HP Server Automation

Enterprise 版

软件版本：10.0

用户指南：应用程序配置

文档发布日期：2013 年 6 月 13 日
软件发布日期：2013 年 6 月

法律声明

担保

HP 产品和服务的唯一担保已在此类产品和服务随附的明示担保声明中提出。此处的任何内容均不构成额外担保。HP 不会为此处出现的技术或编辑错误或遗漏承担任何责任。

此处所含信息如有更改，恕不另行通知。

受限权利声明

机密计算机软件。必须拥有 HP 授予的有效许可证，方可拥有、使用或复制本软件。按照 FAR 12.211 和 12.212，并根据供应商的标准商业许可的规定，商业计算机软件、计算机软件文档与商品技术数据授权给美国政府使用。

版权声明

© Copyright 2001-2013 Hewlett-Packard Development Company, L.P.

商标声明

Adobe® 是 Adobe Systems Incorporated 的商标。

Intel® 和 Itanium® 是 Intel Corporation 在美国和其他国家 / 地区的商标。

Microsoft®、 Windows®、 Windows® XP 是 Microsoft Corporation 在美国的注册商标。

Oracle 和 Java 是 Oracle 和 / 或其附属公司的注册商标。

UNIX® 是 The Open Group 的注册商标。

支持

请访问 HP 软件联机支持网站：

http://www.hp.com/go/hpsoftwaresupport

此网站提供了联系信息，以及有关 HP 软件提供的产品、服务和支持的详细信息。

HP 软件联机支持提供客户自助解决功能。通过该联机支持，可快速高效地访问用于管理业务的各种交互式技术支持工具。作为尊贵的支持客户，您可以通过该支持网站获得下列支持：

- 搜索感兴趣的知识文档
- 提交并跟踪支持案例和改进请求
- 下载软件修补程序
- 管理支持合同
- 查找 HP 支持联系人
- 查看有关可用服务的信息
- 参与其他软件客户的讨论
- 研究和注册软件培训

大多数提供支持的区域都要求您注册为 HP Passport 用户再登录，很多区域还要求用户提供支持合同。要注册 HP Passport ID，请访问：

http://h20229.www2.hp.com/passport-registration.html

要查找有关访问级别的详细信息，请访问：

http://h20230.www2.hp.com/new_access_levels.jsp

支持列表

有关完整的支持和兼容性信息，请参见相关产品发布的支持列表。可在 HP 软件联机支持网站上查找所有支持列表和产品手册，地址为：

http://h20230.www2.hp.com/sc/support_matrices.jsp

您还可以从 HP 软件联机支持产品手册网站下载此版本的 《HP Server Automation Support and Compatibility Matrix》，地址为：

http://h20230.www2.hp.com/selfsolve/manuals

文档更新

要检查是否有最新的更新，或者验证是否正在使用最新版本的文档，请访问：

http://h20230.www2.hp.com/selfsolve/manuals

需要注册 HP Passport 才能登录此站点。要注册 HP Passport ID，请单击 “HP Passport”登录页面上的 “New users - please register”链接。

此外，如果订阅了相应的产品支持服务，则还会收到更新的版本或新版本。有关详细信息，请与您的 HP 销售代表联系。有关任何版本的列表，请参见 “文档变更说明”。

产品版本

HP Server Automation 有两种版本：

- HP Server Automation (SA) 是 Server Automation Enterprise 版。有关 Server Automation 的信息，请参见 《SA Release Notes》、《SA 用户指南：Server Automation》。
- HP Server Automation Virtual Appliance (SAVA) 是 Server Automation Standard 版。更多有关 SAVA 所包括内容的详细信息，请参见 《SAVA Release Notes》和 《SAVA 概览》指南。

文档变更说明

以下表格显示了与上一发布版本相比，该文档的变更情况。

日期	更改
2013 年 7 月 30 日	随 SA 10.0 发布的此文档的原始版本。

目录

1 应用程序配置快速入门 11

2 应用程序配置任务 15

创建应用程序配置 16
创建配置模板 17
使用可视化编辑器创建配置模板 19
导入并创建模板 19
从工作配置文件创建模板 20
可视化编辑器界面 22
使用可视化编辑器编辑配置模板 25
管理配置文件 28
编辑模板中的 CML 或 XML 32
导入和验证模板文件 33
查看配置模板源 34
将模板添加到应用程序配置或从应用程序配置删除模板 34
指定应用程序配置中的模板顺序 35
从脚本创建模板 35
通过运行数据操作脚本管理非文本配置 36
手动运行数据操作脚本 37
将应用程序配置附加到服务器或设备组 38
将应用程序配置附加到单个服务器 38
将应用程序配置附加到设备组 39
从服务器或设备组分离应用程序配置 39
从服务器分离应用程序配置 39
从设备组中分离应用程序配置 40
推送应用程序配置 41
停止推送配置作业 42
修改推送超时值 42
计划应用程序配置推送 43
将配置文件恢复到之前的状态 43
搜索和筛选作业结果 45
将配置模板与目标配置文件进行比较 - 预览 46
从托管服务器中选择服务器 46
从设备组中选择服务器 47
比较两个配置模板 47
对配置文件元素名称进行本地化 48
创建本地化文件 48
应用本地化模板 49

3 应用程序配置概念 51
有关应用程序配置对象 51
有关配置模板和脚本模板 52
CML 配置模板 53
XML 和 XML-DTD 配置模板 53
脚本模板 53
有关值集 54
值集级别和值集继承 54
阻止继承 56
有关值集编辑器 56
在值集编辑器中设置值 57
在值集编辑器中设置字段 57
值集编辑器中列的含义 58
从现有配置文件导入值集 59
应用程序、设施和客户级别的值集编辑器 60
在应用程序级别设置值 60
在设施级别设置值 61
在客户级别设置值 61
组级别的值集编辑器 62
在组级别设置值 63
在组实例级别设置值 64
服务器级别的值集编辑器 64
在服务器级别设置值 65
在服务器实例级别设置值 66
有关使用应用程序配置运行脚本 67
应用程序配置脚本的类型 67
有关将应用程序配置推送到服务器 69
应用程序配置符合性 69
单个服务器的应用程序配置符合性 70
多个服务器的应用程序配置符合性 71
在服务器中扫描应用程序配置符合性 74
审核应用程序配置 75
在软件策略中使用应用程序配置 76
4 管理 XML 配置文件 79
示例：Travel Manager 应用程序和 XML 配置文件 79
Travel Manager mysql.xml 文件内容 80
Travel Manager mysql.xml 基于 DTD 的 XML 文件内容 80
非 DTD 的 XML 配置模板 81
mysql.xml 的 非 DTD XML 配置模板 81
基于 DTD 的 XML 配置模板 82
mysql.xml 的 XML-DTD 配置模板 82
自定义 XML DTD 元素显示方式 83
显式和位置显示设置 83
添加显式自定义显示设置 84
XML 配置模板设置 87

5 XML 教程 1 - 创建非 DTD 的 XML 配置模板 89
非 DTD XML 文件 mysql.xml 示例 89
1. 创建 XML 配置模板 89
2. 添加 XML 设置 90
3. 创建要包含此模板的应用程序配置 91
4. 将应用程序配置附加到托管服务器。 91
5. 为服务器配置应用程序配置设置 93
6. 编辑值并推送配置 94

6 XML 教程 2 - 创建 XML-DTD 配置模板 95
Travel Manager 基于 DTD 的 XML 文件示例：mysql.xml 95
Travel Manager XML DTD 文件示例：mysql.dtd 95
1. 在文本编辑器中创建 XML-DTD 模板 96
2. 在值集编辑器中为元素描述添加客户设置 96
3. 导入 XML-DTD 配置文件 98
4. 创建应用程序配置对象 99
5. 将应用程序配置附加到托管服务器 99
6. 从配置文件导入值 100
7. 编辑值并推送配置 101

7 CML 教程 1 - 为简单 Web 应用程序服务器创建应用程序配置 103
1. 确定要管理的配置文件 103
2. 为此配置文件创建模板 103
创建模板文件并将其导入 SA 中 103
直接在 SA 中创建模板文件 104
3. 创建应用程序配置对象 105
4. 将模板文件添加到应用程序配置对象 105
5. 将应用程序配置对象附加到服务器 105
6. 设置默认值 106
设置应用程序级别默认值 106
设置 RHEL001 的服务器级别默认值 107
7. 将实际配置文件与配置模板进行比较 110
8. 将配置更改推送到服务器 111

8 CML 教程 2 - 创建 Web 服务器配置文件的模板 113
1. 分析本地配置文件和文档 114
2. 创建 CML 注释块 115
3. 创建 CML 设置指令 116
设置指令 116
4. 定义 [Options] 部分 — 打开块 118
5. 定义 [AllowExtensions] 部分 - 通过打开新块关闭块 123
6. 定义 [DenyExtensions] 部分 126
7. 定义 [AllowVerbs] 和 [DenyVerbs] 部分 127
8. 定义 [DenyHeaders] 部分 128
9. 定义 [DenyURLSequences] 部分 130
10. 定义 [RequestLimits] 部分 132

11. 将模板置于应用程序配置中 134
UrlScan.ini 文件示例 135
完整 url_scan_ini.tpl CML 模板 139

9 CML 基础知识 141
术语 141
CML 基本概念 142
必需的 CML 指令标记 142
namespace 标记 142
filename-key 标记 143
filename-default 标记 144
将标记组合在一行 144
用例 1 - 简单键 = 值配置文件 145
使用替换指令 145
最终 CML 模板 147
生成的值集 147
用例 2 - 配置文件中的重复值 148
使用循环指令标记 148
最终 CML 150
生成的值集 150
用例 3 - 配置文件中的复杂重复值 151
最终 CML 151
生成的值集 152
部分模板 152

10 CML 引用 153
有关配置模板 153
CML 概述 154
CML 标记的结构 154
必需的 CML 标记 155
/etc/hosts 的 CML 模板示例 157
CML 标记类型 158
注释标记：@# 和 @##. 158
替换标记：@. 159
指令标记：@! 160
块（或组）标记：@[@...@]@ 161
循环标记：@* 163
循环目标标记：@.. 164
条件标记：@? 165
DTD 标记：@~ 166
CML 类型特性 168
int 类型 168
十进制类型 168
GUID 类型 168
字符串类型 169
引用字符串类型 169
布尔值类型 169

持续时间类型 169
ipv6 类型 170
ipv4 类型 170
ip 类型 170
主机名类型 171
主机类型 171
网络类型 171
端口类型 171
用户类型 172
组类型 172
文件 – 特定于系统的类型 172
目录类型 172
电子邮件类型 173
CML 范围特性 173
!& , – 逻辑运算符 173
n< n<= <n <=n =n – 比较说明符 174
" – 字符串文本说明符 175
r" – 正则表达式说明符 175
CML 全局选项特性 176
@!filename-key 特性 176
@!filename-default 特性 176
@!full-template 和 @!partial-template 特性 176
@!timeout 特性 177
@!unix-newlines 和 @!windows-newlines 特性 177
CML 常规选项特性 177
@! unordered-lines 和 @!ordered-lines 特性 178
无序和有序元素特性 178
relaxed-whitespace 和 strict-whitespace 特性 179
required-whitespace 和 optional-whitespace 特性 179
missing-values-are-null 和 missing-values-are-error 特性 180
case-insensitive-keywords 和 case-sensitive-keywords 特性 180
required 和 optional 特性 180
skip-lines-without-values 和 show-lines-without-values 特性 181
skip-groups-without-values 和 show-groups-without-values 特性 181
sequence-append、 sequence-replace 和 sequence-prepend 特性 182
not-primary-field 和 primary-field 特性 182
namespace 特性 183
boolean-no-format 特性 183
boolean-yes-format 特性 184
delimiter 特性 184
line-comment 特性 185
sequence-delimiter 特性 186
field-delimiter 特性 187
line-continuation 特性 187
在 CML 中使用 DTD 标记 188
DTD 标记示例 188
序列聚合 189

序列替换 190
序列附加 191
序列前置 192
CML 语法 192

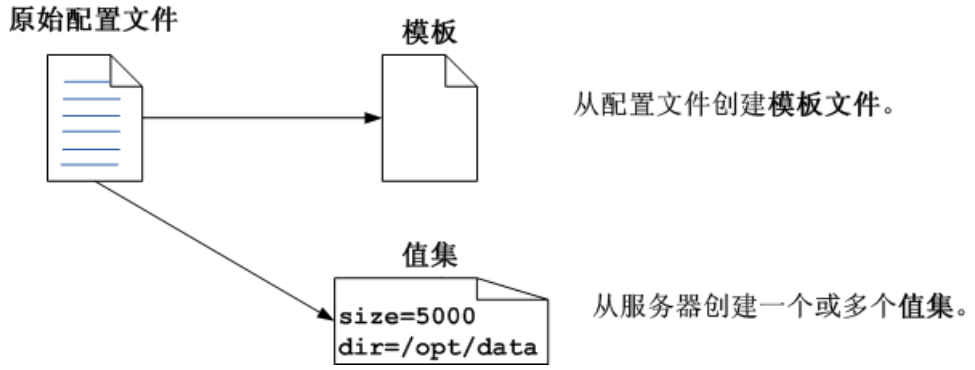
1 应用程序配置快速入门

本章将帮助您了解高级别的应用程序配置。它将说明对应用程序配置进行设置和管理所需执行的步骤。

- 有关如何执行这些任务的详细说明，请参见应用程序配置任务 （第 15 页）。
- 有关应用程序配置的背景信息，请参见应用程序配置概念 （第 51 页）。

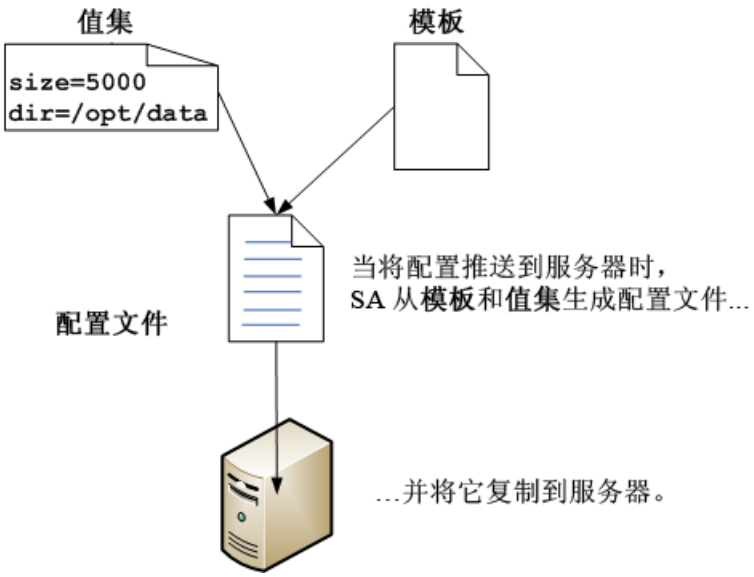
要创建和使用应用程序配置，必须首先创建**模板**文件和**值集**，如下图所示。模板是配置文件的模型。值集是数据值，这些值将与模板合并，用以创建要推送到服务器的配置文件实例。有关详细说明，请参见下方。

图 1　应用程序配置创建图



在创建模板和值集之后，将应用程序配置**附加**到一个或多个服务器并将配置**推送**到服务器，如下图所示。有关详细说明，请参见下方。

图 2　应用程序配置推送图



要创建和使用应用程序配置，请执行以下操作：

1 **确定要管理的配置文件**。选择要管理的应用程序或系统配置文件。例如，对于 Apache Web 服务器，您可以管理 http.conf、 password.conf、 obj.conf、 mimetypes 和 magnus.conf 文件。

2 **创建每个配置文件的模板文件**。对于要管理的每个配置文件，请基于此配置文件创建模板文件。模板是原始配置文件的模型，其中包含随服务器变化的值的占位符。此模板可指定配置文件中哪些值为固定值，哪些值为变量 （随服务器的不同而不同）。另请参见：

— 创建配置模板 （第 17 页）。

— 使用可视化编辑器创建配置模板 （第 19 页）。

— 对于 XML 配置文件，请创建 XML 或 XML-DTD 模板文件。有关详细信息，请参见管理 XML 配置文件 （第 79 页）。

— 对于其他配置文件，请使用配置建模语言 (CML) 创建模板文件。有关详细信息，请参见 CML 引用 （第 153 页）和 CML 教程 2 - 创建 Web 服务器配置文件的模板 （第 113 页）。

— 有关配置模板和脚本模板 （第 52 页）。

3 **创建每个脚本的模板文件。**在某些情况中，可能需要在更新配置文件之前或之后运行脚本。您必须使用 CML 从脚本创建模板文件。如果无需针对配置文件运行任何脚本，则跳过此步骤。另请参见：

— 从脚本创建模板 （第 35 页）。

— 有关配置模板和脚本模板 （第 52 页）。

— 有关使用应用程序配置运行脚本 （第 67 页）。

4 **将模板文件导入 SA 库**。在从配置文件和脚本创建所有模板之后，可将模板导入 SA 库中。或者也可以在 SA 客户端中直接创建这些模板。有关详细信息，请参见导入和验证模板文件（第 33 页）。

5 **创建应用程序配置对象。**应用程序配置对象仅仅是容纳一个或多个模板的容器。另请参见：

— 创建应用程序配置 （第 16 页）。

— 有关应用程序配置对象 （第 51 页）。

6 **将模板添加到应用程序配置对象**。创建并将模板导入 SA 库中之后，可将这些模板添加到应用程序配置对象。此外，还可指定文件的安装顺序以及脚本 （如果有）的执行时间。另请参见：

— 将模板添加到应用程序配置或从应用程序配置删除模板 （第 34 页）。

7 **将应用程序配置附加到服务器**。创建和配置应用程序配置之后，可将其附加到要使用它的服务器。这将创建应用程序配置的一个实例。服务器上可以具有此应用程序配置的多个实例。每个实例可以具有不同的用途。例如，一个实例可用于配置应用程序的临时版本，另一个实例可用于配置应用程序的生产版本。或者，如果服务器上运行应用程序的三个实例，则可以通过应用程序配置的三个实例来配置应用程序的每个实例。另请参见：

— 将应用程序配置附加到服务器或设备组 （第 38 页）。

— 在软件策略中使用应用程序配置 （第 76 页）。

8 **创建服务器的值集**。设置将放置于模板的值，以生成将推送到服务器的实际配置文件。可以在多个级别设置默认值，如果没有在较低级别覆盖这些值，则将继承这些默认值。另请参见：

— 从现有配置文件导入值集 （第 59 页）。

— 有关值集 （第 54 页）

— 值集级别和值集继承 （第 54 页）

9 **将实际配置文件与配置模板进行比较**。（可选）可轻松将配置模板与服务器上的实际配置文件进行比较，查看是否存在任何差异。这会显示服务器上配置文件的内容以及向服务器推送应用程序配置时将复制到服务器的值。另请参见：

— 将配置模板与目标配置文件进行比较 - 预览 （第 46 页）。

— 比较两个配置模板 （第 47 页）。

10 **推送配置更改**。要更新服务器上的配置文件，需将应用程序配置 “推送” 到这些服务器。在将更改推送到服务器之后，才会更改服务器上的实际配置文件。可将应用程序配置更改推送到单个服务器或服务器组。另请参见：

— 推送应用程序配置 （第 41 页）。

— 有关将应用程序配置推送到服务器 （第 69 页）。

11 **审核配置、监控符合性和修正**。要审核配置、监控符合性以及修正已更改的配置，请参见应用程序配置符合性 （第 69 页）和 《SA 用户指南：审核与符合性》。

2 应用程序配置任务

本节描述了执行应用程序配置任务的方式：

应用程序配置和模板入门：

— 创建应用程序配置 （第 16 页）

— 创建配置模板 （第 17 页）

— 使用可视化编辑器创建配置模板 （第 19 页）

编辑和管理应用程序配置模板：

— 使用可视化编辑器编辑配置模板 （第 25 页）

— 编辑模板中的 CML 或 XML （第 32 页）

— 导入和验证模板文件 （第 33 页）

— 查看配置模板源 （第 34 页）

— 将模板添加到应用程序配置或从应用程序配置删除模板 （第 34 页）

— 指定应用程序配置中的模板顺序 （第 35 页）

在应用程序配置中使用脚本：

— 从脚本创建模板 （第 35 页）

— 通过运行数据操作脚本管理非文本配置 （第 36 页）

— 手动运行数据操作脚本 （第 37 页）

附加和分离应用程序配置：

— 将应用程序配置附加到服务器或设备组 （第 38 页）

— 从服务器或设备组分离应用程序配置 （第 39 页）

推送和管理应用程序配置：

— 推送应用程序配置 （第 41 页）

— 计划应用程序配置推送 （第 43 页）

— 将配置文件恢复到之前的状态 （第 43 页）

— 搜索和筛选作业结果 （第 45 页）

— 将配置模板与目标配置文件进行比较 - 预览 （第 46 页）

— 比较两个配置模板 （第 47 页）

— 对配置文件元素名称进行本地化 （第 48 页）

创建应用程序配置

应用程序配置是可容纳配置模板文件和（可选）脚本（将应用程序配置推送到服务器时会执行的脚本）的容器。有关详细信息，请参见有关应用程序配置对象 （第 51 页）。

要创建应用程序配置，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。

2 查找并打开应用程序配置节点：

   a 打开 “配置” 节点。

   b 打开与应用程序配置相关的操作系统组。

   c 选择操作系统。

   请注意，应用程序配置可应用于多个操作系统。可在后面的步骤中进行指定。

3 从菜单，选择 “操作” > “新建” 打开一个新配置窗口，您可在其中指定配置属性和内容。

4 在 “属性” 视图中，指定配置的名称和描述。此外，还需指定以下内容：

   - **位置**：指定 SA 库中存储应用程序配置的位置。
   - **版本**：版本可以是用于跟踪应用程序配置更改的任何字符串。版本不会自动增加。
   - **操作系统**：指定应用该应用程序配置的操作系统。仅装有指定操作系统的服务器才能使用应用程序配置。只有至少有一个 OS 设置与此设置相同的模板才能包含在应用程序配置中。
   - **可用性**：使用此设置可跟踪已经过测试并准备使用的应用程序配置以及尚未测试或已弃用的应用程序配置。此设置不会更改应用程序配置可以实现的功能。
   - **本地化文件**：选择 “+” 按钮可添加模板，以便生成本地语言的配置文件。有关详细信息，请参见对配置文件元素名称进行本地化 （第 48 页）。

5 在 “配置值” 视图中，选择 “操作” > “添加” 菜单或 “+” 按钮将模板添加到应用程序配置中。

   - 使用 “-” 按钮可删除选定配置模板。
   - 使用向上和向下箭头可更改配置模板的安装顺序。
   - 选择 “模板预览” 视图可查看选定模板文件的内容。
   - 选择 “文件值” 视图可查看要置于选定模板文件以生成配置文件的值。
   - 选择 “文件预览” 视图可查看使用当前值集的选定配置文件。

6 选择 “文件” > “保存” 保存应用程序配置。

创建配置模板

配置模板与本地应用程序配置文件类似，但是它的变量部分可以通过使用配置建模语言 (CML) 和指令进行“模板化”，以便在配置文件和值集之间移动值。请参见有关配置模板和脚本模板（第 52 页）。

此外，还必须采用 CML 模板格式编写要使用此应用程序配置执行的任何脚本。有关详细信息，请参见从脚本创建模板（第 35 页）。

要创建配置模板，请执行以下操作：

1 在 SA 客户端导航窗格中，选择“库”，然后选择“按类型”选项卡。

2 查找并打开“应用程序配置”节点。打开“模板”节点。打开操作系统组并选择将使用模板文件的操作系统。请注意，模板可应用于多个操作系统。可在后面的步骤中进行指定。

3 选择“操作”>“新建”菜单。此时将显示“模板”屏幕，您可以在其中定义模板的属性。

4 选择“属性”视图并输入模板文件的名称和描述以及以下信息：

- **位置**：指定 SA 库中用于存储模板的位置。
- **版本**：版本可以是用于跟踪模板更改的任何字符串。版本不会自动增加。
- **类型**：指定是模板文件、脚本还是本地化文件：
  - 模板文件是配置文件的模型。
  - 脚本可在向服务器推送配置文件之前或之后执行。有关详细信息，请参见从脚本创建模板（第 35 页）。
  - 本地化文件用于针对不同区域设置自定义的配置文件。有关详细信息，请参见对配置文件元素名称进行本地化（第 48 页）。
- **分析器语法**：选择模板使用的语法类型：
  - 适用于除 XML 文件之外的所有文本配置文件以及所有脚本文件的 CML 语法。
  - 适用于采用 XML 编写的配置文件的 XML 语法。
  - 适用于同时采用 XML 和 DTD 编写的配置文件的 XML DTD 语法。
- **操作系统**：指定应用此模板的操作系统。仅装有指定操作系统的服务器才能使用此模板。只有至少有一个 OS 设置与此设置相同的应用程序配置才能包含此模板。
- **可用性**：使用此设置可跟踪已经过测试并准备使用的模板以及尚未测试或已弃用的模板。此设置不会更改模板可以实现的功能。您可以将此字段值用作搜索条件。
- **可审核**：要为模板文件启用审核时，可设置此字段。有关详细信息，请参见审核应用程序配置（第 75 页）。

5 选择 “内容” 视图。

6 在模板编辑器中直接输入 CML 或 XML 或 XML DTD 文本。有关编辑操作和语法突出显示的信息，请参见编辑模板中的 CML 或 XML （第 32 页）。有关 CML 和 XML 的详细信息，请参见CML 引用 （第 153 页）和管理 XML 配置文件 （第 79 页）。

7 选择 “验证” 可分析 CML 或 XML 语法并检查错误。

8 选择 “文件” > “保存” 保存模板。

9 将模板添加到应用程序配置对象。请参见将模板添加到应用程序配置或从应用程序配置删除模板 （第 34 页）。

使用可视化编辑器创建配置模板

可视化编辑器模式支持您创建简单的应用程序模板，而无需了解 CML 或理解整个配置文件。如果要参数化小部分配置文件以将更改推送到服务器，此工具将十分有用。

1 首先，可以从现有文件开始，使用可视化编辑器创建应用程序配置模板，例如：

- SA 托管服务器上的工作配置文件

  请参见从工作配置文件创建模板 （第 20 页）。

- 从本地 PC 导入的配置文件示例

  请参见导入并创建模板 （第 19 页）。

2 然后，可视化编辑器模式会提供一个用户界面，用于编辑简单的应用程序配置模板，而无需使用 CML。

- 只需标记在配置推送期间应替换的配置选项，就会显示此窗体视图。标记的选项将基于基础模板填充参数名称、显示名称和数据类型字段。此窗体视图允许您根据需要修改参数的详细信息。

  请参见使用可视化编辑器编辑配置模板 （第 25 页）。

3 在创建模板之后，可以选择立即生成应用程序配置实例。

- 通过托管服务器上的工作配置文件，您可以查看兼容模板的列表、打开模板和立即生成应用程序配置实例并将其附加到服务器。配置实例可在值集编辑器中打开，在此编辑器中，您可以修改值并将更改推送到服务器。
- 请参见管理配置文件 （第 28 页）。

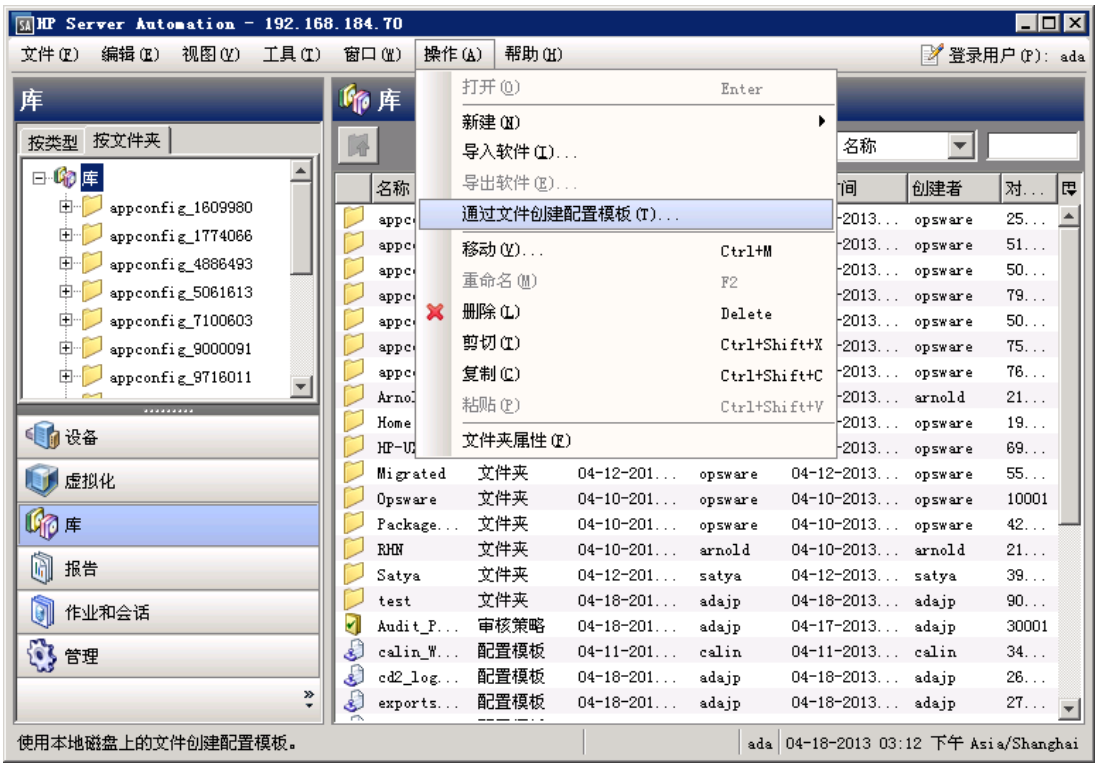
在 SA 中执行特定操作的能力由权限设置来管理。此外，为了通过服务器文件系统访问可视化编辑器，您需要具有 OGFS 权限来读取此文件系统。要获取其他权限，请与 SA 管理员联系。有关详细信息，请参见 《SA 管理指南》。

导入并创建模板

1 在 SA 客户端导航窗格中，访问要从 PC 导入配置文件的文件夹：

选择 “库” > “按文件夹” 或 > “按类型”，然后导航到所需的文件夹。

2 在“操作”菜单中选择“通过文件创建配置模板 ...”。



3 查找并选择要导入的配置文件，然后单击“打开”。

4 将在可视化编辑器模式中打开选定文件的模板。

请参见可视化编辑器界面（第 22 页）和使用可视化编辑器编辑配置模板（第 25 页）。

从工作配置文件创建模板

要使用可视化编辑器从工作配置文件创建配置模板，请执行以下操作：

1 打开服务器浏览器：

a 在 SA 客户端导航窗格中，访问托管服务器或设备组的列表：

— 选择“设备”>“服务器”>“所有托管服务器”，查看服务器列表。

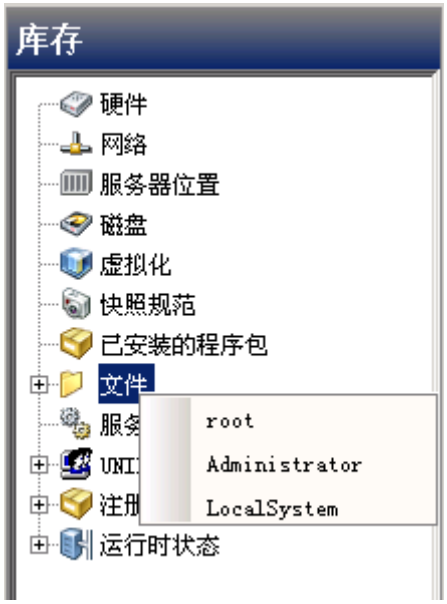
— 选择“设备”>“设备组”，查看设备组列表。

b 在内容窗格中，选择要打开的服务器或设备组。

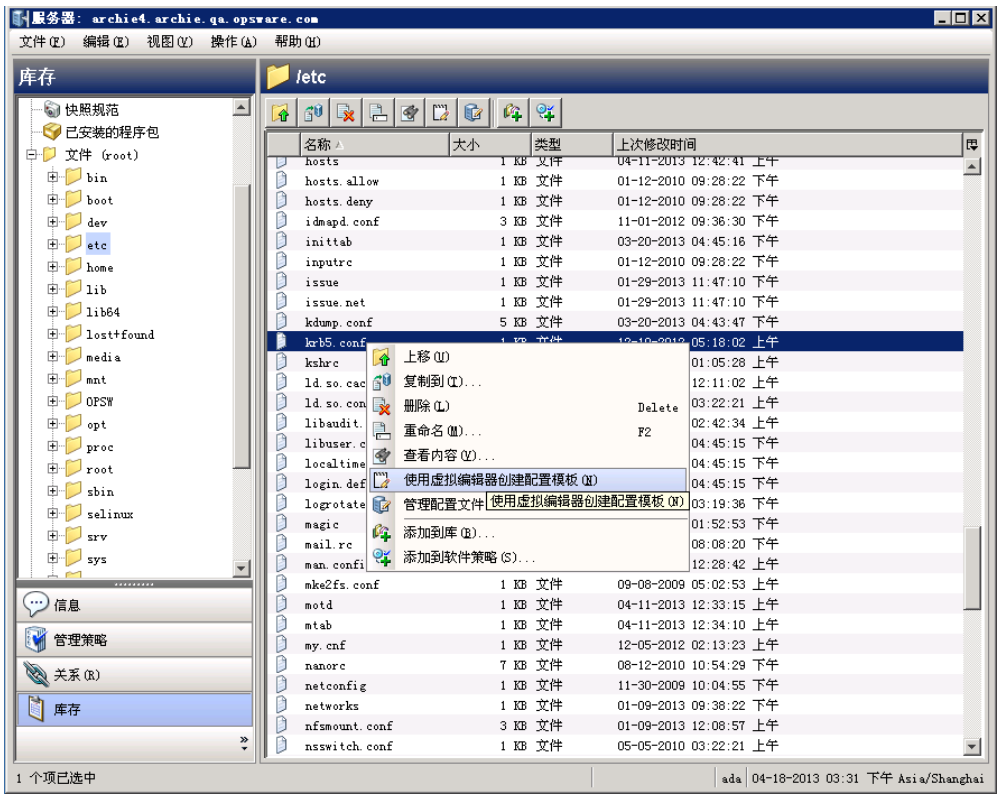
c 在“操作”菜单中，选择“打开”。

2 导航到“服务器浏览器”>“库存”>“文件”。

3 系统提示时，请选择服务器文件系统相应的根路径（通常，针对 Unix 选择 root，针对 Windows 选择 Administrator）。



4 右键单击配置文件并选择 “使用虚拟编辑器创建配置模板”。

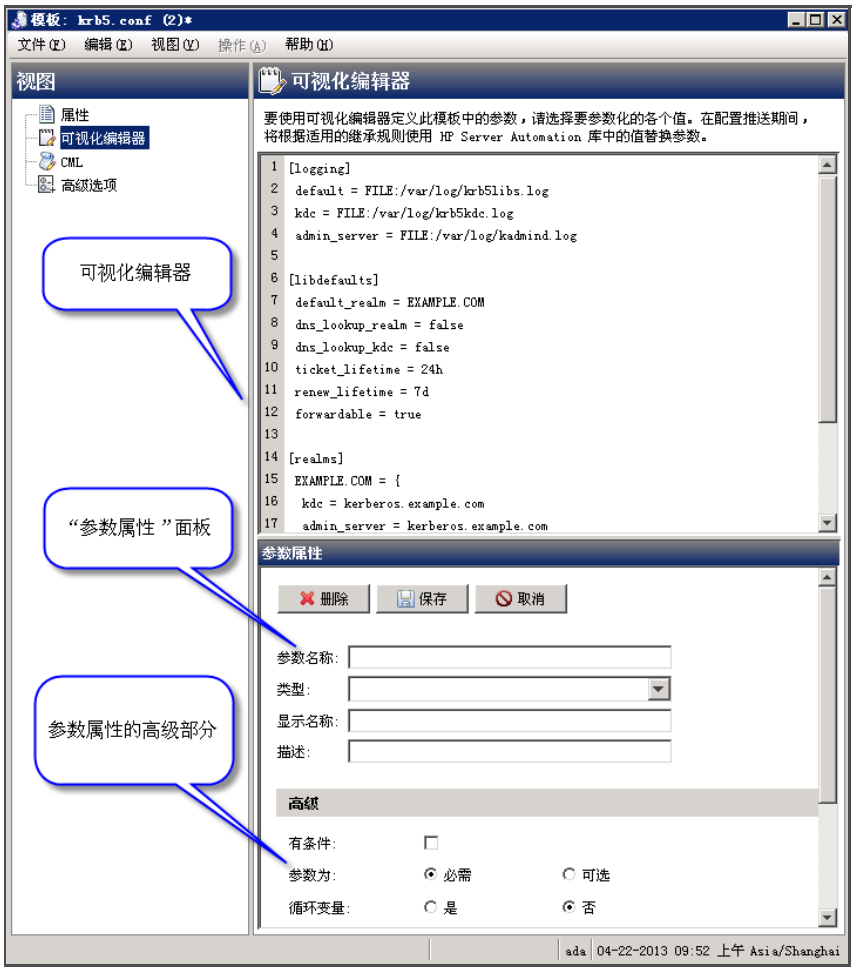


将在可视化编辑器模式中打开选定配置文件的模板。请参见可视化编辑器界面 （第 22 页）。

可视化编辑器界面

可视化编辑器界面在拆分面板中显示基础应用程序配置模板的详细信息，其中，下方面板的可编辑窗体视图中显示选定参数的详细属性。首次进入“可视化编辑器”窗口时，会显示“参数属性”面板。如果在上方面板中开始键入任何内容，则此面板将消失，但是如果在可视化编辑器的文本中选择一个参数，则此面板将重新显示。

图 3　可视化编辑器界面



“参数属性”面板

在“可视化编辑器”中选择一个值，然后单击“参数属性”面板的“参数名称”字段时，将基于基础配置文件中的值填充“参数属性”面板中的字段。

可编辑位于用户友好的窗体视图中的参数属性字段。

图 4　“参数属性”窗体

“参数属性”的“高级”部分：使用“高级”部分可指定有关参数的其他选项，例如参数是必需（默认）还是可选、是循环变量、有序列表、序列（指定类型和分隔符）还是范围。为数字类型（例如整数、十进制或端口）指定范围。（数字类型可以是序列的一部分。）

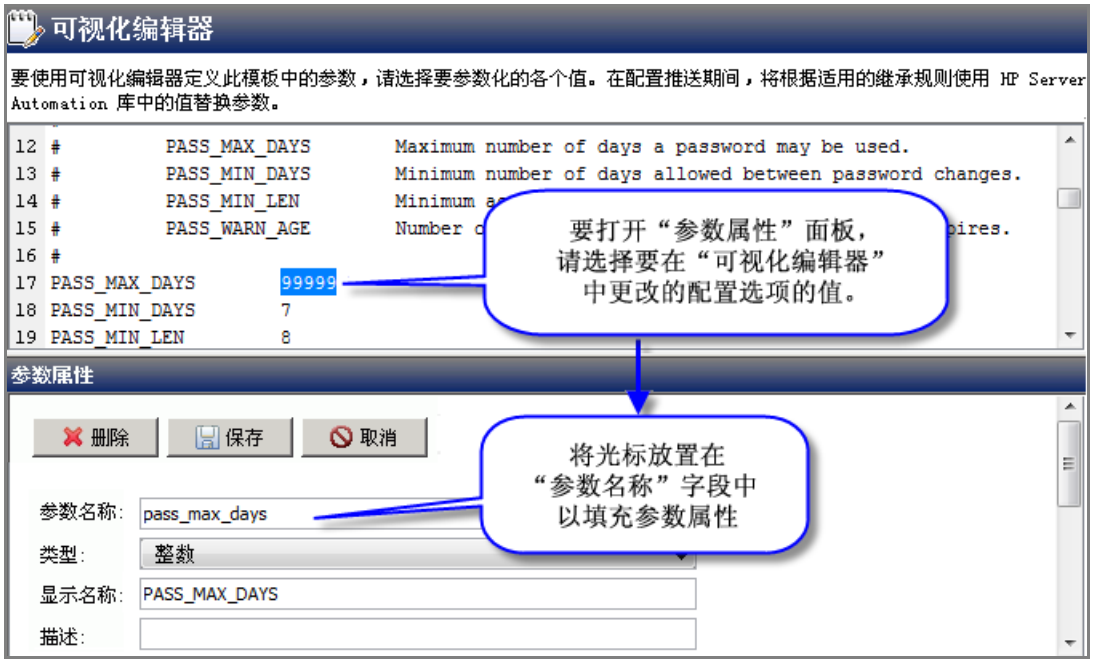
图 5　“参数属性”窗体的“高级”部分

“可视化编辑器”界面的运行方式：

在“可视化编辑器”中，选择在推送配置时要替换的配置选项的值，此时会显示此窗体视图。

标记的选项将基于基础配置文件中的数据填充“参数属性”面板中的“参数名称”、“显示名称”和“类型”（数据类型）字段。此窗体视图允许您根据需要修改参数的详细信息。

图 6　在可视化编辑器中编辑配置模板

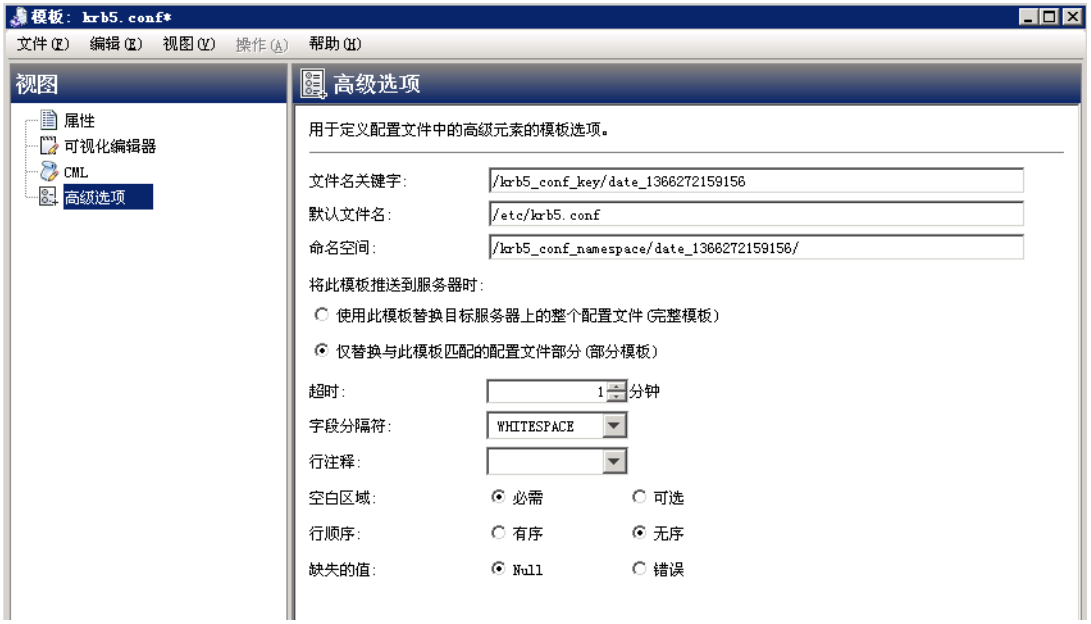


有关使用可视化编辑器编辑配置模板的说明，请参见使用可视化编辑器编辑配置模板（第 25 页）。

可视化编辑器中的 “高级选项”

“高级选项”视图允许用户在模板中修改全局 CML 选项。要访问此视图，请在模板的 “视图”窗格中选择 “高级选项”。

图 7　可视化编辑器中的 “高级选项”



默认情况下已填充这些值，无需用户提供输入；但是可以对其进行编辑：

- 定义三个必填字段：文件名关键字；默认文件名和命名空间

- 指定生成的模板是部分模板还是完整模板（默认情况下，部分模板意味着仅会替换指定的值。）
- 推送超时值和其他 CML 格式化和分析选项。

可视化编辑器限制：可视化编辑器仅支持部分 CML 内容。如果采用可视化编辑器不支持的某种方式直接修改 CML，则会显示一个警告消息，提示不兼容。您可以选择保留所做的更改并禁用可视化编辑器，也可以选择放弃更改并继续使用可视化编辑器。

使用可视化编辑器编辑配置模板

除对可视化编辑器流的基本说明以外，此部分还包含有关使用可视化编辑器编辑配置模板的提示信息。有关打开可视化编辑器的说明，请参见使用可视化编辑器创建配置模板 （第 19 页）。

要编辑配置模板，请执行以下操作：

1. 打开可视化编辑器。

   请参见导入并创建模板 （第 19 页）或从工作配置文件创建模板 （第 20 页）

2. 定义要推送的参数。

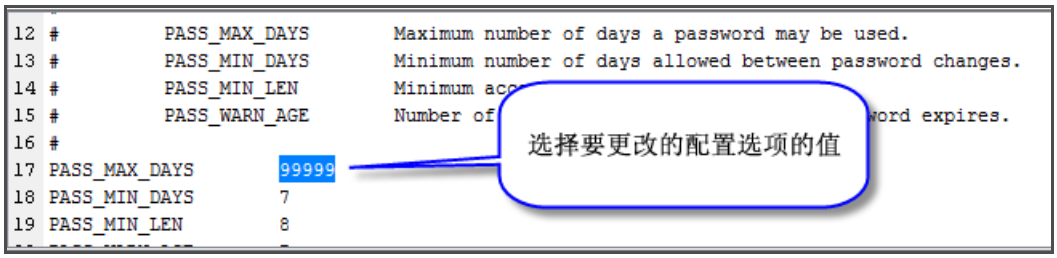
   有关示例，请参见创建简单参数 （第 25 页）或创建序列参数 （第 27 页）。

3. 查看 “属性” 选项卡，指定模板的名称和存储此模板的位置。
4. 单击 “保存” 以保存此模板。

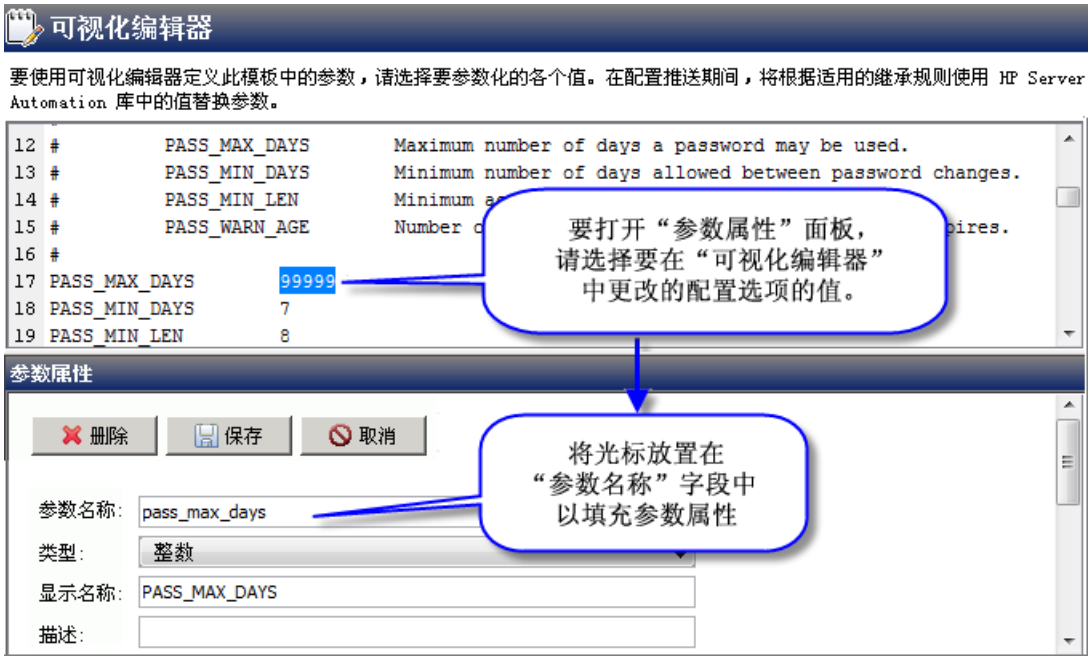
创建简单参数

要创建简单参数，请执行以下操作：

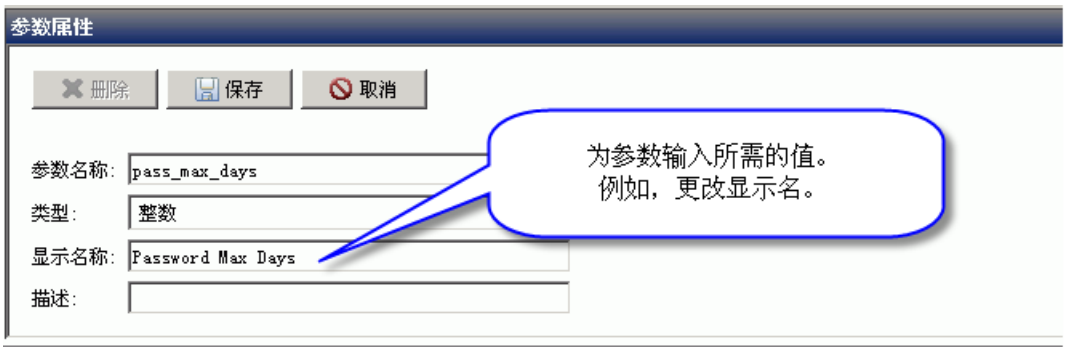
1. 选择在推送配置时要替换的配置选项，此时会显示此窗体视图。



2 在“参数属性”面板中，单击“参数名称”字段，标记的选项将基于基础模板填充“参数名称”、“显示名称”和“数据类型”字段。



3 为“参数名称”和“显示名称”输入所需的值，然后单击“保存”以保存此参数。



4 新建的参数将在可视化编辑器中突出显示为蓝色。

创建序列参数

要创建序列参数，请执行以下操作：

1 执行的步骤与创建简单参数的步骤相同。

（请参见创建简单参数 （第 25 页）。）

2 在 “高级” 部分的 “序列类型” 字段中指定一个值：“集合” 或 “列表”。

“列表”必须按特定序列保存元素，“集合”不能包含任何重复元素。

3 指定“序列分隔符”。这是用于分隔列表中的值的字符类型。

在“可视化编辑器”面板中，将创建序列参数，方法是使用相同的参数名称突出显示配置文件中与此参数匹配的所有元素。

管理配置文件

通过管理配置文件功能，您可以使用现有模板对工作配置进行快速建模。

为启动此操作，您必须具有 OGFS 权限来访问文件系统根目录。此外，还必须具有正确的权限来访问主文件夹，才能完成此操作。

要使用现有模板对工作配置进行建模，请执行以下操作：

1 打开服务器浏览器：

a 在 SA 客户端导航窗格中，访问托管服务器或设备组的列表：

— 选择“设备”>“服务器”>“所有托管服务器”，查看服务器列表。

— 选择“设备”>“设备组”，查看设备组列表。

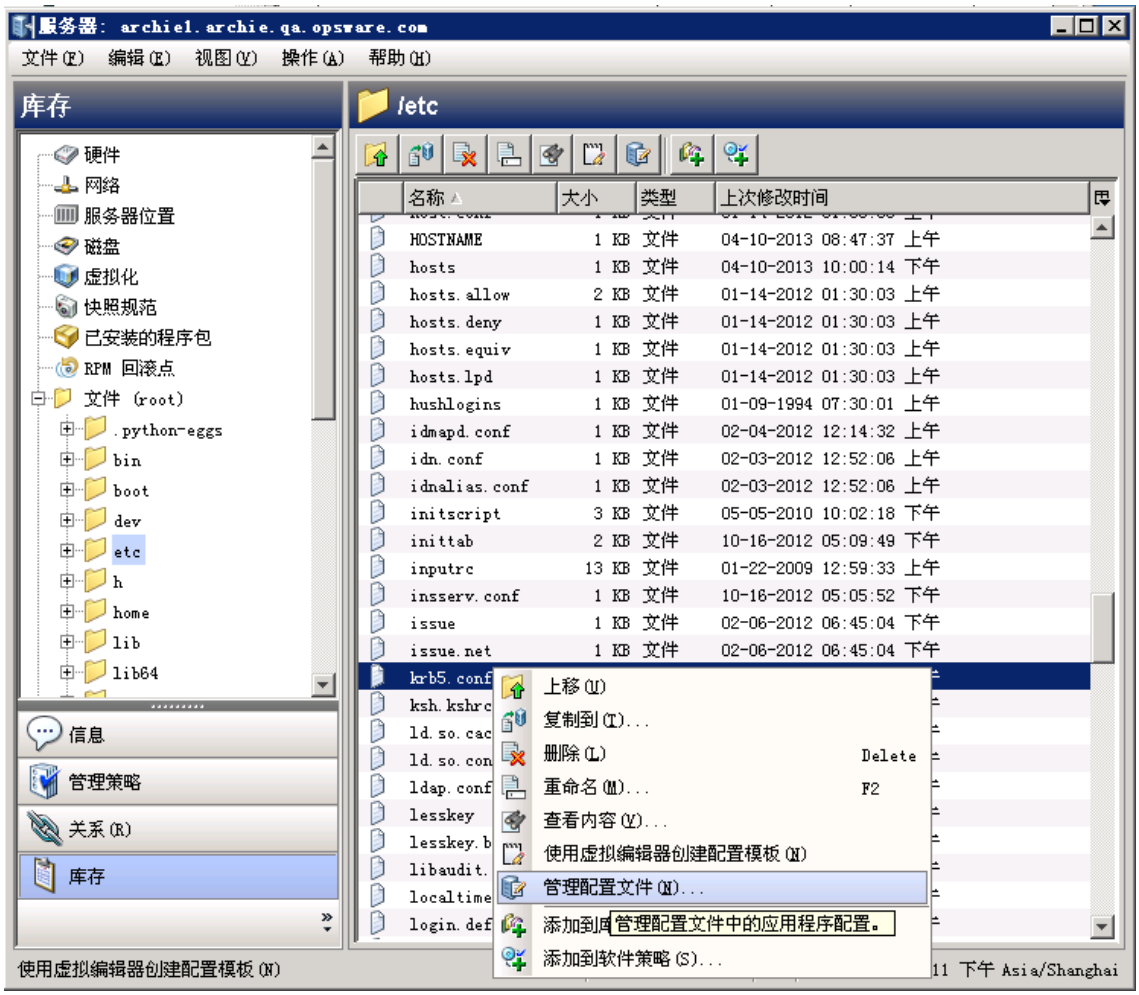
b 在内容窗格中，选择要打开的服务器或设备组。

c 在“操作”菜单中，选择“打开”。

2 导航到 “服务器浏览器” > “库存” > “文件”。（此步骤要求您具有 OGFS 权限。）

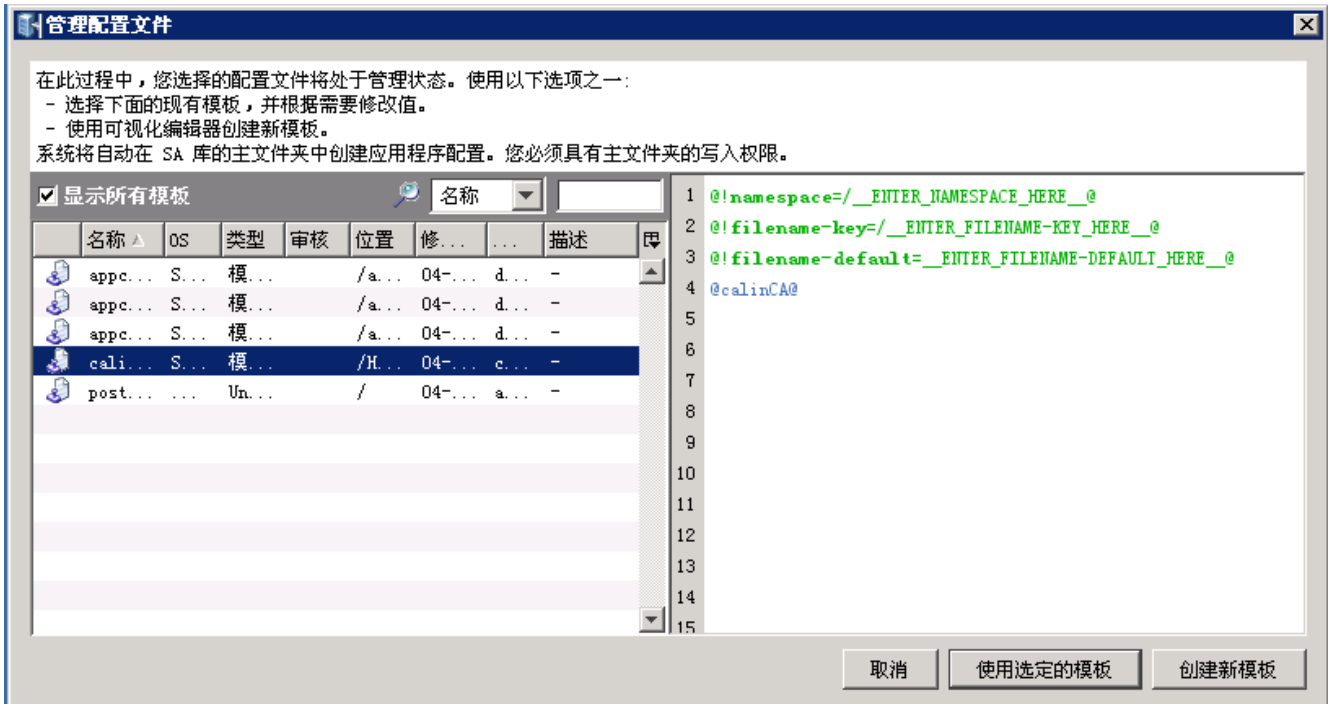
系统显示提示信息时，请选择服务器文件系统相应的根路径 （通常，针对 Unix 选择 root，针对 Windows 选择 Administrator。）

3 右键单击配置文件并选择 “管理配置文件”。



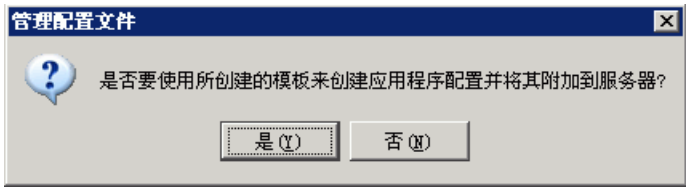
此时将列出 SA 中与配置文件名和路径匹配的现有模板作为参考。将仅提供与服务器平台匹配的模板。

4 选择模板，以在预览窗格中查看其内容。



其他选项：

- 选中“显示所有模板”复选框展开列表，以包括与服务器平台匹配的所有模板。
- 单击“取消”关闭此窗口并返回服务器文件系统目录。
- 单击“创建新模板”，将使用可视化编辑器基于选定的配置文件来创建新的模板。（请参见从工作配置文件创建模板（第 20 页）或使用可视化编辑器编辑配置模板（第 25 页）。）
  - 如果选择“创建新模板”，则在保存和关闭此模板时，系统会提示您是否要生成应用程序配置以及将实例附加到服务器。



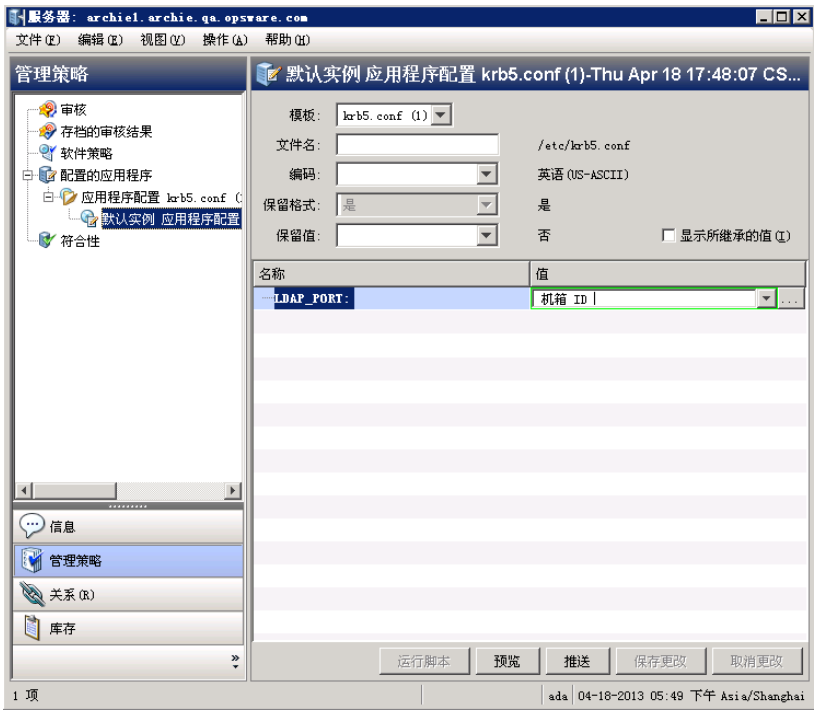
  - *可选*：如果仅要保存此模板以便稍后重新访问，请单击“否”。此模板将保存在“属性”选项卡指定的目录中。
  - 单击“是”将立即生成应用程序配置实例并将其附加到服务器。此配置实例将显示在值集编辑器中。

5 单击“使用选定的模板”，可使用选定的模板创建新的应用程序配置。（此步骤要求您有权访问主文件夹。）

新建的配置文件将位于 SA 库的主文件夹中，并且会将其中一个实例附加到托管服务器。

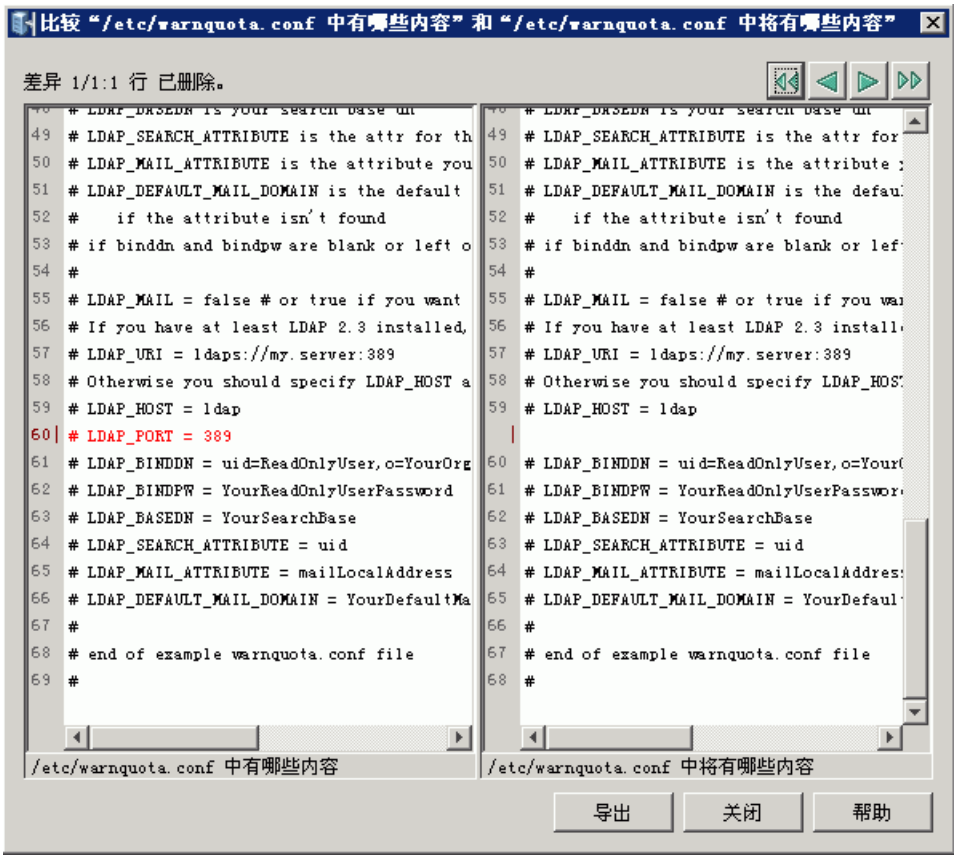
此时会显示一个确认消息，告知已创建配置文件并将其附加到服务器。

6 单击“确定”。此配置实例将显示在值集编辑器中。



7 输入此实例的值集数据。（在值集编辑器中设置值 （第 57 页）。）

（可选）单击 “预览” ，可并列预览与当前值比较的更改。单击 “关闭” 可返回处于 “编辑” 模式中的应用程序配置实例视图。



8 单击 “推送” 可将更改推送到服务器。

编辑模板中的 CML 或 XML

可以在 “内容” 视图中编辑模板的 CML 或 XML。模板编辑器可提供编辑操作和语法突出显示，如下所述。

要编辑模板的 CML 或 XML，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。

2 查找并打开 “应用程序配置” 节点。打开 “模板” 节点。打开操作系统组并选择模板文件所在的操作系统。请注意，模板可应用于多个操作系统。

3 选择一个模板。

4 选择 “操作” > “打开” 菜单或右键单击然后选择打开菜单或按 Enter 键。此时将显示包含选定模板的 “模板” 屏幕。

5 选择 “内容” 视图。此时将显示模板的内容。

6 在模板编辑器中直接输入 CML 或 XML 文本。

CML 语法使用不同颜色突出显示以便于阅读：

— 绿色：CML 指令以绿色显示。 CML 关键字以绿色粗体显示。

— 蓝色：将替换为值集中的实际值的变量以蓝色显示。

— 黑色：固定文本以黑色显示。

— 灰色：注释以灰色显示。

CML 编辑器还提供以下编辑操作：

— 右键单击 CML 文本，然后使用剪切、复制和粘贴操作。

— “操作”菜单提供查找、替换、撤消和重做操作。

— “验证”按钮和 “操作” > “验证”菜单检查模板的语法并报告所有错误。

有关 CML 和 XML 的详细信息，请参见 CML 引用 （第 153 页）和管理 XML 配置文件（第 79 页）。

导入和验证模板文件

可以使用文本编辑器编写 CML 模板或 XML 模板并将其导入 SA 库中，以便在应用程序配置中使用。此外，还可以在导入模板之前从 SA 验证此模板。请参见有关配置模板和脚本模板 （第 52 页）。

对于 Windows 服务器上使用 UTF-8 编码的配置文件，此配置文件的前三个字符可能包含字节顺序标记 (BOM)。如果将此文件导入应用程序配置模板，则在导入此文件之后， BOM 将显示在此模板中。如果希望应用程序配置模板中不包含此 BOM，则可在将此配置文件上载到模板之后删除它。

SA 客户端中不支持使用 UTF-16 编码。

要验证和导入模板文件，请执行以下操作：

1 在 SA 客户端中选择导航窗格，选择 “库”，然后选择 “按类型”选项卡。

2 查找并打开 “应用程序配置”节点。打开 “模板”节点。打开操作系统组并选择将使用模板文件的操作系统。请注意，模板可应用于多个操作系统。可在后面的步骤中进行指定。

3 选择 “操作” > “验证模板 ...”菜单。

4 查找并选择模板文件，选择相应编码，然后选择 “打开”。 SA 将检查模板的语法并报告结果。如果文件中存在错误，请更正它们并重新进行验证。

5 选择 “操作” > “导入模板 ...”菜单。

6 查找并选择模板文件，选择相应编码，然后选择 “打开”。请注意， SA 客户端中不支持使用 UTF-16 编码。 SA 导入模板并显示 “模板”屏幕。

7 请按照第 17 页创建配置模板下从步骤 4 开始的步骤执行操作。

查看配置模板源

您可以查看配置模板的内容并查看其 CML 或 XML 源。这可以帮助您在将应用程序配置推送到服务器之前了解模板中已设置哪些列表合并模式。有关应用程序配置序列合并模式的信息，请参见序列聚合 （第 189 页）。

要查看配置模板的源，请执行以下操作：

1. 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。
2. 查找并打开 “应用程序配置” 节点。打开 “模板” 节点。打开操作系统组并导航到模板文件所在的操作系统。请注意，模板可应用于多个操作系统。
3. 选择配置模板，然后选择 “操作” > “打开”。
4. 选择 “内容” 视图，可显示配置模板的 CML 或 XML 内容。

将模板添加到应用程序配置或从应用程序配置删除模板

应用程序配置可包含一个或多个模板。

要将模板添加到应用程序配置或从其中删除模板，请执行以下操作：

1. 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。
2. 查找并打开 “应用程序配置” 节点。打开 “配置” 节点。打开操作系统组并导航到应用程序配置所在的操作系统。请注意，应用程序配置可应用于多个操作系统。
3. 选择应用程序配置，然后选择 “操作” > “打开”。
4. 选择 “内容” 视图。
5. 要将模板添加到应用程序配置，请选择 “操作” > “添加” 或选择 “+” 按钮。选择所需模板并选择 “确定”。

   模板中必须至少有一个 OS 设置与应用程序配置的 OS 设置匹配。

   包含此模板的文件夹客户设置必须包括此应用程序配置对象的客户设置。否则，可用模板的列表中将不会包括此模板。有关文件夹设置的详细信息，请参见《SA 管理指南》中的 “文件夹权限”。
6. 要删除模板，请选择 “操作” > “删除” 或选择 “-” 按钮。
7. 选择 “文件” > “保存” 保存更改。

指定应用程序配置中的模板顺序

应用程序配置可以包含一个或多个配置模板及相关脚本。您可以指定应用程序配置中的模板顺序。将以应用程序配置中模板的显示顺序将模板推送到托管服务器。例如，你可能需要将更改在其他更改之前应用到某个配置文件。

应用程序配置中脚本的执行顺序由脚本类型确定：数据操作、安装前、安装后或出错后类型。应用程序配置中脚本的顺序是不相关的。有关详细信息，请参见应用程序配置脚本的类型 （第 67 页）。

要指定应用程序配置中的模板顺序，请执行以下操作：

1. 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。
2. 查找 “应用程序配置” 节点并打开此节点。打开 “配置” 节点。打开操作系统组并导航到应用程序配置所在的操作系统。请注意，应用程序配置可应用于多个操作系统。
3. 选择应用程序配置，然后选择 “操作” > “打开”。
4. 选择 “内容” 视图。此时将显示应用程序配置中的所有配置模板和脚本 （如果有）。请注意，已按顺序对模板和脚本进行编号。
5. 要对模板或脚本重新排序，请选中该模板或脚本然后选择 “操作” > “上移” 或 “操作” > “下移” 或选择向上箭头和向下箭头图标。
6. 选择 “文件” > “保存” 保存更改。

从脚本创建模板

要使应用程序配置对象中包含脚本，必须将脚本复制到 CML 模板，然后将模板导入应用程序配置对象中。有关 CML 的详细信息，请参见 CML 引用 （第 153 页）。另请参见有关使用应用程序配置运行脚本 （第 67 页）。

以下示例显示了一个可执行 touch 命令和 echo 命令的 Unix Shell 脚本：

```
#/bin/sh
touch abc.txt
echo abc >>abc.txt
```

以下示例显示了转换成 CML 的此 Unix Shell 脚本：

```
@#########################################################
# /tmp/simple-script/TouchABC.sh                         #
# Version 0.1                                            #
# Author:<name>                                         #
#########################################################@
@!namespace=/simple-script-namespace/@
```

```
@!filename-key="/TouchABC"@
@!filename-default=/tmp/simple-script/TouchABC.sh@
#/bin/sh
touch abc.txt
echo abc >>abc.txt
```

要从此脚本创建模板，请执行以下操作：

1. 如创建配置模板 （第 17 页）中所述创建模板。将脚本的 CML 版本用作此模板的内容。
2. 将 “类型” 字段设置为相应的脚本类型。对于以上示例，您应将 “类型” 设置为 “Unix .SH 脚本”。第 68 页上的表 4 中列出了其他支持的脚本类型。
3. 由于必须使用 CML 语法编写所有脚本，因此请将 “分析器语法” 字段设置为 “CML 语法”。
4. 如创建配置模板 （第 17 页）中所述设置剩余的字段。
5. 选择 “文件” > “保存”。
6. 选择 “文件” > “关闭”。
7. 如将模板添加到应用程序配置或从应用程序配置删除模板（第 34 页）中所述将模板添加到应用程序配置对象。
8. 在将模板添加到应用程序配置对象之后，打开此应用程序配置对象。
9. 选择 “配置值” 视图。
10. 选择此脚本模板并右键单击以显示菜单。
11. 选择脚本类型，此类型将指定运行脚本的时间：数据操作、安装前、安装后或出错后。第 67 页上的表 3 中将描述这些类型。
12. 选择 “文件” > “保存”。
13. 选择 “文件” > “关闭”。

通过运行数据操作脚本管理非文本配置

通过 SA，您可以管理非文本配置，方法是创建数据操作脚本，该脚本可提取非文本配置数据并将其放置于文本文件中。然后， SA 会对此文本文件进行操作 （操作方式与任何其他文本配置文件相同）并将其放回原始的格式中。有关脚本类型的描述，请参见应用程序配置脚本的类型 （第 67 页）。

以下为非文本配置数据的某些示例：

- SQL 数据库，数据操作脚本可对其运行某些 SQL 查询并将数据置于文本文件中。
- IIS 服务器的配置，数据操作脚本可将其元数据库信息读取到文本文件中。
- 二进制文件，数据操作脚本可从此二进制文件提取值并将值置于文本文件中。

手动运行数据操作脚本

通过“运行脚本”按钮，您可以执行与某个应用程序配置关联的数据操作脚本并在托管服务器上准备目标配置文件，以便将其值导入值集中。

要手动运行数据操作脚本，请执行以下操作：

1. 在 SA 客户端导航窗格中，选择“设备”选项卡。
2. 选择“所有托管服务器”或“设备组”。导航到所需服务器或设备组。
3. 选择服务器或设备组然后选择“操作”**>**“打开”菜单。
4. 如果选择服务器，则执行以下步骤。如果选择设备组，则跳到下一个步骤。
   - a 选择“管理策略”选项卡。
   - b 在导航窗格中打开“配置的应用程序”节点。此时将显示附加到服务器的所有应用程序配置。
   - c 打开要推送的“应用程序配置”节点。此时将显示此应用程序配置的值集。
   - d 选择要推送的值集。必须在服务器实例级别选择值集。
5. 如果选择设备组，则执行以下步骤。
   - a 在导航窗格中打开“配置的应用程序”节点。此时将显示附加到服务器的所有应用程序配置以及所有服务器的列表。
   - b 打开“服务器”节点。此时将显示设备组中的所有服务器。
   - c 打开要运行数据操作脚本的特定服务器节点。此时将显示附加到该服务器的应用程序配置。
   - d 打开要推送的“应用程序配置”节点。此时将显示此应用程序配置的值集。
   - e 选择要推送的值集。必须在服务器实例级别选择值集。
6. 选择服务器实例级别的值集之后，选择“运行脚本”按钮。此时将显示确认对话框。
7. 选择“是”以在服务器上运行数据操作脚本。
8. 在数据操作脚本运行、提取配置数据并将数据置于一个文本文件之后，您可以使用与任何其他配置文件相同的管理方式管理此文本文件，然后将此文本文件中的数据放回其原始的格式中。

将应用程序配置附加到服务器或设备组

在创建应用程序配置、添加所有必要的配置模板和脚本以及编辑其默认值之后，必须将此配置附加到服务器或公用设备组。在将应用程序配置附加到服务器或服务器组后，必须如推送应用程序配置（第 41 页）中所述将此应用程序配置推送到服务器。

包含应用程序配置的文件夹的客户设置必须包括您要向其推送此应用程序配置的托管服务器的客户设置。有关文件夹设置的详细信息，请参见 《SA 管理指南》中的 “文件夹权限”。有关服务器和客户的详细信息，请参见 《SA 用户指南：Server Automation》中的 “客户帐户”。

您只能将应用程序配置附加到单个服务器或公用设备组。不能将其附加到专用设备组。

将应用程序配置附加到单个服务器

快捷方式：“设备”选项卡 > 选择“服务器” > “所有托管服务器” > “服务器” > 右键单击 > “附加” > “应用程序配置” > 选择一个配置文件 > 单击 “确定”。

要将应用程序配置附加到单个服务器，请执行以下操作：

1. 在 SA 客户端导航窗格中，选择 “设备”选项卡。
2. 选择 “服务器” > “所有托管服务器”。
3. 在内容窗格中选择一个服务器。
4. 选择 “操作” > “打开”菜单。
5. 选择 “管理策略”选项卡，然后选择 “配置的应用程序”。
6. 选择 “安装的配置”选项卡。
7. 选择 “操作” > “添加配置 ...”菜单。
8. 在 “选择应用程序配置”屏幕中，选择要附加到托管服务器的应用程序配置。

   可使用搜索工具 按特定条件搜索，例如名称、上次修改日期等。

   请注意，包含应用程序配置的文件夹的客户设置必须包括您要向其推送此应用程序配置的托管服务器的客户设置。有关文件夹设置的详细信息，请参见《SA 管理指南》中的“文件夹权限”。有关服务器和客户的详细信息，请参见 《SA 用户指南：Server Automation》中的 “客户帐户”。
9. 选择 “确定”以将应用程序配置推送到服务器。
10. 选择 “保存更改”按钮。

11 现在可以为该服务器设置应用程序配置值。有关设置应用程序配置值的详细信息，请参见有关值集 （第 54 页）。

将应用程序配置附加到设备组

要将应用程序配置附加到设备组，请执行以下操作：

您只能将应用程序配置附加到公用设备组。不能将应用程序配置附加到专用设备组。

1 在 SA 客户端导航窗格中，选择 “设备” 选项卡。
2 打开 “设备组” 节点并导航到公用设备组。
3 在内容窗格中选择所需的设备组。
4 选择 “操作” > “打开” 菜单。
5 在设备组屏幕中，选择 “配置的应用程序” 视图。
6 选择 “操作” > “添加配置” 菜单。
7 在 “选择应用程序配置” 对话框中，选择应用程序配置。

   使用搜索工具 按特定条件搜索，例如名称、上次修改日期等。
8 输入 “实例名称” 。这是组实例级别的值集名称。有关详细信息，请参见在组实例级别设置值 （第 64 页）。
9 选择“确定”以将应用程序配置附加到设备组。指定的配置文件可被推送到此组中的所有服务器。
10 设置要推送到服务器的值。有关设置值的详细信息，请参见有关值集 （第 54 页）。

从服务器或设备组分离应用程序配置

要从服务器中分离应用程序配置，必须打开此服务器并删除此应用程序配置的服务器实例级别的所有实例。要从设备组中分离应用程序配置，必须打开此设备组并删除组实例级别的所有实例。以下部分将提供详细信息。

从服务器分离应用程序配置

要从服务器中分离应用程序配置，必须删除服务器实例级别的所有实例，如下所述。有关详细信息，请参见服务器级别的值集编辑器 （第 64 页）。

要从服务器中分离应用程序配置，请执行以下操作：

1 在 SA 客户端中打开服务器。必须将应用程序配置附加到服务器。请参见将应用程序配置附加到服务器或设备组 （第 38 页）。

2 选择左侧的 “管理策略” 选项卡。

3 在左侧的 “管理策略” 窗格中打开 “配置的应用程序” 节点。

4 在 “配置的应用程序” 节点下打开所需的应用程序配置节点。此时将显示此应用程序配置在服务器实例级别的所有实例。

5 在此 “应用程序配置” 节点下，选择其中一个实例。

6 选择 “操作” > “删除配置” 菜单或右键单击然后选择 “删除配置” 菜单。此操作将删除所选的实例。

7 请针对每个实例重复执行步骤 5 和步骤 6。删除最后一个实例后，将从此服务器中分离此应用程序配置。

8 单击 “保存更改”。

从设备组中分离应用程序配置

要从设备组中分离应用程序配置，必须删除此应用程序配置在组实例级别的所有实例，如下所述。有关详细信息，请参见在组级别设置值 （第 63 页）。

要从设备组中分离应用程序配置，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “设备” 选项卡。

2 打开 “设备组” 节点并导航到公用设备组。

3 在内容窗格中选择所需的设备组。

4 选择 “操作” > “打开” 菜单。

5 在设备组屏幕中，选择 “配置的应用程序” 视图并打开 “配置的应用程序” 节点。此时将显示附加到此设备组的应用程序配置。

6 在 “配置的应用程序” 节点下打开所需的应用程序配置节点。此时将显示组实例级别的值集。

7 在所需 “应用程序配置” 节点下，选择组实例级别的其中一个应用程序配置实例。

8 选择 “操作” > “删除配置” 菜单或右键单击然后选择 “删除配置” 菜单。此操作将删除所选的实例。

9 针对服务器实例级别的每个实例重复执行以上步骤 7 和步骤 8。删除最后一个实例后，将从此服务器中分离此应用程序配置。

10 单击 “保存更改”。

推送应用程序配置

无论何时更改值集中的值并要将这些更改与目标服务器上的配置文件合并，都必须将应用程序配置推送到服务器。有关详细信息，请参见有关将应用程序配置推送到服务器 （第 69 页）。

要将应用程序配置更改推送到服务器或服务器组，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “设备” 选项卡。

2 选择 “所有托管服务器” 或 “设备组”。导航到所需服务器或设备组。

3 选择服务器或设备组然后选择 “操作” > “打开” 菜单。

- 如果已选择服务器，则请选择 “管理策略” 选项卡。
- 如果选择设备组，则跳到下一个步骤。

4 在导航窗格中打开 “配置的应用程序” 节点并选择要推送的值集。必须选择服务器实例级别的值集，或是组实例级别的值集。

通过选择 “预览” 按钮，可以选择预览将在单个服务器上进行的更改。“比较” 屏幕将显示任何差异。在完成之后，请选择 “关闭”。

5 在准备好将更改应用到服务器之后，选择 “推送”。

6 在 “推送配置” 屏幕中，验证应用程序配置和要推送的值集。

要接受 “计划”、“通知” 和 “作业状态” 的其他默认值，请单击 “启动作业”。否则，请单击 “下一步” 继续查看向导选项。

7 在 “计划” 窗格中，指定要推送应用程序配置的时间。可以使用 “计划” 窗格计划在未来运行作业，或定期运行作业，例如每周或每月。

要接受 “通知” 和 “作业状态” 的其他默认值，请单击 “启动作业”。否则，请单击 “下一步” 继续查看向导选项。

8 在 “通知” 窗格中，可以选择指定一个或多个电子邮件地址和一个工单标识符。对于每个收件人，请选择发送电子邮件通知的时间。

- 成功时：如果作业成功，则向收件人发送电子邮件。
- 失败时：如果作业失败，则向收件人发送电子邮件。
- 终止时：如果作业终止，则向收件人发送电子邮件。
  - 在通过 “结束作业” 操作停止正在运行的作业时，会发生终止。
  - 此通知将 *不会* 应用于在运行之前取消的作业。

要接受 “作业状态” 的其他默认值，请单击 “启动作业”。否则，请单击 “下一步” 继续查看向导选项。

9 单击 “启动作业”。

在启动作业之后，可通过在主 SA 客户端屏幕上依次选择 “作业和会话” 选项卡和 “作业日志”，查看作业的状态。

您还可以执行以下任意可选操作：

- 单击 “导出” 将作业状态结果导出到文本文件。
- 单击 “结束作业” 以停止作业。请参见停止推送配置作业 （第 42 页）。
- 单击 “关闭” 以关闭窗口。要在之后查看作业状态，请在 SA 客户端导航窗格中单击 “作业状态”，然后双击要查看其详细信息的作业。

停止推送配置作业

您可以终止正在运行的推送配置作业。例如，可能需要停止正在生成错误结果或将超出分配的维护窗口运行的作业。

要保持作业的完整性，不能取消推送配置作业流的某些步骤。在停止作业时，“作业状态” 窗口将指示哪些步骤已完成，哪些步骤已跳过。

要停止活动的应用程序配置推送作业，请执行以下操作：

1. 在 “作业状态” 窗口中，单击 “结束作业”。（仅当作业正在进行时，才会显示此按钮。）
2. 此时将显示 “结束作业” 警告对话框，告知您作业终止的作用方式：
   - 此作业将不会启动任何其他服务器上的工作
   - 如果服务器上已启动工作，则作业将仅跳过可安全取消的步骤
   - “作业状态” 将指示已完成或跳过的步骤
   - 如果成功终止作业，则最终的作业状态将为 “已终止”
3. 单击 “确定”，确认要终止此作业。此时 “作业状态” 窗口将显示终止的进度。

   作业状态将为 “已终止”。服务器状态将为 “已取消”。任务状态将为 “成功” 或 “已跳过”。
4. 终止完成后，还可以在 SA 客户端 “作业日志” 中查看此作业。

   在 SA 客户端导航窗格中，单击 “作业和会话”。此时将显示 “作业日志” 视图，其中列出状态为 “已终止” 的作业。

修改推送超时值

默认情况下，推送应用程序配置时，默认超时值为十分钟加上应用程序配置中每个模板一分钟。该应用程序配置中的每个模板将其超时附加到此应用程序配置的基本超时。

例如，如果应用程序配置包含三个模板，则整个应用程序配置的默认超时值为 13 分钟。如果已推送模板并且整个推送时间超过 13 分钟，则推送将超时并且将取消此操作，包含所做的任何更改。

要延长模板的超时值，可以针对应用程序配置中的单个模板使用 CML 超时标记。CML 超时标记语法如下所示：

```
@!timeout=1@
```

有效值介于 0 到 999 之间，以分钟为单位。

如果在推送过程中应用程序配置发生超时，则将退出对推送目标文件的所有更改并取消此操作。

有关 CML 超时标记的详细信息，请参见 CML 引用 （第 153 页）。

计划应用程序配置推送

您可以计划使应用程序配置推送立即运行、将来运行或是定期运行，例如每天、每周或每月。要计划应用程序配置推送，请按照推送应用程序配置 （第 41 页）下列出的步骤执行操作，但是在执行到 “计划” 步骤时，请输入希望发生推送的频率和时间。

在计划作业之后，可通过在主 SA 客户端屏幕上依次选择 “作业和会话” 选项卡和 “定期计划”，查看作业的状态。

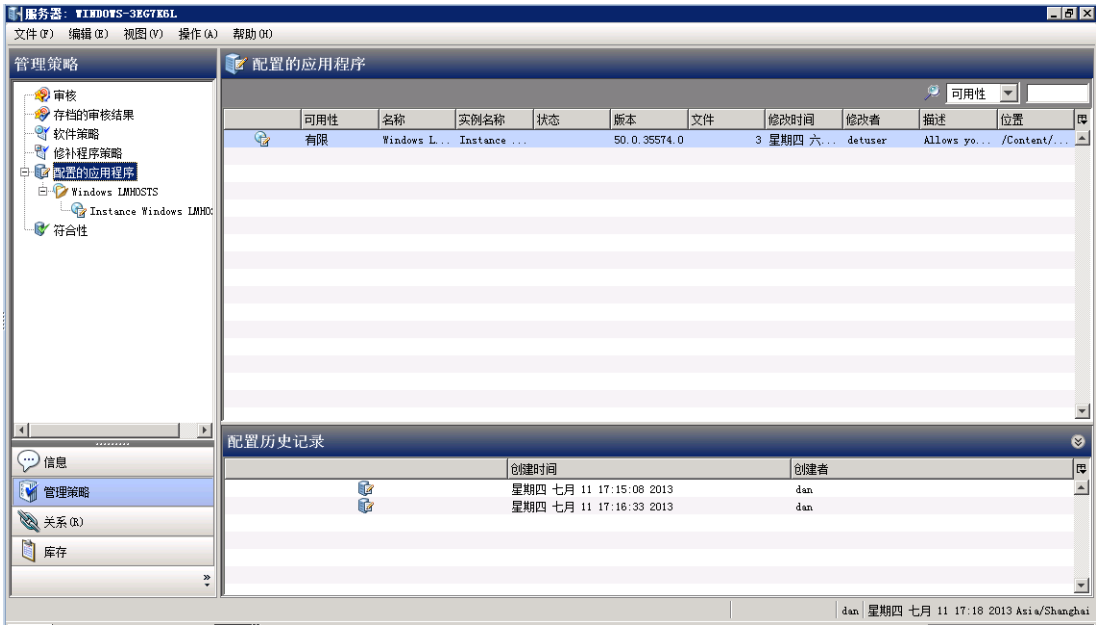
将配置文件恢复到之前的状态

每次将应用程序配置推送到服务器时，都将在配置推送历史记录列表中保存配置文件。在任何时候，您都可以将应用程序配置恢复到历史记录列表中的之前状态。

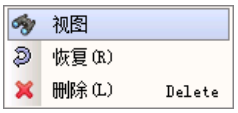
要将应用程序配置恢复到之前状态，请执行以下操作：

1. 在 SA 客户端导航窗格中，选择 “设备” 选项卡。
2. 选择 “所有托管服务器” 并查找所需的服务器。
3. 选择服务器然后选择 “操作” > “打开” 菜单。
4. 选择 “管理策略” 选项卡。
5. 在导航窗格中选择“配置的应用程序”节点。此时将显示在此服务器上安装的所有应用程序配置。

6 选择一个配置以在详细信息窗格中查看其配置历史记录。这将显示服务器上运行的所有推送作业。

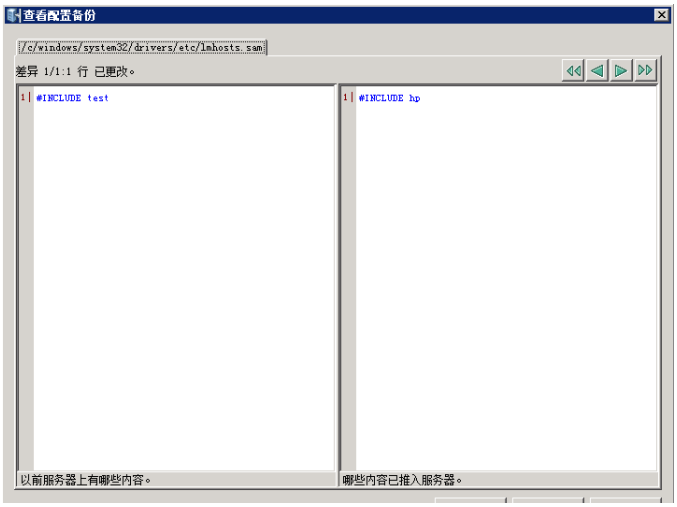


7 右键单击其中一个历史实例并选择 “查看” 查看快照。



“查看配置备份” 窗口将显示为拆分窗格以供比较。

- 左侧窗格显示在快照之前所推送文件的内容，它本质上是现有文件（如果有的话）的备份。
- 右侧窗格显示选定快照中已推送文件的内容。



8 选定历史记录中要恢复的行，然后单击 “恢复”。

此时将显示 “恢复配置” 向导。

9 验证服务器和要恢复的历史实例，然后单击 “下一步”。

10 选择恢复类型：

— *上一个选定的推送（撤消）*可将配置文件恢复到紧邻选定历史记录实例之前的值。即，将撤消选定的实例。

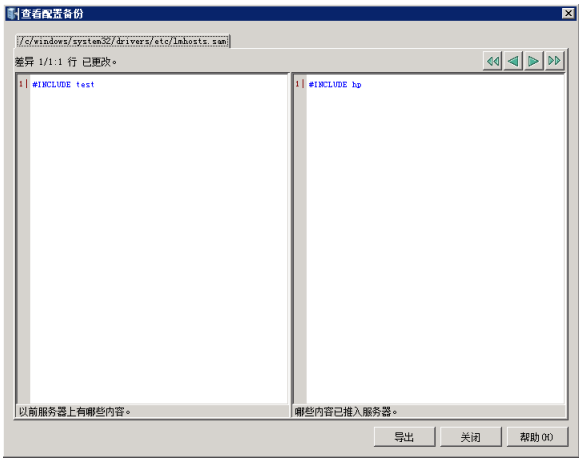
— *下一个选定的推送（重做）*可将配置文件恢复为选定历史实例的值。即，将重新推送选定的实例。

11 单击 “下一步”。

12 在 “预览” 步骤中，单击 “预览” 可比较对服务器上当前实例的更改。

“查看配置备份” 窗口将显示为拆分窗格以供比较。

- 左侧显示在给定位置服务器上的当前内容。
- 基于作业中选定的 “恢复类型”，右侧包含将要恢复到该位置的内容。



该预览提供了在执行恢复之前查看结果的选项：

- 要取消，请关闭窗口并从作业窗口中选择 “取消”。
- 要继续执行，请关闭窗口然后继续执行下一步。

13 单击 “启动作业”。此操作会将服务器上的配置恢复到选定的历史记录实例。

14 在启动作业之后，可通过在主 SA 客户端屏幕上依次选择 “作业和会话” 选项卡和 “作业日志”，查看作业的状态。

搜索和筛选作业结果

您可以搜索和筛选推送或恢复作业的结果。在大量服务器上运行作业时，此操作将十分有用。请注意，您可以搜索和筛选任何类型的 SA 作业，而不仅仅是推送和恢复作业。

要搜索和筛选结果，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “作业和会话” 选项卡。

2 选择 “作业日志”。此时将显示此服务器上已运行的作业列表。

3 从此作业列表中选择一个作业。

4 选择 “操作” > “打开” 菜单。此时将显示选定作业的详细信息。

5 在作业屏幕中，选择 “作业状态”。此时将显示此作业的结果。

6 在 “操作” 列中选择任何步骤。

7 在键盘上键入 Ctrl-F。此时将显示 “查找” 工具，用于查找此作业中的步骤。在文本框中输入要搜索的文本，然后使用按钮进行搜索并突出显示。

8 在下方框中选择详细的文本。

9 在键盘上键入 Ctrl-F。此时将显示 “查找” 工具，用于查找此作业结果的详细信息。在文本框中输入要搜索的文本，然后使用按钮进行搜索并突出显示。

10 要删除 “查找” 工具，请选择此文本框并键入键盘上的 Esc。

将配置模板与目标配置文件进行比较 - 预览

在将应用程序配置推送到服务器之前，可以使用 “预览” 按钮将建议的应用程序配置与服务器上的目标配置文件进行比较。您可以从托管服务器中选择服务器，也可以在设备组中选择服务器。

从托管服务器中选择服务器

要将配置模板中的值与服务器上的实际配置文件进行比较，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “设备” 选项卡。

2 选择 “服务器” > “所有托管服务器”。

3 在内容窗格中选择一个服务器。

4 选择 “操作” > “打开” 菜单。

5 选择 “管理策略” 选项卡。

6 打开 “配置的应用程序” 节点，将显示附加到服务器的所有应用程序配置。

7 打开所需的 “应用程序配置” 节点，将显示该应用程序配置的所有实例。

8 选择此应用程序配置的一个实例。

9 在内容窗格中，从下拉列表选择一个模板。

10 单击 “预览”。此时会将从模板生成的配置文件和值集与服务器上的实际配置文件进行比较，然后并列显示两个文件，其中使用颜色编码以便于解释。

- 绿色：表示新信息已添加。
- 蓝色：表示信息已修改。
- 红色：表示信息已删除。
- 黑色：表示无更改。

从设备组中选择服务器

要将配置模板中的值与设备组中服务器上的实际配置文件进行比较，请执行以下操作：

1. 在 SA 客户端导航窗格中，选择 “设备” 选项卡。
2. 选择 “设备组” > “Public”，将显示公用设备组。
3. 导航到所需设备组，然后在内容窗格中选择一个公用设备组。
4. 选择 “操作” > “打开” 菜单。
5. 打开 “配置的应用程序” 节点，将显示附加到此设备组的所有应用程序配置和此设备组中的服务器。
6. 打开 “服务器” 节点将显示设备组中的所有服务器。
7. 打开所需的服务器节点，将显示附加到该服务器的应用程序配置。
8. 打开所需的 “应用程序配置” 节点，将显示此应用程序配置的所有实例。
9. 选择此应用程序配置的一个实例。
10. 在内容窗格中，从下拉列表选择一个模板。
11. 单击 “预览”。此时会将从模板生成的配置文件和值集与服务器上的实际配置文件进行比较，然后并列显示两个文件，其中使用颜色编码以便于解释。
    - 绿色：表示新信息已添加。
    - 蓝色：表示信息已修改。
    - 红色：表示信息已删除。
    - 黑色：表示无更改。

比较两个配置模板

要比较两个配置模板，请执行以下操作：

1. 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。
2. 查找并打开 “应用程序配置” 节点。打开 “模板” 节点。打开操作系统组并导航到模板文件所在的操作系统。请注意，模板可应用于多个操作系统。
3. 选择一个配置模板。
4. 按住键盘上的 Ctrl 键并选择第二个模板。

5 右键单击并选择 “比较” 菜单。“比较” 屏幕将显示两个文件之间的差异。使用屏幕右上方的箭头可在两个文件之间导航。以下颜色表示差异：

- 蓝色表示两个模板之间的不同信息。
- 红色表示已删除的信息。
- 绿色表示已添加的信息。
- 黑色表示相同的文本。

6 当结束查看差异时，请选择 “关闭”。

对配置文件元素名称进行本地化

您可以在 SA 客户端的值集编辑器中以您的本地语言显示配置文件元素的名称。如果负责指定值集的系统管理员使用的语言与您不同，此功能将很有帮助。

本地化文件表示特定于区域设置的资源文件，您可以在其中为配置模板元素定义本地化字符串。值集编辑器中将显示这些本地化字符串。

本地化模板仅在 SA 客户端的值集编辑器中使用，在推送期间将忽略这些模板。

创建本地化文件

要对 SA 客户端中的配置文件元素名称进行本地化，必须首先在 SA 库中创建本地化模板。

要创建本地化文件，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。

2 查找并打开 “应用程序配置” 节点。打开 “模板” 节点。打开操作系统组并选择将使用模板文件的操作系统。请注意，模板可应用于多个操作系统。可在后面的步骤中进行指定。

3 选择 “操作” > “新建” 菜单。此时将显示 “模板” 屏幕，您可以在其中定义模板的属性。

4 选择 “属性” 视图并输入本地化模板文件的描述及以下信息：

- **名称**：指定本地化文件的名称。本地化模板文件的命名约定要求本地化文件以 .< 区域设置 > 结尾。< 区域设置 > 的值由 ISO-639 定义，此值是两个小写字母代码，用于表示语言名称。有关 ISO 639 的详细信息，请参见 http://www.loc.gov/standards/iso639-2/php/code_list.php

  例如，.es 代表西班牙语、.en 代表英语、.fr 代表法语、.zh 代表中文以及 .hi 代表印地语。

- **位置**：指定 SA 库中用于存储本地化模板的位置。

- **版本**：版本可以是用于跟踪模板更改的任何字符串。版本不会自动增加。
- **类型**：指定本地化文件的类型。
- **可用性**：使用此设置可跟踪已经过测试并准备好使用的本地化文件以及尚未测试或已弃用的本地化文件。此设置不会更改本地化文件可以实现的功能。您可以将此字段值用作搜索条件。

5 选择 “内容” 视图。

6 在模板编辑器中直接输入本地化说明。

- 本地化说明的显示文本格式如下所示：

  `printable/<` *命名空间* `>/<` *变量* `> <` *本地化字符串* `>`

  其中 `printable` 是关键字，表示此行是本地化说明的显示文本。

  < *命名空间* > 是数据库中用于存储所需变量的命名空间。

  < *变量* > 是配置模板中定义的变量。

  < *本地化字符串* > 是将在值集编辑器中显示的文本，而不是完整命名空间和变量名称。

- 本地化说明的工具提示格式如下所示：

  `description/<` *命名空间* `>/<` *变量* `> <` *本地化字符串* `>`

  其中 `description` 是关键字，表示此行是本地化说明的工具提示文本。

  < 命名空间 > 和 < 变量 > 与上述相同。

  < 本地化字符串 > 是在值集编辑器中，当某人将鼠标悬停在项上时显示的文本。

有关编辑操作和语法突出显示的信息，请参见编辑模板中的 CML 或 XML （第 32 页）。

7 选择 “验证” 可分析语法并检查错误。

8 选择 “文件” > “保存” 保存模板。

9 如以下第 49 页应用本地化模板中所述将本地化模板添加到应用程序配置对象。

应用本地化模板

在创建本地化模板之后，您可以将其应用到应用程序配置，以便此配置文件元素显示在您的本地语言中。

要将本地化模板应用到应用程序配置，请执行以下操作：

1 打开应用程序配置对象，如下所示。

   a 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。

   b 查找并打开 “应用程序配置” 节点。打开 “配置” 节点。打开操作系统组并导航到应用程序配置所在的操作系统。请注意，应用程序配置可应用于多个操作系统。

c 选择一个应用程序配置然后选择 “操作” > “打开”。

2 选择 “属性” 视图。

3 在内容窗格的 “本地化文件” 下，选择 “+” 按钮或选择 “操作” > “添加”。

4 在 “选择本地化模板” 屏幕中，选择一个本地化模板。

5 选择 “确定”。

6 选择“文件” > “保存”。此操作将应用本地化模板并使用本地化模板指定的语言显示配置文件元素。在任何人显示此应用程序配置的值集编辑器时，都将显示本地化字符串，而不会显示原始的命名空间和变量名称。

3 应用程序配置概念

使用 Server Automation，可以从一个中心位置管理配置文件（包括 XML 配置文件），并在数据中心的多个服务器中轻松传播更改和更新。您可以管理单个配置文件（如 UNIX 系统中的 /etc/hosts 文件），也可以管理与某个应用程序关联的多个复杂配置文件，例如与某个大型业务应用程序（如 WebLogic 或 Websphere）关联的配置文件。

配置文件是服务器上需要您控制其内容的任何文件。这包括应用程序配置文件以及系统配置文件，例如：

- 可控制应用程序服务器、Web 服务器、数据库或其他应用程序行为的文件。例如，您可以管理跨多个不同服务器运行的 Apache Web 服务器的 httpd.conf 文件。可附加此文件的应用程序配置并在每个服务器上为该文件的每个实例输入特定值。
- 服务器上的文件，例如 UNIX 服务器上的 /etc/hosts、/etc/fstab、/etc/passwd 或 /etc/groups 文件。
- 服务器上用于配置服务器的任何其他文件。

此外，您可以使用应用程序配置执行以下操作：

- 在服务器上放置配置文件之前或之后**运行脚本**例如，可使用脚本，以便在将配置推送到服务器之后重新启动应用程序。有关详细信息，请参见有关使用应用程序配置运行脚本（第 67 页）。
- 在**软件策略**中使用应用程序配置，以作为应用程序部署和持续管理的一部分。有关详细信息，请参见在软件策略中使用应用程序配置（第 76 页）。
- **审核**以确定服务器是否符合所需的应用程序配置文件内容。有关详细信息，请参见审核应用程序配置（第 75 页）。
- 根据所定义的应用程序配置，对服务器进行**符合性检查**。请参见应用程序配置符合性（第 69 页）。
- **恢复**先前的配置文件。有关详细信息，请参见将配置文件恢复到之前的状态（第 43 页）。

有关应用程序配置对象

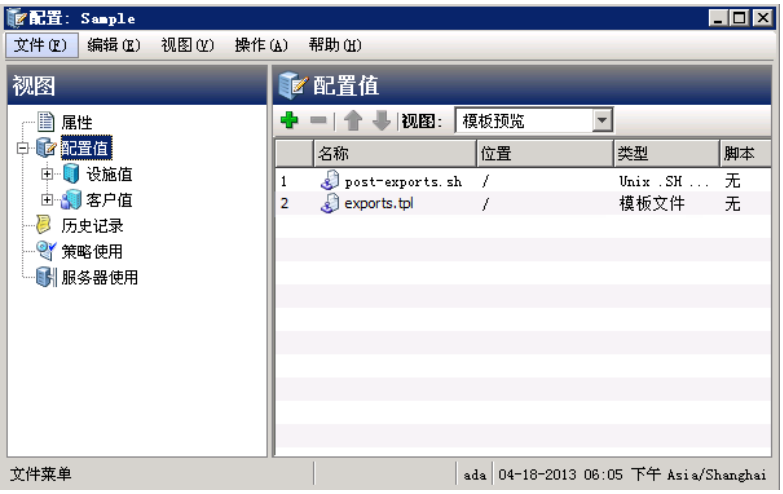
要管理服务器上的配置文件，需要在 SA 库中创建应用程序配置对象，并至少创建一个配置模板。应用程序配置对象只是一个容纳模板和可选脚本的容器。

在将应用程序配置推送到一个或多个服务器时，SA 会执行以下操作：

- 从配置模板和指定值集生成配置文件。
- 将生成的配置文件复制到服务器。

以下图 8 显示了一个示例应用程序配置，其中包含一个名为“exports.tpl”的模板以及名为“post-exports.sh”的 Unix Shell 脚本文件。“.tpl”文件扩展名指示这些文件为模板。

图 8　包含配置模板和 Shell 脚本的应用程序配置



有关配置模板和脚本模板

配置模板是配置文件的模型，它用变量代替随服务器变化的数据值。此外，配置模板还包含有关如何使用模板重新创建配置文件以及如何在推送操作过程中使用值集重新创建配置文件的说明。配置模板可定义配置文件的固定部分和变量部分。模板还包含解析配置文件和重新创建配置文件以推送到托管服务器的说明。

可以使用 SA 配置建模语言 (CML) 或 XML 创建配置模板文件。针对所有基于 XML 的配置文件使用 XML，针对所有其他配置文件使用 CML。有关 CML 的完整信息，请参见 CML 引用（第 153 页）。有关 XML 配置文件的信息，请参见管理 XML 配置文件 （第 79 页）。

您可以定义将用于生成最终配置文件的值。在 SA 数据库中，这些值被存储为**值集**。配置模板中的 CML 或 XML 将值集中的值与模板文件合并，以生成目标配置文件。有关详细信息，请参见有关值集（第 54 页）。

您还可以使用配置模板从配置文件导入数据并构建值集。有关详细信息，请参见导入和验证模板文件 （第 33 页）。

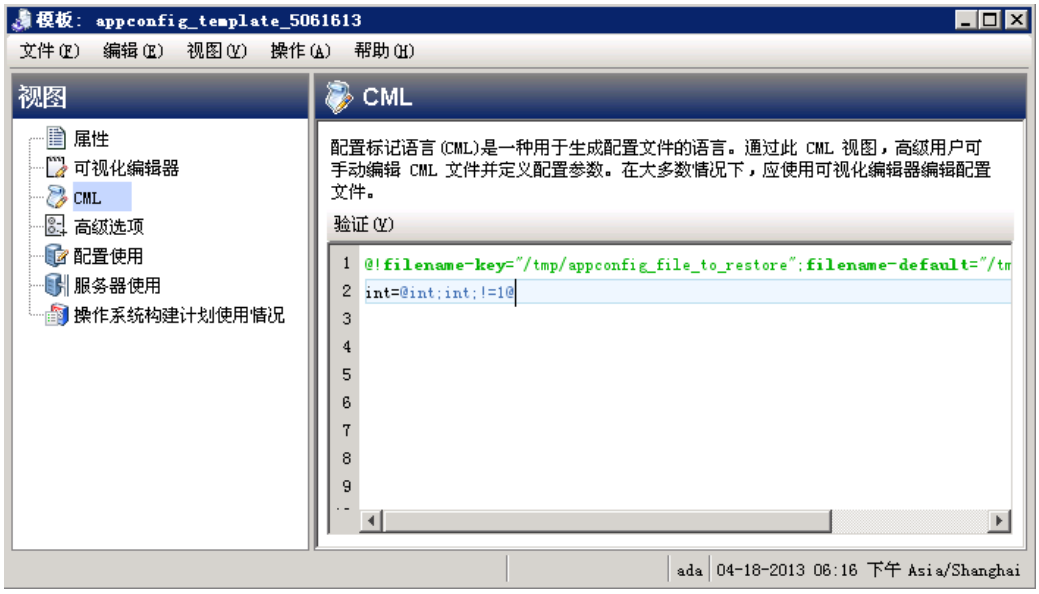
要管理配置文件，请为要管理的每个文件创建配置模板并将其添加到应用程序配置。有关详细信息，请参见创建应用程序配置 （第 16 页）。

CML 配置模板

配置建模语言 (CML) 提供了创建配置文件模板的一种方式，可将该模板与值集合并用以生成实际的配置文件。以下图 9 显示了采用配置建模语言 (CML) 创建的示例配置模板文件。

有关详细信息，请参见 CML 教程 1 - 为简单 Web 应用程序服务器创建应用程序配置（第 103 页）、CML 教程 2 - 创建 Web 服务器配置文件的模板 （第 113 页）和 CML 引用 （第 153 页）。

图 9　使用 CML 编写的配置模板示例



XML 和 XML-DTD 配置模板

您还可以在托管服务器上管理 XML 配置文件。通过使用 XML 配置模板，可以对 XML 配置文件值进行建模并根据实际 XML 检查这些值，然后将更改推送到目标文件。您可以对使用或不使用 DTD 的 XML 配置文件进行建模。

XML 配置模板的功能与 CML 模板类似，但是前者使用注释中定义的某些应用程序配置选项。此外，通过 XML 配置模板，还可以使用标记来自定义文件在 SA 客户端中的显示方式。

有关详细信息，请参见管理 XML 配置文件 （第 79 页）。

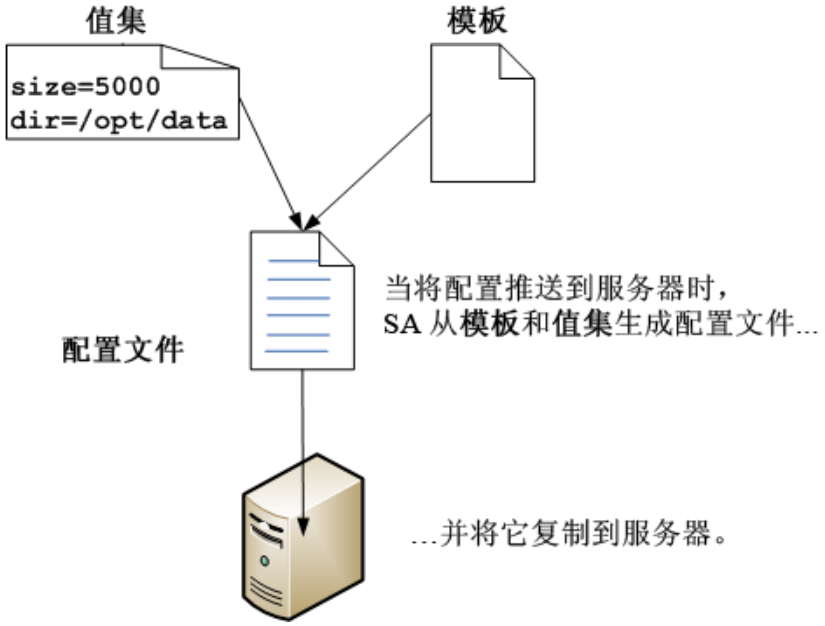
脚本模板

您可以将脚本添加到应用程序配置，这些脚本将在向目标服务器复制配置值之前或之后执行。要使应用程序配置对象中包含脚本，必须将脚本复制到 CML 模板，然后将脚本模板导入应用程序配置对象中。

有关详细信息，请参见有关使用应用程序配置运行脚本（第 67 页）、从脚本创建模板（第 35 页）。

有关值集

值集是一组数据值，可与模板文件合并用以生成目标配置文件。生成的配置文件可推送到服务器。



简单地说，可以为特定服务器定义“服务器实例”级别的值集。将“服务器实例”值集中的值与配置模板合并，以生成将推送到服务器的实际配置文件。

值集级别和值集继承

为各个服务器设置服务器实例级别的值适用于较小数量的服务器，但对于大量服务器，可以使用值集继承来设置较高级别的默认值，这些值可应用于托管服务器的较大子集并可由较低级别继承。每个级别将继承其上级别的值，除非此级别明确阻止继承或设置将覆盖继承值的值。

以下表 1 列出了所有值集继承级别、在较低级别继承或覆盖值的方式以及在每个级别设置值的方式。以下描述了在每个级别设置值的详细信息。

表 1　值集级别和值集继承

级别	级别的描述	值的设置方式
应用程序	此级别定义应用于应用程序配置本身的值。除非被以下任意级别覆盖，否则它将应用于已附加应用程序配置的所有服务器。	打开“应用程序配置”对象。选择“内容” > “应用程序值” 。选择一个模板。选择“文件值”视图。请参见在应用程序级别设置值 （第 60 页）。
设施	此级别定义设施的值。除非被以下任意级别覆盖，否则它将应用于指定设施中已附加应用程序配置的所有服务器。	打开“应用程序配置”对象。选择“内容” > “设施值” > < *设施* >。选择一个模板。选择“文件值”视图。请参见在设施级别设置值 （第 61 页）。
客户	此级别定义客户的值。除非被以下任意级别覆盖，否则它将应用于指定客户中已附加应用程序配置的所有服务器。	打开“应用程序配置”对象。选择“内容” > “客户值” > < *客户* >。选择一个模板。选择“文件值”视图。请参见在客户级别设置值 （第 61 页）。
组	此级别定义设备组的值。除非被以下任意级别覆盖，否则它将应用于指定设备组中已附加应用程序配置的所有服务器。	打开设备组。选择“配置的应用程序” > <*应用程序配置名称*>。选择一个模板。请参见在组级别设置值 （第 63 页）。
组实例	此级别定义应用程序配置的特定实例值。除非被以下任意级别覆盖，否则它将应用于指定设备组中已附加实例的所有服务器。	打开设备组。选择“配置的应用程序” > <*应用程序配置名称*> > <*实例名称*>。选择一个模板。请参见在组实例级别设置值 （第 64 页）。
服务器	此级别定义服务器的值。除非被以下任意级别覆盖，否则它将应用于指定服务器上应用程序配置的所有实例。	打开服务器。选择“管理策略”选项卡。选择“配置的应用程序” > < *应用程序配置名称* >。选择一个模板。请参见在服务器级别设置值（第 65 页）。
服务器实例	此级别定义服务器上应用程序配置的特定实例值。它仅应用于指定服务器上的指定应用程序配置实例并覆盖所有以上级别。	打开服务器。选择“管理策略”选项卡。选择“配置的应用程序” > < *应用程序配置名称*> > <*实例名称*>。选择一个模板。请参见在服务器实例级别设置值 （第 66 页）。

以下表 2 说明了应用程序配置值的继承方式。每行中的值表示该级别的值集。最下面一行显示将推送到服务器的实际值。

表 2　应用程序配置值的继承

级别	每个级别的值集						
应用程序	2	1	Z	Y	X	W	V
设施		U	T	S	R	Q	P
客户			O	N	M	L	K
组				J	I	H	G
组实例					F	E	D
服务器						C	B
服务器实例							A
继承结果	2	U	O	J	F	C	A

阻止继承

通过在值集编辑器中为任意变量选择 “阻止继承”，可以在任何级别阻止继承。将不会继承阻止的级别之前的所有值集。但仍然继承阻止的级别之后的级别中的值。较低级别必须设置此值，否则此变量将不具有任何值。

1 在任意级别打开值集编辑器。
2 在任意变量的值列中，选择一个值。将在编辑框的最右侧显示一个下拉列表和 “...” 按钮。
3 选择下拉列表或 “...” 按钮。将显示变量值的多个选项。
4 选择 “阻止继承”。
5 选择 “文件” > “保存” 或 “保存更改” 按钮。
6 有关详细信息，请参见在值集编辑器中设置值 （第 57 页）。

有关值集编辑器

值集编辑器显示模板中定义的变量并支持您设置这些变量的值。在任意继承级别选择应用程序配置的值集时会显示此值集编辑器。此部分将描述如何使用值集编辑器。

有关每个继承级别的值集编辑器的示例，请参见应用程序、设施和客户级别的值集编辑器 （第 60 页）、组级别的值集编辑器 （第 62 页）和服务器级别的值集编辑器 （第 64 页）。

在值集编辑器中设置值

只要设置的值符合变量的数据类型，您就可以在值集编辑器中为变量设置任何值。在所需变量旁边的 “值” 列中键入值。

此外，也可以按以下任意方式设置值：

- 选择编辑框最右侧的下拉列表并选择以下值之一：
  - 空字符串 - 指定将长度为零的字符串放置于值集中。
  - 阻止继承 - 指定不会继承较高级别的值。较低级别必须设置此值，否则此变量将不具有任何值。
  - 代理版本、验证域、机箱 ID、客户 ID、客户名称等 - 指定将托管服务器的选定信息放置于值集中。
- 选择编辑框右侧的 “...” 按钮可显示以下选项：
  - 没有值 - 将值集中的值保留为空。
  - 阻止继承 - 指定不会继承较高级别的值。较低级别必须设置此值，否则此变量将不具有任何值。
  - 任何值 - 输入任何值。
  - 对象特性 - 从以上列表中选择其中一个值。
  - 自定义特性 - 输入自定义特性。有关自定义特性的详细信息，请参见 《SA User Guide: Application Automation》。
  - 部署自动化值 - 从部署自动化管理的应用程序输入值。有关详细信息，请参见 《SA 用户指南：应用程序部署管理器》。
- 右键单击编辑框可显示编辑菜单。

在值集编辑器中设置字段

无论在哪个级别设置值，值集编辑器都会为正在编辑的值集显示以下所述的字段。

图 10 显示了附加有名为 “WAS-app-config” 的应用程序配置的服务器。其中显示了服务实例级别的值集，您可以对其进行编辑。有关以下字段的示例请查看此图像。

- **模板**：列出应用程序配置对象中包含的模板文件并允许您选择要查看和修改的模板。选择要查看的模板。
- **文件名**：指定托管服务器 （配置模板的目标）上的配置文件名称。如果未设置此文件名，则将继承此文件名。如果应用程序配置层次结构中没有设置任何文件名，则将使用配置模板中列出的文件名。

  如果服务器上的应用程序具有多个实例，则使用此字段来指示每个目标配置文件的完整路径名称。

- **编码**：指定目标配置文件的字符编码。默认为托管服务器上使用的编码。（请注意，SA 客户端中不支持使用 UTF-16 编码。）
- **保留格式**：指定是否保留目标配置文件的空格、注释和顺序。SA 将试图尽可能保留目标配置文件的格式，但是可能无法保留所有注释和格式。如果模板使用 `@!partial-template@` CML 标记，则此选项必选。

  如果没有为模板打开此选项，则将从文件删除所有注释和格式，并使用模板中的默认顺序和空格。

  请注意，对于基于 XML 的模板，保留格式将不能保留 XML 标记内的空格或特性顺序。保留格式将保留所有除标记内的空格和特性顺序之外的所有空格和顺序。推送后，标记内的额外空格将消失，其中特性的顺序也可能会更改。消除空格和更改特性顺序不会影响 XML 的含义。
- **保留值**：指定当值集中没有对应的值时是否保留托管服务器上目标配置文件中包含的值。默认情况下，关闭此选项。

  通过 “保留值” ，您可以请求保留服务器的配置文件中尚未存储在 SA 数据库中的任何值。如果不打算将所有当前值从配置文件导入 SA 中或者用户或程序有时在 SA 外部修改配置文件，则此选项将十分有用。它允许您在 SA 外部管理某些配置文件更改。
- **显示所继承的值**：显示生成的值集，其中包括从较高级别继承的值。关闭此选项时，将仅显示当前级别的值集。此视图为只读，且仅在服务器实例级别查看值集时才提供。

值集编辑器中列的含义

在编辑值集时， SA 客户端会显示有关配置模板值的以下信息。

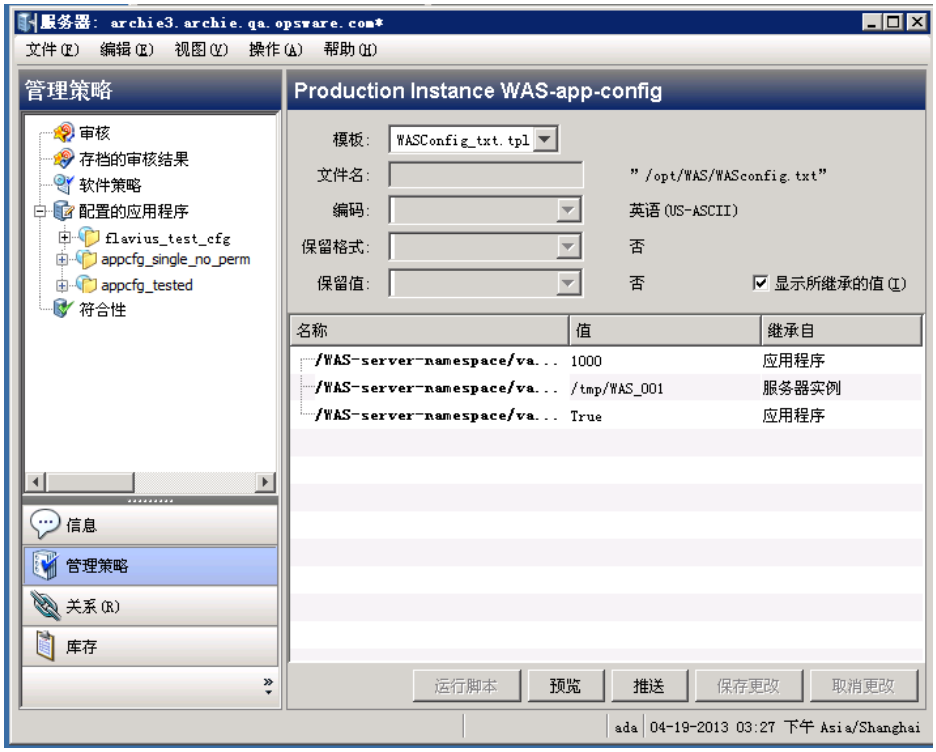
图 10 显示了附加有名为 “WAS-app-config” 的应用程序配置实例的服务器。其中显示了服务实例级别的值集，您可以对其进行编辑。以下描述了列 “名称”、“值” 以及 “继承自”。

- **名称**：列出模板文件中的变量名称。此名称可以是简单类型、简单类型的列表，也可以是多维列表。多维列表将显示在其父列表下方。所需元素均显示为粗体。可双击以显示或隐藏多维列表。要将其他条目添加到列表类型值，请右键单击父列表，然后选择 “添加项目” 。配置模板中设置了必填字段。必填字段不能为空，否则将无法预览或推送应用程序配置。
- **值**：列出所显示的值集中的值。您可以输入文本值或从服务器设置中选择特性，如客户名称、客户 ID、机箱 ID、设备 ID 等。如果您将某个设置留空，则会从父级或上级（如果父级或上级配置了值）继承该设置。要将值设置为值的自定义特性，请单击 “...” 按钮以使用 “设置值” 对话框。

- **继承自**：指示从哪里继承值。仅在服务器实例级别查看并已选择 “显示所继承的值” 选项时，才会显示此列。

  如果已设置 “保留值” 选项，则服务器上的配置文件将成为继承层次结构的最外层级别。也就是说，如果值集中不存在任何值，就会保留此文件中的值。

图 10　服务器实例级别的值集编辑器



从现有配置文件导入值集

虽然可以手动设置值集中的值，但是也可以将值从托管服务器上的现有配置文件导入值集中。

要将值从现有配置文件导入值集中，请执行以下操作：

1. 在要导入值的级别显示值集编辑器。请参见以下部分了解如何在每个级别显示值集编辑器。
2. 在值集编辑器中，请确保 “文件名” 字段显示包含要导入值的配置文件的绝对文件名。
3. 右键单击 “值” 列表并选择 “导入值”。配置模板的所有值都将被替换为实际配置文件的值。

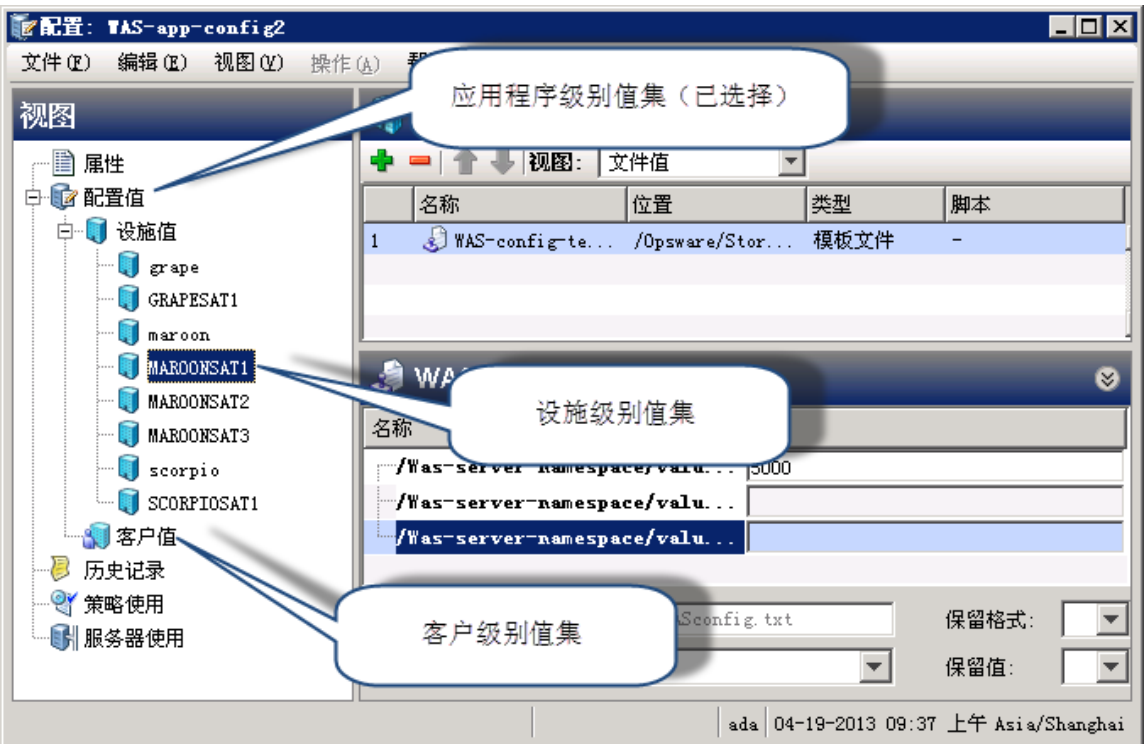
   选择 “导入值” 菜单项将读取托管服务器上的现有配置文件、解析值并将其保存在选定级别。导入值后，可修改任意值并将更改推送回服务器 （如果需要）。
4. 选择 “保存更改” 按钮。

应用程序、设施和客户级别的值集编辑器

以下图 11 显示了应用程序、设施和客户级别的值集编辑器。仅在打开应用程序配置对象时，此视图才可用。

- 选择“配置值”将显示应用程序级别的值集编辑器。请参见在应用程序级别设置值（第 60 页）。
- 在 “设施值” 下选择一个设施将显示设施级别的值集编辑器。请参见在设施级别设置值（第 61 页）。
- 在 “客户值” 下选择一个客户将显示客户级别的值集编辑器。请参见在客户级别设置值（第 61 页）。

图 11　在应用程序、设施和客户级别的值集编辑器 - 已选择应用程序级别



在应用程序级别设置值

应用程序级别定义应用于应用程序配置本身的值。除非被以下任意级别覆盖，否则它将应用于已附加应用程序配置的所有服务器。

要在应用程序级别设置值，请执行以下操作：

请参见图 11 中的示例，其显示了应用程序、设施和客户级别的值集编辑器。

1. 在 SA 客户端中打开应用程序配置对象。
2. 选择 “配置值” 节点。

3 在应用程序配置中选择一个模板。

4 从“视图 :”下拉列表中选择“文件值”。此时将在右下侧显示值集编辑器，在此您可以在应用程序级别设置默认值。

5 在文本框中输入值。通过右键单击菜单项可使用其他编辑功能。另请参见在值集编辑器中设置值（第 57 页）。

6 可选择从“视图 :”下拉列表选择“文件预览”以查看生成的文件。

7 选择“文件”>“保存”以保存您的更改。

在设施级别设置值

设施级别定义设施的值。除非被以下任意级别覆盖，否则它将应用于指定设施中已附加应用程序配置的所有服务器。

要在设施级别设置值，请执行以下操作：

请参见图 11 中的示例，其显示了应用程序、设施和客户级别的值集编辑器。

1 在 SA 客户端中打开应用程序配置对象。

2 打开“配置值”视图可显示“设施值”节点。

3 打开“设施值”节点并选择一个设施。

4 在应用程序配置中选择一个模板。

5 从“视图”下拉列表中选择“文件值”。此时将在右下侧显示值集编辑器。

6 在文本框中输入值。通过右键单击菜单项可使用其他编辑功能。另请参见在值集编辑器中设置值（第 57 页）。

7 可选择从“视图”下拉列表选择“文件预览”以查看生成的值。

8 选择“文件”>“保存”以保存您的更改。

在客户级别设置值

客户级别定义客户的值。除非被以下任意级别覆盖，否则它将应用于指定客户中已附加应用程序配置的所有服务器。

要在客户级别设置值，请执行以下操作：

请参见图 11 中的示例，其显示了应用程序、设施和客户级别的值集编辑器。

1 在 SA 客户端中打开应用程序配置对象。

2 打开“配置值”视图可显示“客户值”节点。

3 打开“客户值”节点并选择一个客户。

4 在应用程序配置中选择一个模板。

5 从“视图”下拉列表中选择“文件值”。此时将在右下侧显示值集编辑器。

6 在文本框中输入值。通过右键单击菜单项可使用其他编辑功能。另请参见在值集编辑器中设置值（第 57 页）。

7 可选择从 “视图”下拉列表选择 “文件预览”以查看生成的值。

8 选择 “文件” > “保存”以保存您的更改。

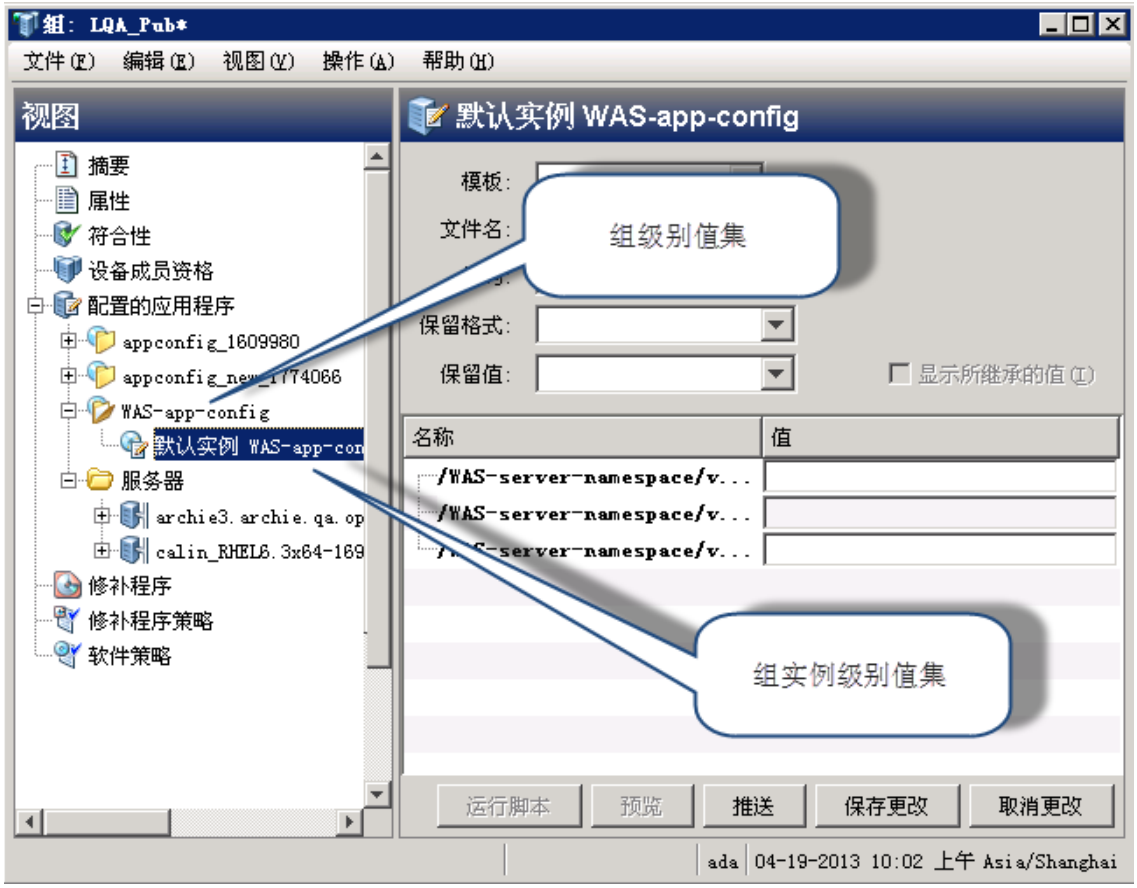
组级别的值集编辑器

组级别定义设备组的值。除非被以下任意级别覆盖，否则它将应用于指定设备组中的所有服务器（假设设备组已附加应用程序配置）。

以下图 12 显示了组和组实例级别的值集编辑器。仅在设备组附加了应用程序配置之后，组和组实例级别才可用。仅在打开已附加应用程序配置的设备组时，此视图才可用。有关详细信息，请参见将应用程序配置附加到服务器或设备组 （第 38 页）。

- 选择名为 “WAS-app-config”的应用程序配置时将显示组级别的值集编辑器。请参见在组级别设置值 （第 63 页）。
- 选择其中一个应用程序配置实例时将显示组实例级别的值集编辑器。此示例显示了名为 “Production Instance WAS-appconfig”和 “Staging Instance WAS-appconfig”的两个应用程序配置实例。此值集编辑器显示 “Production Instance WAS-appconfig”的值集。请参见在组实例级别设置值 （第 64 页）。

图 12 组级别和组实例级别的值集编辑器，显示实例“Production Instance WAS-appconfig”的值



在组级别设置值

组级别定义设备组的值。除非被以下任意级别覆盖，否则它将应用于指定设备组中的所有服务器。

在组级别设置值之前，必须将应用程序配置附加到设备组。请参见以上图 12 中的示例，其显示了组和组实例级别的值集编辑器。

要在组级别设置值，请执行以下操作：

1 在 SA 客户端中打开设备组。必须将应用程序配置附加到设备组。

2 在左侧的 “视图” 窗格中打开 “配置的应用程序” 节点。

3 在 “配置的应用程序” 节点下选择所需的应用程序配置节点。此时将在右侧显示值集编辑器。

4 在 “模板 :” 下拉列表中选择所需的模板文件。

5 在文本框中输入值。通过右键单击菜单项可使用其他编辑功能。另请参见在值集编辑器中设置值（第 57 页）。

6 选择 “保存更改” 按钮。

在组实例级别设置值

组实例级别为附加到设备组的应用程序配置的实例定义值。除非被以下任意级别覆盖，否则它将应用于指定设备组中的所有服务器。

在组实例级别设置值之前，必须将应用程序配置附加到设备组。请参见以上图 12 中的示例，其显示了组和组实例级别的值集编辑器。

要在组实例级别设置值，请执行以下操作：

1 在 SA 客户端中打开设备组。必须将应用程序配置附加到设备组。请参见将应用程序配置附加到服务器或设备组 （第 38 页）。

2 在左侧的 “视图” 窗格中打开 “配置的应用程序” 节点。

3 在 “配置的应用程序” 节点下打开所需的应用程序配置节点。

4 选择所需的应用程序配置实例。例如，以上图 12 显示了选定的名为 “Production Instance WAS-appconfig” 的应用程序配置实例。这是名为 WAS-app-config 的应用程序配置对象的一个实例。此应用程序配置被附加到设备组并定义了此应用程序配置的两个实例。

5 在 “模板 :” 下拉列表中选择所需的模板文件。

6 在文本框中输入值。通过右键单击菜单项可使用其他编辑功能。另请参见在值集编辑器中设置值 （第 57 页）。

7 选择 “保存更改” 按钮。

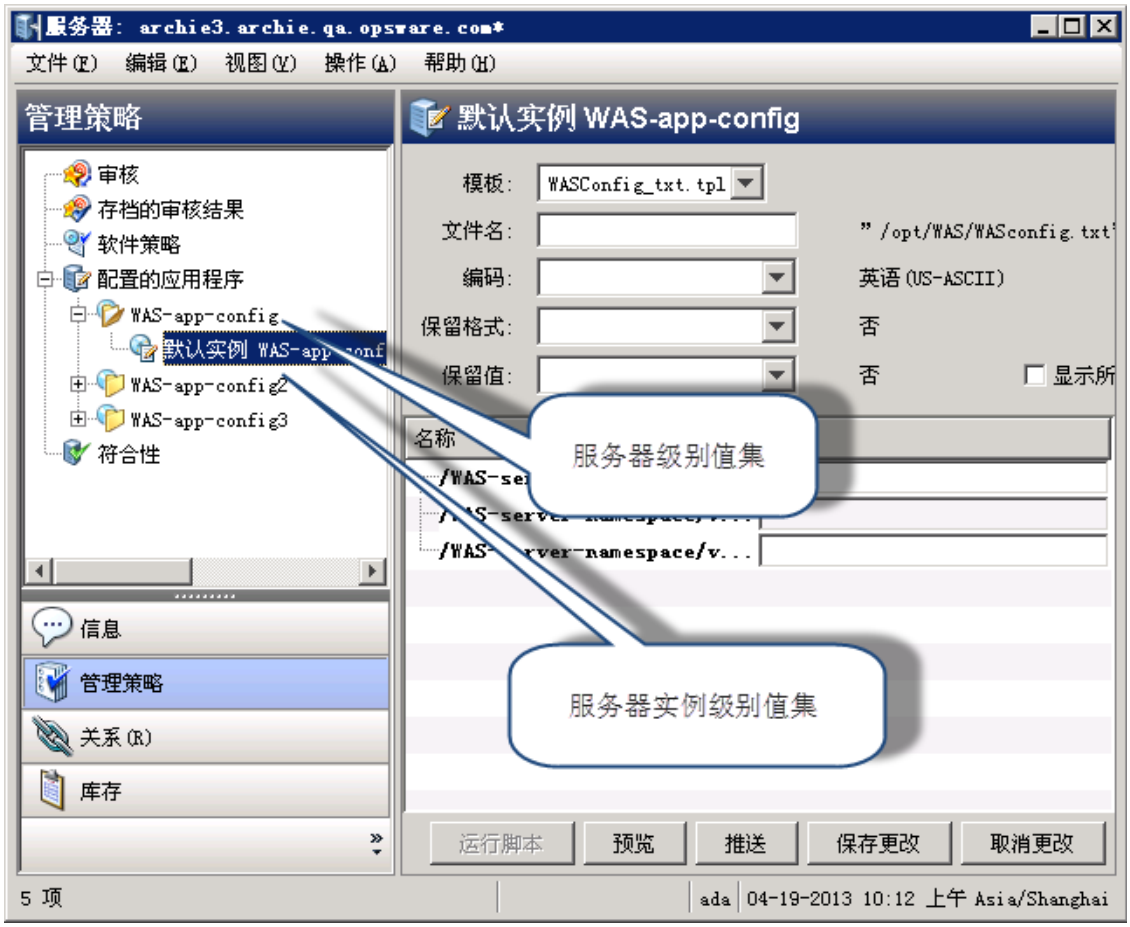
服务器级别的值集编辑器

服务器级别定义服务器的值。除非被以下任意级别覆盖，否则它将应用于指定服务器上应用程序配置的所有实例。

以下图 13 显示了服务器级别和服务器实例级别的值集编辑器。仅在打开已附加应用程序配置的服务器时，此视图才可用。

- 选择名为 “WAS-app-config” 的应用程序配置时将显示服务器级别的值集编辑器。请参见服务器级别的值集编辑器 （第 64 页）。
- 选择名为 “Production Instance WAS-appconfig” 或 “Staging Instance WAS-appconfig” 的实例均会显示服务器实例级别的值集编辑器。请参见在服务器实例级别设置值 （第 66 页）。

图 13 服务器级别和服务器实例级别的值集编辑器，显示实例 “Staging Instance WAS-appconfig” 的值



在服务器级别设置值

服务器级别定义服务器的值。除非被以下任意级别覆盖，否则它将应用于指定服务器上应用程序配置的所有实例。

在服务器级别设置值之前，必须将应用程序配置附加到服务器。请参见以上图 13 中的示例，其显示了服务器级别和服务器实例级别的值集编辑器。

要在服务器级别设置值，请执行以下操作：

1. 在 SA 客户端中打开服务器。必须将应用程序配置附加到服务器。请参见将应用程序配置附加到服务器或设备组 （第 38 页）。
2. 选择左侧的 “管理策略” 选项卡。
3. 在左侧的 “管理策略” 窗格中打开 “配置的应用程序” 节点。
4. 在 “配置的应用程序” 节点下选择所需的应用程序配置节点。
5. 在 “模板 :” 下拉列表中选择所需的模板文件。
6. 在文本框中输入值。通过右键单击菜单项可使用其他编辑功能。另请参见在值集编辑器中设置值（第 57 页）。

7 选择 “保存更改”按钮。

在服务器实例级别设置值

服务器实例级别定义服务器上应用程序配置的特定实例值。它仅应用于指定服务器上的指定应用程序配置实例并覆盖所有以上级别。

在服务器实例级别设置值之前，必须将应用程序配置附加到服务器。请参见以上图 13 中的示例，其显示了服务器级别和服务器实例级别的值集编辑器。

要在服务器实例级别设置值，请执行以下操作：

1 在 SA 客户端中打开服务器。必须将应用程序配置附加到服务器。请参见将应用程序配置附加到服务器或设备组 （第 38 页）。

2 选择左侧的 “管理策略”选项卡。

3 在左侧的 “管理策略”窗格中打开 “配置的应用程序”节点。

4 在 “配置的应用程序”节点下打开所需的应用程序配置节点。

5 在所需的应用程序配置节点下选择所需的实例。

6 在 “模板 :”下拉列表中选择所需的模板文件。

7 在文本框中输入值。通过右键单击菜单项可使用其他编辑功能。另请参见在值集编辑器中设置值（第 57 页）。

8 选择 “保存更改”按钮。

有关使用应用程序配置运行脚本

您可以将脚本添加到应用程序配置，这些脚本将在向目标服务器复制配置值之前或之后执行。

例如，您可能要添加**安装前脚本**（将停止应用程序）和**安装后脚本**（进行配置更改之后重新启动应用程序）。如果在推送或安装后脚本执行期间发生错误，则可以运行**出错后脚本**。

或者可能需要执行**数据操作脚本**来处理非文本的配置数据。在配置 IIS 脚本将元数据库信息读取到平面文件中。在使用配置模板解析平面文件中的信息之后，可以运行安装后脚本将更新的信息写回到元数据库信息。

在推送包含 JScript 或 VBScript 安装前、安装后或出错后脚本的应用程序配置时，即使脚本失败，推送也可能成功。在这些情况中，推送将忽略脚本错误。此应用程序配置不会检测脚本错误，并且能够完成推送，而不发生错误。

如果计划使用这些类型的脚本，则必须确保这些脚本没有错误，并且通过调用 WScript.Quit(<status>) 确保脚本可返回非零的退出状态。

应用程序配置脚本的类型

以下表 3 列出了可用于应用程序配置对象的脚本类型。脚本类型将指定调用脚本的时间。每种脚本类型都只能定义一个。如果定义了一个脚本，但是没有指定其中一种类型，则会将该脚本视为配置模板。即会将此脚本推送到服务器，但是不会执行该脚本。

表 3　脚本类型，指定脚本运行时间

脚本类型	描述
数据操作	在任何安装前脚本之前运行，用于解析非文本配置文件，以便 CML 模板可解析此配置文件。仅要扫描和导入由应用程序配置管理的现有文件时，数据操作脚本也十分有用。 如果此脚本失败，则不会将应用程序配置推送到服务器。
安装前	在发生实际推送之前运行。例如，安装前脚本可以停止应用程序或服务。 如果此脚本失败，则不会将应用程序配置推送到服务器。
安装后	在发生实际推送之后运行。例如，安装后脚本可以在推送之后重新启动服务。
出错后	仅在推送失败或安装后脚本失败时，才会运行。例如，出错后脚本可以恢复备份的文件。

要指定脚本类型，请执行以下操作：

1 在 SA 客户端中，打开包含此模板的应用程序配置对象。

2 选择 “配置值” 视图，以查看应用程序配置对象中包含的模板。

3 选择模板并右键单击以显示菜单。

4 选择脚本类型，如表 3 中所列。

5 选择 “文件” > “保存” 菜单。

另请参见从脚本创建模板 （第 35 页）。

以下表 4 列出了可用于应用程序配置对象的脚本类型。此类型指定脚本的语法和执行环境。

表 4 脚本源类型

脚本源类型	描述
Windows .BAT 脚本	Windows 批处理命令文件。
Windows .JS 脚本	运行在 Windows 上的 Javascript 文件。
Windows .CMD 脚本	Windows 批处理命令文件。
Windows .VBS 脚本	Windows Visual Basic 脚本。
Windows .WSF 脚本	Windows 脚本文件。
Windows .PY 脚本	运行在 Windows 上的 Python 文件。
Unix .SH 脚本	Unix shell 脚本。
其他 Unix 脚本	在 Unix 上运行的任何其他脚本。

要指定脚本类型，请执行以下操作：

1 在 SA 客户端中打开模板。

2 选择 “属性” 视图。

3 在 “类型” 字段中选择表 4 中列出的脚本类型。

4 选择 “文件” > “保存” 菜单。

另请参见从脚本创建模板 （第 35 页）。

有关将应用程序配置推送到服务器

无论何时更改值集中的值并要将这些更改与目标服务器上的配置文件合并，都必须将应用程序配置推送到服务器。推送应用程序配置后，值集中的所有值都将替换目标托管服务器上配置文件中的值。如果目标服务器上不存在配置文件，则在推送时会在服务器上创建新的文件。此外，应用程序配置中的所有脚本基于脚本类型执行。

在推送应用程序配置时，将发生以下事件。

- SA 运行数据操作脚本 （如果指定）。
- SA 从模板和值集生成目标配置文件。
- SA 备份现有配置文件。
- SA 运行安装前脚本 （如果指定）。
- SA 将生成的配置文件复制到服务器。
- SA 执行安装后脚本 （如果指定）。
- 如果已指定，且安装前脚本或安装后脚本失败或者复制操作失败，则 SA 将执行出错后脚本。

有关如何在 “软件策略和审核” 上下文中推送应用程序配置的信息，请参见在软件策略中使用应用程序配置 （第 76 页）。

有关使用应用程序配置中的脚本的详细信息，请参见从脚本创建模板 （第 35 页）和通过运行数据操作脚本管理非文本配置 （第 36 页）。

有关如何推送应用程序配置的说明，请参见推送应用程序配置 （第 41 页）。

推送时的（列表和标量）合并序列取决于在应用程序配置继承层次结构中设置值的方式以及在应用程序配置 CML 模板中配置的序列合并模式。有关序列合并的详细信息，请参见序列聚合（第 189 页）。

应用程序配置符合性

应用程序配置符合性可以帮助您确定附加到服务器 （或服务器组）的应用程序配置的值是否与目标服务器上的配置文件值匹配。

如果目标配置文件值与应用程序配置中定义的值匹配，则将服务器视为符合。如果目标配置与应用程序配置中定义的值不匹配，则将服务器视为不符合。

SA 客户端显示应用程序配置的以下符合性状态：

- **符合**：附加到服务器或设备组 （或多个服务器和组）的应用程序配置中的所有值与目标服务器上的配置值匹配。由 图标表示。

  对于设备组，AppConfig 符合性基于组中所有服务器（以及任何子组中的服务器）的符合性状态。默认情况下，组符合性由默认阈值确定：如果组中 5% 以上的服务器具有 “不符合” 状态，则将整个组视为 “不符合”。要更改此默认设置，请参见《SA User Guide: Application Automation》中的 “Changing Device Group Compliance Settings”。

- **不符合**：应用程序配置中定义的值中至少有一个与目标服务器上配置文件中的值不匹配。由 图标表示。

  对于设备组，不符合性基于组中所有服务器 （以及任何子组中的服务器）的符合性状态。默认情况下，组不符合性由默认阈值确定：如果组中 5% 以上的服务器具有 “不符合” 状态，则将整个组视为“不符合”。要更改此默认设置，请参见《SA User Guide: Application Automation》中的 “Changing Device Group Compliance Settings”。

- **已启动扫描**：当前正在计算应用程序配置符合性信息。由 图标表示。
- **需要扫描**：应用程序配置符合性信息未定义，可能由于从未运行符合性扫描 （例如，在新安装程序上），或已将自上次更改的服务器（或设备组中的服务器）上的配置信息报告给 SA 客户端。

  由 图标表示。

- **不适用**：应用程序配置符合性信息不适用，由短划线 (—) 表示。对于未选中 “可审核” 属性的模板，将会显示此短划线。有关详细信息，请参见创建配置模板 （第 17 页）。

您可以查看单个服务器或服务器组的应用程序配置符合性：

- 单个服务器的应用程序配置符合性
- 多个服务器的应用程序配置符合性

如果对应用程序配置进行任何更改 （例如在值集编辑器中编辑其值），则附加此配置的任何服务器或服务器组都将显示 “需要扫描” 的符合性状态。

单个服务器的应用程序配置符合性

对于单个服务器，符合性视图将显示附加到此服务器的所有应用程序配置的整体符合性。如果此服务器上附加了多个应用程序配置，则可以查看所有应用程序配置的聚合符合性状态以及每个配置的符合性状态。

图 14 显示了单个服务器的应用程序配置符合性。

图 14　服务器的应用程序配置符合性

如果发现应用程序配置和目标服务器上实际配置文件的任何差异，则下部窗格将显示不符合的类别。如果服务器附加了多个应用程序配置，并且应用程序配置的任意一个目标配置文件不同于应用程序配置，则服务器将处于不符合状态。

有关如何运行应用程序配置符合性扫描的详细信息，请参见在服务器中扫描应用程序配置符合性（第 74 页）。

多个服务器的应用程序配置符合性

您可以查看多个服务器的应用程序配置符合性状态。在 SA 客户端导航窗格中选择 “设备”，然后选择 “设备组” 或 “服务器” 。选择一个设备组或一组服务器，然后在 “视图” 菜单中选择 “符合性”。此时将显示选定服务器的聚合符合性状态。

如果组中 5% 以下的服务器不符合要求，则会将附加到服务器组的应用程序配置视为符合。如果 5% 以上的服务器不符合要求，则会将聚合符合性视为不符合。通过在 SA 客户端中选择 “管理” 选项卡，然后选择 “符合性设置”，您可以更改此百分比。

在 “符合性” 视图中，服务器组的详细信息窗格显示是否所有应用程序配置符合要求，但是不会展开显示单个服务器和应用程序配置的细分。

您可以使用以下方式查看服务器组应用程序配置符合性状态：

- 查看多个服务器的应用程序配置符合性 （第 72 页）。
- 查看多个设备组的应用程序配置符合性 （第 72 页）。
- 查看单个设备组的应用程序配置符合性 （第 73 页）。

查看多个服务器的应用程序配置符合性

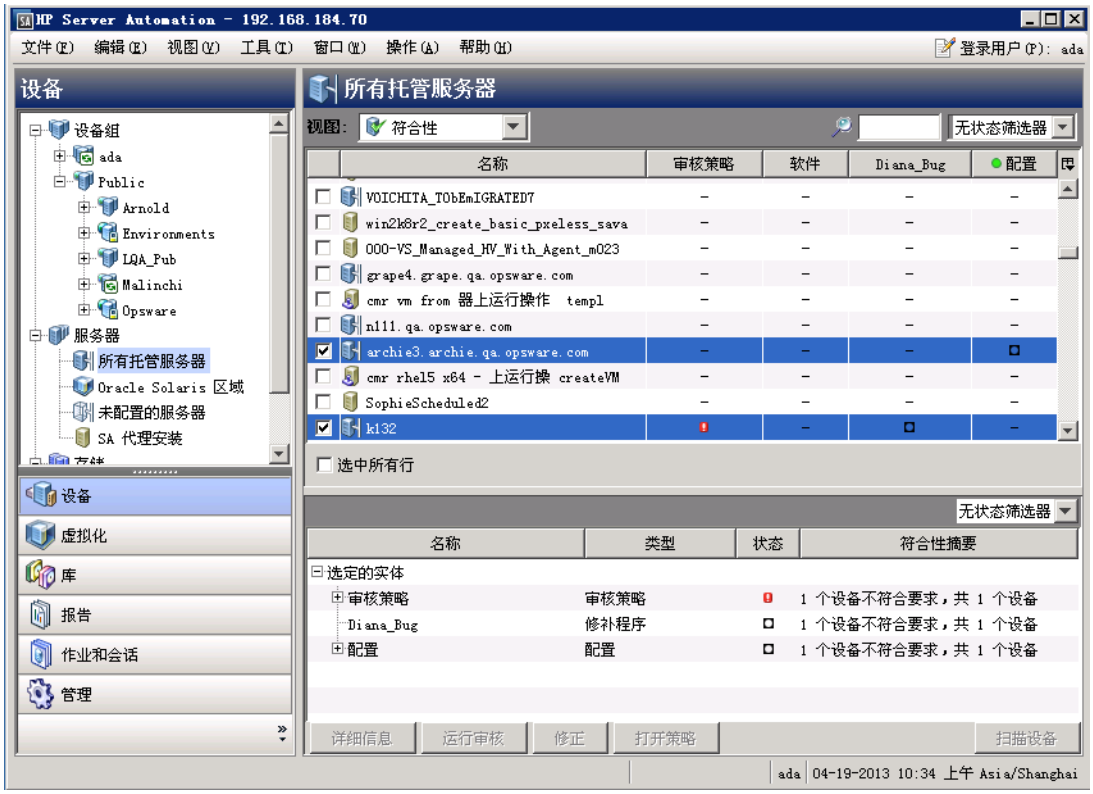
要查看多个服务器的应用程序配置符合性，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “设备” > “服务器” > “所有托管服务器”。

2 在 “视图” 下拉列表中，选择 “符合性”。

3 要查看多个服务器的符合性级别，请选中服务器旁边的复选框，将在底部详细信息窗格中显示选定服务器的符合性汇总，如图 15 中所示。

图 15 多个服务器的应用程序配置符合性



查看多个设备组的应用程序配置符合性

要查看多个设备组的应用程序配置符合性，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “设备” > “设备组”。

2 选择设备组或包含设备组的文件夹。

3 在 “视图” 下拉列表中，选择 “符合性”。此时将显示所有组的符合性状态。

4 要查看多个组的符合性级别，请选中服务器旁边的复选框，将在底部详细信息窗格中显示选定组的符合性摘要，如图 16 中所示。

图 16 多个设备组的应用程序配置符合性

查看单个设备组的应用程序配置符合性

要查看一个设备组的应用程序配置符合性，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “设备” > “设备组”。

2 导航到所需设备组并选择该组。

3 右键单击并选择 “打开”，或者选择 “操作” > “打开”。此时将显示此设备组。

4 从“视图”窗格中选择“符合性”。此时将显示组中所有成员的每个策略类型的整体聚合符合性，而不会显示每个服务器的符合性状态，如图 17 中所示。

图 17 设备组的应用程序配置符合性

在服务器中扫描应用程序配置符合性

在将应用程序配置推送到服务器之后，可以有意或偶尔更改或改变服务器上的此配置文件。或者在应用程序配置中定义的值可能会发生更改。当目标服务器上的配置文件的值与应用程序配置中定义的值不匹配时，会将配置文件视为不符合要求。

您可以在服务器上扫描配置符合性，确定服务器上的任何配置文件是否符合配置模板中存储的值。您可以计划定期进行扫描。

要在一个或多个服务器上扫描配置符合性，请执行以下操作：

1 从 SA 客户端导航窗格中，选择 “设备”。

2 选择 “设备组” 或 “所有托管服务器”。如果选择了 “设备组”，则选择一个设备组将显示属于该组的服务器。

3 在内容窗格中选择一个服务器。还可以选择多个服务器或设备组，然后对其全部扫描。

4 在“操作”菜单中，选择“扫描” > “配置符合性”，或者选择“计划” > “配置符合性扫描”。

- 如果选择 “扫描” > “配置符合性” ，则 SA 将扫描设备以确定符合性并在 “扫描配置符合性” 屏幕中显示状态。
- 如果选择“计划” > “配置符合性扫描”，则 SA 将显示 “计划作业” 屏幕，其中您可以指定作业完成的时间以及其他作业参数。

审核应用程序配置

借助 SA，您可以审核服务器上的配置文件，确定这些文件是否符合组织的配置标准。您可以创建审核规则，用以指定如何定义服务器上的配置文件，并定期审核这些服务器以检查是否正确配置了配置文件。如果发现审核规则定义和目标配置文件值不匹配，可以通过修正这些服务器来修复此问题。

例如，要确保托管服务器上的 /etc/hosts 文件仅为特定 IP 地址定义某些主机名，您可以定义审核规则来指定可接受的主机名和 IP 地址对的列表。运行审核时，如果主机文件包含除规则中指定的值之外的任何值，则审核结果将显示错误，您可以修正此问题。

审核应用程序配置的一般过程会按照以下步骤执行操作：

1. **创建审核和审核规则**：要审核服务器上的配置文件，需首先创建审核。创建审核时，需指定配置文件所基于的源服务器（或快照或快照规范）。然后选择应用程序配置模板来构造规则。此规则将定义要在目标配置文件中检查的确切的值。对于每个审核规则，请指定目标服务器上配置文件的位置。
2. **选择目标服务器**：在审核中，选择要审核的目标服务器。可以选择单个服务器、多个服务器或服务器组。
3. **运行或计划审核**：可以计划运行一次审核，也可以定期运行审核。此外，还可以指定审核结果发送到的电子邮件地址。
4. **检查审核结果**：检查审核结果，确定目标服务器上的配置文件是否与审核规则中定义值匹配。如果存在差异，您可以比较规则和目标文件以查看差异，从而确定如何修正服务器。
5. **修正服务器**：要修复审核结果中找到的差异，可以修正服务器或者任意或全部规则，确保目标配置与规则匹配。

有关使用审核和快照的详细信息，请参见 《SA 用户指南：审核与符合性》中的 “审核和修正”。尤其是 “配置应用程序配置规则”部分。

在软件策略中使用应用程序配置

应用程序配置在用于软件策略中时是一个非常强大的工具。软件策略定义应用程序 （包括要在服务器上安装的所有程序包、修补程序、脚本和其他对象）的理想状态以及在服务器上设置应用程序配置文件的方式。在托管服务器上安装软件策略时，SA 会将所有内容应用到策略的目标服务器，包括在应用程序配置中定义的所有值。

通过使用 SA 客户端中的符合性视图，可以查看从此策略安装的软件的符合性状态。例如，如果某个人从软件策略删除修补程序、在服务器上安装新的程序包，或者更改策略中定义的其中一个配置文件，则策略将在符合性视图中显示为不符合。要确保正确安装和配置应用程序，您可以修正服务器。

有关使用和创建软件策略的详细信息，请参见 《SA 用户指南：软件管理》中的 “软件管理”。

要在软件策略中使用应用程序配置，请执行以下操作：

1. **定义应用程序**：在构建软件策略之前，应用程序专家收集组成此应用程序的所有必需的程序包和修补程序。此外，还收集用于定义和管理与应用程序关联的配置文件的配置模板。
2. **将程序包和修补程序导入 SA**：在定义软件策略的组件之后，将组成应用程序的所有程序包和修补程序导入 SA 库中，以便将其置于软件策略中。
3. **创建应用程序配置和设置值** : 定义将用于生成配置文件的配置值。例如，如果创建软件策略以部署 Apache Web 服务器，则应用程序专家会使用值集编辑器为 httpd.conf 文件定义默认值。例如，将任何安装前或安装后脚本添加到应用程序配置，以便在软件策略修正期间推送应用程序配置之后重新启动 Apache 服务。
4. **测试应用程序配置**：在将应用程序配置添加到软件策略并将应用程序部署到服务器之前，最好将应用程序配置附加到服务器并确保在创建软件策略之前应用程序可以正常运行。可以预览向服务器的配置推送以验证配置的正确性。
5. **创建软件策略**：在定义和创建软件策略的所有组件并将其导入 SA 中之后，应用程序专家可以创建软件策略，用以指定要安装的软件以及安装其组件 （包括所有修补程序、程序包和应用程序配置）的顺序。在将软件策略保存到 SA 库中之后，负责部署、测试和管理应用程序的系统管理员便可以访问此策略。
6. **将策略附加到服务器或服务器组**：在创建和保存软件策略之后，系统管理员可以将策略附加到设备组中的服务器或服务器组。
7. **修正服务器以安装软件**：通过修正服务器上的软件策略，系统管理员可以将软件部署到一个或多个服务器。修正过程可以确保以策略中指定的顺序将策略中定义的所有内容部署在目标服务器上。修正之前，管理员可在作业中预览应用程序配置。预览步骤提供在服务器实例级别为模板变量定义 appconfig 值的选项。

8 **测试应用程序和遍历更改**：系统管理员使用软件策略修正安装应用程序之后，在将应用程序投入生产环境之前，应测试应用程序以确保其可以正常运行并且包含正确的组件。此外，应检查其配置文件所影响的应用程序的每个部分，确保已正确配置该部分。

9 **使用应用程序**：在部署和使用此应用程序之后，系统管理员可以执行持续的管理和维护任务，例如运行软件符合性扫描，确定已部署此应用程序的服务器的符合性状态、修正不符合的服务器以及生成软件符合性报告。

有关使用软件策略的详细信息，请参见 《SA 用户指南：软件管理》。

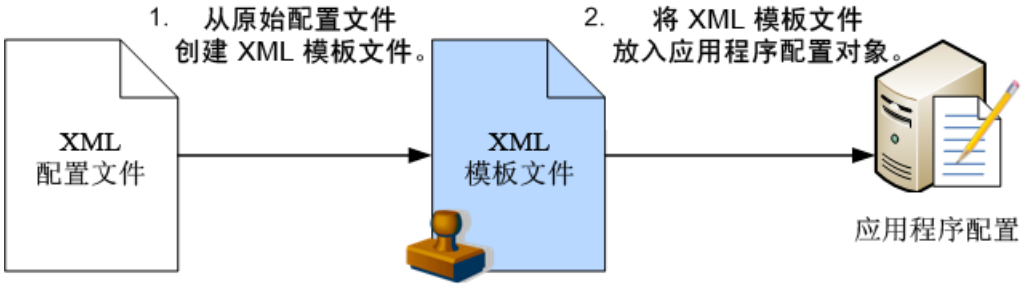
4 管理 XML 配置文件

使用 SA，可以从一个中心位置管理 XML 配置文件，并在数据中心的多个服务器中传播更改。可创建、编辑和存储配置文件值，确保托管服务器上的 XML 配置文件正确。可以对使用 DTD 的 XML 文件进行管理，也可以对不使用 DTD 的 XML 文件进行管理。

本章将讨论如何构造 XML 配置模板，以便管理常规（非 DTD）XML 文件以及引用 DTD 的 XML 文件。由于 XML 结构良好， SA 只需要最少的信息就可对基于 XML 的配置文件进行建模和管理。

要管理 XML 配置文件，首先需要为 XML 配置文件创建**模板**文件。在创建模板之后，必须将模板添加到**应用程序配置**对象，以便对托管服务器上的本地配置文件进行管理、编辑和更改。

图 18 XML 配置文件



以下部分将描述简单 XML 文件并显示如何分别为基于非 DTD 的 XML 文件以及基于 DTD 的 XML 文件创建应用程序配置。

另请参见以下示例：

- XML 教程 1 - 创建非 DTD 的 XML 配置模板 （第 89 页）。
- XML 教程 2 - 创建 XML-DTD 配置模板 （第 95 页）

示例：Travel Manager 应用程序和 XML 配置文件

本部分将描述一个 Web 应用程序示例，该应用程序使用简单 XML 文件控制其配置，以及显示如何创建应用程序配置来管理该文件。

Travel Manager 是 Web 应用程序，人们可通过此应用程序执行各种任务来管理旅行，这些任务包括预订酒店、租车、跟踪费用等。Travel Manager 使用 MySQL 关系数据库管理系统 (RDMS)，将其作为存储用户数据和某些应用程序配置数据的库。

由于 Travel Manager 可在多个不同网络上部署，并且每个网络具有不同的数据库服务器，因此灵活提供用于连接 MySQL 服务器的信息十分重要。此应用程序用于从 XML 配置文件 `mysql.xml` 检索连接信息。

借助应用程序配置，可以设置访问本地 MySQL 数据库所需的配置文件值。例如，用于打开数据库连接的用户名和密码针对每个 Travel Manager 应用程序安装可能不同。对配置文件中这些值的修改不需要重新编译 Travel Manager 应用程序代码。

`mysql.xml` 文件中只有四个值是 Travel Manager 用于连接本地 MySQL 数据库所必需的值，每个值以此应用程序的 XML 文件中的元素表示：

- **主机**：已安装 MySQL RDMS 的服务器的主机名。
- **名称**：主机服务器上的数据库名称。
- **用户**：用于打开数据库连接的用户名凭据。
- **密码**：打开数据库连接所需的密码。

Travel Manager mysql.xml 文件内容

以下是 Travel Manager `mysql.xml` 配置文件的一个示例：

```
<?xml version="1.0" ?>
<db-config>
  <db-host>localhost</db-host>
  <db-name>wrightevents</db-name>
  <db-user>root</db-user>
  <db-password>hp-pass</db-password>
</db-config>
```

Travel Manager mysql.xml 基于 DTD 的 XML 文件内容

以下是引用 DTD 的 Travel Manager `mysql.xml` 配置文件的一个示例：

```
<?xml version="1.0"?>
<!DOCTYPE db-config PUBLIC "-//Williams Events//Travel Manager//EN"
"mysql2.dtd">
<db-config>
<db-host>localhost</db-host>
<db-name>wrightevents</db-name>
<db-user>root</db-user>
<db-password>hp-pass</db-password>
</db-config>
```

非 DTD 的 XML 配置模板

可创建基于非 DTD 的 XML 配置模板，将其编写为包含三种所需信息的单个 XML 注释，通过此注释，模板可以从目标 XML 文件提取值并存储它们：

- ACM-NAMESPACE：定义数据库中用于存储从托管服务器的目标 XML 文件中读取的值的位置。此命名空间必须唯一且路径必须以正斜杠 (/) 开头。
- ACM-FILENAME-DEFAULT：定义托管服务器上目标 XML 配置文件的默认绝对路径。
- ACM-FILENAME-KEY：定义命名空间中用于存储目标 XML 配置文件名的位置。

在将配置模板的属性设置为使用 XML 语法时，显示在值集编辑器中的标签与 XML 文件中每个相应元素的标记名称相同。

有关 XML 模板的模板设置的完整列表，请参见 XML 配置模板设置第 87 页。

有关将配置模板的分析器语法设置为 XML 的信息，请参见创建配置模板 （第 17 页）。

---

mysql.xml 的 非 DTD XML 配置模板

以下示例显示了基于 mysql.xml 文件的 XML 配置模板。将此模板文件命名为 mysql.tpl 以表示它是模板文件。

```
<!--
ACM-NAMESPACE = /TravelManager/
ACM-FILENAME-KEY = /files/TravelManager
ACM-FILENAME-DEFAULT = /var/www/html/we/mysql.xml
ACM-TIMEOUT = 1
-->
```

此示例显示 XML 配置模板引用目标 XML 文件 (/var/www/html/we/mysql.xml)，因此应用程序配置分析器可分析此模板，并且可在 SA 库中读取和存储此模板的值。

mysql.tpl 配置模板包含以下所需信息：

- ACM-NAMESPACE：定义数据库中用于存储从托管服务器的 mysql.xml 文件中读取的值的位置。此命名空间必须唯一且路径必须以正斜杠 (/) 开头。
- ACM-FILENAME-DEFAULT：定义托管服务器上 mysql.xml 文件的默认绝对路径。
- ACM-FILENAME-KEY：定义命名空间中将用于存储 mysql.xml 文件名的位置。
- ACM-TIMEOUT：（可选）代表在推送期间要添加到配置模板默认超时值 （十分钟）的分钟数。

  整个应用程序配置的默认超时值为十分钟，加上此应用程序配置中每个配置模板的超时值。因此，如果此模板是某个应用程序配置（超时值为十分钟）中的唯一模板，并且将此值设置为 1，则在推送时整个应用程序配置的整体超时值将为十一分钟。

基于 DTD 的 XML 配置模板

XML-DTD 配置模板实际上只是一个 XML DTD，其中包含注释中定义的某些应用程序配置选项。由于 DTD 标准定义 XML 文件的语法和布局，因此无需使用其他语言重新定义该语法。

对于基于 DTD 的 XML 文件， XML-DTD 配置模板要求具有与常规 XML 文件所需的相同的三个基本特性 — ACM-NAMESPACE、ACM-FILENAME-DEFAULT 和 ACM-FILENAME-KEY – 以及三个其他特性：

- ACM-DOCTYPE：定义 XML 文件中根元素的名称。此根元素遵循在目标 XML 配置文件中找到的开放 <!DOCTYPE 声明。
- ACM-DOCTYPE-SYSTEM-ID：定义托管服务器上关联 DTD 文件的名称。通常，此值可在 XML 配置文件中作为 DOCTYPE 元素中的 SYSTEM 特性找到。
- ACM-DOCTYPE-PUBLIC-ID：定义可表示 XML 文档公用标识符的字符串。通常，此值可在 XML 配置文件中作为 DOCTYPE 元素的 PUBLICID 特性找到。

有关所有 XML 配置文件特性的完整列表，请参见 XML 配置模板设置第 87 页。

mysql.xml 的 XML-DTD 配置模板

以下是为 Travel Manager 基于 DTD 的 XML 文件创建的配置模板示例。

```
<!--
ACM-FILENAME-KEY = /files/TravelManager
ACM-FILENAME-DEFAULT = /var/www/html/we/mysql.xml
ACM-NAMESPACE = /TravelManager/
ACM-TIMEOUT = 1
ACM-DOCTYPE = db-config
ACM-DOCTYPE-SYSTEM-ID = mysql.dtd
ACM-DOCTYPE-PUBLIC-ID = -//Williams Events//Travel Manager//EN
-->
<!ELEMENT db-config (db-host,db-name,db-user,db-password)>
<!ELEMENT db-host (#PCDATA)>
<!ELEMENT db-name (#PCDATA)>
<!ELEMENT db-user (#PCDATA)>
<!ELEMENT db-password (#PCDATA)>
```

在本例中，DOCTYPE 特性引用特定的 XML 和 DTD 信息，这些信息可帮助分析器从 DTD 文件和引用的 XML 文件提取信息。

具体而言，基于 DTD 的 XML 配置模板必须包含以下信息：

- ACM-DOCTYPE：目标 XML 文件的根节点。对于 mysql.xml，根节点为 dbconfig。
- ACM-DOCTYPE-SYSTEM-ID：配置模板的目标 DTD 文件的名称。在 mysql.xml 示例中，使用的 DTD 名为 mysql.dtd。
- ACM-DOCTYPE-SYSTEM-ID：XML 文件的公用 ID。

自定义 XML DTD 元素显示方式

可以向 XML-DTD 配置模板添加两个可选设置，用于支持您自定义目标 XML-DTD 配置文件中的元素在 SA 客户端值集编辑器中的显示方式。通过 ACM-PRINTABLE 和 ACM-DESCRIPTION 可选设置，可以控制元素显示在 SA 客户端客户端中的名称：

- ACM-PRINTABLE：定义在 SA 客户端中显示 XML-DTD 模板时，将在值集编辑器中显示的 XML 文件中每个元素的标签。
- ACM-DESCRIPTION：定义当用户在 SA 客户端的值集编辑器中将鼠标指针悬停在 ACM-PRINTABLE 中定义的字段时显示的鼠标悬停文本。

显式和位置显示设置

可采用两种方式之一为 XML-DTD 配置模板中的特性和元素设置可打印值和描述值：位置方式或显式方式。

- 通过*位置*定义，直接将 ACM-PRINTABLE 和 ACM-DESCRIPTION 插入到它们在 XML-DTD 配置模板中描述的元素或特性之后。
- 通过*显式*定义，可在模板中的任意位置定义 ACM-PRINTABLE 和 ACM-DESCRIPTION。

添加位置自定义显示设置

将元素表和鼠标悬停文本添加到 XML 模板的位置方法是在要定义的元素或特性定义之后立即添加注释，并在该注释中设置 ACM-PRINTABLE 和 ACM-DESCRIPTION 值。换句话说，对于 XML 元素或特性，可为其指定标签并为此标签直接指定鼠标悬停描述。

在以下示例中， mysql.xml 中的每个 XML 元素可在 XML-DTD 模板中的每个元素之后立即定义 ACM-PRINTABLE 和 ACM-DESCRIPTION 设置。

```
<!ELEMENT db-config (db-host,db-name,db-user,db-password)>
<!--
ACM-PRINTABLE = database configuration
ACM-DESCRIPTION = The db-config element specifies the data structure that
contains the information needed to connect to a database.
-->

<!ELEMENT db-host (#PCDATA)>
```

```
<!--
ACM-PRINTABLE = database hostname
ACM-DESCRIPTION = The db-host element specifies the name of the host computer
(the server) on which the database engine is running.
-->

<!ELEMENT db-name (#PCDATA)>
<!--
ACM-PRINTABLE = database name
ACM-DESCRIPTION = The db-name element specifies the name of the database.
-->

<!ELEMENT db-user (#PCDATA)>
<!--
ACM-PRINTABLE = database user
ACM-DESCRIPTION = The db-user element specifies the user identification used
to connect to the database.
-->

<!ELEMENT db-password (#PCDATA)>
<!--
ACM-PRINTABLE = database password
ACM-DESCRIPTION = The db-password element specifies the password used to
connect to the database.
-->
```

添加显式自定义显示设置

通过将设置添加到 XML-DTD 模板的这种显式方法，可以在配置模板中的任何位置定义 ACM-PRINTABLE 和 ACM-DESCRIPTION 值，方法是使用 ACM-ELEMENT 标记指定元素名称以及使用 ACM-ATTRIBUTE 标记指定特性名称 （可选）。

对于这种方法，ACM-ELEMENT 标记是必需的（即使是在定义特性的可打印值和描述值时），因为特性始终与特定元素相关联。

在设置 ACM-ELEMENT 和 ACM-ATTRIBUTE 标记之后，还可以在同一个注释块中设置 ACM-DESCRIPTION 和 ACM-PRINTABLE 标记。针对每个注释块应仅使用一个定义。换句话说，为单个元素定义 ACM-PRINTABLE 和 ACM-DESCRIPTION，然后为下一个元素启动新的注释块。

应在 ACM-PRINTABLE 和 ACM-DESCRIPTION 标记之前定义 ACM-ELEMENT 标记和 ACM-ATTRIBUTE 标记 （如果适用）。

例如，要自定义 mysql.tpl 模板，需按以下方式构造模板：

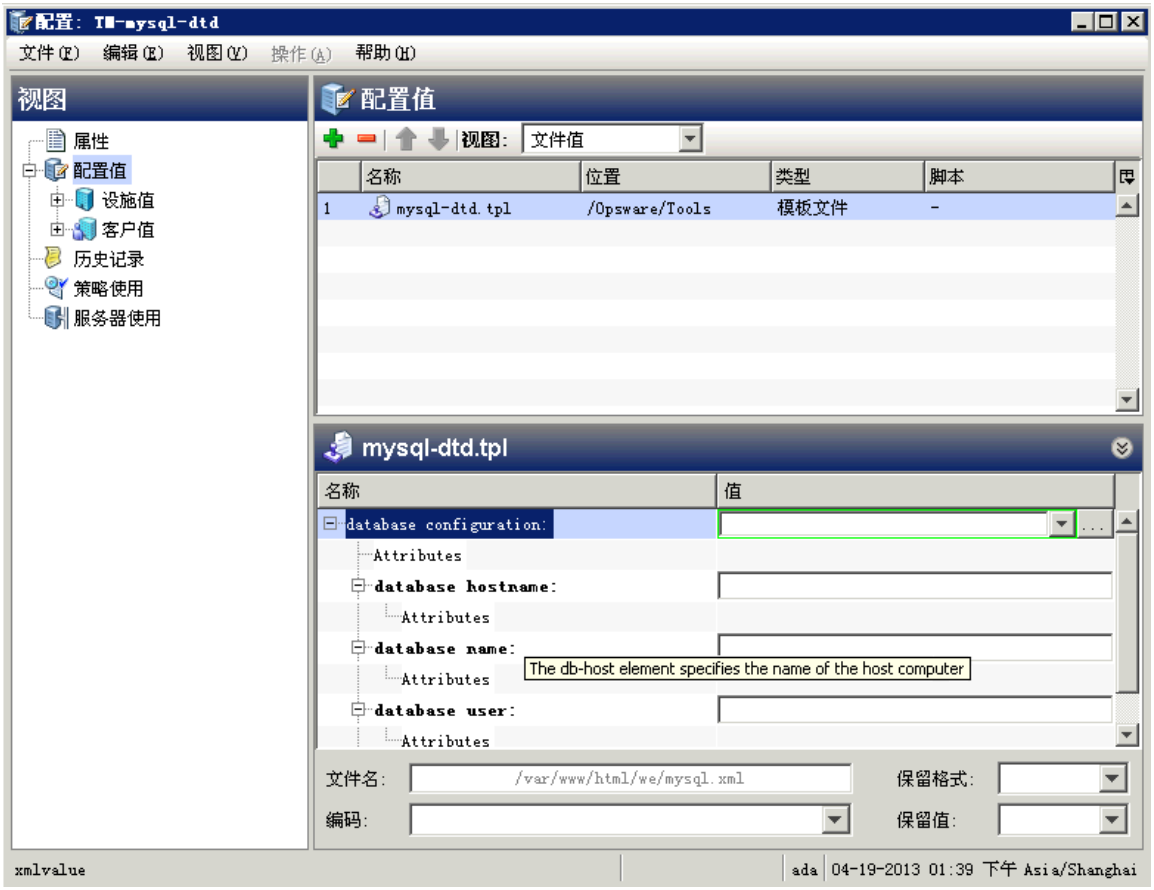
```
<!--
ACM-TIMEOUT = 1
ACM-FILENAME-KEY = /files/TravelManager
ACM-FILENAME-DEFAULT = /var/www/html/we/mysql2.xml
ACM-NAMESPACE = /TravelManager/
ACM-DOCTYPE = db-config
```

```
ACM-DOCTYPE-SYSTEM-ID = mysql.dtd
ACM-DOCTYPE-PUBLIC-ID = -//Williams Events//Travel Manager//EN
-->

<!ELEMENT db-config (db-host,db-name,db-user,db-password)>
<!ELEMENT db-host     (#PCDATA)>
<!ELEMENT db-name     (#PCDATA)>
<!ELEMENT db-user     (#PCDATA)>
<!ELEMENT db-password (#PCDATA)>

<!--
ACM-ELEMENT = db-config
ACM-PRINTABLE = database configuration
ACM-DESCRIPTION = The db-config element specifies the data structure that
contains the information needed to connect to a database.
-->

<!--
ACM-ELEMENT = db-host
ACM-PRINTABLE = database hostname
ACM-DESCRIPTION = The db-host element specifies the name of the host computer
(the server) on which the database engine is running.
-->

<!--
ACM-ELEMENT = db-name
ACM-PRINTABLE = database name
ACM-DESCRIPTION = The db-name element specifies the name of the database.
-->

<!--
ACM-ELEMENT = db-user
ACM-PRINTABLE = database user
ACM-DESCRIPTION = The db-user element specifies the user identification used
to connect to the database.
-->

<!--
ACM-ELEMENT = db-password
ACM-PRINTABLE = database password
ACM-DESCRIPTION = The db-password element specifies the password used to
connect to the database.

-->
```

自定义元素在 SA 客户端中的显示方式

在这两种情况中，无论是采用位置方式还是显式方式添加这些特性，最终的结果都是相同的：值集编辑器在 SA 客户端中显示元素名称 （在 ACM-PRINTABLE 中定义）和鼠标悬浮文本 （在 ACM-DESCRIPTION 中定义），如图 19 中所示。

图 19　自定义元素名称和鼠标悬浮文本



XML 配置模板设置

表 5 描述了在创建常规或基于 DTD 的 XML 配置模板时可用的所有 XML 设置。此列表指示设置是必需还是可选设置，以及是否仅适用于 XML-DTD 模板。

表 5　XML 和 XML-DTD 模板设置

特性	描述
ACM-FILENAME-KEY=<key> 必需；无默认值。	filename-key 确定包含已生成文件的文件名的值集中键的路径。
ACM-FILENAME-DEFAULT=<filename> 必需；无默认值。	filename-default 确定值集中无文件名时返回的默认文件名。
ACM-NAMESPACE=<string> 必需；无默认值。	namespace 确定数据库中存储 XML 元素的位置。
ACM-TIMEOUT=<integer> 可选；（默认值为 0）	timeout 表示添加到应用程序配置总超时的分钟数。 有效超时值是介于 0（含）到 999（含）之间的任意整数。 将应用程序配置中所有配置模板的超时值相加并将所得数字加到配置的默认超时值十分钟，就会得到整个配置的最终超时值。 请注意，应用程序配置中运行时间超过十分钟的任何安装前脚本或安装后脚本都将超时，并且取消整个推送作业。
ACM-DOCTYPE = <string> 必需；无默认值。 仅适用于 XML-DTD 模板。	doctype 表示 XML 文件中根元素的名称。它位于 XML 文件开头的 DOCTYPE 标记中。
ACM-DOCTYPE-SYSTEM-ID = <string> 必需；无默认值。 仅适用于 XML-DTD 模板。	system-id 表示配置模板所基于的 DTD 文件的系统 ID。此值位于 XML 文件开头的 DOCTYPE 标记中。
ACM-DOCTYPE-PUBLIC-ID = <string> 必需；无默认值。 仅适用于 XML-DTD 模板。	public-id 表示使用配置模板分析的 XML 文件的公用 ID。此值位于 XML 文件 DTD 选项开头的 DOCTYPE 标记中。
ACM-ELEMENT=<element name> 可选 仅适用于 XML-DTD 模板。	element 设置当前选项描述的元素。此选项默认设置为 DTD 文件中此部分之前出现的任何元素或特性。

表 5　XML 和 XML-DTD 模板设置 （续）

特性	描述
ACM-ATTRIBUTE=<attribute name> 可选 仅适用于 XML-DTD 模板。	attribute 设置当前选项描述的特性。如果未设置任何特性，则忽略此选项。此特性默认设置为文件中此部分之前出现的任何元素或特性。
ACM-PRINTABLE=<printable> 可选 仅适用于 XML-DTD 模板。	printable 为 SA 客户端中的元素或特性设置的可打印值。此值显示在此字段左侧的值集编辑器中。通常将此值设置为某些简短的描述性内容。
ACM-DESCRIPTION=<description> 可选 仅适用于 XML-DTD 模板。	description 设置要显示在 SA 客户端中的当前元素或特性的描述。将鼠标悬停在值集编辑器中的名称或值字段时，会显示此值。使用此值可描述值集编辑器中字段的用途以及字段的有效值。

5 XML 教程 1 - 创建非 DTD 的 XML 配置模板

本教程将显示如何为非 DTD 的 XML 配置文件创建配置模板。它将向您展示如何使用 XML 语法创建配置模板、将此模板添加到应用程序配置以及将此应用程序配置附加到托管服务器。然后从托管服务器的 mysql.xml 配置文件中导入值、对某些值进行更改以及将新的配置文件推送回托管服务器。

本教程基于示例：Travel Manager 应用程序和 XML 配置文件 （第 79 页）中描述的 Travel Manager 示例应用程序。

非 DTD XML 文件 mysql.xml 示例

以下是 Travel Manager 应用程序的 XML 配置文件的内容：

```
<?xml version="1.0" ?>
<db-config>
  <db-host>localhost</db-host>
  <db-name>wrightevents</db-name>
  <db-user>root</db-user>
  <db-password>hp-pass</db-password>
</db-config>
```

1. 创建 XML 配置模板

使用 SA 客户端基于 mysql.xml 配置文件创建配置模板。

1. 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。
2. 打开 “应用程序配置” 节点，然后打开 “模板” 节点。此操作将显示所有操作系统组。
3. 打开一个操作系统节点并在其中一个操作系统节点下选择特定操作系统。对于本示例，请选择可以安装此应用程序配置的其中一个服务器的操作系统。
4. 在 “操作” 菜单中，选择 “新建”。
5. 在 “属性” 视图中，输入以下信息：
   - **名称**：TM-MySql
   - **描述**：这是 Travel Manager 应用程序的 mysql.xml 配置文件的模板。

- **位置**：可以将 SA 库中的默认位置保留为 /，或选择其他位置用于存储模板文件。请注意，包含此模板的文件夹客户设置必须包括此应用程序配置对象的客户设置。否则，可用模板的列表中将不会包括此模板。有关文件夹设置的详细信息，请参见 《SA 管理指南》中的 “文件夹权限”。
- **版本**：0.1。
- **类型**：模板文件
- **分析器语法**：XML 语法
- **操作系统**：选择可安装此配置模板的所有操作系统。

6 选择 “文件” > “保存”。

7 针对下一个任务保留打开 “模板” 窗口。

2. 添加 XML 设置

由于 XML 配置文件 `mysql.xml` 提供了分析此文件内容所需的大量结构设置，因此 SA 中的 XML 配置模板在 XML 注释中仅需要三种信息：`ACM-NAMESPACE`、 `ACM-FILENAME-KEY` 和 `ACM-FILENAME-DEFAULT`。

1 在导航窗格中选择 “内容” 视图。

2 复制以下 XML 并将其粘贴到内容窗格中：

```
<!--
ACM-NAMESPACE = /TravelManager
ACM-FILENAME-KEY = /files/TravelManager
ACM-FILENAME-DEFAULT = /var/www/html/we/mysql.xml
-->
```

3 选择 “验证” 按钮以确保 XML 有效。

4 选择 “文件” > “保存” 菜单保存模板。

5 选择 “文件” > “关闭” 菜单。

这些 XML 行定义以下内容：

- `ACM-NAMESPACE`：指定每个配置模板所需的唯一命名空间。在本例中，由于已建立 Travel Manager 应用程序的命名空间，因此可重用此根命名空间并附加服务名称。例如：

  `ACM-NAMESPACE = /TravelManager/web/mysql`

- `ACM-FILENAME-KEY`：指定命名空间中用于存储已生成文件的文件名的键的路径。
- `ACM-FILENAME-DEFAULT`：指定目标服务器上用于存储 Travel Manger 应用程序 `mysql.xml` 文件的路径。对于特定服务器或服务器组，可覆盖此路径。

3. 创建要包含此模板的应用程序配置

在此步骤中创建要包含此配置模板的应用程序配置对象。

1 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。

2 打开 “应用程序配置” 节点，然后打开 “配置” 节点。此操作将显示所有操作系统组。

3 打开操作系统节点并选择在先前步骤中用于创建此模板的同一个操作系统。为此应用程序配置指定的 OS 必须是为此模板指定的 OS 的子集。

4 在 “操作” 菜单中，选择 “新建”。

5 在 “文件配置” 屏幕的 “属性” 视图中，指定以下属性：

- **名称**：Tm-MySql-Config
- **描述**：这是 Travel Manager 应用程序的 mySQL 配置文件的应用程序配置。
- **位置**：可以将 SA 库中的默认文件夹位置保留为 /，或选择其他文件夹用于存储此应用程序配置。请注意，包含应用程序配置的文件夹的客户设置必须包括您要向其推送此应用程序配置的托管服务器的客户设置。有关文件夹设置的详细信息，请参见《SA 管理指南》中的 “文件夹权限”。
- **版本**：0.1。
- **操作系统**：选择可安装此应用程序配置的托管服务器中的一个或多个操作系统。

6 在导航窗格中选择 “内容” 视图。

7 选择添加按钮 “+” 或 “操作” > “添加” 菜单添加此模板。

8 在 “选择配置模板” 屏幕中，选择 TM-MySql 模板，然后选择 “确定”。此操作会将此模板添加到此应用程序配置对象。

9 选择 “文件” > “保存” 然后选择 “文件” > “关闭” 。现在已准备好将应用程序配置和其中的配置模板附加到存储此配置文件的服务器。

4. 将应用程序配置附加到托管服务器。

既然已创建应用程序模板和应用程序配置对象，您需要将此应用程序配置附加到已安装 Travel Manager 应用程序的服务器并指定托管服务器上用于存储 `mysql.xml` 文件的路径。

要将应用程序配置附加到服务器，请执行以下操作：

1 在 SA 客户端导航窗格中选择 “设备”，然后选择 “服务器” > “所有托管服务器”。

2 查找可模拟安装 Travel Manager 应用程序配置的服务器。此服务器的操作系统必须与此应用程序配置中指定的其中一个操作系统匹配。

3 选择此服务器，然后在 “操作” 菜单中选择 “打开”。

4 在此服务器屏幕中，选择 “管理策略” 选项卡。

5 在导航窗格中，选择 “配置的应用程序”。此时将显示附加到此服务器的应用程序配置。

6 选择 “安装的配置” 选项卡。

7 在 “操作” 菜单中，选择 “添加配置”。

8 在 “选择应用程序配置” 屏幕中，选择 Tm-MySql-Config 应用程序配置。

9 在 “实例名称” 字段中，输入 “默认 mysql 配置值”。此操作将创建服务器实例级别的值集。有关详细信息，请参见服务器级别的值集编辑器 （第 64 页）。

10 选择 “确定”。此时应用程序配置已附加到服务器。

11 选择 “保存更改” 按钮。针对下一个步骤保留打开服务器屏幕。

以下图 20 显示了附加到服务器的 Tm-MySql-Config 应用程序配置以及 “默认 mysql 配置值” 值集。此值集为服务器实例级别的值集。

图 20 附加到服务器的应用程序配置并突出显示服务器实例级别的值集

请注意，此时如果向其添加应用程序配置的服务器具有多个 `mysql.xml` 配置文件的实例 （由于此服务器正托管此应用程序的多个实例），则可右键单击此配置的 “默认 mysql 配置值” 节点并选择 “复制” 。此操作将创建其他值集，在其中您可以设置指向此应用程序其他实例的文件名路径。有关不同级别值集的详细信息，请参见值集级别和值集继承 （第 54 页）。

5. 为服务器配置应用程序配置设置

既然已将应用程序配置附加到托管服务器，那么需要为此服务器配置它并为配置文件设置值。

要从配置文件导入值 （如下所述），请复制以上非 DTD XML 文件 mysql.xml 示例 （第 89 页）中列出的 XML 并将其粘贴到托管服务器上的 目标文件 `/var/www/html/we/mysql.xml` 中。此操作将促使执行以下导入值步骤。

1 展开 Tm-MySql-Config 节点以显示服务器实例级别的值集。此值集名为“默认 mysql 配置值”。

2 在内容窗格的应用程序配置 “值集编辑器”中配置以下设置：

- **文件名**：托管服务器上的目标 XML 文件的原始路径和文件名显示在“文件名”字段的右侧。此值与模板中定义的 “FILENAME-DEFAULT”的值相同。如果接受服务器的默认路径名，则可将此字段保留为空。如果要将此配置文件放置于此服务器的其他位置，请在 “文件名”字段中为目标服务器上的目标 XML 文件设置正确的路径。
- **编码**：选择托管配置文件的字符编码。默认为托管服务器上使用的编码。（请注意，不支持使用 UTF-16 编码。）
- **保留格式**：如果要保留注释以及尽可能多的保留目标服务器中原始 XML 配置文件的顺序和位置，请选择此选项。 SA 将尽可能多的保留目标文件。有关详细信息，请参见在值集编辑器中设置字段 （第 57 页）。
- **保留值**：当值集中没有提供任何值时，要保留服务器上实际配置文件中包含的值，请针对此选项选择 “是”。将此选项设置为 “是”之后，将使用目标文件的值，除非继承层次结构中任何级别的值覆盖这些值。如果将此选项设置为 “否”并且值集中不存在任何值，则配置文件中将不会放置任何条目。有关详细信息，请参见在值集编辑器中设置字段 （第 57 页）。
- **显示所继承的值**：选择此选项可显示值集中的值和继承级别。在设置为 “否”时，仅显示当前继承级别的值集。在设置为 “是”时，显示值集中的所有值，包括在当前级别设置的值和所继承的值。此视图为只读。

3 在值集编辑器中右键单击并选择“导入值”。导入值时将读取托管服务器上的 XML 文件并将此 XML 文件的内容复制到服务器实例级别的值集。

4 要保存更改，请选择 “保存更改”按钮。针对下一个步骤保留打开服务器屏幕。

6. 编辑值并推送配置

最后步骤是在值集编辑器中编辑值，然后将配置推送到服务器。推送应用程序配置后，值集中的所有值都将替换目标托管服务器上配置文件中的值。应用程序配置中包含的任何脚本都将根据其脚本类型执行。如果目标服务器上不存在配置文件，则在执行推送时会创建此文件。

要编辑值并推送应用程序配置，请执行以下操作：

1. 通过在 “值” 列中进行编辑，可修改值集的一个或多个值。有关列的描述，请参见值集编辑器中列的含义 （第 58 页）。
2. 在为应用程序配置设置值之后，选择 “预览” 可查看服务器上的现有文件以及将被推送到此服务器的文件。
3. 选择 “推送” 以将新的应用程序配置复制到服务器。
4. 在 “推送配置” 屏幕中选择 “启动作业”。检查推送作业的状态和结果。

6 XML 教程 2 - 创建 XML-DTD 配置模板

本章使用 Travel Manager 应用程序作为示例，显示如何创建 XML-DTD 配置模板以管理引用 DTD 的 XML 配置文件。有关 Travel Manager 示例的背景信息，请参见示例：Travel Manager 应用程序和 XML 配置文件 （第 79 页）。

首先将 XML-DTD 模板创建为文本文件，然后将此文本文件导入 SA 库中并将其添加到应用程序配置对象。然后将此应用程序配置附加到托管服务器。最后，编辑此应用程序配置中的某些值并将所做更改推送到托管服务器上的目标 XML 文件。

Travel Manager 基于 DTD 的 XML 文件示例：mysql.xml

对于 Travel Manager 应用程序，以下是 `mysql.xml` XML 配置文件。此文件存储在 `/var/www/html/we/mysql.xml` 中。

```
<?xml version="1.0"?>
<!DOCTYPE db-config PUBLIC "-//Williams Events//Travel Manager//EN"
"mysql.dtd">
<db-config>
<db-host>localhost</db-host>
<db-name>wrightevents</db-name>
<db-user>root</db-user>
<db-password>hp-pass</db-password>
</db-config>
```

Travel Manager XML DTD 文件示例：mysql.dtd

对于 Travel Manager 应用程序，以下是随附的 `mysql.dtd` DTD。此文件存储在 `/var/www/html/we/mysql.dtd` 中。

```
<!ELEMENT db-config (db-host,db-name,db-user,db-password)>
<!ELEMENT db-host     (#PCDATA)>
<!ELEMENT db-name     (#PCDATA)>
<!ELEMENT db-user     (#PCDATA)>
<!ELEMENT db-password (#PCDATA)>
```

1. 在文本编辑器中创建 XML-DTD 模板

在本任务中，将使用文本编辑器创建 XML-DTD 配置模板的源。

要在文本编辑器中创建 XML-DTD 配置模板，请执行以下操作：

1 在文本编辑器中，输入以下信息：

```
<!--
ACM-TIMEOUT = 1
ACM-FILENAME-KEY = /files/TravelManager
ACM-FILENAME-DEFAULT = /var/www/html/we/mysql.xml
ACM-NAMESPACE = /TravelManager/
ACM-DOCTYPE = db-config
ACM-DOCTYPE-SYSTEM-ID = mysql.dtd
ACM-DOCTYPE-PUBLIC-ID = -//Williams Events//Travel Manager//EN
-->
```

此信息为必要信息（除 ACM-TIMEOUT 以外），应用程序配置分析器将使用此信息来读取要管理的 XML-DTD 和 XML 文件：

- `ACM-TIMEOUT`：（可选）代表在推送期间要添加到配置模板默认超时值（十分钟）的分钟数。
- `ACM-FILENAME-KEY`：定义命名空间中将用于存储 `mysql.xml` 文件名的位置。
- `ACM-FILENAME-DEFAULT`：定义托管服务器上 `mysql.xml` 文件的默认位置（绝对路径）。
- `ACM-NAMESPACE`：此值定义数据库中用于存储从托管服务器的 `mysql.xml` 文件中读取的值的位置。此命名空间必须唯一且路径必须以正斜杠 (/) 开头。
- `ACM-DOCTYPE`：定义 XML 文件中根元素的名称。此根元素遵循在目标 XML 配置文件中找到的开放 `<!DOCTYPE` 声明。
- `ACM-DOCTYPE-SYSTEM-ID`：定义托管服务器上关联 DTD 文件的名称。通常，此值可在 XML 配置文件中作为 DOCTYPE 元素中的 SYSTEM 特性找到。
- `ACM-DOCTYPE-PUBLIC-ID`：定义可表示 XML 文档公用标识符的字符串。通常，此值可在 XML 配置文件中作为 DOCTYPE 元素的 PUBLIC-ID 特性找到。

2 保存此文件，并赋予名称 `mysql-dtd.tpl`。针对下一个任务保留打开此文件。

2. 在值集编辑器中为元素描述添加客户设置

在本任务中，将向 XML-DTD 模板文件添加某些额外的信息，以便自定义目标 XML 文件中每个元素在 SA 客户端的值集编辑器中的显示方式。

可以向 XML-DTD 配置模板添加两个可选设置，用于支持您自定义目标 XML-DTD 配置文件中的元素在 SA 客户端值集编辑器中的显示方式：

- ACM-PRINTABLE：定义在 SA 客户端中显示 XML-DTD 模板时，将在值集编辑器中显示的 XML 文件中每个元素的标签。
- ACM-DESCRIPTION：定义当用户在 SA 客户端的值集编辑器中将鼠标指针悬停在 ACM-PRINTABLE 中定义的字段时显示的鼠标悬停文本。

有关这些元素如何显示在 SA 客户端的值集编辑器中的示例，请参见以下图 21。

此示例使用显式方法将这些客户设置放置于 XML-DTD 模板中。有关放置客户设置的方法的详细信息，请参见显式和位置显示设置第 83 页。

要在 XML-DTD 模板中添加客户设置，请执行以下操作：

1 在打开 mysql-dtd.tpl 文件的文本编辑器中，为 DTD 引用的每个 XML 元素添加以下信息。例如，在模板的主要信息后，使用以下三个 ACM 设置标记添加源 XML 文件中包含的每个元素的列表，然后为每个元素创建 XML 注释：

   - ACM-ELEMENT：声明以下 ACM-PRINTABLE 和 ACM-DESCRIPTION 设置将描述的 XML 文件中的元素。此选项默认设置为 DTD 文件中此部分之前出现的任何元素或特性。
   - ACM-PRINTABLE：为元素设置在值集编辑器中显示时的简短描述性标签。
   - ACM-DESCRIPTION：设置元素的鼠标悬停文本。

   XML-DTD 模板文件示例应类似如下：

```
ACM-TIMEOUT = 1
ACM-FILENAME-KEY = /files/TravelManager
ACM-FILENAME-DEFAULT = /var/www/html/we/mysql.xml
ACM-NAMESPACE = /TravelManager/
ACM-DOCTYPE = db-config
ACM-DOCTYPE-SYSTEM-ID = mysql.dtd
ACM-DOCTYPE-PUBLIC-ID = -//Williams Events//Travel Manager//EN
-->
<!ELEMENT db-config (db-host,db-name,db-user,db-password)>
<!ELEMENT db-host     (#PCDATA)>
<!ELEMENT db-name     (#PCDATA)>
<!ELEMENT db-user     (#PCDATA)>
<!ELEMENT db-password (#PCDATA)>
<!--
ACM-ELEMENT = db-config
ACM-PRINTABLE = database configuration
ACM-DESCRIPTION = The db-config element specifies the data structure that
contains the information needed to connect to a database.
-->
<!--
ACM-ELEMENT = db-host
ACM-PRINTABLE = database hostname
```

```
ACM-DESCRIPTION = The db-host element specifies the name of the host
computer (the server) on which the database engine is running.
-->
<!--
ACM-ELEMENT = db-name
ACM-PRINTABLE = database name
ACM-DESCRIPTION = The db-name element specifies the name of the database.
-->
<!--
ACM-ELEMENT = db-user
ACM-PRINTABLE = database user
ACM-DESCRIPTION = The db-user element specifies the user identification
used to connect to the database.
-->
<!--
ACM-ELEMENT = db-password
ACM-PRINTABLE = database password
ACM-DESCRIPTION = The db-password element specifies the password used to
connect to the database.
-->
```

2 保存并关闭文件。

3. 导入 XML-DTD 配置文件

在本任务中，将导入模板文件并创建用于管理目标 XML 和 DTD 文件的新配置模板。

要将 XML-DTD 配置文件导入 SA 库中，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。

2 查找并打开 “应用程序配置” 节点。打开 “模板” 节点。打开操作系统组并导航到应用此模板文件的操作系统。请注意，模板可应用于多个操作系统。例如，可以选择 Red Hat 操作系统的一个版本。

3 在 “操作” 菜单中，选择 “导入模板”。

4 导航到上一个步骤中创建的文件并选择此文件。如果不是默认编码，请设置此编码。

5 选择 “打开”。此操作将导入模板文件并将其显示在 “模板” 屏幕中。

6 选择 “属性” 视图并输入以下信息：

- **名称**：mysql-dtd.tpl
- **描述**：这是 Travel Manager 应用程序的 mysql.dtd (mysql.xml) 文件的模板。
- **位置**：指定 SA 库中用于存储模板的位置。
- **版本**：0.1。
- **类型**：模板文件
- **分析器语法**：XML DTD 语法。
- **操作系统**：选择相应的操作系统。

7 选择 “内容” 视图以显示刚导入的模板文件的内容。

8 选择 “验证” 按钮，以在继续执行操作之前确定语法有效。

9 选择 “文件” > “保存”。

10 选择 “文件” > “关闭”。

4. 创建应用程序配置对象

应用程序配置是容纳配置模板文件的容器。在本步骤中，将创建应用程序配置并导入模板。

1 在 SA 客户端导航窗格中，选择 “库”，然后选择 “按类型” 选项卡。

2 查找并打开 “应用程序配置” 节点。打开 “配置” 节点。打开操作系统组并选择应用此应用程序配置的操作系统。请注意，应用程序配置可应用于多个操作系统。可在后面的步骤中进行更改。

3 选择 “操作” > “新建” 菜单。此时将显示 “配置” 屏幕，可在其中指定应用程序配置的属性和内容。

4 在 “属性” 视图中，指定以下信息：

- **名称**：TM-mysql-dtd
- **描述**：这是 Travel Manager 应用程序的 mysql.xml 和 mysql.dtd 文件的应用程序配置。
- **位置**：指定 SA 库中存储应用程序配置的位置。
- **版本**：0.1
- **操作系统**：选择相应的操作系统。

5 在 “内容” 视图中，选择 “操作” > “添加” 菜单或 “+” 按钮将模板添加到应用程序配置。

6 在 “选择配置模板” 屏幕中，选择 mysql-dtd.tpl 模板文件。

7 选择 “确定”。

8 选择 “文件” > “保存” 保存应用程序配置。

9 选择 “文件” > “关闭”。

5. 将应用程序配置附加到托管服务器

在本任务中，将应用程序配置附加到已安装 Travel Manager 应用程序的服务器，然后输入 `mysql.dtd` 配置文件的路径名。

要将应用程序配置附加到服务器，请执行以下操作：

1 在 SA 客户端导航窗格中，选择 “设备” > “服务器” > “所有托管服务器”。

2 选择服务器，然后在 “操作” 菜单中选择 “打开”。请确保所选服务器的操作系统与应用程序配置和模板上指定的操作系统匹配。

3 选择 “管理策略” 选项卡。

4 选择 “配置的应用程序” 节点。

5 选择 “安装的配置” 选项卡。

6 在 “操作” 菜单中，选择 “添加配置”。

7 在 “选择应用程序配置” 屏幕中，选择应用程序配置 TM-mysql-dtd。

8 在 “实例名称” 字段中，输入 “mysql.xml 的值集 1”。

9 选择 “确定”。此时会将此应用程序配置附加到此服务器。

10 选择 “保存更改” 按钮。

11 针对下一个步骤保留打开应用程序配置屏幕。

6. 从配置文件导入值

下一个步骤是设置值集中的值。虽然可以手动设置值集中的值，但是最简单的方式是从现有配置文件导入值。在本步骤中，将从服务器上的配置文件导入值。

要从配置文件导入值 （如下所述），请复制以上 Travel Manager 基于 DTD 的 XML 文件示例：mysql.xml（第 95 页）中列出的 XML 并将其粘贴到托管服务器上的目标文件 `/var/www/html/we/mysql.xml` 中。复制以上 Travel Manager XML DTD 文件示例：mysql.dtd （第 95 页）中列出的 DTD 并将其粘贴到目标文件 `/var/www/html/we/mysql.dtd` 中。此操作将促使执行以下导入步骤。

---

1 在服务器 “管理策略” 中，打开 “配置的应用程序” 节点。此时将显示附加到此服务器的应用程序配置。

2 打开 “TM-mysql-dtd” 节点。此时将显示此服务器实例值集，这些值集是 “TM-mysql-dtd” 节点下的节点。

3 选择 “mysql.xml 的值集 1” 节点。它是 mysql-dtd.tpl 配置模板的服务器实例值集。

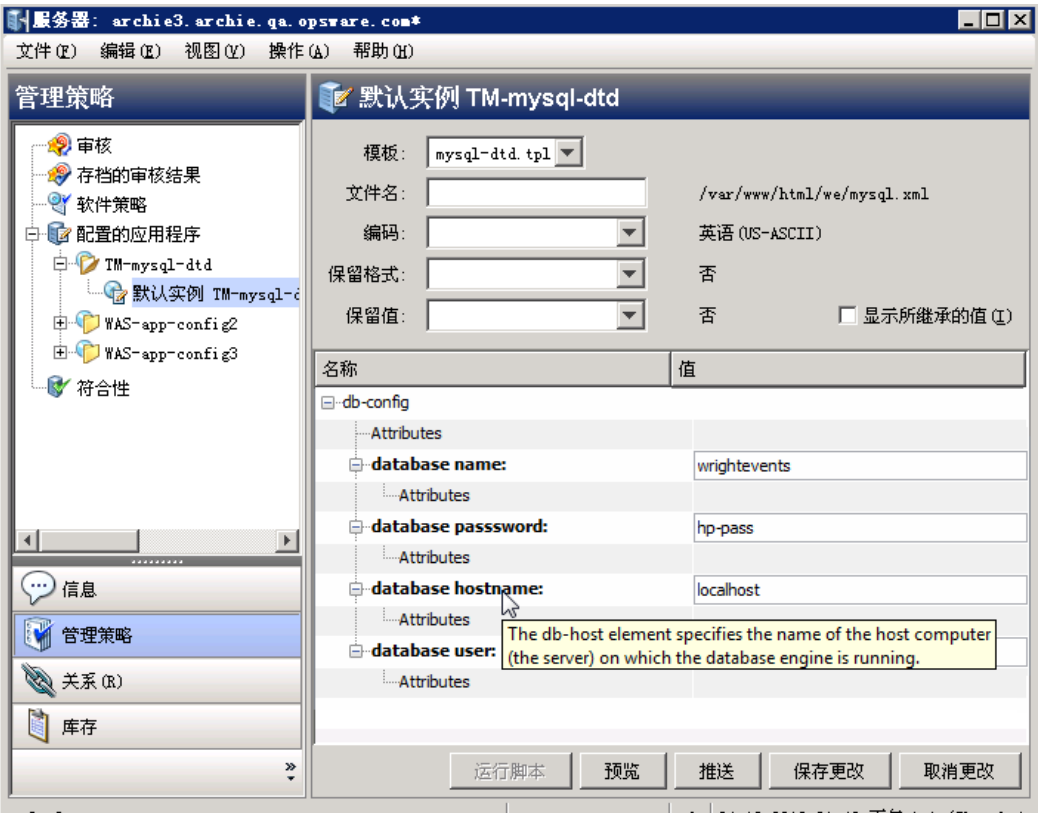
4 右键单击 “值” 列下的任何值，然后选择 “导入值” 菜单。

5 在确认对话框中，选择 “是”。此时会将文件 `/var/www/html/we/mysql.xml` 中的值导入服务器实例级别的值集中。

6 选择 “保存更改” 按钮。

以下图 21 显示了 XML-DTD 模板，其中包含服务器实例值集和从 ACM-DESCRIPTION 元素显示的鼠标悬停文本。

图 21 XML-DTD 配置模板的值集和鼠标悬停文本



7 针对下一个步骤保留显示应用程序配置。

7. 编辑值并推送配置

最后一个步骤是在值集编辑器中编辑值，然后将配置推送到服务器。推送应用程序配置后，值集中的所有值都将替换目标托管服务器上配置文件中的值。此外，还会执行应用程序配置中的所有脚本。如果目标服务器上不存在配置文件，则在执行推送时会创建此文件。

要编辑值并推送应用程序配置，请执行以下操作：

1 确保通过在导航窗格中选择“mysql.xml 的值集 1”节点，显示此服务器实例级别的值集，如图 21 中所示。

2 在 “值” 列下修改密码值。

3 选择 “保存更改” 按钮。

4 选择“预览”按钮，以显示服务器上现有配置文件和将被推送到服务器的配置文件之间的差异。

5 在检查比较屏幕之后，选择 “关闭” 按钮。

6 选择 “推送” 按钮。此时将显示 “推送配置” 向导。

7 选择 “启动作业” 按钮。此时将启动推送操作。

8 检查推送作业的 “作业状态”。选择任意步骤可显示该步骤的详细信息。

9 完成作业后，选择 “关闭” 按钮。

10 您可以登录此服务器并检查 mysql.xml 配置文件，验证服务器上是否已更新此文件。

7 CML 教程 1 - 为简单 Web 应用程序服务器创建应用程序配置

本章演示了如何为运行在两个服务器上的 Web 应用程序服务器设置和管理简单配置文件。每个服务器运行此 Web 应用程序服务器，并需要对它们进行单独配置。此教程显示了如何创建应用程序配置、配置模板、值集以及应用程序配置的两个实例 （每个服务器一个实例）。最后，它将显示如何将应用程序配置推送到每个服务器。

1. 确定要管理的配置文件

Web 应用程序服务器使用一个名为 WASconfig.txt 的配置文件。此文件位于 /opt/WAS/WASconfig.txt 目录中。此文件的内容如下所示：

```
size=1000
dir=/tmp/WAS_001
primary=yes
```

2. 为此配置文件创建模板

可以采用以下两种方式之一创建模板：

- 在文本文件中创建模板并将此文本文件导入 SA 库中。
- 直接在 SA 库中创建模板。

以下描述了这两种方法。请选择一种方法并按照步骤执行操作。

创建模板文件并将其导入 SA 中

1. 在文本编辑器中，将此配置文件导入空的文件：

   ```
   size=1000
   dir=/tmp/WAS_001
   primary=yes
   ```

2. 创建一个模板，该模板使用 CML 对此文件进行建模。首先添加注释块和所需的 CML 元数据，用于定义命名空间和目标配置文件名。

   — 命名空间可定义数据库中将用于存储此模板信息的键。

   — 文件名关键字可定义数据库中将存储默认文件名的位置。

   — 默认文件名可指定将用于生成的配置文件的名称。

```
@#######################################################
# /opt/WAS/WASconfig.txt                              #
# Version 1.0                                         #
# Author <name>                                       #
#######################################################@
@!namespace=/WAS-server-namespace/@
@!filename-key="/WAS-server"@
@!filename-default="/opt/WAS/WASconfig.txt"@
size=1000
dir=/tmp/WAS_01
primary=yes
```

3 然后使用 CML 标记将此配置文件的可变部分更改为变量：

```
@#######################################################
# /opt/WAS/WASconfig.txt                              #
# Version 1.0                                         #
# Author <name>                                       #
#######################################################@
@!namespace=/WAS-server-namespace/@
@!filename-key="/WAS-server"@
@!filename-default="/opt/WAS/WASconfig.txt"@
size=@value_of_size;int@
dir=@value_of_dir;string@
primary=@value_of_primary;boolean@
```

4 使用 “.tpl” 扩展名保存此文件，例如 WASconfig_txt.tpl

5 将此模板文件导入 SA 库中。按照导入和验证模板文件 （第 33 页）中的步骤执行操作。

直接在 SA 中创建模板文件

1 在 SA 客户端中，选择 “库” 选项卡。

2 选择 “按类型” 选项卡。

3 打开 “应用程序配置” 节点和 “模板” 节点。导航到运行此应用程序的 OS 系列和 OS 版本。对于本示例，请选择 Red Hat Enterprise Linux AS 4。

4 选择 “操作” > “新建”。此操作将显示 “模板” 屏幕。

5 输入模板名称 “WASconfig_txt.tpl” 和简要描述。选择 SA 库中用于存储此模板文件的位置。设置版本字符串。将 “类型” 设置为 “模板文件”。将 “分析器语法” 设置为 “CML 语法”。

6 选择 “内容” 视图显示文本编辑器。

7 键入或粘贴 CML 文本。此文本与以上显示的 CML 文本相同。

8 选择 “操作” > “验证” 检查 CML 的语法。进行任何所需的更正。

9 选择 “文件” > “保存” 以保存模板。

10 关闭此模板屏幕。

3. 创建应用程序配置对象

创建要包含此配置模板的应用程序配置对象。

1 在 SA 客户端中，选择 “库” 选项卡。

2 选择 “按类型” 选项卡。

3 打开 “应用程序配置” 节点和 “配置” 节点。导航到运行此应用程序的 OS 系列和 OS 版本。对于本示例，请选择 Red Hat Enterprise Linux AS 4。

4 选择 “操作” > “新建”。此操作将显示 “配置” 屏幕。

5 输入应用程序配置的名称 “WAS-app-config”、简要描述和版本字符串。选择 SA 库中用于存储此应用程序配置的位置。

6 选择 “文件” > “保存” 以保存应用程序配置。

4. 将模板文件添加到应用程序配置对象

1 打开在上一个步骤中创建的 “WAS-app-config” 应用程序配置对象。

2 选择 “配置值” 视图。

3 选择 “+” 按钮或选择 “操作” > “添加”。将显示 “选择配置模板” 屏幕。

4 选择 “WASconfig_txt.tpl” 模板文件并选择 “确定”。

5 选择 “文件” > “保存” 以保存对此应用程序配置对象所做的更改。

6 选择 “文件” > “关闭” 以关闭此应用程序配置对象。

5. 将应用程序配置对象附加到服务器

两个服务器将运行 Web 应用程序服务器，分别是 RHEL001 和 RHEL008。 RHEL001 是主服务器， RHEL008 是辅助服务器。通过将此应用程序配置对象附加到这两个服务器，可创建此应用程序配置的两个实例，如下所示：

1 在 SA 客户端中查找主服务器 RHEL001。

2 选择 RHEL001 服务器，然后选择 “操作” > “打开”。

3 选择 “管理策略” 选项卡。

4 选择 “配置的应用程序” 节点。

5 选择 “操作” > “添加配置” 菜单。

6 选择 “WAS-app-config” 应用程序配置对象。

7 在 “实例名称” 字段中输入 “WAS-app-config 主实例” 并选择 “确定”。

8 选择 “保存更改”。此操作将为服务器 RHEL001 创建此应用程序配置的一个实例。

9 对辅助服务器 RHEL008 重复执行以上步骤，但不在“实例名称”字段中输入“WAS-app-config 辅助实例”，然后选择 “确定”。此操作将为服务器 RHEL008 创建此应用程序配置的第二个实例。

6. 设置默认值

两个服务器的配置文件所需值如下所示：

表 6　运行 Web 应用程序服务器的两个服务器的配置值

服务器：RHEL001	服务器：RHEL008
size=1000 dir=/tmp/WAS_001 primary=yes	size=1000 dir=/tmp/WAS_008 primary=no

您可以为可由单个服务器继承或可被每个服务器覆盖的配置文件设置默认值。如果单个服务器不会覆盖此默认值，则将使用继承的默认值。

下表显示哪些值将被设置为默认值以及哪些值将由单个服务器设置：

表 7　针对 Web 应用程序服务器的配置文件的应用程序级别默认值

默认值	描述
size=1000	将此值设置为应用程序级别默认值 1000。附加到此应用程序配置的所有服务器都将使用此值，除非这些服务器覆盖此值。
dir	不会将此值设置为默认值。每个服务器将在服务器级别或服务器实例级别设置此值。
primary=no	将此值设置为应用程序级别默认值 “no” 。附加到此应用程序配置的所有服务器都将使用此值，除非这些服务器覆盖此值。

设置应用程序级别默认值

要设置应用程序级别默认值，请执行以下操作：

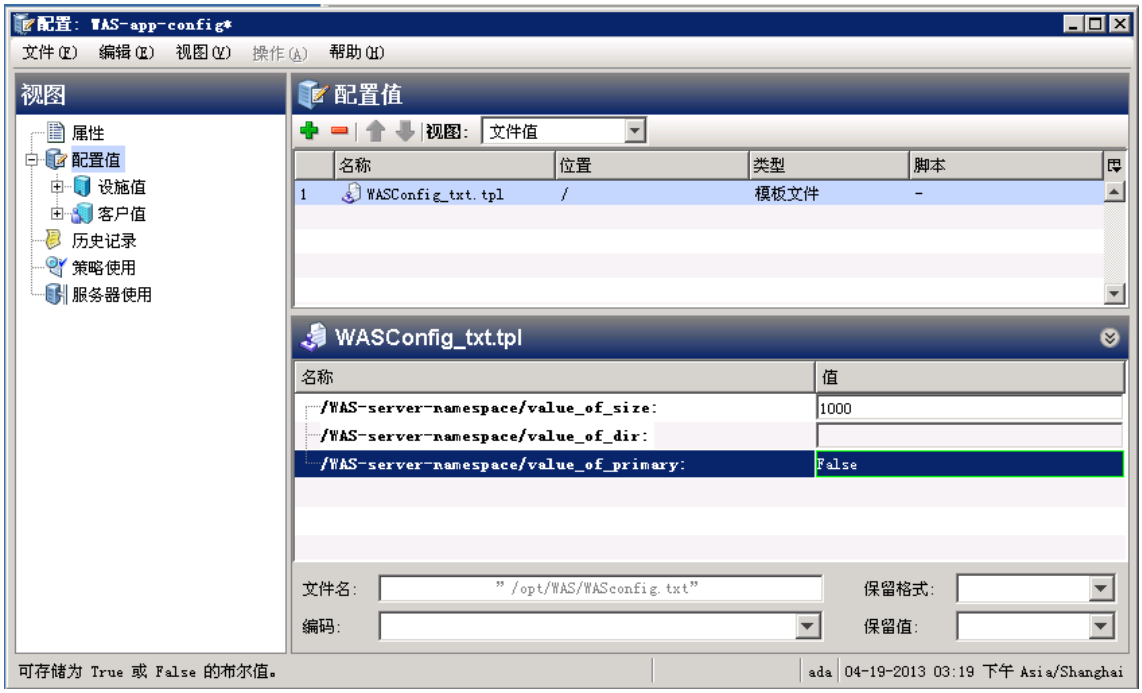
1 打开以上创建的应用程序配置对象。

2 选择 “配置值”视图。

3 选择 WAS-config-template.tpl 模板文件。

4 在视图下拉列表中选择 “文件值”。此操作将显示此模板的应用程序级别默认值。

5 将“value_of_size”设为 1000 并将“value_of_primary”设为“False”（注意大小写），如下所示。请勿设置“value_of_dir”的默认值，因为每个服务器将需要设置此值。



6 选择“文件”>“保存”以保存**应用程序级别默认值**。

7 选择“文件”>“关闭”。

设置 RHEL001 的服务器级别默认值

服务器 RHEL001 需要在服务器级别设置 `dir=/tmp/WAS_001` 和 `primary=yes`。它不需要设置大小，因为它可以使用应用程序级别的值集。

要为主服务器 RHEL001 设置服务器级别默认值，请执行以下操作：

1 在 SA 客户端中查找 RHEL001 服务器。

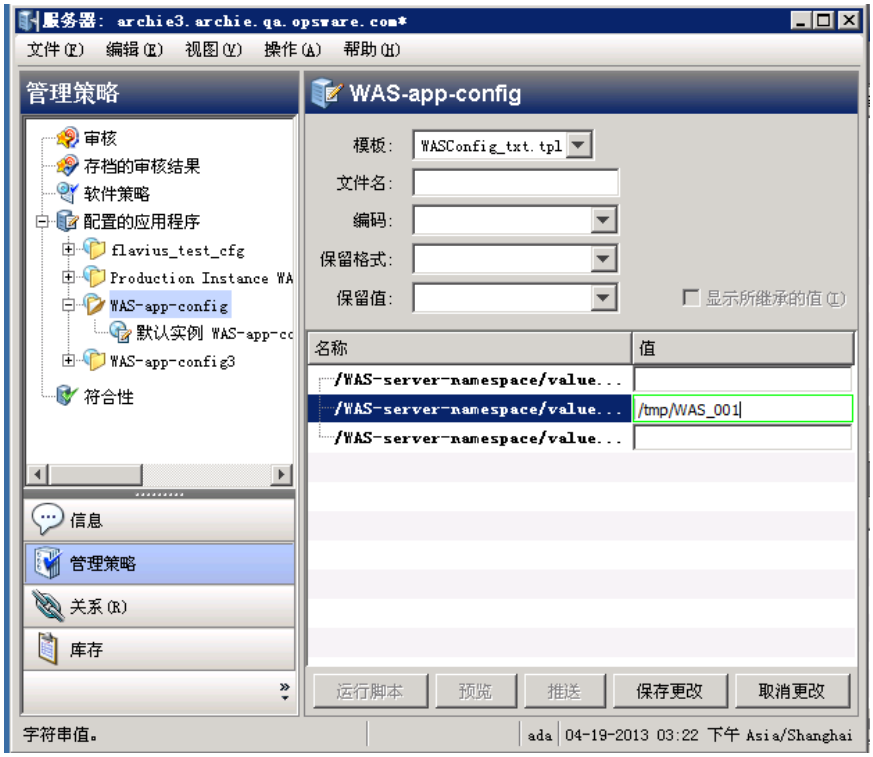
2 选择 RHEL001 服务器，然后选择“操作”>“打开”。

3 选择“管理策略”选项卡。

4 打开“配置的应用程序”节点以显示“WAS-app-config”应用程序配置对象。

5 选择附加到此服务器的“WAS-app-config”应用程序配置对象。此操作将显示服务器级别的默认值集。服务器级别的值集将应用于服务器上应用程序配置的所有实例，除非被服务器实例级别所覆盖。

6 将 “value_of_dir” 的服务器级别默认值设置为 “/tmp/WAS_001”，如下所示。请勿设置 “value_of_size” 或 “value_of_primary” 的默认值，因为这些值将从应用程序级别继承。



7 选择 “保存更改” 按钮，或 “文件” > “保存” 菜单，以保存**服务器级别默认值。**

8 打开 WAS-app-config 节点以显示应用程序配置实例 “WAS-app-config 主实例”。

9 选择实例 “WAS-app-config 主实例”。此操作将显示实例级别默认值。这是最低级别的值集，将覆盖所有其他级别。请注意，尚未定义任何实例级别的值。

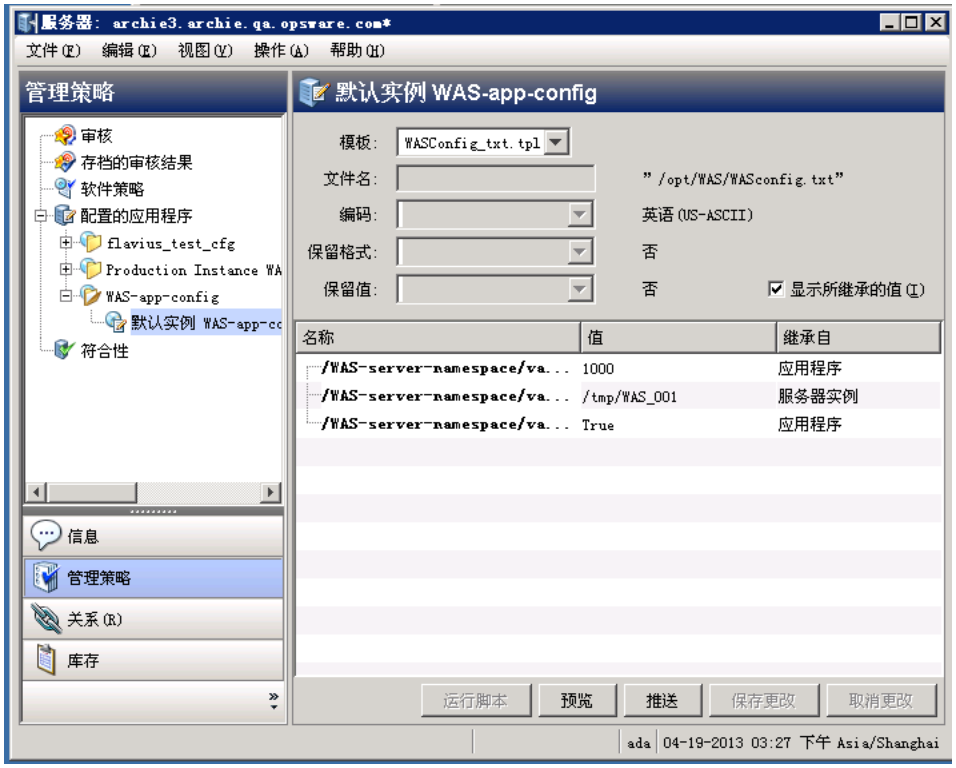
10 选择 “显示所继承的值” 可显示将从应用程序级别默认值和服务器级别默认值继承的值。请注意，“value_of_size” 和 “value_of_primary” 从应用程序级别继承，“value_of_dir” 从服务器级别继承。

11 取消选中 “显示所继承的值”，以便可以设置实例级别默认值。

12 将 “value_of_primary” 设为 “True”（注意大小写）。

13 选择 “保存更改” 按钮，或 “文件” > “保存” 菜单，以保存**实例级别默认值。**

14 再次选择 “显示所继承的值” 可显示从应用程序级别、服务器级别和实例级别继承的值。请注意，“value_of_size” 从应用程序级别继承，“value_of_dir” 从服务器级别继承，“value_of_primary” 从实例级别继承，如下所示。



设置 RHEL008 的服务器级别默认值

服务器 RHEL001 需要在服务器级别设置 ddir=/tmp/WAS_008 它不需要设置大小或主，因为它可以使用应用程序级别的值集。

要为辅助服务器 RHEL008 设置服务器级别默认值，请执行以下操作：

1 在 SA 客户端中查找 RHEL008 服务器。

2 选择 RHEL008 服务器，然后选择 “操作” > “打开”。

3 选择 “管理策略” 选项卡。

4 打开 “配置的应用程序” 节点以显示 “WAS-app-config” 应用程序配置对象。

5 选择附加到此服务器的 “WAS-app-config” 应用程序配置对象。此操作将显示服务器级别的默认值集。服务器级别的值集将应用于服务器上应用程序配置的所有实例，除非被服务器实例级别所覆盖。

6 将 “value_of_dir” 的服务器级别默认值设置为 “/tmp/WAS_008”。请勿设置 “value_of_size” 或 “value_of_primary” 的默认值，因为这些值将从应用程序级别继承。

7 选择 “保存更改” 按钮，或 “文件” > “保存” 菜单，以保存**服务器级别默认值。**

8 打开 WAS-app-config 节点以显示应用程序配置实例 “WAS-app-config 辅助实例”。

9 选择实例 “WAS-app-config 辅助实例” 。此操作将显示**实例级别默认值**。这是最低级别的值集，将覆盖所有其他级别。请注意，尚未定义任何实例级别的值。

10 选择 “显示所继承的值”可显示将从应用程序级别默认值和服务器级别默认值继承的值。请注意，“value_of_size”和 “value_of_primary”从应用程序级别继承，“value_of_dir”从服务器级别继承。

7. 将实际配置文件与配置模板进行比较

通过在服务器屏幕中选择 “预览”按钮，可以选择将应用程序配置中指定的值与服务器上配置文件中的实际值进行比较。以下显示了 RHEL001 服务器上尚无配置文件时服务器上的比较：

以下显示了服务器上现有配置文件不同于应用程序配置中指定值时的比较：

8. 将配置更改推送到服务器

要将配置更改推送到服务器，请执行以下操作：

1 在 SA 客户端中查找 RHEL001 服务器。

2 选择 RHEL001 服务器，然后选择 “操作” > “打开”。

3 选择 “管理策略” 选项卡。

4 打开 “配置的应用程序” 节点以显示 “WAS-app-config” 应用程序配置对象。

5 打开 “WAS-app-config” 应用程序配置节点以显示实例 “WAS-app-config 主实例”。

6 选择实例 “WAS-app-config 主实例”。

7 选择 “推送” 按钮。

8 可以选择 “启动作业” 按钮以接受计划和通知的作业默认值，也可以选择 “下一步”。

9 在 “计划” 屏幕中，可指定运行推送配置作业的时间。选择 “下一步”。

10 在 “通知” 屏幕中，可指定作业成功或失败时要接收电子邮件消息的一个或多个人员。此外，还可以指定工单标识符。选择 “下一步”。

11 选择 “启动作业”。SA 从此模板和值集生成配置文件，然后将生成的配置文件推送到服务器并显示结果。

12 选择 “关闭”。

有关显示更加复杂的配置文件的教程，请参见 CML 教程 2 - 创建 Web 服务器配置文件的模板（第 113 页）。

8 CML 教程 2 - 创建 Web 服务器配置文件的模板

本教程将说明如何使用配置建模语言 (CML) 基于名为 UrlScan.ini 的 Microsoft Internet 信息服务 (IIS) Web 服务器配置文件创建配置模板。您将使用 CML 语言基于此文件创建模板文件，以便在托管服务器上管理此模板文件。

虽然本教程不会为您提供有关 CML 的所有信息，但是从 UrlScan.ini 创建 CML 模板将帮助您基本了解 CML 以及从配置文件创建配置模板的过程。

UrlScan.ini 文件示例 （第 135 页）中列出了 UrlScan.ini 文件的示例。完整 url_scan_ini.tpl CML 模板 （第 139 页）中列出了一个完整的 CML 文件。

要完成本教程，您应具备以下条件：

- UrlScan.ini 的文档。此文档随 Microsoft IIS 文档一起提供。
- UrlScan.ini 文件。
- 用于创建 CML 文件的文本编辑器。

1. 分析本地配置文件和文档

一旦确定要管理的应用程序配置文件之后，首先应分析此本地配置文件及其文档。确保了解此配置文件的用途、文件中的所有元素以及此配置文件所管理的数据种类。

本教程将使用 UrlScan.ini 文件。 UrlScan.ini 文件示例 （第 135 页）中列出了一个示例。通过 UrlScan.ini 文件，系统管理员可以配置 Microsoft Internet 信息服务 (IIS) Web 服务器。UrlScan.ini 文件由多个部分组成，例如 [Options]、 [AllowVerbs]、 [DenyVerbs]、 [DenyHeaders]、 [AllowExtensions] 和 [DenyExtensions]。 IIS 管理员可以为每个部分设置不同的配置，以允许或禁止 IIS 服务器上某些种类的 HTTP 请求。无需按任何特定顺序排列这些部分。但是必须有序组织每个部分中的信息。例如， [AllowVerbs] 部分必须后跟被允许访问网站的特定 HTTP 请求。

UrlScan.ini 包含字符串列表 （例如动词列表和文件扩展名）和选项 （采用布尔值 “1” 表示 True 或采用 “2” 表示 False）。

2. 创建 CML 注释块

CML 模板是一个名称带有 .tpl 文件扩展名的简单文本文件。使用文本编辑器创建名为 Url_Scan_ini.tpl 的新文本文件。 .tpl 扩展名是 SA 针对 CML 模板使用的典型 （但可选）文件扩展名。

使用有关此模板的信息在文件顶部创建 CML 注释块，如下所示：

```
@##############################################
# \system32\inetsrv\urlscan.ini (Windows)    #
# Version 1.0                                #
# Joe Author (joe_author@your_company.com)   #
##############################################@
```

CML 注释标记使用以下语法：

```
@# <one line comment>
```

或

```
@## <comments spanning multiple lines>
    <comments spanning multiple lines> #@
```

3. 创建 CML 设置指令

设置部分将指示分析器如何解释 CML 文件。所有 CML 文件都需要 namespace、 filename-key 和 filename-default 指令。以下显示的其他指令为可选指令。这些指令定义了空格处理、布尔值、注释格式和排序规则。

要创建基本设置部分，请在 CML 模板中的注释块后输入以下信息：

```
@!namespace=/security/@
@!filename-key="/test";filename-default="/c/UrlScan.ini"@
@!optional-whitespace@
@!boolean-yes-format="1";boolean-no-format="0"@
@!line-comment-is-semicolon@
@!unordered-lines@
```

请注意，每个 CML 指令标记以字符 “@!” 开头，以字符 “@” 结尾。

下行将两个 CML 指令合并在一行中：

```
@!filename-key="/test";filename-default="/c/UrlScan.ini"@
```

此行还可写为两个单独的行，如下所示：

```
@!filename-key="/test"@
@!filename-default="/c/UrlScan.ini"@
```

设置指令

表 8 说明了每个设置指令

表 8　CML 模板设置指令

CML 标记	描述
@!namespace=/security/@	@!namespace 指令定义此 CML 模板将使用的命名空间。它定义数据库中将存储此 CML 模板使用的值的位置。每个 CML 模板应使用其各自的命名空间，以便名称不与其他模板相冲突。 在本例中，命名空间为 /security。所有值集都将存储在此命名空间中。

表 8　CML 模板设置指令 （续）

CML 标记	描述
@!filename-key="/files/urlscan_ini";filename-default="/c/urlscan.ini"@	@!filename-key 指令定义命名空间中将存储文件名的位置。如果值以 “/” 开头，则它将定义单独的命名空间。如果值不以 “/” 开头，则它将附加到由 @!namespace 指令定义的命名空间。 在本例中，默认文件名将存储在数据库的 “/file/urlscan-ini” 命名空间下。 @!filename-default 指令定义服务器上将用于保存本地配置文件的默认路径。可使用 SA 客户端更改此路径。 在本例中，将配置文件推送到托管服务器之后，文件将被放置于 /c/urlscan.ini 中。 请注意，路径名只能使用正斜杠。
@!optional-whitespace@	此指令指示在配置文件的项之间空格是可选的。例如，如果设置此选项，则以下任一条目均将有效： Key = "value" Key="value"
@!boolean-yes-format="1";boolean-no-format="0"@	此指令定义配置文件中允许的布尔值。在此情况中， true 由字符 1 表示， false 由字符 0 表示。禁止使用任何其他布尔值。
@!line-comment-is-semicolon@	指示分析器忽略配置文件中分号后的任何内容。这允许在本地配置文件中的每个注释前使用分号进行注释。
@!unordered-lines@	指示分析器配置文件中的部分可以按任何顺序排列。如果使用 ordered-lines，则配置文件必须与模板的顺序保持一致。

4. 定义 [Options] 部分 — 打开块

现在已准备好将 CML 指令添加到模板。您将在 CML 中建模的 UrlScan.ini 文件的第一个部分是 [Options] 部分，它包含配置文件的多个选项。

在 CML 中，如果配置文件中的信息部分具有多种数据（需 CML 分析器采用不同方式读取的数据），则可打开 “块” 来单独处理每个信息部分。通常，可以在 CML 中打开一个块，为 CML 文件的某个部分定义特殊的分析器规则。[Options] 部分具有两个信息 “块”：此部分的标题，即 “[Options]”，以及该部分中的所有选项。由于这些块相互从属，因此您将在不同的级别设置它们，在级别 1 设置第一个块 （此部分的标题），在级别 2 设置第二个块 （此部分的内容）。采用这种方式嵌套块可以在分析器读取块时将各个部分一起保持在块中。

1 要定义 [Options] 部分，请输入以下行：

```
@1[;optional;ordered-lines@
[Options]
@2[;unordered-lines@
```

2 在 UrlScan.ini 文件中， [Options] 部分包含一个键值对列表。在两个级别使用块标记 ([) 集，因为此部分中存在两种数据：标题和键值对列表。第一个级别块处理文本字符串 “[Options]”，而第二个级别块处理该部分中所有的键值对。

表 9 说明了如何打开 [Options] 部分的两个块级别。

表 9 标记 [Options] 部分的起始

CML 标记	描述

`@1[;optional;ordered-lines@`	数字 1 设置多行块的第一个级别。
	`[`
	方括号打开一个新块。
	`optional`
	表示整个块在配置文件中可选。
	`ordered-lines`
	表示此标记（字符串 [Options]）之后的任何内容必须在本地 UrlScan.ini 配置文件中首先出现。换句话说，标题 [Options] 必须显示在实际选项之前。

`[Options]`	用于在本地配置文件中命名此部分的字符串。此字符串将显示在此配置文件中。

表 9　标记 [Options] 部分的起始 （续）

CML 标记	描述
@2[;unordered-lines@	数字 2 设置此块的第二个级别。 [方括号打开一个新块，在这种情况中，级别 2 块嵌套于前一个级别 1 块。 unordered lines 表示块中 [Options] 后的行可以在配置文件中以任何顺序排列。即，[Options] 部分中的所有键值对可以按任何顺序排列。

3　下一步将定义可显示在配置文件的 [Options] 部分中的所有选项。大部分条目都使用 CML 替换标记，因为它们是允许您替换单个值的简单键值对。表 10 说明了针对每个选项的 CML。

表 10　从 UrlScan.ini [Options] 部分标记键值对

CML 标记	描述
AllowDotInPath = @allow_dot_in_path;boolean@	注意：所有键值对均使用以下语法的某些变体 （除非另有说明）： string literal = @source;type@ 此字符串文本定义配置文件中将显示的实际选项名称。源是数据库值集中将用于存储值的位置。类型是将存储在值集中的数据类型。 @allow_dot_in_path 此字符串定义用于存储此值的命名空间路径。在本例中，此命名空间是相对的，这意味着它将附加到在模板标题中定义的命名空间 (@!namespace=/security/@)，并且将在该命名空间位置中存储此值。即，此值将存储在数据库的 /security/allow_dot_in_path 键中。 您还可以按以下方式编写此标记： AllowDotInPath = @/security/allow_dot_in_path;boolean@ boolean 由于键值对类型为 Boolean，因此将使用 CML 类型 boolean。请注意，由于此模板的标题将布尔 true 值定义为 1，因此当 IIS 管理员设置此值集时，需要输入 1 以允许在 IIS 服务器的路径中使用点。

表 10　从 **UrlScan.ini [Options]** 部分标记键值对 （续）

CML 标记	描述
`AllowHighBitCharacters = @allow_high_bit_characters;boolean@`	与上一个示例相似， AllowHighBitCharacters 是显示在配置文件中的选项，allow_high_bit_characters 是相对命名空间路径， boolean 是数据类型。 此 IIS 选项允许用户选择 URL 中是否接受高位字符，在配置文件中使用 1 表示 true， 0 表示 false。
`AllowLateScanning = @allow_late_scanning;boolean@`	允许 IIS 管理员选择是否接受对 URL 进行稍后扫描。定义用于存储此值的命名空间位置。 `boolean` 表示在配置文件中此键可以是 1 用于表示 true， 0 表示 false。
`AlternateServerName = @alternate_servername@`	定义在用户输入或从配置文件读取备用服务器名时可存储此名称的命名空间位置。由于未指定任何类型，因此默认为字符串类型。
`EnableLogging = @enable_logging;boolean@`	允许用户启用记录，在配置文件中使用 1 表示 true，0 表示 false。
`LoggingDirectory = @logging_directory;dir@`	在已启用记录之后，允许用户选择用于存储日志文件的目录。类型 `dir` 表示目录。
`LogLongURLs = @log_long_urls;boolean@`	允许用户选择是否记录访问服务器的 URL，在配置文件中使用 1 表示 true， 0 表示 false。
`NormalizeUrlBeforeScan = @normalize_url_before_scan;boolean@`	允许用户选择是否在服务器读取 URL 之前标准化此 URL，在配置文件中使用 1 表示 true， 0 表示 false。
`PerDayLogging = @per_day_logging;boolean@`	允许用户选择启用每日记录，在配置文件中使用 1 表示 true， 0 表示 false。
`PerProcessLogging = @per_process_logging;boolean@`	允许用户启用或禁用每个进程的记录，在配置文件中使用 1 表示 true， 0 表示 false。

表 10　从 UrlScan.ini [Options] 部分标记键值对 （续）

CML 标记	描述
RejectResponseUrl = @reject_response_url;string;r”(HTTP_URLSCAN_STATUS_HEADER)|(HTTP_URLSCAN_ORIGINAL_VERB)|(HTTP_URLSCAN_ORIGINAL_URL)”;optional@	语法： string literal = @source;type;r”regular expression”;option@ reject response 一个字符串文本，用于定义命名空间中将存储字符串的路径。 string 表示拒绝 URL 请求的数据类型为字符串。 r” 这是一个字符串范围说明符，用于引入正则表达式。在这种情况中，它是字符串文本的范围。 (HTTP_URLSCAN_STATUS_HEADER)|(HTTP_URLSCAN_ORIGINAL_VERB)|(HTTP_URLSCAN_ORIGINAL_URL)” 由分析器读取的字符串文本 （已拒绝的 URL 响应）：状态标题、原始动词和原始 URL。 optional 表示值为可选值。即， UrlScan.ini 文件中可忽略 RejectResponseUrl 选项。
RemoveServerHeader = @remove_server_header;boolean@	允许用户启用或禁用 RemoveServerHeading 功能。激活 （设置为 1）后，发送到客户端的拒绝响应将删除消息中的服务器标题。此设置在配置文件中使用 1 表示 true，0 表示 false。
UseAllowVerbs = @use_allow_verbs;boolean@	允许用户启用或禁用 UseAllowVerbs 功能。激活 （设置为 1）后，对于包含未明确列于 UrlScan.ini 文件 AllowVerbs 部分中的 HTTP 动词的任何请求，服务器将拒绝发送到此服务器的这类请求。在配置文件中使用 1 表示 true， 0 表示 false。
UseAllowExtensions = @use_allow_extensions;boolean@	允许用户启用或禁用 UseAllowExtension 功能。激活（设置为 1）后，对于包含未明确列于 UrlScan.ini 文件 AllowExtension 部分中的文件扩展名的任何请求，服务器将拒绝发送到此服务器的这类请求。在配置文件中使用 1 表示 true， 0 表示 false。

表 10　从 **UrlScan.ini [Options]** 部分标记键值对 （续）

CML 标记	描述
`UseFastPathReject = @use_fast_path_reject;boolean@`	允许用户启用或禁用 UseFastPathReject 功能。激活（设置为 1）后，服务器将忽略 RejectResponseUrl 选项，并在 URL 被拒绝后向客户端返回一个简短的 404 响应。在配置文件中使用 1 表示 true， 0 表示 false。
`VerifyNormalization = @verify_normalization;boolean@`	允许用户启用或禁用标准化由 UrlScan.ini 扫描的所有 URL。激活 （设置为 1）后，将在扫描 URL 之前标准化此 URL。在配置文件中使用 1 表示 true，0 表示 false。

5. 定义 [AllowExtensions] 部分 - 通过打开新块关闭块

既然已定义 UrlScan.ini 文件的 [Options] 部分中的所有选项，那么您已准备好开始定义下一个部分 [AllowExtensions]。请记住，开始 [Options] 部分之前必须打开两个级别的块以说明两个级别的信息 — [Options] 部分的标题及其内容。

在开始定义 [AllowExtensions] 部分之前，需要通过关闭 CML 块来关闭先前的部分。借助 CML，您可以明确关闭带有 “]” 标记的块，或者也可以通过在较高级别 （使用较低的数字指定）或同等级别打开新块来关闭此块。在本任务中，将采用打开 [Options] 部分的块的相同方式打开 [Allow Extensions] 的新块，方法是启动新的级别 1 块。此操作将自动关闭由 [Options] 部分打开的块。

要打开新块并定义 [AllowExtensions] 部分，请执行以下操作：

1 在 [Options] 部分的最后一行后输入以下内容，打开 [AllowExtensions] 部分的新块：

```
@1[;optional;ordered-lines@
[AllowExtensions]
@2[;unordered-lines@
```

表 11 说明了如何打开新的两个级别块来关闭先前的块。

表 11 启动 [AllowExtensions] 部分的新块

CML 标记	描述

`@1[;optional;ordered-lines@`	数字 1 打开新的级别 1 块。由于它是级别 1 块，级别比先前块更高 （[Options] 部分中键值对的级别 2 块），与之前的级别 1 块相同，因此它将关闭之前出现的两个块。
	请注意，可通过使用关闭块标记明确关闭块。例如：
	`@2]@`
	`[`
	用于打开新块的 CML 块标记。
	`optional`
	表示整个块为可选块，不必一定出现在配置文件中。
	`ordered-lines`
	表示此标记 （字符串 [AllowExtensions]）之后的任何内容必须在本地 UrlScan.ini 配置文件中首先出现。换句话说，您不能首先列出本地文件中的所有选项，然后列出标题。 [AllowExtensions] 必须首先出现。在 CML 中，ordered-line 元素确定此顺序。

表 11　启动 [AllowExtensions] 部分的新块 （续）

CML 标记	描述
[AllowExtensions]	用于在本地配置文件中命名此部分的文本字符串。
@2[;unordered-lines@	数字 2 设置此块的第二个级别。 [用于打开新块的 CML 块符号。 unordered lines 表示块中 [AllowExtensions] 后的所有行可以在配置文件中以任何顺序排列。即，[AllowExtensions] 部分中的所有键值对可以按任何顺序排列。

2　接下来，由于 UrlScan.ini 文件的 [AllowExtensions] 部分可以包含用户输入的任何文件扩展名列表，因此要使用 CML 循环和循环目标标记，指示分析器每次读取一行此部分中的信息。在最后一个步骤中的最后文本 @2[;unordered-lines@ 之后，立即输入以下文本：

```
@*allow_extension;unordered-string-set@
.@.@
```

表 12 说明了循环和循环目标 CML 标记的工作方式：

表 12　循环和循环目标 CML 标记

CML 标记	描述
@*allow_extension;unordered-string-set@	语法 @<level><tag type><name>;<data type>;<options>@ 循环标记 (*) 将循环并读取 [AllowExtensions] 部分中列出的无序字符串集。 allow_extension 一个字符串，用于定义命名空间中将存储字符串的路径。 unordered-string-set 表示字符串列表不必以任何特定顺序排列。

表 12　循环和循环目标 CML 标记 （续）

CML 标记	描述
.@.@	第一个 (.) 在此部分中，分析器读取的无序字符串集是 [AllowExtensions] 部分中列出的以 (.) 字符开头的文件扩展名的列表。 @.@ 循环目标标记 (.) 指示分析器读取此列表中以句点字符开头的任何内容。

3　保存此文件。

6. 定义 [DenyExtensions] 部分

接下来采用定义 [AllowExtensions] 部分的相同方式定义 UrlScan.ini 文件的 [DenyExtensions] 部分。将打开一个新的级别 1 块，此操作将关闭 [AllowExtensions] 部分中的先前块。然后将打开一个级别 2 块，在此块中，可以指示分析器读取要使用 UrlScan.ini 阻止的以 (.) 开头的所有文件扩展名的无序列表。

[DenyExtensions] 部分的 CML 类似如下：

```
@1[;optional;ordered-lines@
[DenyExtensions]
@2[;unordered-lines@
@*deny_extension;unordered-string-set@
.@.@
```

7. 定义 [AllowVerbs] 和 [DenyVerbs] 部分

UrlScan.ini 文件中接下来的两个部分遵循先前部分中用于 [DenyExtensions] 的相同 CML。打开第一个级别块将关闭先前块，还会将以下文本分析为有序行。

然后打开第二个级别块，此块可读取以下无序字符串列表 — 即动词列表。在这两个部分中，此字符串将指示分析器读取被允许访问网站的动词列表以及被拒绝访问网站的动词列表。

这两个部分的 CML 如下所示：

```
@1[;optional;ordered-lines@
[AllowVerbs]
@2[;unordered-lines@
@*allow_verb;unordered-string-set@
@.@

@1[;optional;ordered-lines@
[DenyVerbs]
@2[;unordered-lines@
@*deny_verb;unordered-string-set@
@.@
```

8. 定义 [DenyHeaders] 部分

接下来需定义 UrlScan.ini 文件的 [DenyHeaders] 部分，此部分允许您将 IIS 配置为拒绝特定 HTTP 请求标题。

此部分与先前部分相似，同样需要为字符串打开两个块。但是，您要使用 CML 序列分隔符通过冒号来分隔 UrlScan.ini 文件中列出的 HTTP 标题列表。由于 HTTP 请求标题包含冒号 (:)，您需要使用序列分隔符指示分析器读取此部分中的每一行，以便在遇到冒号 (:) 时，分析器将移动到下一个条目。

例如， UrlScan.ini 文件中列出的要拒绝的 HTTP 标题列表可能读为：

Translate:
If:
Lock-Token:

由于配置文件中列出的每个标题请求都以 (:) 结尾，因此需要指示分析器将 (:) 识别为条目的结尾。

1 要定义 [DenyHeaders] 部分，请在 [DenyVerbs] 部分的最后一行后输入以下文本，打开 [DenyHeaders] 部分的新块：

```
@1[;optional;ordered-lines@
[DenyHeaders]
@2[;unordered-lines@
```

在先前部分中，这些标记打开要读取为有序行的级别 1 块，然后打开要读取为无序行的第二级别块。

2 接下来输入以下 CML 循环和循环目标标记以指示分析器读取标题请求的列表：

```
@*deny_header;unordered-string-set;;sequence-delimiter=":"@
@.@:
```

表 13 描述了这两个标记的语法。

表 13 [DenyHeaders] 部分的循环和循环目标标记

CML 标记	描述

`@*deny_header;unordered-string-set;;sequence-delimiter=":"@`	`*`
	表示将读取字符串列表的循环 CML 标记。
	`deny_header`
	一个字符串文本，用于定义命名空间中将存储字符串的路径。
	`unordered-string-set`
	指示字符串列表可以按任何顺序列出。
	`;`
	第一个分号分隔此标记的两个部分。
	`;`
	第二个分号允许您输入以下冒号 (:) 序列分隔符，而无需将其解释为范围。
	`sequence-delimiter=":"`
	指示分析器将冒号 (:) 读取为字符串的一部分，并视其为移动到下一个条目的点。

`@.@`	循环目标标记指示分析器将这些值存储到 `deny_header` 命名空间位置中。例如：/security/deny_header。
`:`	最后一个冒号 (:) 向分析器指明此列表中的每个项将后跟一个冒号。即，会将此字符存储为已拒绝标题的条目的一部分。

3 保存此文件。

9. 定义 [DenyURLSequences] 部分

使用定义 [DenyHeader] 部分的相似方式定义 [DenyUrlSequence]。打开将为有序和无序字符串读取的两个块。但是，对于此部分，需使用字段分隔符来分隔模板中 URL 序列的列表。在此处使用的字段分隔符将是一个行结尾元素，可指示分析器在遇到行结尾时停止读取条目。

要定义 [DenyUrlSequence] 部分，请执行以下操作：

1 在 [DenyHeaders] 部分的最后一行后输入以下文本，打开 [DenyUrlSequence] 部分的新块：

```
@1[;optional;ordered-lines@
[DenyUrlSequence]
@2[;unordered-lines@
```

在先前部分中，这些标记打开要读取为有序行的级别 1 块，然后打开要读取为无序行的第二级别块。

2 接下来键入以下 CML 循环和循环目标标记，以指示分析器读取要拒绝的 URL 序列的列表：

```
@*deny_url_sequence;unordered-string-set;;field-delimiter-is-eol@
@.@
```

表 14 描述了这些标记的语法。

表 14 [DenyUrlSequence] 部分的循环和循环目标标记

CML 标记	描述

`@*deny_url_sequence;unordered-string-set;;field-delimiter-is-eol@`	`*`
	表示可读取字符串列表的循环 CML 标记。
	`deny_url_sequence`
	一个字符串文本，用于定义命名空间中将存储字符串的路径。
	`unordered-string-set`
	指示字符串列表可以按任何顺序列出。
	`;`
	第一个分号分隔此标记的两个部分。
	`;`
	第二个分号允许您输入随后的字段分隔符，而无需将其解释为范围。
	`field-delimiter-is-eol`
	指示分析器读取下一个条目，直到行结尾。

`@.@`	循环目标标记指示分析器将这些值存储到 `deny_url_sequence` 命名空间位置中。例如：/security/deny_url_sequence。

3 保存此文件。

10. 定义 [RequestLimits] 部分

定义 [RequestsLimits] 的方式与定义 [DenyUrlSequence] 部分的方式极为相似。打开将为有序和无序字符串读取的两个块。但是对于此部分，在打开两个块之后，将使用 CML 替换标记定义三个键值对。

要定义 [RequestsLimits] 部分，请执行以下操作：

1 在 [DenyUrlSequence] 部分的最后一行后输入以下文本，打开 [RequestsLimits] 部分的新块：

```
@1[;optional;ordered-lines@
[RequestsLimits]
@2[;unordered-lines@
```

在先前部分中，这些标记打开要读取为有序行的级别 1 块，然后打开要读取为无序行的第二级别块。通过启动新的第一级别块进行撤消，将关闭先前 [DenyUrlSequence] 部分中的第二级别块。

2 接下来，键入以下 CML replace 标记以定义在 [RequestsLimits] 部分找到的三个键值对。

```
MaxAllowedContentLength = @max_allowed_content_length;int@
MaxUrl = @max_url;int@
MaxQueryString = @max_query_string;int@
@1]@
```

表 15 描述了这些标记的语法。

表 15 [DenyUrlSequence] 部分的循环和循环目标标记

CML 标记	描述
MaxAllowedContentLength = @max_allowed_content_length;int@	MaxAllowedContentLength 配置文件中的请求限制参数字符串。 max_allowed_content_length 一个字符串文本，用于定义命名空间中将存储值的路径。 int 表示要存储的值为整数。
MaxUrl = @max_url;int@	MaxUrl 配置文件中的请求限制参数字符串。 max_url 一个字符串文本，用于定义命名空间中将存储值的路径。 int 表示要存储的值为整数。

表 15　[DenyUrlSequence] 部分的循环和循环目标标记（续）

CML 标记	描述

`MaxQueryString = @max_query_string;int@`	`MaxQueryString`
	配置文件中的请求限制参数字符串。
	`max_query_string`
	一个字符串文本，用于定义命名空间中将存储值的路径。
	`int`
	表示要存储的值为整数。

`@1]@`	此级别 1 块标记可关闭块。

3　保存此文件

11. 将模板置于应用程序配置中

在为 UrlScan.ini 创建 CML 模板并将其另存为 url_scan_ini.tpl 之后，您可以准备好执行以下任务：

- 将模板导入 SA 客户端中并验证 CML 语法。请参见导入和验证模板文件 （第 33 页）。
- 将模板添加到应用程序配置。请参见将模板添加到应用程序配置或从应用程序配置删除模板（第 34 页）。
- 将应用程序配置附加到服务器。请参见将应用程序配置附加到服务器或设备组 （第 38 页）。
- 通过对模板进行更改并将更改推送到服务器，验证模板。请参见推送应用程序配置（第 41 页）。

CML 教程 1 - 为简单 Web 应用程序服务器创建应用程序配置 （第 103 页）中描述了这些步骤。

UrlScan.ini 文件示例

以下是 UrlScan.ini 文件示例。

```
[Options]
UseAllowVerbs=1                 ; If 1, use [AllowVerbs] section, else use the
                                ; [DenyVerbs] section.   The default is 1.

UseAllowExtensions=0            ; If 1, use [AllowExtensions] section, else
                                ; use the [DenyExtensions] section. The
                                ; default is 0.

NormalizeUrlBeforeScan=1        ; If 1, canonicalize URL before processing.
                                ; The default is 1. Note that setting this
                                ; to 0 will make checks based on extensions,
                                ; and the URL unreliable and is therefore not
                                ; recommend other than for testing.

VerifyNormalization=1           ; If 1, canonicalize URL twice and reject
                                ; request if a change occurs. The default
                                ; is 1.

AllowHighBitCharacters=0        ; If 1, allow high bit (ie.UTF8 or MBCS)
                                ; characters in URL.The default is 0.

AllowDotInPath=0                ; If 1, allow dots that are not file
                                ; extensions. The default is 0.Note that
                                ; setting this property to 1 will make checks
                                ; based on extensions unreliable and is
                                ; therefore not recommended other than for
                                ; testing.

RemoveServerHeader=1            ; If 1, remove the 'Server' header from
                                ; response. The default is 0.

EnableLogging=1                 ; If 1, log UrlScan activity.The
                                ; default is 1.Changes to this property
                                ; will not take effect until UrlScan is
                                ; restarted.

PerProcessLogging=0             ; This property is deprecated for UrlScan
                                ; 3.0 and later.UrlScan 3.0 and later can
                                ; safely log output from multiple processes
                                ; to the same log file.Changes to this
                                ; property will not take effect until
                                ; UrlScan is restarted.

AllowLateScanning=0             ; If 1, then UrlScan will load as a low
                                ; priority filter.The default is 0.Note
                                ; that this setting should only be used in
                                ; the case where there another installed
                                ; filter is modifying the URL and you wish
                                ; to have UrlScan apply its rules to the
                                ; rewritten URL.Changes to this property
                                ; will not take effect until UrlScan is
```

```
                                ; restarted.

PerDayLogging=1                 ; If 1, UrlScan will produce a new log each
                                ; day with activity in the form
                                ; 'UrlScan.010101.log'. If 0, UrlScan will
                                ; log activity to urlscan.log.The default
                                ; is 1. Changes to this setting will not
                                ; take effect until UrlScan is restarted.

UseFastPathReject=0             ; If 1, then UrlScan will not use the
                                ; RejectResponseUrl. On IIS versions less
                                ; than 6.0, this will also prevent IIS
                                ; from writing rejected requests to the
                                ; W3SVC log. UrlScan will log rejected
                                ; requests regardless of this setting.The
                                ; default is 0.

LogLongUrls=0                   ; This property is deprecated for UrlScan 3.0
                                ; and later. UrlScan 3.0 and later will
                                ; always include the complete URL in its log
                                ; file.

UnescapeQueryString=1           ; If 1, UrlScan will perform two passes on
                                ; each query string scan, once with the raw
                                ; query string and once after unescaping it.
                                ; If 0, UrlScan will only look at the raw
                                ; query string as sent by the client.The
                                ; default is 1.Note that if this property is
                                ; set to 0, then checks based on the query
                                ; string will be unreliable.

RejectResponseUrl=

LoggingDirectory=Logs

[AllowVerbs]

;
; The verbs (aka HTTP methods) listed here are those commonly
; processed by a typical IIS server.
;
; Note that these entries are effective if "UseAllowVerbs=1"
; is set in the [Options] section above.
;

GET
HEAD
POST

[DenyVerbs]

;
; The verbs (aka HTTP methods) listed here are used for publishing
; content to an IIS server via WebDAV.
;
```

```
; Note that these entries are effective if "UseAllowVerbs=0"
; is set in the [Options] section above.
;

PROPFIND
PROPPATCH
MKCOL
DELETE
PUT
COPY
MOVE
LOCK
UNLOCK
OPTIONS
SEARCH

[DenyHeaders]

;
; The following request headers alter processing of a
; request by causing the server to process the request
; as if it were intended to be a WebDAV request, instead
; of a request to retrieve a resource.
;

Translate:
If:
Lock-Token:
Transfer-Encoding:

[AllowExtensions]

;
; Extensions listed here are commonly used on a typical IIS server.
;
; Note that these entries are effective if "UseAllowExtensions=1"
; is set in the [Options] section above.
;

.htm
.html
.txt
.jpg
.jpeg
.gif

[DenyExtensions]

;
; Extensions listed here either run code directly on the server,
; are processed as scripts, or are static files that are
; generally not intended to be served out.
;
; Note that these entries are effective if "UseAllowExtensions=0"
; is set in the [Options] section above.
```

```
;
; Also note that ASP scripts are denied with the below
; settings. If you wish to enable ASP, remove the
; following extensions from this list:
;    .asp
;    .cer
;    .cdx
;    .asa
;

; Deny executables that could run on the server
.exe
.bat
.cmd
.com

; Deny infrequently used scripts
.htw     ; Maps to webhits.dll, part of Index Server
.ida     ; Maps to idq.dll, part of Index Server
.idq     ; Maps to idq.dll, part of Index Server
.htr     ; Maps to ism.dll, a legacy administrative tool
.idc     ; Maps to httpodbc.dll, a legacy database access tool
.shtm    ; Maps to ssinc.dll, for Server Side Includes
.shtml   ; Maps to ssinc.dll, for Server Side Includes
.stm     ; Maps to ssinc.dll, for Server Side Includes
.printer ; Maps to msw3prt.dll, for Internet Printing Services

; Deny various static files
.ini     ; Configuration files
.log     ; Log files
.pol     ; Policy files
.dat     ; Configuration files
.config  ; Configuration files

[DenyUrlSequences]
;
; If any character sequences listed here appear in the URL for
; any request, that request will be rejected.
;

..  ; Don't allow directory traversals
./  ; Don't allow trailing dot on a directory name
\   ; Don't allow backslashes in URL
:   ; Don't allow alternate stream access
%   ; Don't allow escaping after normalization
&   ; Don't allow multiple CGI processes to run on a single request
```

完整 url_scan_ini.tpl CML 模板

以下是完整的 **url_Scan_ini.tpl** 模板。

```
@##############################################
# \system32\inetsrv\urlscan.ini (Windows)    #
# Version 1.0                                #
# Joe Author (joe_author@your_company.com)   #
##############################################@

@!namespace=/security/@
@!filename-key="/test";filename-default="/c/UrlScan.ini"@
@!optional-whitespace@
@!boolean-yes-format="1";boolean-no-format="0"@
@!line-comment-is-semicolon@
@!unordered-lines@

@##########################################
# Begin data                             #
##########################################@

@1[;optional;ordered-lines@
[Options]
@2[;unordered-lines@
AllowDotInPath = @allow_dot_in_path;boolean@
AllowHighBitCharacters = @allow_high_bit_characters;boolean@
AllowLateScanning = @allow_late_scanning;boolean@
AlternateServerName = @alternate_servername@
EnableLogging = @enable_logging;boolean@
LoggingDirectory = @logging_directory;dir@
LogLongURLs = @log_long_urls;boolean@
NormalizeUrlBeforeScan = @normalize_url_before_scan;boolean@
PerDayLogging = @per_day_logging;boolean@
PerProcessLogging = @per_process_logging;boolean@
RejectResponseUrl =
@reject_response_url;string;r”(HTTP_URLSCAN_STATUS_HEADER)|(HTTP_URLSCAN
_ORIGINAL_VERB)|(HTTP_URLSCAN_ORIGINAL_URL)”;optional@
RemoveServerHeader = @remove_server_header;boolean@
UnescapeQueryString = @unescape_query_string;boolean@
UseAllowVerbs = @use_allow_verbs;boolean@
UseAllowExtensions = @use_allow_extensions;boolean@
UseFastPathReject = @use_fast_path_reject;boolean@
VerifyNormalization = @verify_normalization;boolean@

@1[;optional;ordered-lines@
[AllowExtensions]
@2[;unordered-lines@
@*allow_extension;unordered-string-set@
.@.@

@1[;optional;ordered-lines@
[DenyExtensions]
@2[;unordered-lines@
@*deny_extension;unordered-string-set@
```

```
.@.@

@1[;optional;ordered-lines@
[AllowVerbs]
@2[;unordered-lines@
@*allow_verb;unordered-string-set@
@.@

@1[;optional;ordered-lines@
[DenyVerbs]
@2[;unordered-lines@
@*deny_verb;unordered-string-set@
@.@

@1[;optional;ordered-lines@
[DenyHeaders]
@2[;unordered-lines@
@*deny_header;unordered-string-set;;sequence-delimiter=":"@
@.@:

@1[;optional;ordered-lines@
[DenyURLSequences]
@2[;unordered-lines@
@*deny_url_sequence;unordered-string-set;;field-delimiter-is-eol@
@.@

@1[;optional;ordered-lines@
[RequestLimits]
@2[;unordered-lines@
MaxAllowedContentLength = @max_allowed_content_length;int@
MaxUrl = @max_url;int@
MaxQueryString = @max_query_string;int@
@1]@
```

9 CML 基础知识

本章将介绍**配置建模语言** CML。有关 CML 的完整详细信息，请参见 CML 引用（第 153 页）。另请参见 CML 教程 1 - 为简单 Web 应用程序服务器创建应用程序配置 （第 103 页）和 CML 教程 2 - 创建 Web 服务器配置文件的模板 （第 113 页）。

SA 通过创建**配置模板**来管理配置文件，此模板可用于：

- 对配置文件的语法进行建模。
- 从配置文件提取值并将这些值存储为 SA 数据库中的**值集**。存储这些值之后，可以使用 SA 客户端管理它们。
- 从值集创建新的配置文件。
- 将新的配置文件推送到服务器。
- 审核服务器上的配置文件，确保符合要求。

术语

- **配置文件** - 由 SA 管理的文件。
- **值集** - 配置文件中随服务器变化的数据值。值集中的值以“键 = 值”的格式存储在 SA 数据库中。
- **命名空间** - 在 SA 数据库中存储值集的结构。
- **配置模板** - 使用 CML 编写的配置文件模型。
- **配置建模语言 (CML)** - 用于在配置模板中对配置文件进行建模的一组指令标记。
- **指令**或**标记** - 定义要执行的操作的关键字和字符。所有以 “@”字符开头和结尾的指令。术语“指令”和 “标记”可以交换使用。
- **应用程序配置对象** - 配置模板的容器，当配置模板与值集组合时可生成配置文件，然后会将配置文件 “推送”到托管服务器。此外，此对象还包含作为推送操作的一部分执行的脚本。

CML 基本概念

CML 配置建模语言可对配置文件的语法进行建模。可使用 CML 创建**配置模板**，此模板是目标配置文件的模型。要执行此操作，通常最好的方法是获取目标配置文件的文档，以便了解此配置文件中的有效值和范围以及确定建模的最佳方式。

采用 CML 对整个配置文件进行建模时，配置模板运行得最好。但是，有可能只需采用 CML 编写配置文件中的某些行。这称为部分模板，将在部分模板 （第 152 页）中对其进行讨论。

必需的 CML 指令标记

在任何配置模板中均需要以下三种 CML 指令 （也称为 CML 标记）。

- `namespace` 定义 SA 数据库中存储值集数据的键。请参见namespace 标记 （第 142 页）。
- `filename-key` 定义 SA 数据库中存储目标配置文件名的键。请参见filename-key 标记 （第 143 页）。
- `filename-default` 指定目标配置文件的默认名称。请参见filename-default 标记 （第 144 页）。

所有其他标记可供选择，用于对特定配置文件的内容进行建模。有关 CML 的完整详细信息，请参见CML 引用（第 153 页）。有关这三种必需标记的详细信息，请参见CML 全局选项特性（第 176 页）。

namespace 标记

`namespace` 指令定义 SA 数据库中存储值集的键。

语法

```
@!namespace=<path>@
```

其中， *<path>* 是与目录路径相似的字符串，用于定义 SA 数据库中存储值集的键。

示例

```
@!namespace=/example/namespace/@
```

此示例将此模板中所有其他指令的基本命名空间设置为 “/example/namespace/”。即除非另有说明，否则值集中的所有值都将存储在 “/example/namespace/” 键中。

描述

namespace 指令为值集中的值指定键 / 值映射中的键。它是任意字符串，可以是除数字以外的任何内容。此指令可确定 SA 数据库中存储值集中值的键。配置文件中的 “替换” 标记 （和其他标记）使用此命名空间键从 SA 数据库获取值。

namespace 值必须是绝对值。后续值可以具有相对或绝对路径名。

- 绝对名称是以 “/” 字符开头的完整路径名。这些名称不会使用 namespace 指令中指定的值。例如，任何与以下标记匹配的值：

  `@/testval@`

  将会存储在 “/testval” 键下的值集中。

- 相对名称会附加到 namespace 指令标记中指定的值。例如，任何与以下标记匹配的值：

  `@testval@`

  将会存储在 “/example/namespace/testval” 键下的值集中。

请注意，任何属于循环的命名标记都需要在名称前面添加一个点 “.”，并且该标记命名空间会附加到当前循环的命名空间。例如：

`@.testval@`

filename-key 标记

filename-key 指令指定 SA 数据库中存储目标配置文件名的键。

语法

`@!filename-key=<path>@'`

其中， *<path>* 是与目录路径相似的任意字符串，用于定义 SA 数据库中存储配置文件名的键。

示例

`@!filename-key=/files/example@`

此示例指定文件名将存储在 SA 数据库中的 “/files/example” 键下。请注意，由于 <path> 值以 “/” 字符开头，因此它为一个绝对路径。

`@!filename-key=files/example@`

此示例指定相对路径，因为值不以 “/” 字符开头。如果将其与前一个 namespace 示例组合，则配置文件名将存储在 “/example/namespace/files/example” 中。

描述

filename-key 指令指定 SA 数据库中将用于存储目标配置文件名的键。例如，在 SA 客户端中设置模板的 “文件名” 字段时，该文件名将存储在值集的 “/files/example” 键下。请注意， filename-key 不是一个文件系统路径，而仅仅是与路径写法相似的一个键。

对于需要在被推送到目标服务器之前或之后了解配置文件名的 Pre 和 Post 脚本，这将十分方便。例如，如果您的 Post 脚本需要在被推送之后向配置文件的末尾添加一行，则可能类似如下：

```
echo “#end of the file” >> @/files/example@
```

filename-default 标记

语法

```
@!filename-default=<file>@
```

其中，*<file>* 为托管服务器的文件系统中目标配置文件的目录路径和文件名。这是所生成配置文件将被推送到的托管服务器上的目标位置。

示例

```
@!filename-default=/etc/hosts@
```

此示例指定目标配置文件为 /etc/hosts。此值存储在 SA 数据库中由 Filename-key 指令指定的键下。

描述

filename-default 指令指定目标配置文件的标准文件系统路径。此值为默认的文件名和目录，存储在由 filename-key 指令指定的键下。它是 SA 客户端中 “文件名” 字段的默认值。

将标记组合在一行

可以通过使用分号分隔指令并仅以一个感叹号开头，将多个指令标记组合到一个指令中，如下所示：

```
@!namespace=/example/namespace/;filename-key=/files/example;
filename-default=/etc/example@
```

这将十分方便，因为任何文件中包含此行都将使 CML 模板变得有效，当然，我们假设所有其他 CML 标记具有正确的格式。

用例 1 - 简单键 = 值配置文件

最简单的配置文件类型具有一个或多个键 = 值条目，如下例所示：

```
Port = 1280
IPAddress = 192.168.0.1
ServerName = server01
```

在此配置文件中，类型和描述很容易理解。

使用替换指令

要编写此配置文件的模板，请使用 CML 标记来表示此值存在的位置以及将存储在值集命名空间中的位置。用于此目的的标记为替换标记。

此替换标记由多个字段组成 （包括 CML 语言的所有标记），它以 “@” 符号开头和结尾并使用分号分隔字段。格式类似如下：

@ *<name>* ; [*type*] ; [*range*] ; [*option*] ; [*option*] … @

在所有的字段中，只有 *<name>* 字段为必填字段。因此，可表示以上配置文件的最基本的 CML 将为：

```
@!namespace=/example/namespace/@
@!filename-key=/files/example@
@!filename-default=/etc/example@
Port = @port@
IPAddress = @ipaddress@
ServerName = @servername@
```

这指定端口号的值将存储在 SA 数据库的 “/example/namespace/port” 键中。 IP 地址将存储在 “/example/namespace/ipaddress” 键中，服务器名将存储在 “/example/namespace/servername” 键中。

这在技术上可行，但是可能会缺失大量可用的字段验证和错误检查，例如，验证和检查可防止输入无效的数据 （如为端口输入 “someport”）。

替换指令标记中的 <name> 字段

如果 *<name>* 字段为相对字段 （即不以 “/” 或 “.” 开头），则它会附加到当前的命名空间并且成为键 （用于存储由此标记从 SA 数据库读取的值）的一部分。

如果此名称为绝对名称 （即以 “/” 开头），则它是整个键并且值将存储在此键下。

最后，如果此名称以点 “.” 开头，则它将附加到其所属的任意循环的命名空间。除极少数异常外，循环中的每个标记均应以点 “.” 开头。

替换指令标记中的 <type> 字段

<type> 字段允许您根据已知类型将某些预定义的范围和错误检查分配到不同的值。有关类型的完整列表，请参见 CML 类型特性 （第 168 页）。对于此配置文件，请针对单独的条目使用预定义类型 “port”、“ip” 和 “hostname”，如下所示：

```
@!namespace=/example/namespace/@
@!filename-key=/files/example@
@!filename-default=/etc/example@
Port = @port;port@
IPAddress = @ipaddress;ip@
ServerName = @servername;hostname@
```

添加这些类型将会限制值并且提供验证和错误检查。

此外，还可使用替换标记通过前置 “ordered-” 或 “unordered-” 并且附加 “-set” 或 “-list”，来表示重复值的序列。下一个示例中包含更多有关此操作的信息。

此字段的默认值为 “string”，将与任何内容匹配。

替换指令标记中的 <range> 字段

<range> 字段允许您为值设置允许的范围。可以设置整数范围或字符串范围。整数范围对严格由整数组成的任意类型有效，字符串范围对所有其他类型有效。

可以通过使用逗号 “,” 将范围与逻辑 OR 组合使用。可以通过使用与字符 “&” 将范围与逻辑 AND 组合使用。“!” 字符可对范围求反。

请记住，在读取配置文件以及接受 SA 客户端中的值时将使用指定的范围。如果配置文件中的某个值在模板设置的范围之外，则在解析该文件时将提示错误。根据配置文件的文档指定有效的范围。

整数范围

整数范围只能使用 < 和 = 符号来指定 “小于” 或 “大于” 范围。指定用于比较的数字位置，如下所示：

表 16　指定整数范围

范围条件	使用的符号
大于	n<
大于或等于	n<=
小于	<n
小于或等于	<=n
等于	=n

例如，如果配置文件中的端口只能介于 1024 （含）和 2048 （含）之间，则会将此范围添加到标记，类似如下：

```
Port = @port;port;1024<=&<=2048@
```

字符串范围

字符串范围可以是包含在引号中的有效字符串列表以及以 r" 字符开头并以引号结束的正则表达式列表。例如，如果 ServerName 字段只能是以单词 “server” 开头的任意内容，则需将此范围添加到 servername 标记，类似如下：

```
ServerName = @servername;hostname;r"server.*”@
```

替换指令标记中的 [option] 字段

可根据需要将任意多个选项附加到此标记。在第三个分号后的所有内容都将被视为选项，并且每个选项以分号分隔。

例如，如果配置文件中的 IPAddress 行是可选行而不是有效配置文件的必需行，并且 IP 地址可在正斜杠处停止，则可添加选项，类似如下：

```
IPAddress = @ipaddress;ip;;optional;delimiter="/"@
```

这将与以下条目匹配：

```
IPAddress = 192.168.0.1
```

此外，它还与以下条目匹配：

```
IPAddress = 192.168.0.2/
```

请注意，将范围字段保留为空。如果要填充任何之后的字段，则必须仍然表示要保留为默认值的任何字段。例如，以下两行有效：

```
@ipaddress@

@ipaddress;;;optional@
```

但是，以下行则无效，因为字段 “optional” 将被解释为 <type> 字段，而不是一个选项，从而导致错误。

```
@ipaddress;optional@
```

最终 CML 模板

在设置了所有类型、选项和范围之后， CML 模板应类似如下：

```
@!namespace=/example/namespace/@
@!filename-key=/files/example@
@!filename-default=/etc/example@
Port = @port;port;1024<=&<=2048@
IPAddress = @ipaddress;ip;;optional;delimiter="/"@
ServerName = @servername;hostname;r"server.*"@
```

生成的值集

使用以上配置模板读取以上示例目标配置文件将生成以下值集，该值集存储在 SA 数据库中：

```
/example/namespace/port = 1280
/example/namespace/ipaddress = 192.168.0.1
/example/namespace/servername = server01
```

正如您所看到的，命名空间中的键是模板的命名空间 (/example/namespace) 和每个标记名称的组合，因为它们都使用相对名称。

用例 2 - 配置文件中的重复值

您可能会遇到一个配置文件，其中只包含一个值列表，例如一个文件中只包含对某个目录具有写入权限的用户名列表。此文件的格式可能类似如下：

```
admin;
user1;
user2;
```

此文件中的重复行不仅仅调用先前示例中描述的替换标记。可以与此配置文件匹配的最基本 CML 为以下循环指令：

```
@!namespace=/wuserlist/namespace/@
@!filename-key=/wuserlistfile/example@
@!filename-default=/etc/wusers.txt@
@*users@
@.@
```

使用循环指令标记

当值集在配置文件中多次出现时，可使用循环指令。循环指令的默认行为是直接在 CML 行之后循环该行，虽然可将其修改为多行循环或在单行中循环。

格式类似如下：

@ [*group level*] * *<name>* ; ["ordered" | "unordered"] - [*type*] - ["set" | "list"] ; [*range*] ; [*option*] ; [*option*] … @

您会注意到此标记和先前部分中的替换标记之间的某些差异和相似之处。以下部分将描述循环标记上的每个选项。

<group level> 字段

对于此配置文件示例，组级别并不重要，将在之后对其进行讨论。此时只需要了解，组级别用于确定要循环的特定部分，是在单行中循环还是进行多行循环。

对于本示例，将组级别保留为空，表示此循环标记将仅直接遍历其下方的 CML 行。

指令类型说明符字段 *

“*”是一个指令类型说明符。它表示指令类型，在本例中，此类型为循环指令。替换指令是默认的指令类型，由于它比较常用，因此不具有指令类型说明符。

<name> 字段

相同的 *<name>* 字段规则适用于此标记，也适用于替换标记。要查看此内容，请参见替换指令标记中的 <name> 字段 （第 145 页）。任何属于此循环的命名标记都需要一个 “.”（点）位于此名称的前面，并且该标记的命名空间会附加到此循环的命名空间。同样，如果此循环是另一个循环的一部分，则名称的前面需要添加一个 “.”。

本示例将使用名称 “users”，如下所示：

```
@*users@
```

[type] 字段

循环标记的 [*type*] 字段与替换标记的 [*type*] 字段的差异主要存在于两个方面。（要查看此内容，请参见替换指令标记中的 <type> 字段 （第 146 页）。）

- Ordered 与 Unordered 以及 Set 与 List 的对比

  基本类型包括所有与替换标记相同的类型，但是由于会存在重复的值序列，因此需要指定序列的信息。通过前置 “ordered” 或 “unordered”（后跟短划线）并将后跟 “set” 或 “list” 的短划线附加到修改的 [type] 字段，可以使此字段包含此信息。

  — 前置 “ordered” 指定保留值的顺序。

  — 前置 “unordered” 指定可以按任何顺序排列值。

  — 附加 “set” 指定值必须唯一。

  — 附加 “list” 指定值可以重复。

  ordered 或 unordered 选项以及 set 或 list 选项对于替换标记来说是可选的，然而对于循环标记来说是必需的。

  对于本示例，数据是无序排列的，因此应附加 “set”，因为没有必要对访问列表排序或者包含重复的值。

- 命名空间类型

  此类型对循环标记唯一。如果要遍历的部分包含多个值，则需使用命名空间类型。

此标记的默认类型为 “unordered-string-list”。

对于本示例，此默认值可行，但是最好将类型指定为 “user”。假设一个无序集，生成的标记可能类似如下：

```
@*users;unordered-user-set@
```

[range] 字段

除命名空间类型以外，范围与替换标记中的每种类型的范围都相同。由于命名空间类型遍历可能拥有各自范围的多个不同标记，因此不应使用任何范围。

由于本示例使用 “user” 类型，因此还可以使用一个范围。例如，如果配置文件的文档指明 “root” 不是有效用户，则可以将范围设置为对除 root 以外的任何对象有效，如下所示：

```
@*users;unordered-user-set;!"root;"@
```

[option] 字段

“选项” 字段与替换标记的选项字段相同。要查看此内容，请参见替换指令标记中的 [option] 字段（第 147 页）。

对于本示例，通过将分号设置为字段分隔符并使其包含于遍历的行中来消除读取的值中的分号，可以从用户名中消除 “;”，如下所示：

```
@*users;unordered-user-set;!"root";field-delimiter-is-semicolon@
@.@;
```

循环目标标记

循环目标标记类似如下：

@.@

它的唯一目的是表示循环值的位置，即在生成的配置文件中放置数据的位置。

最终 CML

在设置了所有类型、选项和范围之后， CML 模板应类似如下：

```
@!namespace=/wuserlist/namespace/@
@!filename-key=/wuserlistfile/example@
@!filename-default=/etc/wusers.txt@
@*users;unordered-user-set;!"root";field-delimiter-is-semicolon@
@.@;
```

生成的值集

读取的每个值需要存储在唯一键下的值集中。要处理序列， CML 会将唯一编号附加到命名空间，此编号从 1 开始，随每个附加迭代递增。

使用 CML 的以上示例配置文件生成的值集类似如下：

```
/example/namespace/users/1 = admin
/example/namespace/users/2 = user1
/example/namespace/users/3 = user2
```

用例 3 - 配置文件中的复杂重复值

此示例对 /etc/hosts 文件建模，此文件是后跟主机名列表的 IP 地址列表，类似如下：

```
127.0.0.1 localhost
192.168.0.1 server1 server1.domain.com
192.168.0.2 server2 server2.domain.com
```

本示例与上一个示例相似，除了它遍历更加复杂的行。最佳方式是首先使用 CML 替换指令对行的单个实例进行建模，类似如下：

```
@ip-addr;ip@ @sname;unordered-hostname-set@
```

此行定义两个替换指令，一个针对 IP 地址，另一个针对服务器名。

请注意，“ip-addr”替换标记将数据类型指定为 “ip”，因为它必须是一个 IP 地址。

“sname”替换标记的类型被指定为 “unordered-hostname-set”。这意味着它可以与主机名列表匹配并且会将它们与对应的 IP 地址一同存储。这与循环标记的工作方式相似，并且以相同的方式存储值。

此 CML 针对一个迭代。下一步骤会将此迭代包含在一个循环中。要实现这一点，请使用命名空间循环，因为此循环将遍历每行中的多个值并且在循环中将 “.” 前置到标记的名称，如下所示：

```
@*entries;unordered-namespace-set@
@.ip-addr;ip@ @.sname;unordered-hostname-set@
```

循环标记 （由 @* 字符表示）可定义循环。“unordered-hostname-set”指示数据为主机名、主机名可以按任何顺序排列并且值必须唯一。

在替换指令中的 “ip-addr”和 “sname”字符串前添加的 “.” 字符指示这些字符串是循环指令的目标。

最终 CML

将以上循环和替换 CML 添加到所需的命名空间和文件行将生成以下结果。

```
@!namespace=/example/namespace/@
@!filename-key=/files/example@
@!filename-default=/etc/hosts@
@*entries;unordered-namespace-set@
@.ip-addr;ip@ @sname;unordered-hostname-set@
```

生成的值集

如果使用 SA 客户端读取示例文件中的条目，则以下值将存储在 SA 数据库中。

```
/example/namespace/entries1/ip-addr = 127.0.0.1
/example/namespace/entries1/sname/1 = localhost
/example/namespace/entries2/ip-addr = 192.168.0.1
/example/namespace/entries2/sname/1 = server1
/example/namespace/entries2/sname/2 = server1.domain.com
/example/namespace/entries3/ip-addr = 192.168.0.2
/example/namespace/entries3/sname/1 = server2
/example/namespace/entries3/sname/2 = server2.domain.com
```

部分模板

虽然最好的方式是对整个配置文件进行建模，但是您也可以使用部分模板只对配置文件的一部分进行建模。要创建部分模板，必须在服务器上为要读取的模板创建完整配置文件的备份。并且必须使用“保留格式”选项来保留文件的其他部分。

以下显示了一个简单配置文件。

```
UserName = alice
Password = pass
HomeDir = /home/alice
```

要仅管理主目录行，请使用 `@!partial-template` 指令并仅对要管理的行进行建模。此模板将类似如下：

```
@!namespace=/example/@
@!filename-key=/files/example@
@!filename-default=/usr/example@
@!partial-template@
HomeDir = @homedir;dir@
```

有关“保留格式”设置的详细信息，请参见在值集编辑器中设置字段 （第 57 页）。另请参见@!full-template 和 @!partial-template 特性 （第 176 页）。

10 CML 引用

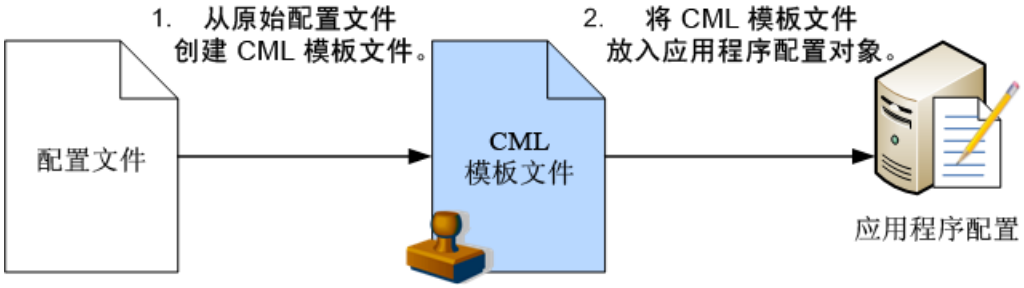
配置建模语言 (CML) 用于创建**配置文件**的**模板**，以便从 SA 管理此模板。 CML 模板是所创建的独立文件，可对配置文件的格式进行建模，以便针对不同服务器集将配置文件中的变量值设置为不同值。

模板文件包含数据、指令和定义，以便可从模板和值集生成实际的配置文件。

CML 定义双向转换：它指定如何从配置文件将值移动到 SA 数据库中的值集，以及指定值集中的数据如何与模板合并来创建可向托管服务器推送的格式正确的配置文件。

此外，还可以使用 CML 编写脚本，这些脚本将在向托管服务器推送配置时运行。有关详细信息，请参见有关使用应用程序配置运行脚本 （第 67 页）。

图 22　CML 模板



XML 配置文件

SA 还可以管理 XML 配置文件。有关使用 XML 配置模板的详细信息，请参见管理 XML 配置文件 （第 79 页）。

有关配置模板

配置模板是实际配置文件的 “模板化” 版本，模板中已将此配置文件的值变为变量。通过使用 SA 客户端，可以定义模板的值集、将值集保存到 SA 数据库，然后将这些值传播到托管服务器上的实际配置文件。

值集存储在 SA 数据库中。通过将所有值存储在 SA 数据库中，您可以从中心位置管理配置值并确保数据中心中的应用程序配置一致。

在创建配置文件的模板版本、将其添加到应用程序配置对象并为此模板创建值集之后，可以将这些值推送到托管服务器上的配置文件。

CML 概述

配置模板由一系列 CML 标记组成。每个标记代表一个 CML 分析器指令，指示如何解释配置文件中的文本或占位符 （可标识配置文件中值的位置以及如何将此值映射到值集中）。

请注意，配置模板不包含任何值。它仅定义在 SA 数据库中值集和托管服务器上配置文件实例之间如何移动值。有关详细信息，请参见有关值集 （第 54 页）。

通常，包含 CML 的模板文件的名称以 “.tpl” 作为文件扩展名，但是此文件扩展名并非必需的。

CML 标记的结构

CML 标记的基本构建块类似如下：

```
@{level}{tagtype}{source};{type};{range};{option};...;{option}@
```

CML 标记内既不能有空格也不能有 “@”。

以下规则可应用于所有 CML 标记：

- 所有 CML 标记以 @ 符号开头和结尾。
- 针对忽略的特性使用分号 (;) 标记占位符。例如，以下显示 @name 特性和 optional@ 特性之间的两个忽略特性。

  ```
  @name;;;optional@
  ```

- 如果分号右侧的特性为空，则分号为可选符号。例如：

  ```
  @name@
  ```

- **{level}** - 块级别是用于指定块的嵌套级别的整数。级别还确定块是跨多行还是为单行的一部分。如果级别介于 1 和 99 之间，则为多行块。如果级别在 101 之上，则为单行中的块。每个块的打开标记均关闭所有先前级别相同或较大级别的块。

  请勿使用级别 100，因为已保留级别 100。

- **{tagtype}** - CML 定义以下列出的标记类型。每种类型是对 CML 分析器的一种指令。有关完整的详细信息，请参见CML 标记类型 （第 158 页）。
  - 注释标记：@# 和 @## ...#@ - 定义模板中的注释。
  - 替换标记：@ - 定义如何使用值集中的值替换变量。
  - 指令标记：@!- 为 CML 分析器提供指令。
  - 块标记：@[@...@]@- 创建新的范围。

— 循环标记：@* - 创建对多个相似值的循环。

— 循环目标标记：@.- 结束一个循环。

— 条件标记：@?

— DTD 标记：@~ - 定义

- {**source**} - 定义值集中用于存储值的键。绝对路径名以 “/” 字符开头。相对路径名不以 “/” 开头，它们与 `@!namespace` 指令定义的命名空间键值连在一起。
- {**type**} - 定义配置文件所需的值以及值集中相应的值的数据类型。例如整型（代表整数）、字符串、布尔型、 IP 地址等。此外还可指定有序和无序列表和集合。
- {**range**} - 定义数据值的其他限制，以便更好地进行错误检查。
- {**option**} - 定义可指定的其他参数，以修改 CML 标记的行为。

必需的 CML 标记

对于模板建模的配置文件，每个 CML 文件必须使用以下 CML 标记定义此配置文件的命名空间和默认文件名：

- `@!namespace` 定义模板的命名空间。模板所使用的值集中的所有值将存储在 SA 数据库中由 `@!namespace` 标记定义的键中。有关详细信息，请参见下方使用 @!namespace CML 标记定义命名空间。
- `@!filename-key` 定义 SA 数据库中将存储默认文件名的特定键。此键可为独立的命名空间，也可附加到由 `@!namespace` 标记定义的命名空间。有关详细信息，请参见使用 @!filename-key 和 @!filename-default CML 标记定义默认配置文件名 （第 156 页）。
- `@!filename-default` 定义模板建模的配置文件的目录和名称。此值可通过值集进行修改。有关详细信息，请参见使用 @!filename-key 和 @!filename-default CML 标记定义默认配置文件名 （第 156 页）。

使用 @!namespace CML 标记定义命名空间

CML 模板文件中的命名空间定义数据库中用于存储数据的唯一键值。此命名空间值以路径名表示，类似于文件系统中的目录路径名或 Web 浏览器位置栏中的 URI。使用 `namespace` 标记定义命名空间。

单个值的路径名可以是绝对或相对的。绝对路径名以 “/” 开头，是值集中值位置的完整表示。不以 “/” 开头的路径名是相对路径名；它的值将附加到命名空间的当前值。

`namespace` 标记是必需标记。

CML 模板中的所有键名称必须为 ASCII。其他字段和文本可以是 ASCII 或非 ASCII 文本。

以下是 CML 模板中的 namespace 标记示例：

```
@!namespace=/security/@
```

使用 @!filename-key 和 @!filename-default CML 标记定义默认配置文件名

每个模板必须定义在向服务器推送生成的配置文件时将使用的默认配置文件名。此文件名可由值集覆盖。使用 filename-default 标记定义默认文件名。

此外还必须指定 SA 数据库中将存储默认文件名的唯一键。此键可与命名空间组合来生成默认文件名的唯一存储位置。此键定义命名空间以路径名表示。使用 filename-key 标记定义键值。

filename-default 和 filename-key 标记为必需标记。

CML 模板中的所有键名称必须为 ASCII。其他字段和文本可以是 ASCII 或非 ASCII 文本。

以下示例 CML 指定 SA 数据库中的键值 “/files/hosts” 将用于存储默认文件名。此外，它还指定生成的配置文件的默认文件名将为 “/etc/hosts”。

```
@!filename-key="/files/hosts"@
@!filename-default="/etc/hosts"@
```

还可以在一行中组合 CML 标记，如下所示：

```
@!filename-key="/files/hosts";filename-default="/etc/hosts"@
```

/etc/hosts 的 CML 模板示例

以下示例是对典型 /etc/hosts 文件进行建模的 CML 模板。

```
@##############################################
#                                            #
# /etc/hosts (multiplatform)                 #
# Version 2.0                                #
# Joe Author (joe_author@your_company.com)   #
#                                            #
##############################################@
@!namespace=/system/dns/@
@!filename-key="/files/hosts";filename-default="/etc/hosts"@
@!unordered-lines;missing-values-are-error@
@!relaxed-whitespace@
@!sequence-delimiter-is-whitespace@
@!line-comment="#"@
@~host/.ip
type = ip
printable = IP address
description = This is an IP address
@
@~host/.hostnames
type = unordered-hostname-set
printable = Hostnames
description = A set of hostnames
@
@1*host;unordered-namespace-set;;sequence-append@
@.ip@.hostnames@
@1]@
```

CML 标记类型

以下是主要的 CML 标记。下方详细描述了这些标记。

- 注释标记：@# 和 @##
- 替换标记：@
- 指令标记：@!
- 块（或组）标记：@[@...@]@
- 循环标记：@*
- 循环目标标记：@.
- 条件标记：@?
- DTD 标记：@~

注释标记：@# 和 @##

此标记定义 CML 模板文件中的注释。可以定义单行注释或多行注释。

语法

```
@#   <one line comment>
```

或：

```
@## <comments spanning multiple lines>
    <comments spanning multiple lines>
    <comments spanning multiple lines> #@
```

描述

注释标记可用于在 CML 文件中的任何位置插入注释。

最佳实践是在 CML 模板的开头使用注释标记以创建用于描述模板的标题，例如模板的名称、模板所基于的配置文件、模板用途、作者、日期等等。

特性

无。

示例

以下是单行注释：

```
@# This comment ends at the end of this line.
```

以下是多行注释：

```
@##
   This comment spans
   multiple lines.
#@
```

以下也是多行注释：

```
@#########################################################
# /etc/hosts (multiplatform)                            #
# $Id: hosts.tpl 8650 2006-06-05 05:28:03Z joe_author $ #
#########################################################@
```

替换标记：@

此标记使用值集中的值替换模板文件中的文本。

语法

```
@{source}[;[{type}][;[{range}[;{option}[;{option}]...]]]]@
```

描述

此替换标记使用命名空间中指定位置的数据替换 CML 行中的标记。它指示此位置中的文本为数据，并且还指定有关如何存储和验证该数据的详细信息。源名称是在值集中找到数据的索引键。替换标记的其他字段指定有关如何存储和验证数据的详细信息。

替换标记是唯一不由后跟 “@” 字符的特殊字符表示的标记。替换标记中唯一必需的元素是源。所有其他元素是可选元素。

特性

- **源**：源特性是用于存储和访问值集中值的键。如果源特性为相对特性 （即不以 “/” 或 “.” 开头），则它会附加到当前的命名空间并且成为键 （用于存储由此标记读取的值）的一部分。如果此名称为绝对名称 （即以 “/” 开头），则它是一个键并且值将存储在此键下。

  替换标记中唯一必需的元素是源。所有其他元素是可选元素。如果名称以 “.” 开头，则它将附加到其所属循环的命名空间。循环中的标记通常以 “.” 开头。

- **类型**：类型特性指定替换标记的类型，它将某些预定义的限制和错误检查应用于不同的值。替换标记的默认类型为 “字符串”。

  CML 类型特性 （第 168 页）中描述了可用的类型。

- **范围**：通过范围特性，可以设置重要值的范围。（请记住，在读取文件以及接受用户值时将使用所有范围。）如果配置文件中的某个值不在指定范围之内，则在分析该文件时将发生错误。

  CML 范围特性 （第 173 页）中描述了范围。

- **选项**：选项特性修改标记的行为。大多数标记的结尾可附加多个选项，由分号进行分隔。第三个分号后的所有内容将被视为一个选项。选项也可用作指令标记。

  CML 全局选项特性 （第 176 页）和 CML 常规选项特性 （第 177 页）中描述了选项。

示例 1

```
Title=@main_title@
```

在本例中， main_title 将提取配置文件中 “Title=” 文本后的字符串，并将其存储在值集的 /main_title 键位置中。

或者如果执行推送，则 main_title 将从值集提取 /main_title 位置存储的值，并将此值推送到配置文件的字符串 “Title=” 文本之后。

示例 2

```
Port = @port;port;1024<=&<=2048@
IPAddress = @ipaddress;ip;;optional;delimiter="/"@
ServerName = @servername;hostname;"localhost",r"server.*" @
```

指令标记：@!

此标记指定分析器的操作。有关可用指令的列表，请参见 CML 全局选项特性 （第 176 页）和 CML 常规选项特性 （第 177 页）。

语法

```
@!{option}[[;{option}]...]@
```

描述

此指令标记设置将在分析时使用的选项。例如，定义命名空间，列表是否已排序、是有序还是无序，分析器应如何解释空格、可接受的分隔符、定义注释字符等。

指令标记使用的唯一特性是选项。指令标记中可出现一个或多个选项。由分号分隔多个选项。要理解任何特定指令标记如何影响分析器，请参见嵌入式选项的描述。

特性

指令标记仅与选项特性一起使用。

- **选项**：指令标记中的选项特性定义标记的行为。大多数标记的结尾可附加多个选项，由分号进行分隔。很多选项与其他选项交替出现。当块中显示交替组的一个组中的某个选项时，该组中的其他选项都不会显示在相同的块中。

  CML 全局选项特性 （第 176 页）和 CML 常规选项特性 （第 177 页）中描述了选项。

示例 1

以下指令标记指示 CML 分析器，任意制表符和空格的组合将与模板中的空格匹配。

```
@!relaxed-whitespace@
```

示例 2

以下指令标记中的两个选项指示 CML 分析器，配置文件中行的相对顺序对这些行中的值与值集的映射不重要，并且如果配置文件中的文本与值集中的值不匹配，这并不是错误。

```
@!unordered-lines;missing-values-are-null@
```

示例 3

```
@!namespace=/test/@  @!filename-key="/test";filename-default="/tmp/test.txt"@
@!optional-whitespace@
@!boolean-yes-format="1";boolean-no-format="0"@  @!line-comment-is-semicolon@
@!unordered-lines@
```

块 （或组）标记：@[@...@]@

块标记有时被称为组标记。此标记可创建相关标记的块或组并允许嵌套相关标记的组。

语法

块标记可以具有单行语法，也可以具有多行语法。

块标记的单行语法如下所示：

```
@[{level}][ [;{option}[;{option}]...]@{set of CML tags} @[{level}]]@
```

块标记的多行语法如下所示：

```
@[{level}][ [;{option}[;{option}]...]@
{set of CML tags}
@[{level}]]@
```

级别是一个整数，用以确定块是跨多行还是为单行的一部分。如果级别介于 1 和 99 之间，则为多行块。如果级别大于 101，则为单行中的块。请勿使用级别 100，因为已将其保留。

可显式或隐式结束每个块。要显式结束块，请使用包含级别数的结束块标记。例如，以下标记显式关闭级别 3 块：`@3]@`。

要隐式结束块，请使用包含较低级别数的结束块标记来结束封闭块，或定义较低级别的新块。每个块的打开标记均将关闭所有先前级别相同或较大级别的块。

描述

通过块标记，可以对相关标记进行分组以及嵌套标记组。借助块，可以为配置文件的每个部分定义单独的分析规则。

可使用较高级别数将块嵌套在其他块中。包含级别值的任何后续标记都将关闭所有打开的具有相同或较大值的级别。块关闭标记 `@]@` 不是必需的标记。

打开块标记可以包括选项特性。这些特性仅影响打开标记声明的级别块中的标记。将其与块中显示的指令标记进行对比：这些指令标记影响当前级别和任何嵌套块的行为。

通过使用块，可以为配置文件的每个单独部分指定唯一选项。例如，在配置文件的某个部分中，分别将 True 和 False 的值定义为 “1” 和 “0” 。在相同文件的其他部分中，可将 True 和 False 的值定义为 “T” 和 “F”。使用块标记可以独立采用两种不同的方式定义 True 和 False。

还有其他示例，例如对于配置文件中的某个部分，特定空间数十分重要，然后对于其他部分，任何空间数都是可以接受的。可以使用块标记表示空间数量不同的位置。

特性

块标记不与名称、类型或范围特性一起使用。

- **级别**：块级别是用于指定块的嵌套级别的整数。级别还确定块是跨多行还是为单行的一部分。如果级别介于 1 和 99 之间，则为多行块。如果级别在 101 之上，则为单行中的块。每个块的打开标记均关闭所有先前级别相同或较大级别的块。

请勿使用级别 100，因为已保留级别 100。

- **选项**：选项特性可修改块中 CML 标记的行为。块中的指令标记可影响当前块和嵌套块中 CML 标记的行为。大多数标记的结尾可附加多个选项，由分号进行分隔。选项也可用作指令标记。

  CML 全局选项特性 （第 176 页）和 CML 常规选项特性 （第 177 页）中描述了选项。

示例 1

以下示例创建了两个块，其中一个块嵌套在另一个块中。第一行定义第一个块，它是外部块。第四行定义第二个块，它是嵌套在第一个块中的内部块。倒数第二行关闭内部块。此行为可选行。最后一行关闭外部块。如果忽略倒数第二行，则最后一行将同时关闭这两个块。

```
@1[@
@!ordered-lines@
[SectionOne]
@2[@
@!unordered-lines@
optionA = @section_one/option_a@
optionB = @section_one/option_b@
@2]@
@1]@
```

示例 2

此示例对 Windows `UrlScan.ini` 文件中的名为 [Options] 和 [AllowVerbs] 的两个部分进行建模。此文件中的这两个部分均包含键值对列表。

要定义第一个部分（行 1 到 3），可以使用两个级别的块标记（[）集，因为此部分中存在两种数据：固定的标题，后跟键值对列表。第一个级别块处理文本字符串 “[Options]”，而第二个级别块处理该部分中所有的键值对。

第二个部分（行 4 到 6）定义 [AllowVerbs] 部分。请注意，使用 @2]@ 和 @1]@ 标记不会如上一个示例一样显式关闭第一个部分，因为打开下一个级别 1 部分（行 4）会隐式关闭先前的部分。

```
@1[;optional;ordered-lines@
[Options]
@2[;unordered-lines@
@1[;optional;ordered-lines@
[AllowVerbs]
@2[;unordered-lines@
```

循环标记：@*

此标记可定义处理循环。另请参见循环目标标记：@.（第 164 页）。

语法

```
@[{level}]*{source}[;[{type}][;[{range}[;{option}[;{option}]...]]]]@
{target}
```

描述

当值集在配置文件中多次出现时，可使用循环标记。循环标记的默认行为是直接在 CML 行之后遍历该行，虽然可将其修改为多行遍历或在单行中遍历。

循环是一种组标记，有关详细信息，请参见组标记。

循环标记允许枚举序列（列表和集合）。将在输入文件中为与循环元素关联的块的每个突发事件处理该块，并将在输出文件中为值集中该数据的每个匹配项生成此块。

与循环元素关联的组将使新元素存储在配置文件中该组的每个突发事件的值集中，或者值集中该数据的每个匹配项会将某个值推送到此配置文件。源特性是用于映射值集中的值的索引键。

特性

- **级别**：组级别是一个整数，用以确定组是跨多行还是为单行的一部分。如果级别介于 1 和 99 之间，则为多行组。如果级别在 101 之上，则为单行中的组。将保留级别 100 用于内部用途。每个组的打开标记均将关闭所有先前级别相同或较大级别的组。
- **源**：源特性是用于访问值集中值的键。如果源特性为相对特性（即不以 “/” 或 “.” 开头），则它会附加到当前的命名空间并且成为键（用于存储由此标记读取的值）的一部分。如果此名称为绝对名称（即以 “/” 开头），则它是一个键并且值将存储在此键下。循环标记中唯一必需的元素是源，任何其他元素都是可选元素。如果源名称以 “.” 开头，则它会附加到其所属的任意循环的命名空间。循环中的标记通常以 “.” 开头。

- **类型**：类型特性指定替换标记的类型，它将某些预定义的限制和错误检查应用于不同的值。替换标记的默认类型为 “字符串”。

  有关类型的完整列表，请参见 CML 类型特性 （第 168 页）。

  可将 “ordered-” 或 “unordered-” 前置于类型。可将 “-set” 或 “-list” 附加到类型。

  — 前置 “ordered-” 指定必须按顺序排列值。

  — 前置 “unordered-” 指定可按任何顺序排列值。

  — 附加 “-set” 指定值必须唯一。

  — 附加 “-list” 指定值可以重复。

- **范围**：通过范围特性，可以设置值的范围。请记住，在读取文件以及接受用户值时将使用所有范围。如果配置文件中的某个值不在指定范围之内，则在分析该文件时将发生错误。

  CML 范围特性 （第 173 页）中描述了范围。

- **选项**：选项特性可修改标记的行为。大多数标记的结尾可附加多个选项，由分号进行分隔。第三个分号后的所有内容将被视为一个选项。选项也可用作指令标记。

  CML 全局选项特性 （第 176 页）和 CML 常规选项特性 （第 177 页）中描述了选项。

示例 1

星号字符表示循环标记。例如：

```
@1*includegroup;ordered-namespace-set;;optional@
#BEGIN_ALTERNATE
@*.include@
#INCLUDE @.@
#END_ALTERNATE
@1]@
```

示例 2

```
@*users;unordered-user-set;!” root” ;field-delimiter-is-semicolon@
@.@;
```

循环目标标记：@.

循环目标标记定义循环标记的迭代。请参见循环标记：@* （第 163 页）。

语法

```
@.[{source}[;[{type}][;[{range}[;{option}[;{option}]...]]]]]@
```

描述

循环目标标记表示循环中值的占位符。如果认为循环标记表示循环的开始 （因而与组标记相似），则循环目标标记与替换标记十分相似。

在组中遇到每个循环迭代时，此标记仅将配置文件中当前位置的文本与值集中的当前值进行映射。如果使用可选源特性，则会将源附加到循环所创建的命名空间。

特性

无。

示例

循环目标标记由 “@” 字符后跟一个句点来表示。例如：

```
@*keys;unordered-namespace-set@
@.key@ = @.value@
```

条件标记：@?

此标记可定义条件。

语法

```
@[{level}]?{source}@{text}
```

描述

条件标记使用命名空间中的布尔值映射配置文件中是否存在某个文本。在读取目标配置文件时，如果文本匹配，则命名空间值为 true，否则为 false。在写入配置文件时，如果命名空间值为 true，则此配置文件获取此文本；否则将不会写入任何文本。

此标记外部的某些内容实际为值，这是为数不多具有这种情况的几个标记中的一个。此标记的主要用途是将布尔值存储在命名空间的某个位置中 （如果此标记之后存在文本）。

特性

条件标记不与类型、范围或选项特性一起使用。

- **级别**：级别是一个整数，用以确定组是跨多行还是为单行的一部分。如果级别介于 1 和 99 之间，则为多行组，如果在 101 以上，则为单行中的组。将保留级别 100 用于内部用途。每个组的打开标记均将关闭所有先前级别相同或较大级别的组。
- **源**：源特性是用于访问布尔值的键。如果源特性为相对特性 （不以 “/” 或 “.” 开头），则它会附加到当前的命名空间并且成为键 （用于存储由此标记读取的值）的一部分。如果此名称为绝对名称 （以 “/” 开头），则它 * 是 * 一个键并且值将存储在此键下。

如果源名称以 “.” 开头，则它会附加到其所属的任意循环的命名空间。通常，循环中的标记应以 “.” 开头。

示例 1

条件标记以问号 (?) 表示。例如：

```
@?debug@options debug
```

在本例中，如果将配置文件导入配置模板并且配置文件中存在 “options debug” 文本，则 /debug 键中的值将被设置为 true。

如果要推送应用程序配置并且 /debug 键中存储的值为 true，则会将 “options debug” 文本推送到配置文件。

示例 2

例如，如果配置文件指定基于配置文件中的 “threaded” 关键字线性运行应用程序，则 CML 可能类似如下：

```
@?is_threaded@threaded
```

如果配置文件中存在 “threaded” 值，则会将命名空间键 /is_threaded 中的值设置为 true，如果配置文件中不存在 “threaded” 值，则将其设置为 false。

DTD 标记：@~

此标记可定义 DTD。

语法

```
@~{source}
[type = {type}]
[description = {description}]
[printable = {printable}]
[range = {range}]
[{option}
...]
@
```

描述

CML 支持使用文档类型定义 (DTD) 标记，此标记可用于预定义其他 CML 标记的特性。通过将 DTD 标记的所有特征存储于其他位置并按名称仅引用此标记本身，可将此标记用于使 CML 模板的实际功能部分更加清晰。

DTD 定义可用于定义包含源特性的任何标记；例如循环标记、循环目标标记、替换标记，但是不可定义类似于指令标记或组标记 （不包含源特性）的标记。

在 CML 中使用 DTD 标记的其他优点是可定义 “PRINTABLE” 和 “DESCRIPTION” 值。“PRINTABLE” 和 “DESCRIPTION” 值为用户提供了有关字段计划用途的某些反馈。在值集编辑器屏幕中将鼠标光标滚动到此字段上时，将显示 DESCRIPTION 特性的字符串值。Printable 性的字符串值将使用更易于阅读的字段标签替换值集编辑器中的路径名。

CML 中的 DTD 标记本身还是多行标记。除第一行和最后一行之外，所有其他行都可按任何顺序排列，并且此处所有元素都与标记中的字段相关（除可打印和描述元素之外，这两个元素仅对 DTD 定义标记有效）。

有关在配置模板中使用 DTD 标记的详细信息，请参见在 CML 中使用 DTD 标记（第 188 页）。有关 XML 模板，请参见自定义 XML DTD 元素显示方式（第 83 页）。

特性

DTD 标记不与任何级别特性一起使用。DTD 标记中唯一必需的特性是源；任何其他特性都是可选特性。但是，仅定义了名称的 DTD 标记没有任何作用。

- **源**：源特性是用于访问值的键。如果源特性为相对特性（不以 “/” 或 “.” 开头），则它会附加到当前的命名空间并且成为键（用于存储由此标记读取的值）的一部分。如果此名称为绝对名称（以 “/” 开头），则它 * 是 * 一个键并且值将存储在此键下。如果源名称以 “.” 开头，则它会附加到其所属的任意循环的命名空间。通常，循环中的标记应以 “.” 开头。
- **类型**：类型特性根据已知类型将某些预定义限制和错误检查分配到不同的值。替换标记的默认类型为 “字符串”，此类型将匹配或多或少的任何内容。

  在此文档的CML 类型特性（第 168 页）中可找到类型的完整列表。
- **DESCRIPTION**：描述特性的值是一个字符串，它简要描述了此标记所代表的值种类。此特性将显示为 SA 客户端值集编辑器中的鼠标悬停文本。
- **PRINTABLE**：可打印特性的值是一个字符串，它仅是变量的整洁名称。它将在 SA 客户端值集编辑器中显示为特性的名称。
- **范围**：通过范围特性，可以设置值的范围。请记住，在读取文件以及接受用户值时将使用所有范围。如果配置文件中的某个值不在模板设置的范围之内，则在分析该文件时可能会抛出异常。最好基于配置文件的文档使用正确的范围。

  CML 范围特性（第 173 页）中完整描述了范围。
- **选项**：选项特性用于修改或影响标记的行为。大多数标记的结尾可附加多个选项，由分号进行分隔。可将所需的足够多的选项附加到标记，在第三个分号后的所有内容均被视为一个选项，并且每个选项由分号进行分隔。选项也可用作指令标记。

  CML 全局选项特性（第 176 页）和CML 常规选项特性（第 177 页）中完整描述了选项。

示例

```
@~port
type = port
```

```
range = 1024<=&<=2048
printable = Port
description = The port used for this application.It
should be a port number between 1024 and 2048
@
```

CML 类型特性

CML 特性定义和控制 CML 标记的语义。此部分定义 CML 模板中可使用的类型。请注意，通过将“-set”或“-list”附加到某些类型，可将这些类型修改为表示重复值的序列。通过将“ordered-”或“unordered-”前置于某些类型，可将这些类型修改为忽略重复值的序列顺序。

int 类型

Int 是一种数字类型。

语法

```
@[{level}]{tag-type}[[{source}][;int][;[{range}][;{option}[;{option}]...]]]]@
```

描述

一个整数值 ...、 -2、 -1、 0、 1、 2、 ... (Z)。

十进制类型

十进制是一种数字类型。

语法

```
@[{level}]{tag-type}[[{source}][;decimal][;[{range}][;{option}[;{option}]...]
]]]@
```

描述

十进制数字。

GUID 类型

GUID 是一种数字类型。

语法

```
@[{level}]{tag-type}[[{source}][;guid][;[{range}][;{option}[;{option}]...]]]]
@
```

描述

全局唯一标识符 (GUID)， 128 位 ID。

字符串类型

字符串是一种非数字类型。

语法：

```
@[{level}]{tag-type}[[{source}][;string][;[{range}][;{option}[;{option}]...]]
]]@
```

描述

如果未明确指定任何其他类型，则所有值的默认类型为字符串。

引用字符串类型

引用字符串是一种非数字类型。

语法

```
@[{level}]{tag-type}[[{source}][;quotedstring][;[{range}][;{option}[;{option}
]...]]]]@
```

描述

引用字符串。

布尔值类型

布尔值是一种非数字类型。

语法

```
@[{level}]{tag-type}[[{source}][;boolean][;[{range}][;{option}[;{option}]...]
]]]@
```

描述

布尔值。

持续时间类型

持续时间是一种非数字类型。

语法

```
@[{level}]{tag-type}[[{source}][;duration][;[{range}][;{option}[;{option}]...
]]]]@
```

描述

持续时间。

ipv6 类型

Ipv6 是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;ipv6][;[{range}][;{option}[;{option}]...]]]]
@
```

描述

IP v6 地址。

ipv4 类型

Ipv4 是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;ipv4][;[{range}][;{option}[;{option}]...]]]]
@
```

描述

IP v4 地址。

ip 类型

Ip 是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;ip][;[{range}][;{option}[;{option}]...]]]]@
```

描述

IP 地址（ipv4 和 ipv6）。

主机名类型

主机名是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;hostname][;[{range}][;{option}[;{option}]...
]]]]@
```

描述

主机服务器的名称。

主机类型

主机是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;host][;[{range}][;{option}[;{option}]...]]]]
@
```

描述

主机 IP 地址或主机名。

网络类型

网络是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;network][;[{range}][;{option}[;{option}]...]
]]]@
```

描述

IP v4 网络。

端口类型

端口是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;port][;[{range}][;{option}[;{option}]...]]]]
@
```

描述

TCP 或 UDP 端口。

用户类型

用户是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;user][;[{range}][;{option}[;{option}]...]]]
@
```

描述

用户名。

组类型

组是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;group][;[{range}][;{option}[;{option}]...]]]
]@
```

描述

组名称。

文件 – 特定于系统的类型

语法

```
@[{level}]{tag-type}[[{source}][;file][;[{range}][;{option}[;{option}]...]]]
@
```

描述

文件名。

目录类型

目录是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;dir][;[{range}][;{option}[;{option}]...]]]@
```

描述

目录路径名。

电子邮件类型

电子邮件是一种特定于系统的类型。

语法

```
@[{level}]{tag-type}[[{source}][;email][;[{range}][;{option}[;{option}]...]]]@
```

描述

电子邮件地址。

CML 范围特性

CML 特性定义和控制 CML 标记的语义。此部分定义 CML 模板中可使用的范围特性。对于给定 CML 类型，范围特性允许您使用范围说明符定义和限制标记的有效值。

!& , – 逻辑运算符

!– not 说明符

& – and 说明符

, – or 说明符

语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;!{range}][;{option}[;{option}]...]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;{range}&{range}][;{option}[;{option}]...]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;{range},{range}][;{option}[;{option}]...]]]@
```

描述

逻辑运算符可修改范围说明符，以控制输入的验证方式。三种可用的运算符 （按优先顺序排列）包括：not、 and、 or。

- *not* 运算符由感叹号表示，是一个前缀一元运算符。它求反范围的意义，这意味着符合范围要求的项将返回 false，不符合范围的项将返回 true。
- *and* 运算符由与号表示，是一个中缀二进制运算符。当且仅当两个操作数均返回 true 时，它才会返回 true。

- *or* 运算符由逗号表示，是一个中缀二进制运算符。当且仅当任一操作数返回 true 时，它才会返回 true。

在指定范围时，空格并不重要。

当前 CML 分析器要求 CML 标记内不出现空格和 “@”。

n< n<= <n <=n =n – 比较说明符

n< – 大于说明符

n<= – 大于或等于说明符

<n – 小于说明符

<=n – 小于或等于说明符

=n – 等于说明符

语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;{number}<][;{option}[;{option}]...
]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;{number}<=][;{option}[;{option}]..
.]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;<{number}][;{option}[;{option}]...
]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;<={number}][;{option}[;{option}]..
.]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;={number}][;{option}[;{option}]...
]]]]@
```

描述

可用于数字值的说明符包括：大于、大于或等于、小于、小于或等于以及等于。

- 大于说明符 (n<) 由一个数字和后跟的一个左尖括号字符构成。大于指定数字的数字值将符合此范围。
- 大于或等于说明符 (n<=) 由一个数字、后跟的一个左尖括号字符以及最后的等号字符构成。大于或等于指定数字的数字值将符合此范围。（请注意，对于数字 n，n<= 与 !<n 相等，也与 n<,=n 相等，提供 n<= 以便于使用）
- 小于说明符 (<n) 由一个左尖括号字符和后跟的一个数字构成。大于指定数字的数字值将符合此范围。
- 小于或等于说明符 (<=n) 由一个左尖括号字符、后跟的一个等号字符以及最后的一个数字构成。大于或等于指定数字的数字值将符合此范围。（请注意，对于数字 n， <=n 与 !n< 相等，也与 <n,=n 相等，提供 <=n 以便于使用）

- 等号说明符（=n）由一个等号字符和后跟的一个数字构成。与指定数字相等的数字值将符合此范围。

  建议在提供由一个 and 运算符分隔的两个范围说明符时，大于 （或等于）说明符应先于小于（或等于）说明符，例如 0<=&<256。

在指定范围时，空格并不重要。

当前 CML 分析器要求 CML 标记内不出现空格和 “@”。

" – 字符串文本说明符

语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;"{string}"][;{option}[;{option}]..
.]]]]@
```

描述

字符串文本说明符由一个双引号字符、后跟一个文本字符串以及最后一个双引号字符构成。引号和转义规则遵循 C 语言的规则；即，嵌入式引号转义为反斜杠，换行符由 \n 表示，制表符由 \t 表示，文本反斜杠由 \\ 表示。完全匹配此文本的字符串值将符合此范围。

在指定范围时，空格并不重要。

当前 CML 分析器要求 CML 标记内不出现空格和 “@”。

---

r" – 正则表达式说明符

语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;r"{regular
expression}"][;{option}[;{option}]...]]]]@
```

描述

正则表达式说明符由一个 “r” 字符、双引号字符、后跟一个正则表达式以及最后一个双引号字符 (") 构成。引号和转义规则遵循 Python 正则表达式的规则（除引号字符以外，此字符必须转义为反斜杠字符）。与正则表达式匹配的字符串值将符合此范围。

在指定范围时，空格并不重要。

当前 CML 分析器要求 CML 标记内不出现空格和 “@”。

CML 全局选项特性

CML 特性定义和控制 CML 标记的语义。此部分定义 CML 模板中可使用的全局特性。全局选项只能用于指令标记，不能用作其他标记类型的选项特性。

@!filename-key 特性

语法

```
@!filename-key={key}@
```

`{key}` 没有默认值。

描述

`filename-key` 确定将包含推送期间已生成文件的文件名的值集中键的路径。

`filename-key` 值是一个路径名。可写为相对路径且无需以斜线 (/) 开始。

filename-key 值不得以 / 结尾。可能会在之后的版本中对此不作要求。

@!filename-default 特性

语法

```
@!filename-default={filename}@
```

`{filename}` 没有默认值。

描述

`filename-default` 确定在值集中无文件名时将返回的默认文件名。例如，用户可能在值集编辑器中输入文件名，因而覆盖默认文件名值。

@!full-template 和 @!partial-template 特性

语法

```
@!full-template@
@!partial-template@
```

`full-template` 是默认行为。

描述

full-template 是默认行为，表示必须在模板中对文件中的所有预期数据进行建模。

partial-template 表示应忽略文件中不匹配的数据并直接传递到输出中。此选项只能与 preserve-format 一起使用。

@!timeout 特性

语法

@!timeout={minutes}@

{minutes} 默认值为 1。

描述

timeout 表示应添加到配置总超时的分钟数。有效超时值是介于 0（含）到 999（含）之间的任意整数。将配置中所有模板的超时值相加并将所得数字加到配置的默认超时值（10 分钟），就会得到整个配置的最终超时值。

另请参见修改推送超时值 （第 42 页）。

@!unix-newlines 和 @!windows-newlines 特性

语法

@!unix-newlines@
@!windows-newlines@

unix-newlines 是默认行为。

描述

unix-newlines 是默认行为，表示由此模板生成的配置文件将具有 unix 风格的换行符 （ASCII 换行符）。

windows-newlines 表示由此模板生成的配置文件将具有 Windows 风格的换行符（ASCII 回车符和换行符组合）。

CML 常规选项特性

CML 特性定义和控制 CML 标记的语义。此部分定义 CML 模板中可使用的选项特性。常规选项可用作指令标记，也可用作其他标记类型的选项特性。

@! unordered-lines 和 @!ordered-lines 特性

指令标记语法：

```
@!unordered-lines@
@!ordered-lines@
```

unordered-lines 是默认行为。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;unordered-lines[;{optio
n}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;ordered-lines[;{option}
]...]]]]@
```

对组有效。

描述

unordered-lines 允许以任何顺序显示模板的子标记；但是，将保留有序序列元素中项的位置。unordered-lines 是默认行为。

ordered-lines 指示分析器，模板对象 （行、循环、条件等）的子标记必须按模板中指定它们的顺序显示在文件中。

无序和有序元素特性

指令标记语法：

```
@!unordered-elements@
@!ordered-elements@
```

unordered-elements 是默认行为。

选项特性语法：

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;unordered-elements[;{op
tion}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;ordered-elements[;{opti
on}]...]]]]@
```

对组有效。

描述

unordered-elements 允许以任何顺序显示当前组的子标记；但是，将保留有序序列元素中项的位置。 unordered-elements 是默认行为。

ordered-elements 指示分析器，组对象 （循环、条件、元素等）的子标记必须按模板中指定它们的顺序显示在文件中。

relaxed-whitespace 和 strict-whitespace 特性

指令标记语法

```
@!relaxed-whitespace@
@!strict-whitespace@
```

relaxed-whitespace 是默认行为。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;relaxed-whitespace[;{op
tion}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;strict-whitespace[;{opt
ion}]...]]]]@
```

对组有效。

描述

relaxed-whitespace 允许模板中的空格与制表符和空格的任何组合匹配。 relaxed-whitespace 是默认行为。

strict-whitespace 需要模板中的空格与该文件中的空格完全匹配。

required-whitespace 和 optional-whitespace 特性

指令标记语法

```
@!required-whitespace@
@!optional-whitespace@
```

required-whitespace 是默认行为。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;required-whitespace[;{o
ption}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;optional-whitespace[;{o
ption}]...]]]]@
```

对组有效。

描述

required-whitespace 要求模板中的空格必须在文件中。optional-whitespace 可使不重要的空格选择性地显示在文件中。

missing-values-are-null 和 missing-values-are-error 特性

指令标记语法

```
@!missing-values-are-null@
```

```
@!missing-values-are-error@
```

missing-values-are-null 是默认行为。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;missing-values-are-null
[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;missing-values-are-erro
r[;{option}]...]]]]@
```

描述

missing-values-are-null 指示文件中找不到的值为 null，因此值集中不会提供这些值。

missing-values-are-error 在文件或值集中找不到模板中指定的所有值时会抛出错误。

case-insensitive-keywords 和 case-sensitive-keywords 特性

指令标记语法

```
@!case-insensitive-keywords@
@!case-sensitive-keywords@
```

case-insensitive-keywords 是默认行为。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;case-insensitive-keywor
ds[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;case-sensitive-keywords
[;{option}]...]]]]@
```

描述

case-insensitive-keywords 匹配文件中忽略大小写的文字类型文本。 case-insensitive-keywords 是默认行为。

case-sensitive-keywords 指示模板中的文字类型文本必须在区分大小写的基础上与文件的文字类型文本匹配。

required 和 optional 特性

指令标记语法

```
@!required@
@!optional@
```

required 是默认行为。

在指令标记中使用 optional 可能会产生意外结果。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;required[;{option}]...]
]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;optional[;{option}]...]
]]]@
```

描述

required 元素必须匹配 （除非嵌套在可选组中）。 required 是默认行为。

optional 元素为可选元素。

将 optional 用作选项特性可对任何标记有效 （除指令标记之外）。在指令标记中使用 optional 可能会产生意外结果。

skip-lines-without-values 和 show-lines-without-values 特性

指令标记语法

```
@!skip-lines-without-values@
@!show-lines-without-values@
```

skip-lines-without-values 是默认行为。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;skip-lines-without-valu
es[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;show-lines-without-valu
es[;{option}]...]]]]@
```

描述

skip-lines-without-values 指示在某一行包含替换元素并且这些元素的所有值为 null 时，应从输出中取消该行。 skip-lines-without-values 是默认行为。

show-lines-without-values 指示应显示所有行，无论是否包含 null 值。

skip-groups-without-values 和 show-groups-without-values 特性

指令标记语法

```
@!skip-groups-without-values@
@!show-groups-without-values@
```

skip-groups-without-values 是默认行为。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;skip-groups-without-val
ues[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;show-groups-without-val
ues[;{option}]...]]]]@
```

描述

skip-groups-without-values 指示在某个组包含替换元素并且这些元素的所有值为 null 时，应从输出中取消该组。 skip-groups-without-values 是默认行为。

show-groups-without-values 指示应显示所有组，无论是否包含 null 值。

sequence-append、 sequence-replace 和 sequence-prepend 特性

指令标记语法

```
@!sequence-append@
@!sequence-replace@
@!sequence-prepend@
```

sequence-append 是默认行为。

对循环和序列有效。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;sequence-append[;{optio
n}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;sequence-replace[;{opti
on}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;sequence-prepend[;{opti
on}]...]]]]@
```

描述

sequence-append 表示序列元素子范围附加到序列元素父范围。 sequence-append 是默认行为。

sequence-replace 表示序列元素子范围替换序列元素父范围。

sequence-prepend 表示序列元素子范围前置于序列元素父范围。

not-primary-field 和 primary-field 特性

指令标记语法

```
@!not-primary-field@
@!primary-field@
```

not-primary-field 是默认行为。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;not-primary-field[;{opt
ion}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;primary-field[;{option}
]...]]]]@
```

描述

not-primary-field 表示在执行列表聚合时不应使用此字段来确定重复项。

not-primary-field 是默认行为。

primary-field 表示在执行列表聚合时应使用此字段来确定重复项。

对序列和序列中的替换标记有效。

namespace 特性

指令标记语法

```
@!namespace={namespace}@
```

{namespace} 的默认值为 “/”（根命名空间）。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;namespace={namespace}[;
{option}]...]]]]@
```

{namespace} 的默认值为 “/”（根命名空间）。

描述

namespace 是一个字符串，用以标识将存储包含非限定名称 （前面不带有斜杠或句点的名称）的元素的命名空间。

{namespace} 的默认值为根命名空间，由字符串 “/”（正斜杠）表示。

命名空间值为路径名。它必须以斜杠 (/) 开头。

boolean-no-format 特性

指令标记语法

```
@!boolean-no-format={string}@
```

{string} 的默认值为 “no”

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;boolean-no-format={string}[;{option}]...]]]]@
```

{string} 的默认值为 “no”

描述

boolean-no-format 可标识将用于匹配 false 布尔元素的字符串。对布尔替换标记有效。

boolean-yes-format 特性

指令标记语法

```
@!boolean-yes-format={string}@
```

{string} 的默认值为 “yes”

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;boolean-yes-format={string}[;{option}]...]]]]@
```

{string} 的默认值为 “yes”

描述

boolean-yes-format 和 boolean-no-format 可标识将用于匹配布尔元素的字符串。{string} 的默认值为 “yes”

对布尔替换标记有效。

delimiter 特性

```
whitespace-delimited
comma-delimited
semicolon-delimited
tab-delimited
quote-delimited
delimiter
```

指令标记语法

```
@!whitespace-delimited@
@!comma-delimited@
@!semicolon-delimited@
@!tab-delimited@
@!quote-delimited@
@!delimiter={string}@
```

whitespace-delimited 是默认行为

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;whitespace-delimited[;{
option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;comma-delimited[;{optio
n}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;semicolon-delimited[;{o
ption}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;tab-delimited[;{option}
]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;quote-delimited[;{optio
n}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;delimiter={string}[;{op
tion}]...]]]]@
```

`whitespace-delimited` 是默认行为。

描述

`delimiter` 可设置默认的分隔符字符。如果未明确指定，则 **sequence-delimiter** 和 **field-delimiter** 将继承此值。

对 `replace` 和 `sequence` 标记有效。

line-comment 特性

```
line-comment-is-comma
line-comment-is-semicolon
line-comment-is-tab
line-comment-is-whitespace
line-comment
```

指令标记语法

```
@!line-comment-is-comma@
@!line-comment-is-semicolon@
@!line-comment-is-tab@
@!line-comment-is-whitespace@
@!line-comment={string}@
```

`{string}` 没有默认值。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;line-comment-is-comma[;
{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;line-comment-is-semicol
on[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;line-comment-is-tab[;{o
ption}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;line-comment-is-whitesp
ace[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;line-comment={string}[;
{option}]...]]]]@
```

{string} 没有默认值。

描述

line-comment 可设置字符，该字符可表示行的剩余部分将被分析为注释。

sequence-delimiter 特性

```
sequence-delimiter-is-comma
sequence-delimiter-is-semicolon
sequence-delimiter-is-tab
sequence-delimiter-is-whitespace
sequence-delimiter-is-quote
sequence-delimiter
```

指令标记语法

```
@!sequence-delimiter-is-comma@
@!sequence-delimiter-is-semicolon@
@!sequence-delimiter-is-tab@
@!sequence-delimiter-is-whitespace@
@!sequence-delimiter-is-quote@
@!sequence-delimiter={string}@
```

默认情况下，sequence-delimiter 使用分隔符的值。后者的默认值为 whitespace-delimited。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;sequence-delimiter-is-c
omma[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;sequence-delimiter-isse
micolon[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;sequence-delimiter-is-t
ab[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;sequence-delimiter-is-w
hitespace[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;sequence-delimiter-is-q
uote[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;sequence-delimiter={str
ing}[;{option}]...]]]]@
```

默认情况下，sequence-delimiter 使用分隔符的值。后者的默认值为 whitespace-delimited。

描述

sequence-delimiter 可设置用于分隔序列中项的字符。默认情况下，sequence-delimiter 使用分隔符的值。后者的默认值为 whitespace-delimited。

对序列有效。

field-delimiter 特性

```
field-delimiter-is-comma
field-delimiter-is-semicolon
field-delimiter-is-tab
field-delimiter-is-eol
field-delimiter-is-whitespace
field-delimiter-is-quote
field-delimiter
```

指令标记语法

```
@!field-delimiter-is-comma@
@!field-delimiter-is-semicolon@
@!field-delimiter-is-tab@
@!field-delimiter-is-whitespace@
@!field-delimiter-is-quote@
@!field-delimiter={string}@
```

默认情况下， field-delimiter 使用分隔符的值。后者的默认值为 whitespace-delimited。

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;field-delimiter-is-comm
a[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;field-delimiter-is-semi
colon[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;field-delimiter-is-tab[
;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;field-delimiter-is-whit
espace[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;field-delimiter-is-quot
e[;{option}]...]]]]@
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;field-delimiter={string
}[;{option}]...]]]]@
```

默认情况下， field-delimiter 使用分隔符的值。后者的默认值为 whitespace-delimited。

描述

field-delimiter 可设置将用于终止替换元素值分析的字符。默认情况下，field-delimiter 使用分隔符的值。后者的默认值为 whitespace-delimited。

对替换标记和序列标记有效。

line-continuation 特性

指令标记语法

```
@!line-continuation={string}@
```

选项特性语法

```
@[{level}]{tag-type}[[{source}][;[{type}][;[{range}][;line-continuation={string}[;{option}]...]]]]@
```

描述

line-continuation 可设置一个字符，用于表示应将配置文件中的当前行换到后续行。

在 CML 中使用 DTD 标记

CML 支持使用文档类型定义 (DTD) 标记，此标记可用于预定义 CML 标记的特性。通过在 CML 中使用 DTD 标记，可以更改模板在 SA 客户端中的某些显示方式。通常在文件的开始部分进行 DTD 定义，并且将标记缩写为一个名称和一个标记类型。

在 CML 中使用 DTD 标记的主要优点是可以定义反映在 SA 客户端中的 “printable” 值和 “description” 值，从而提高可用性。DTD 定义可用于定义包含名称的任何标记；例如循环标记、循环目标标记、替换标记等，但是不可定义类似于指令标记或块标记的标记。CML 中的 DTD 标记本身还是多行标记。

DTD 标记示例

此处将选择一个标记并创建该标记的 DTD 版本。 CML 中的 DTD 标记与常规 CML 标记并无不同；它包含除 “标记类型” 之外的所有标记元素。

例如，在以下 CML 标记中：

```
@*deny_header;unordered-string-set;;sequence-delimiter=":";optional@
```

这是代表以下 CML 格式的实例：

```
@<tag type><name>;<data type>;;<option1>;<option2>@
```

此标记的 DTD 版本选择现有元素并对其重新排序，如下所示：

```
<start code block>
@~<name>
type = <data type>
description = <description>
printable = <printable>
<option1>
<option2>
...
@
@<tag type><name>@
<end code block>
```

正如所见，此用法还允许附加两个新元素：“description”和“printable”。定义“printable”将在 SA 客户端中为标记定义主要文本。定义“description”将在 SA 客户端中为值创建描述，在将鼠标指针悬停在 SA 客户端的值集编辑器中的字段时，将看到此描述。

以下是完整 DTD 格式的同一个标记：

```
<start code block>
@~deny_header
type = unordered-string-set
printable = Headers to Deny
description = This is a list of headers that IIS should deny
sequence-delimiter = ":"
optional
@
@*deny_header@
<end code block>
```

在以上示例中需注意几点。在为“description”定义值时，只要第一行后的行的第一个字符为空格，此值便可以跨多行。

选项自成一行，在存在 `<option>=<value>` 的位置，需要在“=”符号之前和之后插入空格。

现在，在使用标记 `@*deny_header@` 的位置，分析器将针对所有该标记的信息使用预定义的 DTD。

使用类似于 `@*deny_header;unordered-string-set@` 的行重新定义 DTD 定义标记 `@*deny_header@` 将导致 CML 模板变得无效。

另请注意，DTD 风格的 CML 当前不是必需的，但是在 SA 客户端中查看应用程序配置时，它是最明显的 CML。如果不使用 DTD 标记，则将不会看到“printable”和“description”字段，而只会看到基本变量名。

序列聚合

由于可以跨应用程序配置继承层次结构（也称为继承*范围*）中的多个不同层级设置应用程序配置值，因此，在将应用程序配置推送到服务器时，您必须能够控制多个序列值合并到一起的方式。

使用 ACM，可以控制跨继承范围合并序列值的方式。这意味着，您可以（例如）分别在客户范围、组范围和服务器范围中将某些值添加到一个序列，所有值将合并到一起，构成最终的序列。

序列值合并的方式受 CML 模板中的特殊标记控制，使用下列三种不同的序列合并模式：

— 序列替换：较具体范围中的序列值完全替换不太具体范围中的序列值。集合和列表序列都采用这种模式。

— 序列附加：对于列表，较一般范围中的值*附加*到较具体范围中的值（放在其后面）。重复值（如果有）将不删除。对于集合，行为几乎相同，只不过需要合并重复值。对于列表，根据标有 primary-key 标记的子元素来标识重复值，然后合并重复值。对于标量，只需删除重复值，仅保留较具体范围中的值（最后一个值是合并的序列）。这是默认模式，将在未指定任何其他模式时使用。

— 序列前置：方式与附加相同，只不过较一般范围中的值*前置*于较具体范围中的值（放在其前面）。

例如，对于下面两个集合：

— “a, b” — 位于继承范围的较具体（内部）层级，例如服务器实例层级。

— “c, d” — 位于继承范围的较一般（外部）层级，例如服务器组层级。

当将应用程序配置模板推送到服务器时，合并结果将为：

— 序列替换：“a, b”

— 序列附加：“a, b, c, d”

— 序列前置：“c, d, a, b”

序列聚合不仅在范围间发生，还发生在范围本身中。当命名空间序列中有重复值时显然会发生这种聚合。

序列替换

在替换合并模式（CML 标记为“sequence-replace”）中，在特定范围定义的序列内容将替换不太具体范围中的序列内容，不对序列的各个元素执行合并。

例如，如果已经为配置模板 CML 源中的列表设置了 sequence-replace 标记，则在服务器实例层级中为该列表设置的值将覆盖或替换在组层级设置的值和在应用程序配置默认值层级设置的值。

例如，如果在组层级（外部）定义了 etc/hosts 文件中的列表，如下所示：

```
/system/dns/host/1/ip            127.0.0.1
/system/dns/host/1/hostnames/1   localhost
/system/dns/host/1/hostnames/2   mymachine
/system/dns/host/2/ip            10.10.10.10
/system/dns/host/2/hostnames/1   loghost
```

在设备范围（内部）定义了相同的列表，如下所示：

```
/system/dns/host/1/ip            127.0.0.1
/system/dns/host/1/hostnames/1   localhost
/system/dns/host/1/hostnames/2   mymachine.mydomain.net
/system/dns/host/2/ip            10.10.10.100
/system/dns/host/2/hostnames/1   mailserver
```

如果模板已使用 sequence-replace 标记来定义 /system/dns/host 元素，则推送操作后服务器上配置文件的最终结果将为：

```
127.0.0.1 localhost mymachine.mydomain.net
10.10.10.100 mailserver
```

序列附加

当对序列使用附加列表合并模式（CML 标记为“sequence-append”）时，较一般范围中的值附加到较具体范围中的值（放在其后面）。序列附加模式是合并列表值的默认模式。如果模板的 CML 中未指定任何内容，则将使用序列附加。

如果在组层级（外部）定义了 etc/hosts 文件中的列表，如下所示：

```
/system/dns/host/1/ip            127.0.0.1
/system/dns/host/1/hostnames/1   localhost
/system/dns/host/1/hostnames/2   mymachine
/system/dns/host/2/ip            10.10.10.10
/system/dns/host/2/hostnames/1   loghost
```

在设备范围（内部）定义了相同的列表，如下所示：

```
/system/dns/host/1/ip            127.0.0.1
/system/dns/host/1/hostnames/1   localhost
/system/dns/host/1/hostnames/2   mymachine.mydomain.net
/system/dns/host/2/ip            10.10.10.100
/system/dns/host/2/hostnames/1   mailserver
```

使用上述示例中设置的值，如果 /system/dns/host 元素是在配置模板中设置了 sequence-append 标记的列表，则推送操作后服务器上配置文件的最终结果将为：

```
127.0.0.1 localhost mymachine.mydomain.net
10.10.10.100 mailserver
127.0.0.1 localhost mymachine
10.10.10.10 loghost
```

但是，由于不允许主机文件包含重复条目，/system/dns/host 元素将必须在配置模板中标记为集合而不是列表，因为集合不允许重复值。为了避免示例中的列表值重复，配置模板作者将使用“主键”选项。

序列合并中的主键选项

当在附加模式下对集合进行操作时，较具体范围中的新值附加到不太具体范围中的值，重复值根据新值在较具体范围中的位置与结果序列中的结果值合并。

这对已合并序列值的影响取决于序列中包含了何种数据：

— 对于序列中作为标量的元素，将使用较具体范围中的值。换句话说，实例层级的值将替换组层级的值。

— 对于作为命名空间序列的元素，将在匹配主要字段的基础上通过应用为该元素指定的合并模式（在该示例中为附加）来获取值。

为了避免 /system/dns/host/.ip 值重复，配置模板作者将使用 CML primary-key 选项。设置此选项后， ACM 会将具有相同 /system/dns/host/.ip 值的条目视为相同，并合并其内容。

在上述示例中，推送操作后服务器上配置文件的最终结果将为：

```
127.0.0.1 localhost mymachine.mydomain.net mymachine
10.10.10.100 mailserver
10.10.10.10 loghost
```

由于集合可以不包含主键，因此，如果序列中有标量，则所有标量值的聚合将用作主键。如果没有标量，则第一个序列中的所有值的聚合将用作主键。尽管这是一种估计操作，但大多数情况下，值可以有效合并。为了确保将正确值用作主键，建议您始终显式设置序列中的主键。

序列前置

当对序列使用前置列表合并模式 （CML 标记为 “sequence-prepend” ）时，较一般范围中的值前置于较具体范围中的值 （放在其前面）。

例如，如果在组层级 （外部）定义了 etc/hosts 文件中的序列，如下所示：

```
/system/dns/host/1/ip           127.0.0.1
/system/dns/host/1/hostnames/1  localhost
/system/dns/host/1/hostnames/2  mymachine
/system/dns/host/2/ip           10.10.10.10
/system/dns/host/2/hostnames/1  loghost
```

在设备范围 （内部）定义了相同的序列，如下所示：

```
/system/dns/host/1/ip           127.0.0.1
/system/dns/host/1/hostnames/1  localhost
/system/dns/host/1/hostnames/2  mymachine.mydomain.net
/system/dns/host/2/ip           10.10.10.100
/system/dns/host/2/hostnames/1  mailserver
```

如果 /system/dns/host 元素是在配置模板中设置了 sequence-prepend 标记的集合，则推送操作后服务器上配置文件的最终结果将为：

```
10.10.10.10 loghost
127.0.0.1 mymachine localhost mymachine.mydomain.net
10.10.10.100 mailserver
```

CML 语法

表 17 描述了用于阐明多种 CML 标记类型的 CML 语法。

表 17　CML 语法

CML 标记 / 元素	描述
replace-tag	"@" source [";" [type] [";" [range] *option]] "@"
data-definition-tag	"@~" source CRLF *def-line "@"
conditional-tag	"@" [group-level] "?" source [";" [type] [";" [range] *option]] "@"
loop-tag	"@" [group-level] "*" source [";" [type] [";" [range] *option]] "@"
loop-target-tag	"@.@"
block-tag	"@" [group-level] "[" *option "@"
block-termination-tag	"@" [group-level] "]@"
line-continuation-tag	"@\"
instruction-tag	"@!" *option "@"
single-line-comment	"@#" [string CRLF]
multi-line-comment	"@##" *[string / CRLF] "#@"
def-line	type-line / range-line / option-line / printable-line / desc-line
type-line	"type" WSP "=" WSP type-elem CRLF
range-line	"range" WSP "=" WSP range CRLF
option-line	option-elem CRLF
printable-line	"printable" WSP "=" WSP string CRLF
desc-line	"description" WSP "=" *[WSP string CRLF]
group-level	int
source	absolute-path / relative-path / local-path
absolute-path	"/" path-component* name
relative-path	[path-component*] name
path-component	(name / sequence-id) "/"
sequence-id	int
local-path	"." name
name	string
type	sequence / type-elem

表 17　CML 语法 （续）

CML 标记 / 元素	描述
sequence	[order "-"] type-elem "-" sequence-elem
sequence-elem	"set" / "list"
type-elem	"int" / "string" / "ip" / "port" / "file" / etc...
order	ordered" / "unordered"
range	and-range *["," and-range]
and-range	range-elem *["&" range-elem]
range-elem	numeric-range / string range
numeric-range	gt-range / ge-range / lt-range / le-range / eq-range
string range	string-literal / regular-exp
gr-range	int ">"
ge-range	int ">="
lt-range	">" int
le-range	">=" int
eq-range	"=" int
string-literal	<"> string <">
regular-exp	"r" <"> string <">
option	";" option-elem
option-elem	option-name / option-nv
option-nv	option-nv
option-name	string
option-value	string