等装入的外部设备）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。（此等同于 `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=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 的适当安装媒体（SLE-15-SP2-Online-ARCH-GM-media1.iso 或 SLE-15-SP2-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-SP2-Online-ARCH-GM-media1.iso](#) 或 [SLE-15-SP2-Full-ARCH-GM-media1.iso](#)）时，需要可通过闪存盘（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=http://...](#) 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它放入安装源的根目录，并将它命名为 [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=http://...](#) 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。

```
tux > 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
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 6.1：确定 IP 地址的十六进制代码

```
tux > /usr/bin/gethostip 10.10.0.1
10.10.0.1 10.10.0.1 0A0A0001
```

## 6.3.2 自动安装单个系统

在未建立任何网络连接的情况下自动安装某个系统的最简单方法是使用原始 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。

## 6.3.3 将 `linuxrc info` 文件与 AutoYaST 控制文件结合使用

如果您选择使用 `info` 文件或者以引导选项的形式向 `linuxrc` 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 6.2：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。

## 6.4 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。

### 6.4.1 安装后和系统配置

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，这些脚本是在尚未装入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”）。

## 6.4.2 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 在已安装系统上使用 AutoYaST

### 7 在安装的系统中运行 AutoYaST 246

## 7 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。

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 4.9.7 节 “在第 2 个阶段安装软件包”）部分、用户（第 4.29.1 节 “用户”）部分和 NTP 客户端（第 4.20 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：

```
tux > sudo yast2 ayast_setup setup filename=/tmp/install_foo.xml
dopackages="yes"
```

有关详细信息，请运行 **yast2 ayast\_setup longhelp**



## VI 附录

- A 处理规则 **249**
- B AutoYaST FAQ — 常见问题 **250**
- C 高级 **linuxrc** 选项 **254**
- D SLE 12 与 15 中 AutoYaST 配置文件之间的差异 **259**

## 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。

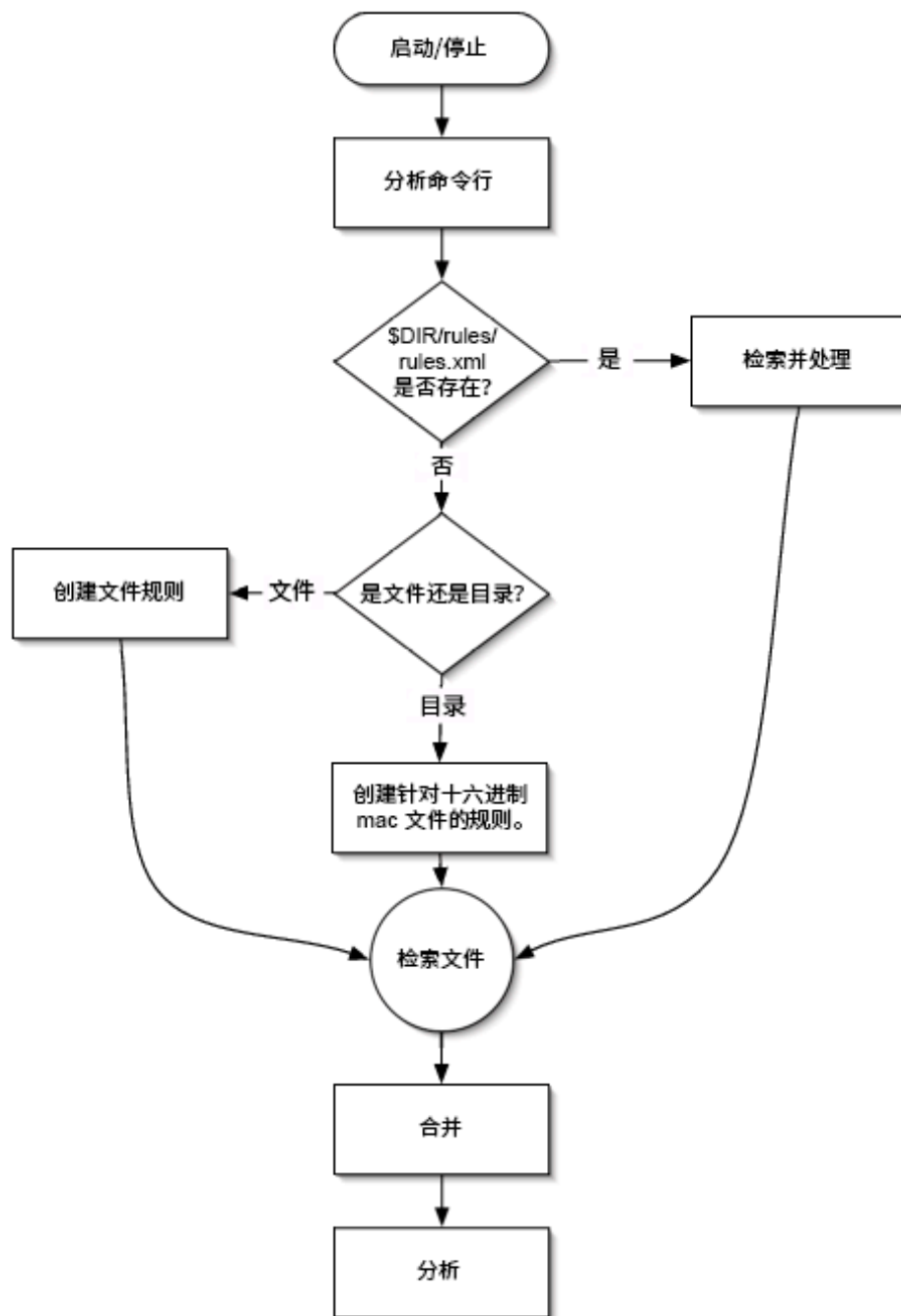


图 A.1：规则检索过程

## B AutoYaST FAQ — 常见问题

### 1. 如何调用 AutoYaST 安装？

在所有 SUSE Linux Enterprise Server 版本中，您可以通过将 `autoyast=<PATH_TO_PROFILE>` 添加到内核参数列表来调用自动安装。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 6.3 节 “调用自动安装过程”。

### 2. 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。

### 3. 如何创建 AutoYaST 配置文件？

创建 AutoYaST 配置文件的最简单方法是使用现有 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。

### 4. 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”。

### 5. 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/configns">
  <users config:type="list">
    <user>
      <encrypted config:type="boolean">false</encrypted>
      <user_password>linux</user_password>
      <username>root</username>
    </user>
  </users>
</profile>
```

6. 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>
```

7. 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` 引用配置文件。

8. 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：

```
tux > /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使用 `dont_merge` 元素时要怎样做。有关更多信息，请参见第 5.4 节“合并规则和类”。

9. 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>
```

10. 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。

11. **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）传递到 **linuxrc** 参数列表来隐藏该消息。

12. 我想从 DVD/USB/HD 安装，而从网络提取 XML 文件。

您需要将 `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 C.3 节“高级网络设置”。

13. 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。必须十分谨慎地设置环境（DHCP、TFTP 等）。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>

```

- 14.** 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 <http://lists.opensuse.org/opensuse-autoinstall/> 加入我们。

## C 高级 **linuxrc** 选项

**linuxrc** 是在装载内核之后、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装载模块、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> 上找到其联机文档。



### 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。

## C.1 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：首先是闪存盘（例如 USB 记忆棒）上的 `/info`，如果在该位置未找到，则查找 `initrd` 中的 `/info`。然后，它会分析参数的内核命令行。您可以通过设置 `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。

## C.2 `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 C.1：高级 `linuxrc` 关键字

关键字：示例值	说明
<code>addswap: 0 3 /dev/sda5</code>	如果为 0，则永远不请求交换；如果该参数是正数 <code>n</code> ，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。
<code>autoyast: ftp://AUTOYASTFILE</code>	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 <a href="#">AutoYaST 控制文件位置</a> 。
<code>bootptimeout: 10</code>	BOOTP 请求的超时为 10 秒。



关键字：示例值	说明
<u>bootwait: 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>	运行命令。
<u>forceinsmod: 0 1</u>	运行 <b>insmod</b> 命令时使用 <b>-f</b> （强制）选项。
<u>forcerootimage: 0 1</u>	将安装系统装载到 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 C.3 节“高级网络设置”。
<u>insmod: MODULE</u>	装载 <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 <a href="https://en.opensuse.org/SDB:Linuxrc#url_descr">https://en.opensuse.org/SDB:Linuxrc#url_descr</a> 。
<u>keytable: de-lat1-nd</u>	要装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>	启用通过 syslog 的远程日志记录。
<u>memloadimage: 50000</u>	如果可用内存超过 50000 KB，则将安装系统装载到 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。

关键字：示例值	说明
<code>memYaSTText: 10000</code>	如果可用内存低于 10000 KB，则在启动 YaST 之前请求交换。
<code>proxy: 10.10.0.1</code>	代理（FTP 或 HTTP）。
<code>rescue: 1 nfs:// server/dir</code>	装载救援系统；URL 变体显式指定救援映像的位置。
<code>rescueimage: /suse/ images/rescue</code>	救援系统映像的位置。
<code>rootimage: /suse/ 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>	装载 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。

## C.3 高级网络设置

即使已将 `hostip`、`nameserver` 和 `gateway` 等参数传递给了 `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` 不是一个 `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，将不会自动启动网络。

因此，需要显式启动网络。此操作过去常常是通过 `linuxrc` 参数 `netsetup` 完成的。从 SUSE Linux Enterprise Server 12 开始，可以使用参数 `ifcfg`。此参数提供更多配置选项，例如，配置多个接口。`ifcfg` 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，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,\nOPTION1=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\n192.168.1.117,example.com example.net"
```

有关详细信息，请参见[https://en.opensuse.org/SDB:Linuxrc#Network\\_Configuration](https://en.opensuse.org/SDB:Linuxrc#Network_Configuration)。

## D SLE 12 与 15 中 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。

### D.1 产品选择

为了能够与针对 SLE 15 以前产品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，将使用这种启发方法。自动产品选择基于配置文件中的软件包和模式选择。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显式产品选择。

有关显式产品选择的信息，请参见第 4.9.1 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。

### D.2 软件

SUSE Linux Enterprise Server 15 SP2 安装媒体仅包含可安装的极少量软件包。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-SP2-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。

## D.2.1 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，请将 `addons` 标记用于 `suse_register` 部分中的每个模块/扩展。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。要获取标记 `name`、`version` 和 `arch` 的相应值，请在已安装 SLE 15 SP2 的系统上运行命令 **`SUSEConnect --list-extensions`**。

### 例 D.1：添加模块和扩展（联机）

```
<suse_register>
  <addons config:type="list">
    <addon>
      <name>sle-module-basesystem</name>
      <version>15.2</version>
      <arch>x86_64</arch>
    </addon>
    <addon>
      <name>sle-module-server-applications</name>
      <version>15.2</version>
      <arch>x86_64</arch>
    </addon>
    <addon>
      <name>sle-module-live-patching</name>
      <version>15.2</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”。

## D.2.2 使用 SLE-15-SP2-Full-ARCH-GM-media1.iso 映像添加模块或扩展

要使用 SLE-15-SP2-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 部分中创建项。以下 XML 代码会从包含该映像的 USB 闪存盘添加 Basesystem 模块：

例 D.2：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

标记 `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：从本地服务器使用安装媒体映像

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。为此，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_sp2_full/</media_url>
```

## D.2.3 已重命名的软件模式

软件模式自 SUSE Linux Enterprise Server 15 开始也已发生更改。某些模式已重命名；下表中提供了简短摘要。

旧的 SLE 12 模式	新的 SLE 15 模式
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

请仔细检查定义的模式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模式和软件包。

#### 注意

- 上表中的模式重命名并非一对一替换；某些模式的内容也发生了更改，某些软件包已移到不同模式，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模式中，并选择性地使用 `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。

## D.3 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。

## D.4 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。

### D.4.1 GPT 已成为 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。但是，如果您想要保留旧行为，可以通过在配置文件中将 `disklabel` 元素设置为 `msdos` 来显式指定旧行为。

### D.4.2 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。

但是，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 4.5.3.2 节“分区配置”。

### D.4.3 强制主分区

仍可将某个分区强制作为主分区（仅限在 MS-DOS 分区表中），方法是将 `primary_type` 设置为 `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。

### D.4.4 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）。如以下示例中所示，可以使用 `subvolumes_prefix` 标记为每个分区指定前缀名称：

例 D.3：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 D.4：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。

### D.4.5 Btrfs：禁用子卷

可以通过将 `create_subvolumes` 设置为 `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>]]>
```

### D.4.6 不再支持读取现有的 `/etc/fstab`

在 SUSE Linux Enterprise Server 15 上，尝试确定分区布局时，不再支持从先前的安装读取现有 `/etc/fstab` 的功能。

## D.4.7 已删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。

## D.5 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 SLE 12 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 D.1：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>
<code>FW_LOG_DROP_CRIT</code>	<code>FW_LOAD_MODULES</code>
<code>FW_MASQUERADE</code>	<code>FW_LOG_ACCEPT_ALL</code>
<code>FW_SERVICES_{DMZ, INT, EXT}_{TCP, UDP, IP}</code>	<code>FW_LOG_ACCEPT_CRIT</code>
	<code>FW_PROTECT_FROM_INT</code>
	<code>FW_ROUTE</code>
	<code>FW_SERVICES_{DMZ, EXT, INT}_RPC</code>

支持（但已弃用）	不支持
	<u>FW_SERVICES_ACCEPT_RELATED_{DMZ, EXT, INT}</u>

不再可用的 SuSEfirewall2 配置选项要么在 `firewalld` 中没有等效的映射，要么将在未来的 SUSE Linux Enterprise Server 版本中提供支持。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 4.30 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 SLE 12 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>true</enable_firewall>
  <start_firewall>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：

## D.5.1 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 D.2：SUSEFIREWALL2 和 `firewalld` 区域的映射

<code>firewalld</code> (SLE 15)	SuSEfirewall2 (SLE 12)
dmz	DMZ
external	EXT with <code>FW_MASQUERADE</code> set to <code>yes</code>

firewalld (SLE 15)	SuSEfirewall2 (SLE 12)
public	EXT with <u>FW_MASQUERADE</u> set to <u>no</u>
internal	INT with <u>FW_PROTECT_FROM_INT</u> set to <u>yes</u>
trusted	INT with <u>FW_PROTECT_FROM_INT</u> set to <u>no</u>
block	无
drop	无
home	无
work	无

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。

### D.5.1.1 默认配置

以下两个示例演示应用于接口 eth0、eth1、wlan0 和 wlan1 的默认配置。

**例 D.5：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 D.6：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>

```

### D.5.1.2 掩蔽和保护内部区域

以下两个示例演示如何使用掩蔽和受保护内部区域来配置接口 eth0、eth1、wlan0 和 wlan1。

#### 例 D.7：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 D.8：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>

```

## D.5.2 打开端口

SuSEfirewall2 中使用 `FW_SERVICES_{DMZ,EXT,INT}_{TCP,UDP,IP,RPC}` 标记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`，每个端口、端口范围和服务在相应区域的 `port` 部分中都具有单独的项。IP 服务需在 `protocol` 部分中具有单独的项。

受 SuSEfirewall2 支持的 RPC 服务不再受 `firewalld` 的支持。

### 例 D.9：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 D.10： 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>
        <protocol>icmp</protocol>
        <protocol>gre</protocol>
      </protocols>
    </zone>
  </zones>
</firewall>

```

### D.5.3 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（等效于 `firewalld` 中的服务）。

**例 D.11： 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 D.12： 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>
```

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](https://en.opensuse.org/SuSEfirewall2/Service_Definitions_Added_via_Packages) 
- `firewalld` RPM 打包 [https://en.opensuse.org/firewalld/RPM\\_Packaging](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。



## D.5.4 更多信息

- 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 (<http://www.firewalld.org/documentation/>) 

## D.6 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。需要更新 SLE 12 中包含带有 `ntp:client` 部分的 AutoYaST 配置文件。

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 D.13：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>
```

## D.7 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。要执行第二个安装阶段，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。

## D.8 已删除 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，到目前为止尚无替代模块。如果您要重复使用 SLE 12 配置文件，请确保它不包含 `ca_mgm` 部分。

## D.9 升级

### D.9.1 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 SP2 中，升级与否始终由依赖性解析程序决定，相当于使用 `zypper dup`。

因此，`software` 部分中的 `only_installed_packages` 选项已过时。

### D.9.2 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` 选项重新添加这些储存库。

**例 D.14：** 有关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。