
附录 A — 配置参考

本节对各种 WebSphere Product Center 属性文件中使用的参数进行概述。

- common.properties
- admin_properties.xml
- docstore_mount.xml
- data_entry_properties.xml

在 WebSphere Product Center 的安装期间，将把许多值赋给参数。如果需要修改任何值，则可以使用本附录中概述的规则来进行那些修改。

文件名：common.properties

在启动时，系统将使用此文件来读入所有系统级别参数。此文件包含数据库层的设置（连接参数）、目录设置、缺省字符集、线程池参数以及其它设置。

注意：common.properties 参数是按它们在文件中的出现顺序列示的。

数据库设置

用途：连接至数据库

db_username — 用来登录到数据库的用户名

db_password — 用来登录到数据库的密码

db_url — 用来连接至数据库的 jdbc url。使用以下格式：jdbc:db2:<db alias>

示例：

db_userName=qal

db_password=qal

db_url=jdbc:db2:qadb

独立的表空间

用途：定义数据库是否具有不同的表空间。对于膝上型计算机或单机系统，此参数很有用。

如果为 false，则 GenSchema.java 忽略所有表空间伪指令

如果为 true，则 GenSchema.java 遵守所有表空间伪指令

示例：

db_tablespaces=true

缺省服务的 DB 参数

用途：定义缺省服务的参数。

值：使用整数值。

示例：

db_maxConnection=8

db_minConnection=2

db_maxConnection_default=4

db_maxConnection_system=4

admin 服务的 DB 参数

用途：定义 admin 服务的参数。

值：使用整数值。

示例：

db_maxConnection_admin=5

db_minConnection_admin=2

db_maxConnection__admin_default=4

db_maxConnection__admin_system=1

appsvr 服务的 DB 参数

用途：定义 appsvr 服务的参数。

值：使用整数值。

示例：

db_maxConnection_appsvr=30

db_minConnection_appsvr=10

db_maxConnection_appsvr_default=26

db_maxConnection_appsvr_system=4

scheduler 服务的 DB 参数

用途：定义 scheduler 服务的参数。

值：使用整数值。

示例：

db_maxConnection_scheduler=40

db_minConnection_scheduler=10

db_maxConnection_scheduler_default=36

db_maxConnection_scheduler_system=4

eventprocessor 服务的 DB 参数

用途：定义 eventprocessor 服务的参数。

值：使用整数值。

示例：

db_maxConnection_eventprocessor=6

db_minConnection_eventprocessor=4

db_maxConnection_eventprocessor_default=2

db_maxConnection_eventprocessor_system=4

queuemanager 服务的 DB 参数

用途：定义 queuemanager 服务的参数。

值：使用整数值。

示例：

db_maxConnection_queuemanager=12

db_minConnection_queuemanager=4

db_maxConnection_queuemanager_default=6

db_maxConnection_queuemanager_system=6

workflow 引擎服务的 DB 参数

用途：定义 workflow 引擎服务的参数。

值：使用整数值。

示例:

db_maxConnection_workflowengine=12

db_minConnection_workflowengine=4

db_maxConnection_workflowengine_default=6

db_maxConnection_workflowengine_system=6

DB 连接池信息

用途: 定义 DB 连接池信息的参数。

值: 最大连接时间是以天数计的。

示例:

db_maxConnTime=1

db_numRetry=3

db_retrySleep=

db_class_name=COM.company.db2.jdbc.app.DB2Driver

设置时区

用途: 定义时区属性。

设置属性: timezone

值: 以小时或秒设置偏移量

示例:

要设置 (GMT-08:00) 太平洋时间 (美国和加拿大); Tijuana (以小时计) 为:

timezone=-8

要设置 (GMT-08:00) 太平洋时间 (美国和加拿大); Tijuana (以分钟计) 为:

timezone=-480

每页的行数

用途: 不同表中的每页行数。如果在该用户的用户设置中未指定任何值, 则将读取这些值。如果这些值也未指定, 则使用硬编码的值 20。

值: 输入整数。

示例:

table_rows_per_page_specs_console=20

table_rows_per_page_scripts_console=20

table_rows_per_page_item_set=20

table_rows_per_page_multi_edit=10

table_rows_per_page_alerts_display=25

table_rows_per_page_lookup_table=20

清空性能缓冲区

用途: 定义多久清空一次性能缓冲区。

值: 单位是对该缓冲区执行的写操作的数目, 即, 在每写 100 次之后进行清空。

示例:

db_perf_dump=100

WebSphere Product Center 日志文件

用途: 定义 WebSphere Product Center 日志文件的相对于 $\${TOP}$ 的位置。

值: 输入 WebSphere Product Center 日志文件的目录。

示例:

dbpool_log_file=logs/db_pool/pool_log.txt

每个提供程序的基本目录 (相对于 $\${TOP}$)

用途: 定义每个提供程序的相对于 $\${TOP}$ 的基本目录。

注意: 基本目录必须以 public_html 开头。

值: 输入每个提供程序的相对于 $\${TOP}$ 的基本目录路径。

示例:

supplier_base_dir=/public_html/suppliers/

FTP 根目录

用途: 定义 FTP 的根目录。

值：FTP 根目录的路径必须以正斜杠 “/” 结尾。

示例：

```
ftp_root_dir=/u01/ftp/
```

调度程序信息

用途：Num_threads 是调度程序工作程序线程的最大数目，master_poll_time 和 scheduler_nap_log_factor 用来调整调度程序在启动作业之后并在查找另一个要运行的调度之前要等待的时间长度。master_poll_time 也被用作对 DB 执行的连续查询之间的等待时间长度。

我们的想法是要避免让单个调度程序启动 10 个作业而另一个调度程序只有 1 个作业的情况。

```
nap_time = master_poll_time + scheduler_nap_log_factor*log  
(count_busy_threads)
```

示例：

Log 是以 10 为底的对数。

当有 1 个忙线程时， $\log(1) = 0$ ， $\text{nap_time} = 5000 + 10000 * 0 = 5$ 秒。

当有 10 个忙线程时， $\log(10) = 1$ ， $\text{nap_time} = 5000 + 10000 * 1 = 15$ 秒。

当有 100 个忙线程时， $\log(100) = 2$ ， $\text{nap_time} = 5000 + 10000 * 2 = 25$ 秒。

参数：

master_poll_time

sch_poll_time

num_threads

值：输入整数值。时间是以毫秒计的。

示例：

```
master_poll_time=5000
```

```
scheduler_nap_log_factor=10000
```

```
sch_poll_time=30000
```

```
num_threads=8
```

工作流引擎工作程序线程

用途：定义工作流引擎工作程序线程的最大数目。

参数：wfe_num_threads

值：输入整数值。

示例：

wfe_num_threads=8

工作流引擎 db 事件轮询时间

用途：定义当没有要运行的工作流引擎时在对 DB2 执行的连续查询之间等待的时间长度。

参数：wfe_db_event_poll_time

值：输入整数值。

示例：

wfe_db_evnt_poll_time=1000

工作流引擎信息

用途：定义工作流引擎时间。

参数：wfl_engine_poll_time

值：输入整数值。

示例：

wfl_engine_poll_time=5000

条目处理器设置

用途：在数据条目屏幕中，项被保存下来，并且宏是在后台运行的。

max_threads_entry_processor 应该是 $< (\text{db_maxConnection_appsvr_default}) / 2$ 。

示例：

max_threads_entry_processor=8

max_jobs_entry_processor=64

事件处理器信息

用途：定义事件处理器时间。

参数: `evp_poll_time`

值: 输入整数值。

示例:

`evp_poll_time=5000`

挂装管理器守护进程信息

用途: 定义在将挂装管理器设置为休眠之前要经过的时间量。

参数: `mountmgr_daemon_sleep_time`

值: 输入整数值。

示例:

`mountmgr_daemon_sleep_time=120000`

EmailBean 的 SMTP 地址

用途: 设置 emailbean 的地址。

`smtp_address=mail.company.com`

`from_address=support@company.com`

支持服务电子邮件地址

用途: 设置从“帮助”菜单访问的支持服务电子邮件地址链接。这指的是客户的内部支持服务电子邮件地址，而不是 IBM WebSphere Product Center 支持服务电子邮件地址。

参数: `support_email_address`

值: 设置为内部支持服务电子邮件地址。

`support_email_address=techsupport@company.com`

版本信息

用途: 定义已安装的 WebSphere Product Center 版本。此值是在 WebSphere Product Center 安装期间自动插入的。不要更改此值。

值: 格式包括 <WebSphere Product Center 版本号 mmddyy>

示例:

`version_info=4.2.0.0 041202`

系统的缺省字符集

用途：定义系统的缺省字符集。UTF-8 字符集编码是 WebSphere Product Center 的一项要求。不要更改此值。

示例：

charset_value=UTF-8

charset_name=UTF-8

缺省情况下在下拉字段中指示的字符集

用途：定义在下拉字段中使用的缺省字符集。

参数：Default_charset_value 和 Default_charset_name

Default_charset_value=Cp1252

Default_charset_name=Default (Cp1252 - ANSI Latin 1)

最大项对象数

用途：定义要从数据库访存的以及存储在内存中以便在 ItemSet.java 中导出的项对象的最大数目。

参数：max_fetch_value

值：输入整数值。

示例：

max_fetch_value=100

要合并的最大项数

用途：定义要合并和清除的项的最大数目。

参数：aggregation_queue_size

值：输入整数值。

示例：

aggregation_queue_size=2000

临时表中的最大项数

用途：定义要在临时表中占用空间的项的最大数目。

参数：aggrgation_batch_size

值：输入整数值。

示例:

```
aggregation_batch_size=100
```

要删除 / 回滚的最大项数

用途: 定义可以删除或回滚的项的最大数目。

参数: delete_rollback_batch_size

值: 输入整数值。

示例:

```
delete_rollback_batch_size=100
```

RMI 端口

用途: 标识所使用的 RMI 端口。

参数: rmi_port

值: 输入所使用的期望 RMI 端口的整数值。

示例:

```
rmi_port=17507
```

注意: 可以通过定义变量 <service_name>_rmi_port=<rmi_port> 来将 WebSphere Product Center 服务与特定 RMI 端口绑定。

即, 要将 “scheduler_01” 绑定为使用端口 “12123”, 请定义以下参数:

```
scheduler_01_rmi_port=12123
```

应用程序服务器变量

用途: 定义应用程序服务器变量。

值: 输入所使用的期望端口的整数值。

示例:

```
appsvr_port=7507
```

```
tomcat_ajp12=21507
```

```
tomecat_ajp13=20507
```

语言环境设置

用途：为运行时的语言支持定义一个相对于 `${TOP}` 的目录，该目录包含每个语言环境的 XML 文件。

值：定义的目录必须以 “/” 结尾。输入未定义语言环境时的缺省语言环境。

示例：

```
locale_xml_top=/locales/
```

缺省语言环境设置

用途：定义未指定语言环境时的缺省语言环境。

值：输入语言环境值。

示例：

```
default_locale=en_US
```

刷新作业状态表

用途：作业状态页面的 `http-equiv` 刷新时间间隔。此参数定义作业状态页面的更新频率。

参数：`job_status_refresh=`

值：单位是以秒计的。

示例：

```
job_status_refresh=30
```

临时目录

用途：定义在本地框使用的临时目录的位置。

值：目录必须以 “/” 结尾。

示例：

```
tmp_dir=/local/boxer/tmp/
```

查找下拉框

用途：定义要出现在查找下拉框中的值的数目。

参数：`max_lookup_dropdown=`

值：输入范围内的整数值。

示例：

`max_lookup_dropdown=25`

最大不活动时间间隔

用途：定义最大不活动时间限制，在这段时间过后，将显示一个对话框以自动地与系统断开连接。

参数： `max_inactive_interval=`

值：以秒计的单位。

示例：

`max_inactive_interval=1800`

脉冲更新时间间隔

用途：定义每个 JVM 的脉动信号的时间间隔。

参数： `pulse_update_interval`

值：缺省值设置为 1 分钟（以毫秒计的时间）。

示例：

`pulse_update_interval=60000`

不活动 JVM 监视器时间间隔

用途：定义 JVM 监视器的不活动时间间隔。

参数： `inactive_jvm_monitor_interval`

值：缺省值设置为 5 分钟（以毫秒计的时间）。

示例：

`inactive_jvm_monitor_interval=300000`

压缩 BLOB

用途：定义是否压缩 BLOB。

参数： `gzip_blobs=`

值： `true / false`

示例：

`gzip_blobs=true`

发送外部消息

用途：定义能否发送外部消息。

参数：remote_message_delivery=on

值：on / off

示例：

```
remote_message_delivery=on
```

OS 级别命令

用途：定义用来运行 OS 级别命令的前缀。

参数：exec_prefix

值：对于 UNIX 系统，保留为空白。

示例：

```
exec_prefix=
```

队列管理器守护进程

用途：所有系统使用的队列管理器守护进程是多线程的，缺省情况下设置为 3。

参数：queue_manager_threads

值：将缺省值设置为 3。

示例：

```
queue_manager_threads=3
```

数据条目工作列表大小

用途：设置数据条目屏幕中显示的数据条目工作列表的最大大小。请根据并行用户的数目以及正在访问的项集合的数量来配置此设置。不要超出在以下示例中出现的变量：

参数：worklist_initial_size_limit 和
worklist_initial_size_limit_sort_all_enabled

值：设置整数值。

示例：

```
# Data Entry Work List Max Size
```

```
worklist_initial_size_limit=5000
```

`worklist_initial_size_limit_with_sort_all_enabled=500`

数据条目：在单一 / 多重编辑之间切换前保存

用途：在单一编辑与多重编辑之间进行切换前，必须保存条目。

参数：`must_save_before_switching_single_multi_edit`

值：设置为 `true` 或 `false`。

`false`（旧行为）— 访存丰富搜索、单一编辑和多重编辑属性集合的并集

`true` — 仅访存适用的属性集合（内存占用量较少）

示例：

`must_save_before_switching_single_multi_edit=true`

数据条目：在页面调度前保存条目

用途：在页面调度到后面的或前面的条目之前，必须保存条目。

参数：`must_save_before_paging_entries=true`

值：设置为 `true` 或 `false`。

`false` — 不需要保存

`true` -> 不修改任何不可视的条目（内存占用量较少）

示例：

`must_save_before_paging_entries=true`

启用继承

用途：打开 / 关闭继承功能。还应该启用子规范。（在 4.2.1 中，未提供继承。）

参数：`enable_inheritance`

值：设置为 `yes` 或 `no`。

示例：

`enable_inheritance=yes`

启用子规范

用途：打开 / 关闭子规范功能，该功能用于继承。（在 4.2.1 中，未提供继承。可以在各种与继承无关的数据建模方案中使用子规范。）

参数: enable_subspecs

值: 设置为 yes 或 no。

示例:

enable_subspecs=yes

队列管理器服务

用途: 定义队列管理器参数。

参数: queuemanager_poll_time、queuemanager_num_threads 和 queuemanager_max_jobs

值: 设置为整数值。

示例:

queuemanager_poll_time=5000

queuemanager_num_threads=10

queuemanager_max_jobs=1000

IBM MQ 上的 JMS

用途: IBM MQ 上的 JMS 的实现需要下列参数。

参数: jms_provider 和 jms_receive_timeout

值: 设置为以下示例中显示的值。

示例:

jms_provider=IBM WebSphere MQ

jms_receive_timeout=1000

workflow 条目可编辑

用途: 将 workflow 条目设置为可编辑。

参数: workflow_with_entries_editable

值: 缺省情况下, 带有条目的 workflow 是不可编辑的。

no — 带有条目的 workflow 是不可编辑的

yes — 带有条目的 workflow 是可编辑的

示例:

`workflow_with_entries_editable=no`

JMS 端口和队列设置

用途：设置 WebSphere Product Center 使用的入站 / 出站队列以及设置所使用的 MQ 端口：

参数：`jms_inbound_queue`、`jms_outbound_queue` 和 `mq_port`

示例：

`jms_inbound_queue=WPC.MINIME.QUEUE`

`jms_outbound_queue=WPC.MINIME.QUEUE`

`mq_port=1414`

MQ 设置

用途：MQ 的实现直接需要这些设置，IBM MQ 上的 JMS 也需要这些设置。

参数：`mq_channel`、`mq_hostname` 和 `mq_queuemanager`

`mq_channel=WPC.JAVA.CHANNEL`

`mq_hostname=minime`

`mq_queuemanager=WPC.MINIME.QMGR`

`mq_inbound_queue=WPC.MINIME.QUEUE`

`mq_outbound_queue=WPC.MINIME.QUEUE`

如果缺省值不可接受，则指定下列参数。

`mq_queue_put_open_options=`

`mq_message_put_options=`

`mq_queue_get_open_options=`

`mq_message_get_options=`

脚本编制操作也可以传入这些参数。必须在这里指定整数值。除非要在代码中使用有效值来覆盖缺省值，否则不要将这些参数注释掉。即使指定了这些参数，脚本编制操作传入的值也将覆盖它们。

消息的字符集

用途：设置用于消息的字符集。

参数: mq_use_utf 和 mq_charset

示例:

mq_use_utf=false

mq_charset=819

限制规范的高速缓存大小

用途: 设置在 WebSphere Product Center 的用户界面中显示的静态高速缓存的规范定义的大小。这个值影响所使用的内存量。请根据系统需求来设置高速缓存大小。

参数: max_specs_in_cache

值: 设置整数值。

示例:

max_specs_in_cache=200

启用挂装管理器

用途: 仅当外部进程对文档库中挂装的文件系统目录添加 / 删除文件时, 挂装管理器才有用。

参数: enable_mountmgr

值: true / false

enable_mountmgr=true

设置每次出现的运行规则

用途: 允许每次都计算多次出现字符串枚举规则值。如果不希望这样做, 则设置为 “false”。

参数: run_rule_per_occurrence

run_rule_per_occurrence=true

为组 / 本地化属性创建唯一的标识

用途: 将这些参数设置为 true 将自动地为多次出现的组和本地化属性保存唯一的标识。简而言之, 这允许使用脚本操作 `getChangedAttributesForMultiOccurrence` 来准确地进行区分。

参数: create_unique_key_for_multi_occurrence_groups 和
create_unique_key_for_multi_occurrence_localized

值: true / false

示例:

```
create_unique_key_for_multi_occurrence_groups=true
```

```
create_unique_key_for_multi_occurrence_localized=true
```

限制查找表和视图对象

用途：内存设置，用于限制可以与单一会话存储在一起的查找表和容器视图对象的数目。为了提高性能，这些对象与会话存储在一起，但是，这会增加每个会话的内存占用量，因此，最好设置保守的数值。

参数：max_lookups_in_cache 和 max_ctgviews_in_cache

值：输入整数值。

示例:

```
max_lookups_in_cache=10
```

```
max_ctgviews_in_cache=10
```

```
max_roles_in_cache=50
```

```
max_accesses_in_cache=500
```

启用内存监视

用途：启用对会话内存使用情况进行的内存监视以分析会话的内存占用量。

参数：enable_memorymonitor 和 memorymonitor_interval

值：如果启用的话，设置为“true”并对 interval 输入整数值。

示例:

```
enable_memorymonitor=true
```

```
memorymonitor_interval=50000
```

显示装入屏幕

用途：允许 / 禁止在页面导航之间显示装入屏幕并且在页面装入完成前不允许用户导航到别处。将此参数设置为“true”可以防止过多用户同时对服务器进行操作。

参数：display_loading_screen

值：true / false

示例:

display_loading_screen=false

收集屏幕和作业的概要分析信息

用途：收集屏幕和作业的概要分析信息。

参数：profiling_info_collection_depth=1

值：-1 / 0 / 1 / 2 / 3 / ... / 100（-1 指示不收集任何概要分析信息；深度值以 0 开始）

Profiling_info_collection_depth=1

收集已调度作业的其它概要分析

用途：收集已调度作业的其它概要分析信息。

参数：profiling_scheduled_jobs

值：none / nodebuglog / full

Profiling_scheduled_jobs=full

窗口小部件名

用途：使用窗口小部件长名称。缺省情况下，此参数设置为 false。

参数：debug_use_long_widget_names

值：true / false

debug_use_long_widget_names=false

DB 连接池接口

用途：对于 WebSphere Product Center，ThinPoolDBManager 不再是缺省 DB 连接池接口层。因此，缺省值设置为“false”。要切换回使用 ThinPoolDBManager，请将其设置为“true”。

参数：debug_profile_db_connections

值：true / false

示例：

debug_profile_db_connections=true

同步锁设置

用途：同步锁的设置，同步锁使线程在临界区等待，直到正在对临界区执行操作的当前线程的操作完成为止。每经过一小段时间（由 wait_poll_time 定义），一个等待线程就进行轮询以检查临界区是否空

闲，如果临界区空闲，该线程就锁定临界区以进行使用。线程在被锁定临界区上可以等待的最大时间长度是由 `wait_max_time` 指定的，在这段时间过后，将会因为超时而抛出异常。

参数: `wait_poll_time` 和 `wait_max_time`

值: 输入的值是以毫秒计的。缺省情况下，`wait_poll_time` 是 1 秒，`wait_max_time` 是 1 分钟。

`wait_poll_time=1000`

`wait_max_time=60000`

设置为检查脚本中的变量

用途: 设置为声明要在脚本中使用 `var` 声明的变量。如果设置为 `true`，则必须使用 `var` 来声明脚本变量。

参数: `script_check_variables`

值: `true` / `false`

`script_check_variables=false`

设置为修剪条目属性值

用途: 设置为修剪条目属性值。例如，将 “xyz ” 保存为 “xyz”。

参数: `trim_entry_attribute_values`

值: `true` / `false`

`trim_entry_attribute_values=false`

显示条目属性列表

用途: 设置为在 “高级选择编辑器” 屏幕上显示条目属性列表。

参数: `display_attributes_in_rule_editor`

值: `true` / `false`

`Display_attributes_in_rule_editor=true`

设置不可变规范

用途: 设置 `getCtgSpec` 和 `getCatalogSpec` 的缺省行为以获取不可变规范或可变规范。不可变规范不能被修改，但它们重用共享高速缓存。可变规范可以被修改。

注意: 可以通过将一个布尔值传递给 `getCtgSpec/getCatalogSpec` 来覆盖此选项。

参数: `get_immutable_specs`

值: `true / false`

`Get_immutable_specs=false`

关闭事件激发

用途: 打开和关闭事件激发。将此标志设置为 `false` 将停止激发事件。

参数: `fire_event_processor_events`

值: `true / false`

`fire_event_processor_events=true`

设置在父代下显示的节点的上限

用途: 对在左导航窗格（也称为目录资源管理器）中的一个父节点下显示的节点数设置上限，或者不设置上限（0）。

参数: `leftnav_max_categories` 和 `leftnav_max_items`

值: 输入整数。

`levtnav_max_categories=0`

`leftnav_max_items=100`

设置标准 URL

用途: 设置 Web 站点的标准 URL（包括端口号），用户应该将他们的浏览器指向这个 Web 站点以访问 WebSphere Product Center 实例。不应该包括尾部的 “/” 字符。如果将此值保留为空值，则将根据应用程序服务器的主机名和端口推断此值。对于 WebSphere，不要将此值保留为空值。

参数: `trigo_web_url`

值: 输入带有端口号的 URL。

`trigo_web_url=http://bach.qa.company.com:7507`

文件名: **admin_properties.xml**

用途: 管理实用程序使用此文件来配置 WebSphere Product Center 的集群。

规则：

列示作为集群一部分的不同主机

只能在集群中的机器上启动服务

如果在集群中只有一台主机，则使用“localhost”，否则使用真实的机器名

示例：

```
<!-- %DISCLAIMER% -->
```

```
<admin>
```

```
<!--
```

列示作为集群一部分的不同主机。

只能在集群中的机器上启动服务。

当集群中只有一台主机时，应该只使用“localhost”。

否则，应该使用真实的机器名。

示例：

```
<cluster>
```

```
<host name="server1"/>
```

```
<host name="server2"/>
```

```
<host name="server3"/>
```

```
</cluster>
```

```
<cluster>
```

```
<host name="localhost"/>
```

```
</cluster>
```

```
</admin>
```

文件名： **docstore_mount.xml**

用途：系统挂装管理器需要此文件以获取各种 OS 文件系统挂装点的位置。

示例：

```
<?xml version="1.0"?>
```

```
<mnts>
```

```
<mnt doc_path="/public_html/" real_path="$supplier_base_dir"/>
<mnt doc_path="/ftp/" real_path="$supplier_ftp_dir"/>
</mnts>
```

文件名: data_entry_properties.xml

此文件包含当使用单一编辑时显示在数据条目屏幕中的附加框架的信息。

对于一个公司的给定目录或层次结构，可以定义一组脚本，将执行这些脚本以显示附加框架。

有两种类型的脚本：

- **url:** 脚本将由应该返回 url 的 `getURL(entry)` 函数组成。将添加指向这个 url 的新 `iframe`。
- **内容:** 脚本将由应该返回某些 html 内容的 `getContent(entry)` 函数组成。将添加显示此内容的新 `div`。

对于每个脚本，将需要指定：

type (url/content)

title

path (通常是 /scripts/triggers/<script name>)

要传递给 `iframe` 或 `div` html 元素的某些附加 html 信息 (可选)。

示例:

```
<xml>
```

```
<company code="WPC">
```

```
<catalog name="ctg1">
```

```
<script>
```

```
<type>url</type>
```

```
<extra>height='150'</extra>
```

```
<title>for ctg1 - 1</title>
```

```
<path>/scripts/triggers/test_ctg1b</path>
```

```
</script>
```

```
<script>
```

```
<type>content</type>
```

```
<title>for ctg1 - 1</title>
```

```
path>/scripts/triggers/test_ctg1b</path>
```

```
</script>
```

```
</catalog>

<hierarchy name="h1">
<script>
<type>url</type>
<title>for h2 - 1</title>
<path>/scripts/triggers/test_h2</path>
</script>
</hierarchy>

</company>
</xml>
```