

WebSphere Application Server



高速缓存代理管理指南

版本 6.0

WebSphere Application Server



高速缓存代理管理指南

版本 6.0

注意

在使用本资料及其支持的产品之前，请务必阅读第 253 页的『声明』中的一般信息。

第一版（2004 年 11 月）

此版本适用于：

WebSphere Application Server，版本 6.0

及所有后续发行版和修订版，直到在新版本中另有声明为止。

通过您当地的 IBM 代表或 IBM 分部可订购出版物。

© Copyright International Business Machines Corporation 2004. All rights reserved.

目录

图	ix
关于本书	xi
本书的阅读对象	xi
本书中使用的约定和术语	xi
辅助选项	xi
如何发送您的意见	xii
相关信息	xii
第 1 部分 高速缓存代理入门	1
第 1 章 概述	3
新功能部件	3
第 2 章 使用“配置和管理”表单	7
浏览器要求	7
访问“配置和管理”表单	7
设置管理员密码	9
第 3 章 使用配置向导	11
第 4 章 手工编辑 ibmproxy.conf 文件	13
第 5 章 启动和停止高速缓存代理	15
在 Linux 和 UNIX 系统上自动启动和关闭	15
在 Linux 和 UNIX 系统上手工启动	16
在 AIX 上:	16
在 HP-UX 上:	16
在 Linux 上:	16
在 Solaris 上:	17
作为 Windows 服务启动	17
作为 Windows 应用程序启动	18
使用“开始”菜单	18
使用命令提示符	18
启动多个代理服务器	18
在 Linux 和 UNIX 系统上手工关闭	19
关闭命令的限制	19
在 Windows 系统上手工关闭	20
配置更改后重新启动	20
第 2 部分 配置和调整高速缓存代理进程	21
第 6 章 定义服务器	23
关联的伪指令	24
“配置和管理”表单	24
第 7 章 建立进程所有权	25
关联的伪指令	25
“配置和管理”表单	25

第 8 章 管理连接	27
关联的伪指令	28
“配置和管理”表单	28
第 9 章 调整代理服务器进程	31
设置与性能关联的伪指令	31
检查其它应用程序	31
验证调页空间	31
调整文件系统	32
调整 TCP/IP 配置	32
调整高负载环境的 TCP 时间等待时间间隔 (HP-UX、Linux、Solaris 和 Windows)	32
调整 Linux 内核	33
调整 AIX 线程调整变量	33
第 3 部分 配置高速缓存代理行为	35
第 10 章 管理请求处理	37
启用 HTTP/FTP 方法	37
关联的伪指令	38
“配置和管理”表单	39
定义映射规则	39
映射规则	39
配置代理服务器	41
关联的伪指令	41
“配置和管理”表单	41
启用结点重写 (可选)	41
定义不具有 JunctionPrefix 选项的结点	41
定义具有 JunctionPrefix 选项的结点 (建议的方法)	42
关联的伪指令	43
“配置和管理”表单	43
使用 Cookie 替代 JunctionRewrite	44
用于扩展 JunctionRewrite 功能的样本 transmogriifier 插件	44
第 11 章 管理本地内容的传递	47
定义文档根目录	47
关联的伪指令	47
“配置和管理”表单	47
定义缺省欢迎页面	48
关联的伪指令	48
“配置和管理”表单	48
第 12 章 管理 FTP 连接	49
保护 FTP 文件	49
管理 FTP 服务器登录	49
管理 FTP 目录路径	50
管理 FTP 链接	50

第 13 章 定制服务器处理	51
服务器端包含	51
服务器端包含的注意事项	51
服务器端包含的配置	51
服务器端包含的格式	52
服务器端包含的伪指令	52
定制错误消息	57
实时流式协议 (RTSP) 重定向	58
关于 RTSP 重定向	58
RTSP 限制	58
RTSP 增强	58
配置 RTSP 重定向	58
第 14 章 配置头选项	59
关联的伪指令	59
“配置和管理” 表单	59
第 15 章 关于应用程序编程接口	61
关联的伪指令	61
“配置和管理” 表单	61
<hr/>	
第 4 部分 配置代理服务器高速缓存	63
第 16 章 代理服务器高速缓存概述	65
高速缓存存储	65
高速缓存索引	65
FTP 高速缓存	66
DNS 高速缓存	66
高速缓存排除	66
高速缓存管理	66
第 17 章 配置基本高速缓存	69
1. 启用高速缓存	69
2. 配置高速缓存存储	69
可选定制	70
设置高速缓存	70
保存或装入高速缓存内存到磁盘	71
设置高速缓存过滤器	71
配置用于查询结果和动态生成文件的高速缓存	71
配置文件到期和垃圾收集	71
配置自动预装入	71
配置高速缓存共享	71
配置日志记录	72
第 18 章 控制高速缓存哪些内容	73
配置基于 URL 的高速缓存过滤器	73
高速缓存查询响应	73
查询响应高速缓存的附加要求	74
高速缓存本地提供服务的文件	74
根据部分 URL 高速缓存文件	75
相关的配置文件伪指令	75
第 19 章 维护高速缓存内容	77
文件到期	77
有关高速缓存新鲜度的其它信息	78

有关 FTP 中的日期	78
配置高速缓存新鲜度	79
垃圾收集	80
配置垃圾收集	80

第 20 章 为自动刷新和预装入配置高速缓存代理程序	83
设置服务器主机名	83
用特定文件预装入高速缓存	84
用频繁高速缓存的文件预装入高速缓存	84
挖掘	85
相关代理配置文件伪指令	86
手工启动高速缓存代理程序	87

第 21 章 使用共享高速缓存	89
远程高速缓存访问	89
配置远程高速缓存访问	90
配置因特网高速缓存协议插件	90
配置 ICP 插件	90

第 22 章 动态地高速缓存生成的内容	93
为代理高速缓存配置 IBM WebSphere Application Server	94
在应用程序服务器配置动态高速缓存	94
配置应用程序服务器适配器	94
配置动态高速缓存的高速缓存代理	94
设置 Service 伪指令启用动态高速缓存插件	94
设置 ExternalCacheManager 伪指令以指定文件源	95

第 23 章 调整代理服务器高速缓存	97
选择高速缓存存储介质	97
优化磁盘高速缓存性能	97
高速缓存垃圾收集	97
平台特定的最优化	97
AIX	98
HP-UX 和 Solaris	98
Windows	98

第 5 部分 配置高速缓存代理安全性 99

第 24 章 关于代理服务器安全性	101
--------------------------	------------

第 25 章 服务器保护设置	103
使用“配置和管理” 表单设置保护	103
使用配置文件伪指令设置保护	104
缺省保护设置	105

第 26 章 安全套接字层 (SSL)	107
SSL 握手	107
配置安全远程管理	108
密钥和证书管理	108
认证中心	108
使用 IBM 密钥管理器实用程序	109
创建新的密钥数据库、密码和存储文件	110
接收 CA 证书	114

存储 CA 证书	114
支持的密码规范	115
第 27 章 启用加密硬件的支持	117
第 28 章 使用 Tivoli Access Manager 插件	119
配置	119
使用配置脚本前要执行的步骤	119
使用配置脚本	119
启动高速缓存代理和 Access Manager 插件	120
第 29 章 使用 PAC-LDAP 授权模块	121
概述	121
认证	121
授权	121
轻量级目录访问协议 (LDAP)	121
安装	122
安全 PACD-LDAP 服务器连接的附加要求	123
LDAP 客户机包需要 GSKit	123
必须为 Linux 系统设置 LD_PRELOAD 环境变量	123
编辑 ibmproxy.conf 文件以启用 PAC-LDAP 授权模块	123
编辑 PAC-LDAP 授权模块配置文件	124
paccp.conf	125
pac.conf	125
pacpolicy.conf	125
创建 pac_ldap.cred	126
启动和停止 pacd	126

第 6 部分 监视高速缓存代理 129

第 30 章 配置日志记录 131

关于日志	131
日志文件名和基本选项	131
访问日志过滤器	132
控制所记录的内容的原因	132
配置访问日志过滤器	132
缺省日志设置	133
维护和归档日志	134
日志文件方案	135

第 31 章 使用服务器活动监视器 137

附录 A. 使用高速缓存代理命令 141

cgiparse 命令	142
cgiutils 命令	145
htadm 命令	147
htcformat 命令	150
ibmproxy 命令	152

附录 B. 配置文件伪指令 155

重新启动时不更改的伪指令	155
伪指令概述	155
可接受的值	156

配置文件记录的语法	156
高速缓存代理伪指令	157
AcceptAnything - 服务所有文件	157
AccessLog - 命名访问日志文件的路径	157
AccessLogExcludeMethod - 抑制通过指定方法请求的文件或目录的日志条目	158
AccessLogExcludeMimeType - 禁止特定 MIME 类型的代理访问日志条目	158
AccessLogExcludeReturnCode - 抑制特定返回码的日志条目	159
AccessLogExcludeURL - 抑制指定文件或目录的日志条目	159
AccessLogExcludeUserAgent - 抑制来自特定浏览器的日志条目	159
AddBlankIcon - 指定用于对齐目录列表标题的图标 URL	160
AddDirIcon - 指定目录列表上目录的图标 URL	160
AddEncoding - 为具有特殊后缀的文件指定 MIME 内容编码	161
AddIcon - 将图标绑定到 MIME 内容类型或编码类型	161
AddParentIcon - 指定代表目录列表上父目录的图标 URL	162
AddType - 为具有特殊后缀的文件指定数据类型	162
AddUnknownIcon - 指定目录列表上未知文件类型的图标 URL	163
AdminPort - 指定请求管理页面或表单的端口	164
AggressiveCaching - 为非高速缓存文件指定高速缓存	164
AlwaysWelcome - 指定是否要为欢迎文件搜索所请求的目录	165
appendCRLFtoPost - 附加 CRLF 到 POST 请求	165
ArrayName - 命名远程高速缓存数组	165
Authentication - 定制认证步骤	166
Authorization - 定制授权步骤	166
AutoCacheRefresh - 指定是否要使用高速缓存刷新	167
BindSpecific - 指定服务器是否绑定到一个或所有 IP 地址	167
BlockSize - 指定在高速缓存中的块大小	167
CacheAccessLog - 指定高速缓存访问日志文件的路径	167
CacheAlgorithm - 指定高速缓存算法	168
CacheByIncomingUrl - 指定生成高速缓存文件名的基础	168
CacheClean - 指定要保留高速缓存文件多久	169
CacheDefaultExpiry - 指定文件的缺省到期时间	169
CacheDev - 指定高速缓存的存储设备	169
CacheExpiryCheck - 指定服务器是否返回到期文件	170
CacheFileSizeLimit - 指定要高速缓存文件的最大大小	170
CacheLastModifiedFactor - 指定值以确定到期日期	170
CacheLocalDomain - 指定是否要高速缓存本地域	171

CacheMatchLanguage — 指定返回的高速缓存内容的语言首选项	171
CacheMaxExpiry - 指定高速缓存文件的最大生命周期	172
CacheMemory - 指定高速缓存 RAM	173
CacheMinHold - 指定文件的保留时间	173
CacheNoConnect - 指定独立高速缓存方式	174
CacheOnly - 仅高速缓存其 URL 与模板匹配的那些文件	174
CacheQueries - 指定高速缓存对包含问号 (?) 的 URL 的响应	174
CacheRefreshInterval - 指定重新生效高速缓存对象的时间间隔	175
CacheRefreshTime - 指定何时启动高速缓存代理程序	175
CacheTimeMargin - 指定高速缓存文件的最小生命周期	175
CacheUnused - 指定未使用的高速缓存文件要保留多久	176
Caching - 启用代理高速缓存	176
CdfAware - 指定高速缓存代理的实例作为内容分发框架的部件	176
CdfRestartFile - 指定文件将文件名存储到 url 映射	177
CompressAge - 指定何时压缩日志	177
CompressCommand - 指定压缩命令和参数	177
CompressDeleteAge - 指定何时删除日志	178
ConfigFile — 指定其它配置文件的名称	179
ConnThreads — 指定要用于连接管理的连接线程数	179
ContinueCaching - 指定需要高速缓存一个文件的百分率	179
DefinePicsRule - 提供内容过滤规则	180
DefProt - 为与模板匹配的请求指定缺省保护设置	180
DelayPeriod - 指定请求之间的暂停	182
DelveAcrossHosts - 指定跨域高速缓存	182
DelveDepth - 指定高速缓存时要逐层访问的链接层次	182
DelveInto - 指定高速缓存代理程序是否要逐层访问链接	182
DirBackgroundImage - 为目录列表指定背景图像	183
DirShowBytes - 在目录列表上显示小文件的字节计数	183
DirShowCase - 对目录列表上的文件排序时使用大小写	183
DirShowDate - 在目录列表上显示最后修改日期	184
DirShowDescription - 在目录列表上显示文件的描述	184
DirShowHidden - 在目录列表上显示隐藏文件	184
DirShowIcons - 在目录列表上显示图标	184
DirShowMaxDescrLength - 指定目录列表上描述的最大长度	185
DirShowMaxLength - 指定目录列表上文件名的最大长度	185

DirShowMinLength - 指定目录列表上文件名的最小长度	185
DirShowSize - 在目录列表上显示文件大小	185
Disable - 禁用 HTTP 方法	185
DisInheritEnv - 指定 CGI 程序不继承的环境变量	186
DNS-Lookup - 指定服务器是否查找客户机主机名	186
Enable - 启用 HTTP 方法	187
EnableTcpNodelay — 启用 TCP NODELAY 套接字选项	187
Error - 定制 Error 步骤	187
ErrorLog - 指定记录服务器错误的文件	188
ErrorPage - 为特定错误条件指定定制消息缺省值	189
EventLog - 指定事件日志文件的路径	190
Exec - 运行 CGI 程序以匹配请求	190
ExportCacheImageTo - 将高速缓存内存导出到磁盘	191
ExternalCacheManager - 对 IBM WebSphere Application Server 中的动态高速缓存配置高速缓存代理	191
Fail - 拒绝匹配请求	192
flexibleSocks - 启用灵活的 SOCKS 实现	193
FTPDirInfo - 生成目录的欢迎或描述消息	193
ftp_proxy - 指定 FTP 请求的另一个代理服务器	194
FTPUrlPath - 指定如何解释 FTP URL	194
Gc - 指定垃圾收集	194
GCAdvisor - 定制垃圾收集过程	195
GcHighWater - 指定垃圾收集何时开始	195
GcLowWater - 指定垃圾收集何时结束	195
gopher_proxy - 指定 Gopher 请求的另一个代理服务器	195
GroupId - 指定组标识	196
HeaderServerName - 指定 HTTP 头中返回的代理服务器名称	196
Hostname - 指定服务器的全限定域名或 IP 地址	196
http_proxy - 为 HTTP 请求指定另一个代理服务器	197
HTTPSCheckRoot - 过滤 HTTPS 请求	197
ICP_Address - 指定 ICP 查询的 IP 地址	197
ICP_MaxThreads - 指定 ICP 查询的最大线程	198
Occupier - 指定 Occupier 群集的成员	198
ICP_Port - 指定 ICP 查询的端口号	198
ICP_Timeout - 指定 ICP 查询的最大等待时间	199
IgnoreURL - 指定不刷新的 URL	199
imbeds - 指定是否使用服务器端包含处理	199
ImportCacheImageFrom - 从文件导入高速缓存内存	200
InheritEnv - 指定 CGI 程序继承的环境变量	200
InputTimeout - 指定输入超时	201
JunctionReplaceUrlPrefix — 当与 JunctionRewrite 插件协同使用时, 替换 URL 而不是插入前缀	201
JunctionRewrite - 启用 URL 重写	202

JunctionRewriteSetCookiePath — 当与 JunctionRewrite 插件协同使用时, 重写 Set-Cookie 头中的路径选项	202	PersistTimeout - 指定等待客户机发送另一个请求的时间	218
JunctionSkipUrlPrefix — 当与 JunctionRewrite 插件协同使用时, 跳过已包含前缀的重写 URL	202	PICSDbLookup - 定制 PICS 标号检索步骤	218
KeepExpired - 修改代理高速缓存的垃圾收集	203	PidFile (仅 Linux 和 UNIX) — 指定在其存储高速缓存代理的进程标识的文件	219
KeyRing - 指定到密钥环数据库的文件路径	203	插件模块伪指令	219
KeyRingStash - 指定到密钥环数据库的密码文件的文件路径	203	Port - 指定服务器侦听请求的端口	220
LimitRequestBody — 指定 PUT 或 POST 请求中最大主体大小	204	PostAuth - 定制 PostAuth 步骤	220
LimitRequestFields — 指定客户机请求中的最大标题数	204	PostExit - 定制 PostExit 步骤	221
LimitRequestFieldSize — 指定最大标题长度和请求行	204	PreExit - 定制 PreExit 步骤	221
ListenBacklog - 指定服务器可执行的侦听备用客户机连接数	204	Protect - 为匹配模板的请求激活保护设置	221
LoadInlineImages - 控制嵌入图像的刷新	205	Protection - 在配置文件中定义已命名的保护设置	225
LoadTopCached - 指定要刷新的受欢迎页面数	205	Protection 子伪指令 - 指定一组资源如何受保护	226
LoadURL - 指定要刷新的 URL	205	Proxy - 指定代理协议或逆向代理	228
Log - 定制 Log 步骤	205	ProxyAccessLog - 命名代理访问日志文件的路径	229
LogArchive - 指定日志归档的行为	206	ProxyAdvisor - 定制代理请求的服务	229
LogFileFormat - 指定访问日志格式	206	ProxyForwardLabels - 指定 PICS 过滤	230
LogToGUI (仅 Windows) - 在服务器窗口中显示日志条目	207	ProxyFrom - 指定具有 From: 头的客户机	230
LogToSyslog — 指定是否将访问信息发送到系统日志 (仅 Linux 和 UNIX)	207	ProxyIgnoreNoCache - 忽略重新装入的请求	230
Map - 将匹配请求更改为新的请求字符串	208	ProxyPersistence - 允许持久连接	231
MaxActiveThreads - 指定有效线程的最大数	209	ProxySendClientAddress - 生成 Client IP Address: 头	231
MaxContentLengthBuffer - 指定动态数据的缓冲区大小	209	ProxyUserAgent - 修改 “User Agent” 字符串	231
MaxLogFileSize - 指定每个日志文件的最大大小	209	ProxyVia - 指定 HTTP 头的格式	232
MaxPersistRequest - 指定一个持久连接上接收的最大请求数	210	ProxyWAS - 指定请求是发送到 WebSphere Application Server	232
MaxQueueDepth - 指定队列中最大 URL 数	210	PureProxy - 禁用专用的代理	232
MaxRuntime - 指定高速缓存代理程序运行的最大时间	210	PurgeAge - 指定日志的时效	233
MaxSocketPerServer - 指定服务器的最大打开套接字数	211	PurgeSize - 指定日志压缩文档的大小限制	233
MaxUrls - 指定要刷新的最大 URL 数	211	RCAConfigFile - 指定 ConfigFile 的别名	234
Member - 指定一个数组成员	211	RCAThreads - 指定每个端口的线程数	234
Midnight - 指定用于归档日志的 API 插件	212	ReadTimeout - 指定连接的时间限制	234
NameTrans - 定制名称转换步骤	212	Redirect - 为发送到另一个服务器的请求指定模板	235
NoBG - 在前台运行高速缓存代理进程	213	ReversePass - 拦截自动重定向请求	236
NoCaching - 指定不高速缓存其 URL 与模板匹配的那些文件	213	RewriteSetCookieDomain — 指定需要重写的域模式	236
NoLog - 抑制匹配模板的特定的主机或域的日志条目	214	RTSPEnable - 启用 RTSP 重定向	237
no_proxy - 指定直接连接到域的模板	214	rtsp_proxy_server - 指定用于重定向的服务器	237
NoProxyHeader - 将客户机头指定到块	215	rtsp_proxy_threshold - 指定在重定向到高速缓存之前的请求数	237
NumClients - 指定要使用的高速缓存代理程序线程数	215	rtsp_url_list_size - 指定代理内存中的 URL 数	237
ObjectType - 定制对象类型步骤	215	ScriptTimeout - 指定脚本的超时设置	238
OutputTimeout - 指定输出超时	216	SendHTTP10Outbound - 指定代理请求的协议版本	238
PacFilePath - 指定包含 PAC 文件的目录	216	SendRevProxyName - 在 HOST 头指定高速缓存代理主机名	239
Pass - 指定接受请求的模板	216	ServerConnGCRun - 指定运行垃圾收集线程的时间间隔	239
		ServerConnPool - 将连接合用指定到源服务器	239
		ServerConnTimeout - 指定最大的非活动的周期	240
		ServerInit - 定制服务器初始化步骤	240
		ServerRoot - 指定服务器程序安装的目录	240
		ServerTerm - 定制服务器终止步骤	241
		Service - 定制服务步骤	241

SignificantURLTerminator - 为 URL 请求指定终止代码	242
SMTPServer (仅 Windows) - 设置发送邮件例程的 SMTP 服务器	242
SNMP - 启用和禁用 SNMP 支持	243
SNMPCommunity - 为 SNMP 提供安全性密码	243
SSLCaching - 为安全请求启用高速缓存	243
SSLCertificate - 为证书指定密钥标号	243
SSLCryptoCard - 指定已安装的加密卡	244
SSLEnable - 为安全请求指定在端口 443 上侦听	244
SSLForwardPort - 指定 HTTP SSL 升级的寻址端口	244
SSLOnly - 对 HTTP 请求禁用侦听器线程	245
SSLTunneling - 启用 SSL 隧道	245
SSLVersion - 指定 SSL 的版本	245
SSLV2Timeout - 指定 SSLV2 会话到期之前要等待的时间	245
SSLV3Timeout - 指定 SSLV3 会话到期之前要等待的时间	246

SuffixCaseSense - 指定后缀定义是否区分大小写	246
TLSV1Enable - 启用传送层安全协议	246
Transmogriifier - 定制数据操作步骤	247
TransmogriifiedWarning - 发送警告消息到客户机	247
TransparentProxy - 在 Linux 或 AIX 上启用透明代理	247
UpdateProxy - 指定高速缓存目标	248
UserId - 指定缺省用户标识	248
V2CipherSpecs - 列出 SSL V2 的支持密码规范	249
V3CipherSpecs - 列出 SSL V3 的支持密码规范	249
WebMasterEMail - 设置电子邮件地址以接收选择服务器报告	250
WebMasterSocksServer (仅 Windows) - 设置发送邮件例程的 socks 服务器	250
Welcome - 指定欢迎文件的名称	251

声明	253
商标	254



1. 挖掘 85

关于本书

本前言描述本书面向的读者和目的、它的组织、辅助选项功能、约定和术语，以及相关文档。

本书的阅读对象

《高速缓存代理管理指南》是为熟悉操作系统和因特网服务的、有经验的网络和系统管理员编写的。用户不需要事先了解高速缓存代理。

本书并不支持高速缓存代理的前发行版。

本书中使用的约定和术语

本文档使用下列印刷和键入约定。

表 1. 本书中使用的约定

约定	含义
粗体	当提及图形用户界面 (GUI) 时，粗体用于表明菜单、菜单项、标签、按钮、图标和文件夹。它还用于强调命令名，否则它们可能会与周围的文本相混淆。
等宽字体	表明您必须在命令提示符下输入的文本。等宽字体还表明屏幕文本、代码示例和文件摘录。
斜体	表明您必须提供的变量值 (例如，您为 <i>fileName</i> 提供的文件名)。斜体还表明强调和书标题。
Ctrl- <i>x</i>	其中 <i>x</i> 是键的名称，表明“control-字符”顺序。例如，Ctrl-c 意味着在按下 c 键时已按住 Ctrl 键。
Return	指标有字 Return、字 Enter 或左箭头的键。
%	表示 Linux™ 和 UNIX® 命令行环境提示符，此命令不需要 root 用户特权。
#	表示 Linux 和 UNIX 命令行环境提示符，此命令需要 root 用户特权。
C:\	表示 Windows® 命令提示符。
输入命令	当指示“输入”或“发出”某条命令时，输入此命令然后按 Return 键。例如，指示“输入 ls 命令”意味着在命令提示符下输入 ls ，然后按 Return 键。
[]	括住语法描述中的可选项。
{ }	括住列表，从中您必须选择一个语法描述项。
	分隔语法描述中用 { } (大括号) 括住的选项列表中的项。
...	语法描述中的省略号表明您可以一次或多次重复先前项。示例中的省略号表明为了简明扼要起见而从示例中省略的信息。

辅助选项

辅助选项功能部件帮助有生理残疾的用户 (如行动不便或有视力障碍的用户) 顺利地使用软件产品。以下这些是 WebSphere® Application Server, 版本 6.0 的主要辅助选项功能部件:

- 您可以使用屏幕阅读器软件和数字语音合成器聆听屏幕上显示的内容。您还可以使用语音识别软件（如 IBM ViaVoice）来输入数据并浏览用户界面。
- 您可以使用键盘代替鼠标来操作功能部件。
- 您可以通过标准文本编辑器或命令行界面而不是所提供的图形界面来配置和管理 Application Server 功能部件。要获取有关特殊功能部件的辅助选项的更多信息，请参阅那些功能部件的相关文档。

如何发送您的意见

您的反馈十分重要，它有助于我们提供最准确而高质量的信息。如果您对本书或其它任何有关 WebSphere Application Server 的 Edge components 文档有任何意见：

- 请将您的意见通过电子邮件发送到 fsdoc@us.ibm.com。确保指明书名、书号、WebSphere Application Server 的版本，如果可以的话，还请指明您所评论的文本所在的特定位置（例如，页码或表号）。

相关信息

- 《Edge Components 的概念、规划和安装》，G151-0062-00
- *Programming Guide for Edge Components*, GC31-6856-00
- 《Load Balancer 管理指南》，G151-0064-00
- *IBM WebSphere Edge Services Architecture*
- IBM 主 Web 站点: www.ibm.com/
- IBM® WebSphere Application Server 产品 Web 站点:
www.ibm.com/software/webservers/appserv/
- IBM WebSphere Application Server 库 Web 站点:
www.ibm.com/software/webservers/appserv/library.html
- IBM WebSphere Application Server 支持 Web 站点:
www.ibm.com/software/webservers/appserv/support.html
- IBM WebSphere Application Server 信息中心:
www.ibm.com/software/webservers/appserv/infocenter.html
- IBM WebSphere Application Server Edge Components 信息中心:
www.ibm.com/software/webservers/appserv/ecinfocenter.html

第 1 部分 高速缓存代理入门

这一部分提供高速缓存代理组件的概述，使用“配置和管理”表单和配置向导的说明，手工编辑 `ibmproxy.conf` 文件的说明，以及启动和停止代理服务器的过程。

这一部分包含下列章节：

第 3 页的第 1 章，『概述』

第 7 页的第 2 章，『使用“配置和管理”表单』

第 11 页的第 3 章，『使用配置向导』

第 13 页的第 4 章，『手工编辑 `ibmproxy.conf` 文件』

第 15 页的第 5 章，『启动和停止高速缓存代理』

第 1 章 概述

高速缓存代理拦截来自客户机的数据请求、从托管内容的机器中检索请求的信息并将此内容发回客户机。最常见的情况是，请求针对的是存储在 Web 服务器（也称为源服务器或内容主机）上的文档，并通过超文本传输协议（HTTP）发送请求。但是，您可以配置高速缓存代理处理其它协议，如文件传输协议（FTP）和 Gopher。

在将可高速缓存的内容发送给请求者之前，高速缓存代理会将这些内容存储到本地高速缓存中。可高速缓存内容的示例包含静态 Web 页面和动态生成但极少更改其片段的 JavaServer Pages（JSP）。通过从本地高速缓存中直接发送上述这些内容，高速缓存使得高速缓存代理能够满足针对相同内容的后继请求，与再次从内容主机检索这些内容相比，其速度要快得多。

新功能部件

高速缓存代理管理指南包括新功能部件、新支持的平台和 V6.0 以及 V5.0 后的发行版（5.0.1、5.0.2、5.1 和 5.1.1）的改正更新。这些新功能部件中（对于 V 6.0 和 V5.0 后的发行版）最重要的功能部件在此列出。

注：

1. 要获取更多有关支持的硬件和软件需求的信息，请参阅 *Edge Component* 的概念、规划与安装中『*Edge Component* 的需求』一章。
2. 要获取有关 V6 Edge Components 的系统需求的最当前信息，请访问以下 WebSphere Application Server Web 页面，<http://www.ibm.com/software/webservers/appserv/doc/latest/prereq.html>。

- **支持 Linux for S/390、zSeries、iSeries 和 pSeries**

除了支持在 Linux for Intel 上运行高速缓存代理外，现在代理还在 Linux for S/390、zSeries、iSeries 和 pSeries 上运行。

- **支持 AIX 5.2 和 AIX 5.3**

除了 AIX 5.1 外，现在高速缓存代理还支持 AIX 5.2 和 AIX 5.3。

- **支持 Solaris 9**

除了支持 Solaris 8 外，现在高速缓存代理还支持 Solaris 9。

- **支持 Windows Server 2003**

除了支持 Windows 2000 外，现在高速缓存代理还支持 Windows Server 2003。

- **支持 HP-UX V11i**

除了支持 AIX、Linux、Solaris 和 Windows 系统外，现在高速缓存代理还支持 HP-UX。

- **支持 JDK 1.4.2**

现在支持 32 位 JDK 的新版本：JDK 1.4.2。

- **支持 GSKit 7**

缺省情况下，与高速缓存代理协同提供并安装 GSKit 的新版本：GSKit 7。

- **结点重写的增强和新伪指令**

存在与结点重写相关的增强和新伪指令。要获取更多信息，请参阅：

- 第 41 页的『启用结点重写（可选）』
- 第 201 页的『JunctionReplaceUrlPrefix — 当与 JunctionRewrite 插件协同使用时，替换 URL 而不是插入前缀』
- 第 202 页的『JunctionSkipUrlPrefix — 当与 JunctionRewrite 插件协同使用时，跳过已包含前缀的重写 URL』
- 第 202 页的『JunctionRewriteSetCookiePath — 当与 JunctionRewrite 插件协同使用时，重写 Set-Cookie 头中的路径选项』
- 第 236 页的『RewriteSetCookieDomain — 指定需要重写的域模式』
- 第 228 页的『Proxy - 指定代理协议或逆向代理』

还提供了使用 JunctionRewrite 插件的备用方法。要获取更多信息，请参阅：

- 第 44 页的『使用 Cookie 替代 JunctionRewrite』
- 第 202 页的『JunctionRewrite - 启用 URL 重写』

- **其它新伪指令**

支持下列新伪指令：

- 第 171 页的『CacheMatchLanguage — 指定返回的高速缓存内容的语言首选项』
- 第 187 页的『EnableTcpNodelay — 启用 TCP NODELAY 套接字选项』
- 限制与请求相关的伪指令：
 - 第 204 页的『LimitRequestBody — 指定 PUT 或 POST 请求中最大主体大小』
 - 第 204 页的『LimitRequestFields — 指定客户机请求中的最大标题数』
 - 第 204 页的『LimitRequestFieldSize — 指定最大标题长度和请求行』
- 将高速缓存内存保存或装入到磁盘伪指令：
 - 第 191 页的『ExportCacheImageTo - 将高速缓存内存导出到磁盘』
 - 第 200 页的『ImportCacheImageFrom - 从文件导入高速缓存内存』

- **现有伪指令的增强**

存在下列现有伪指令的增强：

- BindSpecific: OutgoingSrcIp 选项允许代理使用特定源 IP 地址进行出局连接。
- ReversePass: host:port 选项允许代理应用基于后端服务器主机名和端口的不同 ReversePass 规则。

- **为扩展 JunctionRewrite 功能提供的 transmogrifier 插件示例**

对于 JunctionRewrite，现在提供了可以定制的样本代码，它重写 / 解析 HTML 文件中的 JavaScript (SCRIPT) 和 applet (APPLET) 标记块。（JunctionRewrite 插件不能单独处理 JavaScript 中或 Java 参数值中的资源链接。要获取更多的信息，请参阅第 44 页的『用于扩展 JunctionRewrite 功能的样本 transmogrifier 插件』。

- **PACD 守护程序配置相关更改**

要启用匿名绑定，请参阅第 126 页的『创建 pac_ldap.cred』。

对于代理和 LDAP 服务器之间的 SSL 连接，将密钥数据库密码放入 pac_keyring.pwd 文件中。请参阅第 110 页的『创建新的密钥数据库、密码和存储文件』。

- **更改缺省配置以更安全**

在配置文件 (ibmproxy.conf) 中，对缺省设置进行更改以提供更好的安全性。例如，进行更改以禁用 HTTP CONNECTION 和 SSL 隧道传送。不存在此增强的新伪指令。

第 2 章 使用“配置和管理”表单

高速缓存代理附带 HTML 表单，这些表单可用于向请求客户机提供服务，以及用于配置代理服务器。这些表单运行编辑本地代理服务器配置文件（ibmproxy.conf）的 CGI 程序。要使用这些表单，代理服务器必须正在运行且必须配置以从它们驻留的本地目录传递表单。

缺省情况下，高速缓存代理使用 Pass 伪指令（包含在允许访问“配置和管理”表单的 ibmproxy.conf 文件中）进行安装。当客户机从此代理服务器请求缺省主页时，将获得 Frntpage.html。此页面包含到“配置和管理”表单起始页面 wte.html 的超文本链接。

“配置和管理”表单是受保护的，在服务它们前需要客户机认证。要获取设置管理员标识和密码的说明，请参阅第 9 页的『设置管理员密码』。

浏览器要求

用于访问“配置和管理”表单的 Web 浏览器必须支持下列操作：

- *HTML 4.0*: 所有表单写入 HTML 4.0 规范。您的 Web 浏览器必须支持 HTML 4.0 和框架集。
- *Java 1.1 和 JavaScript*: Applet 被写入 Java 1.1 规范。您的 Web 浏览器必须支持遵从 Java 1.1 的 Java 虚拟机。applet 与那些遵从 Java 2.0 规范的 Java 虚拟机不兼容。必须启用 JavaScript 和 Java。
- *256 色*: 运行 Web 浏览器的工作站必须支持至少 256 色。

建议的浏览器是 Mozilla 1.4（对于 Linux 和 UNIX 系统）或 Internet Explorer（对于 Windows 系统）。请参阅 *Edge Components* 的概念、计划和安装以获取其它与浏览器相关的信息来查看“配置和管理”表单。

访问“配置和管理”表单

要访问“配置和管理”表单：

1. 确保代理服务器正在运行。要获取有关启动代理服务器的说明，请参阅第 15 页的第 5 章，『启动和停止高速缓存代理』。
2. 使 HTTP 浏览器导向为请求代理服务器的主页（Frntpage.html）或“配置和管理”起始页面（wte.html）。

注：此页面取决于您代理服务器的实际映射规则，并可根据括号中显示的缺省页面进行变化。

`http://your.server.name[:port][/directory][/page.html]`

其中

- *your.server.name* 是您主机的全路径名（例如，`http://www.ibm.com/`）。
- *[:port]* 如果您的代理服务器不在端口 80 上侦听管理请求，则在服务器名后包括端口号：`http://your.server.name:port`
- *[/directory]* URL 中附加的目录取决于映射规则。

- [/page.html] 只有当 HTML 页面没有作为欢迎页面列出时，需要指定它。要获取欢迎页面的信息，请参阅第 48 页的『定义缺省欢迎页面』。

3. 单击“配置和管理”表单，开始使用服务器配置表单。将提示您输入管理员用户名和密码。输入一个已授权的用户名和密码。高速缓存代理配置客户机窗口打开。

注:

- a. 主页面出现后，可能需要几秒钟以装入左面导航框架的内容。
- b. 在 Windows 2003 系统上，请求管理表单（CGI 脚本）的连接在其完成前可能接收重置。结果，浏览器可能报告未接收到数据或显示页不可用消息。要避免此问题，将 MaxActiveThreads 增加到大于 200 的值，或将 ConnThreads 增加到大于 50 的值，以解决重置连接的问题。请参阅第 209 页的『MaxActiveThreads - 指定有效线程的最大数』和第 179 页的『ConnThreads — 指定要用于连接管理的连接线程数』以获取更多有关这些伪指令的信息。

4. 左边的导航框架显示了配置表单的五个主类别:

- 代理配置
- 高速缓存配置
- 服务器配置
- 服务器活动监视
- 插件配置

单击标题左边的三角形指针以展开该类别中的配置表单列表。单击一个表单，打开它。此表单在输入字段中显示当前的配置值（如果有的话）；如果自安装完成以来您还没有对配置作过更改，这些将会是缺省值。

5. 在任何表单上，输入该特定功能的配置信息。每张表单提供了帮助您决定作什么样的更改的说明。要获取进一步的信息，单击每个表单顶部的帮助图标，即问号（?）。它提供下列链接:


- 字段帮助 - 每个屏幕面板中各个字段的描述
- 如何... - 使用表单完成特定任务的详细步骤
- 索引 - 帮助信息的索引

6. 在填充表单后，单击**提交**，以您所作的更改更新服务器配置。**提交**按钮在每个表单输入字段的下面。如果您不要对您表单上标明的字段进行更改，单击**复位**，表单上的字段将返回它们原来的值。

7. 如果您单击**提交**且您的输入被接受，在上部框架中将显示下列消息:

已成功完成请求的配置更改

如果输入没有被接受，上部框架中显示错误消息，表明哪些设置是不可接受的。

8. 要重新启动代理服务器，单击上部框架中的服务器重新启动图标（）。当代理服务器接收到重新启动命令时，它停止接受来自客户机的请求，但将完成所有已经在处理中的请求。在重新装入更改过的配置文件后，代理再次开始接受客户机请求。

注: 通过使用“配置和管理”表单或编辑 `ibmproxy.conf` 文件来更改某些伪指令时，在更改生效前，需要完全停止服务器，然后再启动，而不仅仅是重新启动。这些伪指令在第 155 页的表 6 中列出。

设置管理员密码

在安装高速缓存代理软件包后，您必须创建管理员标识和密码以访问“配置和管理”表单。缺省代理服务器配置认证用户通过使用 `webadmin.passwd` 密码文件请求“配置和管理”表单，在 Linux 和 UNIX 系统上，此密码文件位于 `/opt/ibm/edge/cp/server_root/protect/` 目录，在 Windows 系统上，它位于 `\Program Files\IBM\edge\cp\etc\` 目录。软件包安装不覆盖现有 `webadmin.passwd` 文件。

使用下列命令添加管理员条目到 `webadmin.passwd` 文件：

- 在 Linux 和 UNIX 系统上：

```
# htadm -adduser /opt/ibm/edge/cp/server_root/protect/webadmin.passwd
```

当提示时，向 `htadm` 程序提供管理员的用户名、密码和真实姓名。

- 在 Windows 系统上：

```
cd "\Program Files\IBM\edge\cp\server_root\protect\  
htadm -adduser webadmin.passwd"
```

当提示时，向 `htadm` 程序提供管理员的用户名、密码和真实姓名。

要获取 `htadm` 命令的详细描述，请参阅第 147 页的『`htadm` 命令』。

第 3 章 使用配置向导

高速缓存代理配置向导允许您快速配置已安装的高速缓存代理。此程序只对那些修改高速缓存代理行为使其能担当用品所必需的基本伪指令进行设置。此代理服务器可能需要其它配置。

要使用高速缓存代理配置向导:

1. 启动配置向导。

在 Windows 系统上: 单击开始 -> 程序 -> **IBM WebSphere** -> **Edge Components** -> 高速缓存代理 -> 配置向导。

在 Linux 和 UNIX 系统上: 输入命令 `/opt/ibm/edge/cp/cpwizard/cpwizard.sh`

2. 选择网络端口, 在其上代理服务器将侦听 HTTP 请求。
3. 输入目标内容服务器的名称。
4. 输入代理服务器管理员的用户标识和密码。

注:

1. 配置向导设置下列伪指令:

```
Port port  
Proxy /* http://content server :port
```

2. 如果使用配置向导来配置代理服务器, 然后启用 SSL, 那么必须为通过端口 443 检索到的代理请求创建一个映射规则。如需了解更多信息, 请参阅第 39 页的『定义映射规则』。

示例:

```
Proxy /* http://content server :443
```

或者

```
Proxy /* https://content server :443
```

限制: 在 Linux 系统上, 键盘快捷方式不为“高速缓存代理配置向导”工作。

第 4 章 手工编辑 ibmproxy.conf 文件

可以通过手工、编辑 `ibmproxy` 配置文件或通过“配置和管理”表单配置高速缓存代理。

- 在 Linux 和 UNIX 系统上，`ibmproxy.conf` 文件位于 `/etc/` 目录。
- 在 Windows 系统上，`ibmproxy.conf` 文件位于 `C:\Program Files\IBM\edge\cp\etc\en_US\`。

配置文件由名为伪指令的语句组成。要更改您的配置，通过修改伪指令来编辑配置文件，并保存您的更改。您几乎可以使用任何文本编辑器（如 `emacs` 和 `vi`）来编辑配置文件。

注：不要使用 Solaris 公共桌面环境（CDE）中包括的文本文件编辑器。Solaris 编辑器有时修改文件的属组，并更改文件链接的特性，使“配置和管理”表单无法写到配置文件中。

当重新启动服务器时，您对配置文件的更改生效，除非您更改了第 155 页的表 6 中标识的一个伪指令。如果您更改了那个列表中的某一伪指令，则您必须停止服务器并再次启动它。要获取指示信息，请参阅第 15 页的第 5 章，『启动和停止高速缓存代理』。

第 155 页的附录 B，『配置文件伪指令』描述每个配置文件伪指令和语法的详细信息。

第 5 章 启动和停止高速缓存代理

高速缓存代理被设计成能作为后台进程连续地运行，极少需要操作员干涉。通常，代理服务器在机器引导期间启动，只在需要维护时才停止。若有必要，代理服务器可以手工启动。代理服务器也可使用重新启动指令，它能有效的停止并启动代理服务器而无需中断活动客户机的连接。

在 Linux 和 UNIX 系统上自动启动和关闭

在 Linux 和 UNIX 系统上，安装高速缓存代理时，**ibmproxy** 初始化脚本及其关联的符号链接位于相应的 `/etc/` 目录中。然后这些脚本被集成到操作系统的启动和关机例程中。您可以通过编辑 **ibmproxy** 脚本和更改 **ibmproxy** 命令选项来更改自动重新启动的配置设置。

注: Solaris 文件描述符限制

由于 Solaris 全系统对文件描述符的限制，高速缓存代理初始化脚本可能无法设置期望的最大文件描述符数。如果全系统最大值小于高速缓存代理初始化脚本中的设置，则使用全系统限制。您可以更改文件描述符限制以避免可能由于值太低（小于 1024）而导致的代理性能问题。发出 **ulimit** 命令以查看当前可用的描述符数。如果值小于 1024，则增加文件描述符限制。要将文件描述符限制增加到 1024，在 `/etc/system` 文件中添加下列行：

```
set rlim_fd_cur=0x400
```

禁用自动启动和关闭

要禁用自动启动和关闭：

- 在 AIX 系统上，从初始化文件除去 **ibmproxy** 命令。
- 在 HP-UX 系统上，除去下列到 **ibmproxy** 的链接：
 - `/sbin/rc1.d/K154ibmproxy`
 - `/sbin/rc2.d/S880ibmproxy`
- 在 Linux 系统上，除去对运行级别子目录中 `/etc/rc.d/init.d/ibmproxy` 的符号链接。

在 SuSE Linux 上，除去下列到 **ibmproxy** 的链接：

- `/etc/rc.d/rc3.d/S20ibmproxy`
- `/etc/rc.d/rc3.d/K20ibmproxy`
- `/etc/rc.d/rc4.d/S20ibmproxy`
- `/etc/rc.d/rc4.d/K20ibmproxy`
- `/etc/rc.d/rc5.d/S20ibmproxy`
- `/etc/rc.d/rc5.d/K20ibmproxy`

在 Red Hat Linux 上，除去下列到 **ibmproxy** 的链接：

- `/etc/rc.d/rc0.d/K54ibmproxy`
- `/etc/rc.d/rc1.d/K54ibmproxy`

- /etc/rc.d/rc2.d/K54ibmproxy
- /etc/rc.d/rc6.d/K54ibmproxy
- /etc/rc.d/rc3.d/S88ibmproxy
- /etc/rc.d/rc5.d/S88ibmproxy
- 在 Solaris 系统上，除去 **ibmproxy start** 命令及其两个 kill 脚本，按如下所示：
 - 从 /etc/rc2.d 目录删除 S88ibmproxy。
 - 从 /etc/rc0.d 目录删除 K54ibmproxy。
 - 从 /etc/rc1.d 目录删除 K54ibmproxy。

在 Linux 和 UNIX 系统上手工启动

无论使用何种启动方法（直接从命令提示符启动或从脚本启动），最终都将调用 **ibmproxy** 命令。要获得 **ibmproxy** 命令的详细描述，请参阅第 152 页的『**ibmproxy** 命令』。以下只是一些最常用自变量的示例。

在 AIX 上:

- 要使用 **startsrc** 命令启动缺省语言环境的代理服务器，输入下列内容：
`startsrc -s ibmproxy`
- 要使用 **startsrc** 命令启动除缺省语言环境外的任何语言环境的代理服务器，输入下列内容：
`startsrc -s ibmproxy -e "LC_ALL=locale"`
- 要使用缺省运行时设置启动代理服务器而不使用 **startsrc** 命令，输入下列内容：
`ibmproxy`

在 HP-UX 上:

- 要通过运行初始化脚本启动代理服务器，在根提示符下输入下列内容：
`/sbin/init.d/ibmproxy start`
- 要作为后台进程启动代理服务器而不运行初始化脚本，在根提示符下输入下列内容：
`/usr/sbin/ibmproxy`
- 要作为前台进程启动代理服务器而不运行初始化脚本，在根提示符下输入下列内容：
`/usr/sbin/ibmproxy -nobg`

在 Linux 上:

- 要通过运行初始化脚本启动代理服务器，在根提示符下输入下列内容：
`/etc/rc.d/init.d/ibmproxy start`
- 要作为后台进程启动代理服务器而不运行初始化脚本，在根提示符下输入下列内容：
`/usr/sbin/ibmproxy`
- 要作为前台进程启动代理服务器而不运行初始化脚本，在根提示符下输入下列内容：
`/usr/sbin/ibmproxy -nobg`

- 要使用已有 SQUID 配置文件 (squidConfig.file) 启动代理服务器, 在根提示符下输入下列内容:

```
squidConfig.file -r /etc/errors_icons.conf
```

其中, *errors_icons.conf* 文件标识了在浏览目录时哪些图标用于指定的文件类型。

在 Solaris 上:

- 要通过运行初始化脚本启动代理服务器, 在根提示符下输入下列内容:

```
/etc/init.d/ibmproxy start
```
- 要作为后台进程启动代理服务器而不运行初始化脚本, 在根提示符下输入下列内容:

```
/usr/sbin/ibmproxy
```

- 要作为前台进程启动代理服务器而不运行初始化脚本, 在根提示符下输入下列内容:

```
/usr/sbin/ibmproxy -nobg
```

作为 Windows 服务启动

如果高速缓存代理作为 Windows 服务安装, 则它象任何其它 Windows 服务一样启动:

1. 单击开始 -> 设置 (对于 Windows 2000) -> 控制面板。
2. 在控制面板窗口, 双击管理工具 -> 服务。
3. 在服务窗口中, 突出显示高速缓存代理。
4. 单击启动以启动高速缓存代理服务。

如果高速缓存代理作为服务安装, 则它可配置为在 Windows 启动时自动启动。在该情况下, 您无须在代理可服务请求之前登录。要使您的代理自动启动:

1. 单击开始 -> 设置 (对于 Windows 2000) -> 控制面板。
2. 在控制面板窗口, 双击管理工具 -> 服务。
3. 在服务窗口中, 突出显示高速缓存代理。
4. 单击自动单选按钮, 然后单击启动以在 Windows 启动时自动启动高速缓存代理服务。

刷新 PATH 环境变量

如果高速缓存代理在服务窗口中标记为已启动, 但代理实际不起作用, 则可能是在代理安装后未重新启动机器所致。如果高速缓存代理服务设置为与桌面交互, 重新启动发生故障也可能导致下列错误消息在弹出框中出现: 消息目录错误: 消息目录无法装入或无效

此机器必须重新启动, 这样 Windows 注册表中的 PATH 环境变量值才会刷新。如果未刷新注册表, 则 PATH 变量可能会显示正确的高速缓存代理和 GSK7 路径, 但却不能正常运作。

注: 当高速缓存代理和另一个应用程序 (如网络文件系统) 作为服务运行时, 对于 Windows 系统存在潜在冲突。高速缓存代理有时无法解释包含远程驱动器 (此驱动器从属于也作为服务运行的文件系统应用程序) 的路径。

如果在 Windows PATH 环境变量中文件系统服务路径出现在高速缓存代理服务路径的前面，则会发生此问题。改变 PATH 变量以把文件系统服务放到靠近设置的末尾处，就能解决此问题。

此问题不影响由应用程序（这些应用程序不作为 Windows 服务运行）控制的远程驱动器。例如，高速缓存代理能访问其它 Windows 机器（这些机器通过局域网（LAN）是可见的）上的共享驱动器。

作为 Windows 应用程序启动

使用“开始”菜单

当高速缓存代理作为 Windows 应用程序安装时，安装过程将创建一个高速缓存代理条目作为开始菜单的子菜单。要作为应用程序启动高速缓存代理，单击开始 -> 程序 -> IBM WebSphere -> Edge Components -> 高速缓存代理。

此启动过程使用当前配置设置来运行代理服务器。如果您想要在启动时指定其它设置，则使用命令启动过程（请参阅下一节）。

使用命令提示符

要从任何 Windows 或 DOS 命令提示符启动服务器，使用 **ibmproxy** 命令。如果您在安装服务器后尚未关机和重新启动 Windows，则输入此命令的全路径名，按如下所示（缺省情况下）：

```
c:\Program Files\IBM\edge\cp\bin\ibmproxy.exe
```

ibmproxy 命令使用当前配置设置来启动服务器。如果您在安装后尚未更改服务器配置，则当前配置基于您在安装期间输入的信息和缺省选项。

ibmproxy 命令把服务器作为应用程序启动，即使您已经安装了高速缓存代理来作为服务运行。要强制服务器作为应用程序运行，您还可指定命令选项 **-noservice**。其它命令选项在运行时更改配置设置。

启动多个代理服务器

代理服务器的多个实例可以并发运行，但是每个实例必须在一个单独的端口上侦听。在 AIX 系统上，使用 SRC 仅可启动一个实例。必须为服务器的所有实例都指定唯一的配置文件，因为该配置文件标识了一个端口号，并且此号对于特定机器上的每个服务器必须都不同。要启动服务器的附加实例（当至少一个实例已在运行时），输入下列命令：

- 在 Linux 和 UNIX 上：

```
ibmproxy -r other_config_file
```

- 在 Windows 上：

```
ibmproxy -noservice -r other_config_file
```

其中 *other_config_file* 是唯一的配置文件。

当启动服务器的多个实例时，记录为每个实例显示的进程标识。这些标识是停止服务器的特定实例所必需的。

注：在运行服务器的多个实例的 Linux 系统上，命令 `/etc/rc.d/init.d/ibmproxy stop` 将停止启动的最后一个服务器。其它实例必须分别停止。请参阅『在 Linux 和 UNIX 系统上手工关闭』以获取相关信息。

在 Linux 和 UNIX 系统上手工关闭

要停止服务器：

- 您必须是启动进程的用户或超级用户 `root`。
- 您使用的方法必须与启动服务器的方法相同。下表列出了启动方法及其关联的停止方法。

表 2. Linux 和 UNIX 系统的启动和停止方法

启动方法	停止方法
从 <code>/etc/inittab</code> (在 AIX 上)	输入 <code>stopsrc -s ibmproxy</code>
从 <code>/sbin/init.d</code> (在 HP-UX 上)	输入 <code>/sbin/init.d/ibmproxy stop</code>
从 <code>/etc/rc.d/init.d</code> (在 Linux 上)	输入 <code>/etc/rc.d/init.d/ibmproxy stop</code>
<code>ibmproxy</code>	<ol style="list-style-type: none"> 1. 查找 ibmproxy 进程标识：在 AIX 上，输入 <code>ps -aef grep "ibmproxy"</code>。在 Linux 上，输入 <code>ps -aux grep ibmproxy grep server_ID</code>。在 Solaris 和 HP-UX 上，输入 <code>ps -ef grep "ibmproxy"</code> 2. 停止 ibmproxy 进程：输入 <code>kill process_id</code> <p>要停止此机器上的所有服务器：输入 <code>killall ibmproxy</code></p>
<code>ibmproxy -nobg</code>	输入 <code>ctrl-c</code>
<code>ibmproxy -r -other_config_file</code> (在 AIX 上)	输入 <code>stopsrc -s ibmproxy -p process_id</code>
<code>ibmproxy -r -other_config_file</code> (在 Linux 上)	<ol style="list-style-type: none"> 1. 查找 ibmproxy 进程标识：输入 <code>ps aux grep ibmproxy grep process_id</code> 2. 停止 ibmproxy 进程：输入 <code>kill process_id</code>

要在根提示符下停止服务器，输入：

- 在 AIX 上： `stopsrc -s ibmproxy`
- 在 HP-UX 上： `/sbin/init.d/ibmproxy stop`
- 在 Linux 上： `/etc/rc.d/init.d/ibmproxy stop`
- 在 Solaris 上： `/etc/init.d/ibmproxy stop`

关闭命令的限制

您在使用关闭命令时可能会遇到下列限制：

- **AIX、HP-UX 和 Linux**

在 AIX、HP-UX 和 Linux 系统上，停止高速缓存代理系统的命令有时仅关闭高速缓存代理进程。导致此行为的 AIX 命令是 **stopsrc -s ibmproxy** 命令。导致此行为的 HP-UX 和 Linux 命令是 **ibmproxy -stop** 命令。

关闭代理服务器后，由 LDAP 服务器使用的 PACD 进程可能会保持运行。使用 **kill** 命令可以安全关闭 PACD 进程，按如下所示：

```
kill -15 PACD_process_ID
```

- **Solaris**

在 Solaris 系统上发出 **ibmproxy -stop** 命令产生的效果与在其它操作系统上发出此命令不同。因为 Solaris 代码中的限制，在 Solaris 平台上使用 **ibmproxy -stop** 时不执行服务器终止插件步骤。

此限制对代理服务器软件及客户实现的插件具有蕴含式。

关闭代理服务器后，由 LDAP 服务器使用的 PACD 进程可能会继续运行。使用 **kill** 命令可以安全关闭 PACD 进程，按如下所示：

```
kill -15 PACD_process_ID
```

在 Windows 系统上手工关闭

您可以用停止其它 Windows 程序的相同方法来停止高速缓存代理服务器。

如果代理作为服务安装：

1. 单击开始 -> 设置（对于 Windows 2000） -> 控制面板。
2. 在控制面板窗口中，双击管理工具 -> 服务。
3. 在服务窗口中，突出显示高速缓存代理。
4. 单击停止以停止高速缓存代理服务。

如果代理不作为服务安装，则执行下列任何操作停止高速缓存代理：

- 单击右上角的 **x** 图标。
- 从文件菜单，单击退出。
- 按 **Alt + F4**。

配置更改后重新启动

更改服务器配置（通过使用“配置和管理”表单或编辑 `ibmproxy.conf` 文件）后，您必须在更改生效之前重新启动服务器。在大多数情况下，您可以重新启动服务器，而无需首先停止它。但是，一些设置不是简单地重新启动一次就可刷新的。要获取更多的信息，请参阅第 155 页的表 6。

要在不首先停止服务器的情况下重新启动它，请单击任何“配置和管理”表单上的**重新启动**按钮，或输入下列命令：`ibmproxy -restart`

第 2 部分 配置和调整高速缓存代理进程

这一部分解释高速缓存代理组件如何与操作系统、计算机硬件和网络进行交互。还提供了配置此交互作用的过程。代理服务器配置的这些元素通常是由系统管理员管理的，且必须小心地协调网络资源（如，IP 地址和主机名）和系统资源（如可用内存和 CPU 周期）。

这一部分包含下列章节：

第 23 页的第 6 章，『定义服务器』

第 25 页的第 7 章，『建立进程所有权』

第 27 页的第 8 章，『管理连接』

第 31 页的第 9 章，『调整代理服务器进程』

第 6 章 定义服务器

高速缓存代理通常作为主机计算机系统（它配置为作为网络服务器执行）上的后台进程运行。此进程与主机计算机系统上的一个或所有活动网际协议（IP）地址关联（绑定到）。它侦听指定端口上的各种因特网协议（如 FTP 和 HTTP），并对与其行为配置一致的这些请求执行操作。（要获得更多信息，请参阅第 35 页的第 3 部分，『配置高速缓存代理行为』。）

缺省情况下，高速缓存代理假设了其主机计算机系统的名称。您可考虑指定代理服务器的主机名来覆盖此缺省行为。为了将高速缓存代理绑定到一特定 IP 地址，必须更改代理服务器的主机名以与该 IP 地址相同。

注：在代理服务器尝试绑定到 IP 地址的事件中，尚未将主机名设置成有效的 IP 地址，绑定将失败且代理服务器将侦听所有有效的 IP 地址。

代理服务器的主机名不会影响如何得出客户机流量。代理服务器不会将其自己的主机名与 HTTP 请求头中的主机名自变量的值相比较。代理服务器的主机名有时合并并在动态生成的本地内容页面中，如错误消息。它也可作为 HTTP 头中的 Via 自变量的值传回请求的客户机。

可配置代理服务器以替换具有代理服务器主机名（将请求传递到目标服务器之前的主机名）的请求客户机的主机名。这样做可强制目标服务器维护到代理服务器的通信通道，而不是建立一个与客户机的直接连接。

通过指定主机计算机系统上代理服务器文件的物理位置来定义代理服务器进程，代理服务器中的名称引用它自己的名称，且侦听端口与 ServerRoot、Hostname 和 Port 伪指令的值相同。如果主机有多个 IP 地址，代理服务器可通过将 BindSpecific 伪指令的值设置为开，且 Hostname 伪指令的值设置为与 IP 地址相同的值来绑定到一特定地址。

管理端口提供了访问“配置和管理”表单和维护服务器的方法。要通过管理端口提供对代理服务器的访问，为 AdminPort 伪指令指定一个值。管理端口上接收到的请求并不与标准端口上接收到的请求一起排队。可以写映射规则以允许通过此端口访问“配置和管理”表单。

当启用 BindSpecific 伪指令时，高速缓存代理与从 Hostname 伪指令的值派生出的 IP 地址一起绑定到 Port 伪指令指定的端口。AdminPort 伪指令指定的端口绑定到所有在系统上可用的 IP 地址。

要覆盖正在运行服务器的缺省名称（如 IBM-PROXY 或 IBM_HTTP_SERVER），为 HeaderServerName 伪指令指定一个值。该值将植入 HTTP 响应服务器字段。

要提高代理性能，可将 PureProxy 伪指令的值设置为开。这将完全禁用所有高速缓存功能。

关联的伪指令

下列伪指令定义了代理服务器进程:

- 第 196 页的『Hostname - 指定服务器的全限定域名或 IP 地址』
- 第 240 页的『ServerRoot - 指定服务器程序安装的目录』
- 第 196 页的『HeaderServerName - 指定 HTTP 头中返回的代理服务器名称』
- 第 167 页的『BindSpecific - 指定服务器是否绑定到一个或所有 IP 地址』
- 第 220 页的『Port - 指定服务器侦听请求的端口』
- 第 164 页的『AdminPort - 指定请求管理页面或表单的端口』
- 第 232 页的『PureProxy - 禁用专用的代理』

如需了解更多信息, 请参阅第 13 页的第 4 章, 『手工编辑 ibmproxy.conf 文件』。

“配置和管理” 表单

下列“配置和管理”表单将编辑关联伪指令的值:

- 服务器配置 -> 基本设置 -> 主机名
- 服务器配置 -> 基本设置 -> 服务器根
- 服务器配置 -> 基本设置 -> 缺省端口号
- 服务器配置 -> 基本设置 -> 管理员端口号
- 服务器配置 -> 基本设置 -> 绑定选项
- 代理配置 -> 代理性能 -> 作为纯代理运行

注: 您不能使用“配置和管理”表单来编辑 HeaderServerName 伪指令。

如需了解更多信息, 请参阅第 7 页的第 2 章, 『使用“配置和管理”表单』。

第 7 章 建立进程所有权

当除超级用户 `root` 外的用户启动高速缓存代理时，该用户维护与代理服务器关联的所有进程的所有权。然而，如果高速缓存代理是由超级用户 `root` 启动的，代理服务器中的设置用户标识函数将读取 `ibmproxy.conf` 文件中的 `UserId` 和 `GroupId` 伪指令，并将进程所有权更改为指定用户和组。这样做可限制文件访问和保护计算机系统。如果更改了 `UserId` 或 `GroupId` 伪指令，您必须更新代理服务器使用的日志目录和其它文件（例如，访问控制列表（ACL））的所有权和许可权。

注：在 Linux 系统上，只有担负侦听连接责任的进程和线程将更改它们的所有权。工作流程中负担其它活动责任的进程和线程仍归 `root` 用户所有。所有进程和线程会接收到进程标识（PID）号。`ps` 命令将列出所有的进程标识，而不管它们是否与进程或线程关联。

通过指定用户标识、组标识和文件位置（在其中进程标识作为 `UserID`、`GroupID` 和 `PidFile` 伪指令的值记录的文件）来建立代理服务器进程的所有权。

要强制代理服务器进程作为前台进程运行，将 `NoBG` 伪指令的值设置为开。

关联的伪指令

下列伪指令将建立代理服务器进程的所有权：

- 第 248 页的『`UserId` - 指定缺省用户标识』
- 第 196 页的『`GroupId` - 指定组标识』
- 第 213 页的『`NoBG` - 在前台运行高速缓存代理进程』
- 第 219 页的『`PidFile`（仅 Linux 和 UNIX）— 指定在其存储高速缓存代理的进程标识的文件』

如需了解更多信息，请参阅第 13 页的第 4 章，『手工编辑 `ibmproxy.conf` 文件』。

“配置和管理” 表单

下列“配置和管理”表单将编辑关联伪指令的值：

- 服务器配置 -> 基本设置 -> 用户标识
- 服务器配置 -> 基本设置 -> 组标识
- 服务器配置 -> 基本设置 -> 进程标识文件位置

注：您不能使用“配置和管理”表单来编辑 `NoBG` 伪指令。

如需了解更多信息，请参阅第 7 页的第 2 章，『使用“配置和管理”表单』。

第 8 章 管理连接

高速缓存代理将产生一个新线程以处理每个客户机请求。如果没有可用的线程，则代理服务器会挂起请求直至更多线程可用。随着活动线程数的增加，代理服务器将消耗更多内存。指定活动线程的最大数以作为 `MaxActiveThreads` 伪指令的值。

侦听备用是在拒绝与新客户机连接之前，服务器记录的客户机连接的挂起请求数。此设置基于服务器可在几秒内处理的请求数。在超时之前，服务器必须响应客户机连接。指定备用中可挂起的最大连接数，以作为 `ListenBacklog` 伪指令的值。

此代理服务器能维护持久客户机或服务器的连接。通过使用持久连接，服务器接受来自客户机的多个请求并通过相同的 `TCP/IP` 连接发送响应。持久连接的使用减少了客户机的等待时间并降低了代理服务器上的 `CPU` 负载，而服务器内存开销也只略微增加。当服务器不为每个请求和响应建立一个单独的 `TCP/IP` 连接时，整个吞吐量将增加，并且当连接持久时，可以最有效地使用 `TCP/IP` 连接。

服务器端连接池允许重用代理服务器和源服务器之间已有的连接，以应用服务器端上持久连接的有点。每个重用的连接节省三个 `TCP` 信息包（两个用于建立连接的三路握手信息包，以及一个用于关闭连接的三路握手信息包）。服务器端连接池的优点包括：

- 更少的网络拥塞（通过将连接的打开和关闭最小化）
- 更少的在路由器、客户机和服务器中使用的 `CPU` 时间
- 客户机和服务器上更少的内存使用
- 高速缓存失配时，更快的代理响应（由于避免了连接的打开和关闭）

注：只建议在受控的环境中使用连接池。在源服务器不遵守 `HTTP 1.1` 时，它可能会降低性能。注意，正确设置源服务器也是关键。下面是来自 `Apache 1.3.19` 配置文件的一个简单示例：

- `#KeepAlive`: 是否允许持久连接（每个 `#connection` 有多个请求）。给 `deactivate#` 设置为 `Off`
- `KeepAlive On`
- `#MaxKeepAliveRequests`: 在持久连接期间允许的最大请求数。设置为 `0` 以允许无限数量。保留该数为高以获得最佳 `performance#`
- `Max KeepAliveRequests 0`
- `#KeepAliveTimeout`: 等待来自同一 `connection#` 上的相同客户机的下一个请求的时间秒数
- `KeepAliveTimeout 240`

这些设置使连接只要使用就保持至 `Web` 服务器的连接，并允许代理而不是源服务器来管理连接。因此，只在需要的范围内合用连接。

当启用服务器端连接池时，将合用至源服务器的 `HTTP` 连接。在代理的 `SSLEnable` 伪指令设置为开时，`SSL` 连接在配置中也是合用的。

通过指定任何时刻每个服务器保持的最大空闲套接字数、设置终止空闲持久连接前服务器等待的时间，以及垃圾收集线程检查超时连接的时间间隔（缺省值为 2 分钟）来配置如何维护连接池。

定义各种保持打开的连接的时间，以作为 InputTimeout、OutputTimeout、PersistTimeout、ReadTimeout 和 ScriptTimeout 伪指令的值。

关联的伪指令

下列伪指令管理与代理服务器进程的连接：

- 第 209 页的『MaxActiveThreads - 指定有效线程的最大数』
- 第 179 页的『ConnThreads - 指定要用于连接管理的连接线程数』
- 第 204 页的『ListenBacklog - 指定服务器可执行的侦听备用客户机连接数』
- 第 231 页的『ProxyPersistence - 允许持久连接』
- 第 210 页的『MaxPersistRequest - 指定一个持久连接上接收的最大请求数』
- 第 239 页的『ServerConnPool - 将连接合用指定到源服务器』
- 第 211 页的『MaxSocketPerServer - 指定服务器的最大打开套接字数』
- 第 240 页的『ServerConnTimeout - 指定最大的非活动的周期』
- 第 239 页的『ServerConnGCRun - 指定运行垃圾收集线程的时间间隔』
- 第 218 页的『PersistTimeout - 指定等待客户机发送另一个请求的时间』
- 第 201 页的『InputTimeout - 指定输入超时』
- 第 234 页的『ReadTimeout - 指定连接的时间限制』
- 第 216 页的『OutputTimeout - 指定输出超时』
- 第 238 页的『ScriptTimeout - 指定脚本的超时设置』

如需了解更多信息，请参阅第 13 页的第 4 章，『手工编辑 ibmproxy.conf 文件』。

“配置和管理” 表单

下列“配置和管理”表单将编辑关联伪指令的值：

- 服务器配置 -> 系统管理 -> 性能 -> 活动线程的最大数
- 服务器配置 -> 系统管理 -> 性能 -> 侦听备用大小
- 代理配置 -> 代理性能 -> 允许持久连接
- 服务器配置 -> 系统管理 -> 性能 -> 最大请求数
- 服务器配置 -> 系统管理 -> 性能 -> 存留超时
- 服务器配置 -> 系统管理 -> 超时 -> 输入超时
- 服务器配置 -> 系统管理 -> 超时 -> 读超时
- 服务器配置 -> 系统管理 -> 超时 -> 输出超时
- 服务器配置 -> 系统管理 -> 超时 -> 脚本超时
- 服务器配置 -> 系统管理 -> 超时 -> 存留超时

注:

1. 您不能使用“配置和管理”表单来编辑 ServerConnPool、MaxsocketPerServer、ServerConnTimeout 或 ServerConnGCRun 伪指令。
2. PersistTimeout 可从服务器配置 -> 系统管理 -> 性能表单或服务器配置 -> 系统管理 -> 超时表单来进行编辑。

如需了解更多信息, 请参阅第 7 页的第 2 章, 『使用“配置和管理”表单』。

第 9 章 调整代理服务器进程

通过适当地设置和调整系统，可以显著地改进高速缓存代理的性能。下列是改进设置和调整的一些建议。

设置与性能关联的伪指令

下列伪指令会明显影响代理服务器进程的性能：

- 第 232 页的『PureProxy - 禁用专用的代理』。此功能部件通过完全禁用高速缓存来提高系统性能。
- 第 231 页的『ProxyPersistence - 允许持久连接』。此功能部件启用客户机和服务器以维护打开的连接。持久连接减少了从代理服务器请求文档的延迟时间，但需要增加网络带宽和每个连接的专用服务器线程。如果您的设置限制了可用线程数，则不要允许持久连接。

下列“配置和管理”表单字段将编辑关联伪指令的值：

- 代理配置 -> 代理性能：作为纯代理运行
- 代理配置 -> 代理性能：允许持久连接

检查其它应用程序

检查运行在您系统上的服务或守护程序，并除去那些不需要的以增加可用内存和 CPU 周期。例如，如果您的系统正在运行一个仅服务一些 Web 页面的 Web 服务器，则考虑将此高速缓存代理用作您唯一的 Web 服务器。按如下所示禁用其它 Web 服务器：

- 在 AIX 上：Examine /etc/inittab
- 在 Linux 上：Examine /etc/rc.d/rcx.d 以获取您的系统的缺省运行级别（通常为 2）
- 在 HP-UX 和 Solaris 上：Examine /etc/rcx.d 以获得系统的缺省运行级别（通常为 2）。
- 在 Windows 系统上：
 1. 单击开始 -> 设置（对于 Windows 2000） -> 控制面板 -> 管理工具 -> 服务。
 2. 复查那些不必需但设置为自动的服务。
 3. 将那些服务的启动类型从自动更改为手工。

验证调页空间

确保您的系统有足够的调页空间以进行正确的操作。系统需有两倍于物理内存的调页空间。如有可能，通过多个物理驱动器扩展调页空间。例如，一台内存为 512 MB 的 Netfinity 5000 服务器和共需要 1 GB 的调页空间的五个 SCSI 驱动器，大约每个驱动器有 200 MB。

调整文件系统

高速缓存代理在其操作期间创建和破坏许多文件。如果您的代理服务器记录访问（使用访问日志、代理访问日志或高速缓存访问日志），则将日志导向它们自己的文件系统，以便如果日志意外增长，它们不会使用为另一功能（如高速缓存）准备的空间。

调整 TCP/IP 配置

高速缓存代理对 TCP/IP 配置的更改是敏感的。降低任何操作系统上的 TCP/IP 值可能导致代理服务器以非期待方式执行。更明确地，如果 TCP/IP 值设置得太低，连接可能会由连接到代理服务器的客户机复位连接或由代理连接的源服务器复位连接。特别是对于通过低带宽连接（56700 位 / 秒或更低）连接到代理服务器的客户机，这是真的。如果必须降低 TCP/IP 参数，则需谨慎处理。

调整高负载环境的 TCP 时间等待时间间隔（HP-UX、Linux、Solaris 和 Windows）

TCP 时间等待时间间隔指定在强制关闭之前套接字等待来自发送方的 FIN 信息包的时间长度。在高负载环境中，如果在关闭它们的连接后，TIME_WAIT 状态下有大量套接字保留为暂挂，那么代理服务器可能会出现停止。减少 TCP 时间等待时间间隔将减少暂挂的套接字数，且在高负载环境中，这可避免代理服务器出现停止。建议将此时间间隔设置为 5 秒。

要将 TCP 时间等待时间间隔设置为 5 秒：

- 在 HP-UX 上：

发出下列命令：

```
ndd /dev/tcp -set tcp_time_wait_interval 5000
```

使用“sam”实用程序设置内核参数 max_thread_proc 至少为 2048。

注：还考虑调整以下内核参数：

```
maxfiles、maxfiles_lim、maxproc、shmem、tcp_conn_request_max、tcp_ip_abort_interval、  
tcp_keeplive_interval、tcp_rexmit_interval_initial、tcp_rexmit_interval_max、  
tcp_rexmit_interval_min、tcp_xmit_hiwater_def 和 tcp_recv_hiwater_def。
```

- 在 Linux 上：

发出下列命令：

```
echo "1024 61000" > /proc/sys/net/ipv4/ip_local_port_range  
echo "5" > /proc/sys/net/ipv4/tcp_fin_timeout
```

- 在 Solaris 上：

发出下列命令：

```
ndd /dev/tcp -set tcp_time_wait_interval 5000
```

按如下所示编辑 /etc/system 文件：

```
set tcp:tcp_conn_hash_size=8129
```

- 在 Windows 上：

必须创建设置 TCP 时间等待时间间隔的注册表项。请参阅您的 Windows 文档以获得更多信息。

调整 Linux 内核

在 Linux 内核中的一些限制很低并且可以修改。一些限制可以通过 /proc 文件系统更改，其它限制需要重新编译内核。

注：/proc 文件系统是虚拟的；即，它在磁盘上不物理存在。相反，它充当感触 Linux 内核中的接口。因为它不存在，您的输入值将在重新启动时丢失。因此，将您要对 /proc 文件系统所做的更改放置到 RedHat 上的 /etc/rc.d/rc.local 文件中或 SuSE 上的 /etc/rc.config 文件中。那么在重新启动时都会激活更改。

一些建议如下：

- 缺省情况下，文件描述符最大值为 4096。可以通过将下列内容添加到 rc.local 文件来更改它：

```
echo 32768 > /proc/sys/fs/file-max
```

- 缺省情况下，内节点最大值为 16384。可以通过将下列内容添加到 rc.local 文件来更改它：

```
echo 65536 > /proc/sys/fs/inode-max
```

- 缺省情况下，TCP 和 UDP 端口范围是 1024 到 4999。通过在 rc.local 文件中添加下列内容，可将其更改为 32768 到 61000。

```
echo 32768 61000 > /proc/sys/net/ipv4/ip_local_port_range
```

- 缺省情况下，允许的任务数为 512。如果太多任务在运行，则这将影响进程的线程最大数。通过修改 *YourKernelSource*/include/linux/tasks.h 文件中的 NR_TASKS 值可以将此限制增加为 2048。
- 另外，将 MIN_TASKS_LEFT_FOR_ROOT 的值更改为 24。您必须重新编译您的内核以使此更改生效。

如果您确定要重新构建您的内核，则仅启用那些您肯定需要的选项。如果您不需要特定守护程序，则不要运行它。

调整 AIX 线程调整变量

在 AIX 系统上，可以通过使用系统作用域线程和允许线程使用多个堆来提高高速缓存代理性能。性能与操作系统的多处理能力和底层操作系统的线程调度有关。通过设置以下 AIX 线程调整变量（如下所示）可以提高性能：

```
export AIXTHREAD_SCOPE=S
export SPINLOOPTIME=500
export YIELDLOOPTIME=100
export MALLOCMULTIHEAP=1
```

在启动 /usr/sbin/ibmproxy 或可以添加环境变量到 /etc/rc.ibmproxy（如果使用 **startsrc -s ibmproxy** 启动代理服务器）之前，您可以设置这些环境变量。调整这些线程调整变量之后，在 SMP 系统上，将更加显著改进性能。但是，在某些情况下，在单处理器系统上，也可以显著地改进性能。

注：要获得更多信息，查看 AIX 操作系统文档以获得关于线程调整变量的详细信息。

第 3 部分 配置高速缓存代理行为

这一部分解释高速缓存代理组件如何响应客户机请求，并提供配置此行为的过程。代理服务器配置的这些元素通常是由 Web 管理员管理的，它不会影响主机计算机系统或网络中其它计算机系统上的其它进程。

这一部分包含下列章节：

第 37 页的第 10 章，『管理请求处理』

第 47 页的第 11 章，『管理本地内容的传递』

第 49 页的第 12 章，『管理 FTP 连接』

第 51 页的第 13 章，『定制服务器处理』

第 59 页的第 14 章，『配置头选项』

第 61 页的第 15 章，『关于应用程序编程接口』

第 10 章 管理请求处理

当高速缓存代理接收客户机请求时，如果请求的方法已启用，则对 URL 字段中指定的对象执行方法字段中指定的操作。代理服务器依照管理员定义的一组映射规则来解析 URL。这些映射规则可能会指示高速缓存代理作为 Web 服务器操作并检索本地文件系统的对象，或者作为代理服务器操作并检索源服务器的对象。

本章描述如何启用方法、定义映射规则和配置代理服务器。

启用 HTTP/FTP 方法

到服务器的客户机请求包含一个方法字段，表明服务器要对指定对象执行的操作。

下列是代理服务器支持的方法列表，以及描述如果启用方法，它是如何响应包含方法的客户机请求。

注：对 HTTP 和 FTP 请求，某些方法是相同的。对 HTTP 启用的这些方法也同样能对 FTP 启用。

DELETE

代理服务器将删除由 URL 标识的对象。DELETE 允许客户机从您的高速缓存代理中擦除文件。使用服务器保护设置来定义谁能使用 DELETE，以及对哪些文件使用此操作。要获取详细信息，请参阅第 103 页的第 25 章，『服务器保护设置』。

GET 代理服务器返回任何由 URL 标识的数据。如果 URL 指的是一个可执行程序，代理返回程序输出。此方法可用来处理持久连接。

HEAD 代理服务器仅返回由 URL 标识 HTTP 文档头，而没有文档主体。

OPTIONS

代理服务器返回关于由 URL 标识的请求 - 响应链上通信选项的信息。该方法允许客户机确定与对象关联的选项和要求，或服务器的能力，而不必作用于对象或检索对象。

POST 请求包含数据和 URL。代理服务器接受请求中包含的数据，作为处理数据的 URL 中所标识资源的新附属。该资源可以是一个数据接受程序、一个到其它一些协议的网关或一个接受注释的独立程序。

POST 方法用于处理现有资源的注释。示例包括公布一条消息到公告板、新闻组、邮寄列表或类似的资源组；提供数据块，例如从表单到数据处理程序；或通过附加操作扩展数据库。在高速缓存代理本身，POST 方法用于处理“配置和管理”表单。

此方法可用来处理持久连接。

PUT 请求包含数据和 URL。代理服务器存储 URL 标识的资源中的数据。如果资源已经存在，PUT 将用包含在请求中的数据替换该资源。如果资源不存在，PUT 将创建它，并用包含在请求中的数据植入它。此方法可用来处理持久连接。

启用 PUT 方法允许通过使用 HTTP 和 FTP 将文件写入高速缓存代理。因为 PUT 允许客户机写到高速缓存代理，您需要使用服务器保护设置来定义谁能使用 PUT 和 PUT 能在哪些文件上使用。（请参阅第 103 页的第 25 章，『服务器保护设置』）。

TRACE

代理服务器回送由客户机发送的请求消息。该方法允许客户机查看在请求链的另一端接收到了什么，并使用那些数据来测试或诊断。代理响应的内容类型是 message/http。

关联的伪指令

下列伪指令启用 HTTP/FTP 方法：

- 第 187 页的『Enable - 启用 HTTP 方法』
- 第 185 页的『Disable - 禁用 HTTP 方法』

如需了解更多信息，请参阅第 13 页的第 4 章，『手工编辑 ibmproxy.conf 文件』。

“配置和管理” 表单

下列“配置和管理”表单将编辑关联伪指令的值:

- 服务器配置 -> 请求处理 -> HTTP 方法 -> GET
- 服务器配置 -> 请求处理 -> HTTP 方法 -> HEAD
- 服务器配置 -> 请求处理 -> HTTP 方法 -> POST
- 服务器配置 -> 请求处理 -> HTTP 方法 -> PUT
- 服务器配置 -> 请求处理 -> HTTP 方法 -> DELETE
- 服务器配置 -> 请求处理 -> HTTP 方法 -> OPTIONS
- 服务器配置 -> 请求处理 -> HTTP 方法 -> TRACE

注: 如果禁用 POST 方法, 您就不能使用“配置和管理”表单来配置高速缓存代理。

如需了解更多信息, 请参阅第 7 页的第 2 章, 『使用“配置和管理”表单』。

定义映射规则

映射规则就是配置使高速缓存代理以某种方法处理客户机请求的伪指令, 例如传递到源服务器 (proxied)、重定向或拒绝。正确设置映射规则对高速缓存代理的正确功能相当重要。映射规则会影响下列内容:

- 基本代理功能
- 访问基于浏览器的“配置和管理”表单
- 高速缓存 servlet 结果和其它动态生成的内容的能力

映射规则伪指令使用下列格式:

```
rule template target [IP_address | host_name]:[port]
```

只有那些与给出模板和 IP 端口组合匹配的请求才遵从那个规则。模板可包含通配符, 例如 `https://**/*.asp`。

规则出现在配置文件中的顺序很重要。除了 Map 伪指令外, 一旦请求与模板相匹配, 它就进行处理且不再评估后继规则。Map 伪指令将替换请求中的 URL。此新请求会继续与剩余映射规则相比较。

映射规则

下列映射规则应用于与给出模板匹配的客户机请求:

- **Map** - 重写请求。Map 规则将使用另一个 URL 字符串 (目标) 替换请求 URL (模板)。在此替代后, 会继续将包含新字符串的请求与剩余映射规则进行比较。
- **Pass, Exec** - 提供本地请求服务。Pass 和 Exec 规则处理代理服务器上的请求。Pass 规则将请求 URL (模板) 映射到代理服务器中提供服务的文件 (目标); Exec 规则将请求 URL 映射到代理服务器上运行的 CGI 程序。
- **Fail** - 拒绝请求。Fail 规则将拒绝代理服务器上的请求 (模板)。任何与 Fail 规则模板匹配的请求都不做进一步处理。Fail 规则没有目标自变量。
- **Redirect** - 转发请求。Redirect 规则将请求 (模板) 转发到另一个 Web 服务器 (目标)。因为完整 URL (包含通信协议) 是此规则的目标, 所以它可能会在此重定向期间更改协议, 例如将 SSL 加密添加到 HTTP 请求中。重定向在满足请求之前不会检查高速缓存。

- **Proxy, ProxyWAS** - 代理请求。Proxy 和 ProxyWAS 规则会将请求（模板）传递到另一个服务器（目标）。与简单的 Redirect 规则不同，Proxy 规则允许代理服务器检查高速缓存以满足请求，高速缓存来自源服务器的内容，以及写启用高级功能的 HTTP 头。当源服务器是 WebSphere Application Server 时，使用 ProxyWAS 规则而不使用 Proxy 规则。

下列映射规则应用于源服务器响应:

- **ReversePass** - 自动拦截重定向请求。当响应在其到客户机的途中经过代理服务器时，ReversePass 规则将该来自源服务器的响应与模板匹配。ReversePass 伪指令被设计成检测使客户机能直接与源服务器联系的重定向状态代码。然后指派客户机与目标自变量中定义的服务器进行联系。

下列映射规则应用于 API 应用程序:

- **nameTrans** - 在请求处理“名称转换”步骤期间，接受请求并运行 API 应用程序，该应用程序由替换文件路径定义。
- **service** - 在请求处理“服务”步骤期间，接受请求并运行 API 应用程序，该应用程序由替换文件路径定义。

配置代理服务器

要配置标准代理:

- 将代理服务器端口设置为 80。

Port 80

- 在所有其它规则之前添加 Proxy 规则, 此规则将端口 80 上接收到的所有请求代理到源服务器。

Proxy /* http://our.content.server.com/* :80

- 启用除 80 端口外的管理端口。

AdminPort 8080

这允许端口 80 上所有 HTTP 流量代理到源服务器。管理端口上输入的流量与初始通配符代理规则不匹配, 因此它不受影响。剩余映射规则将用于处理请求。

关联的伪指令

下列伪指令定义映射规则:

- 第 208 页的『Map - 将匹配请求更改为新的请求字符串』
- 第 216 页的『Pass - 指定接受请求的模板』
- 第 190 页的『Exec - 运行 CGI 程序以匹配请求』
- 第 235 页的『Redirect - 为发送到另一个服务器的请求指定模板』
- 第 228 页的『Proxy - 指定代理协议或逆向代理』
- 第 232 页的『ProxyWAS - 指定请求是发送到 WebSphere Application Server』
- 第 236 页的『ReversePass - 拦截自动重定向请求』

如需了解更多信息, 请参阅第 13 页的第 4 章, 『手工编辑 ibmproxy.conf 文件』。

“配置和管理” 表单

下列“配置和管理”表单编辑关联伪指令的值:

- 服务器配置 -> 请求处理 -> 请求路由

注: “配置和管理”表单不支持端口号自变量。

如需了解更多信息, 请参阅第 7 页的第 2 章, 『使用“配置和管理”表单』。

启用结点重写 (可选)

JunctionRewrite 伪指令支持高速缓存代理中的结点重写例程重写来自源服务器的响应, 以确保当使用结点时服务器相对 URL 获得到适当源服务器的映射。同时必须启用结点重新插件。结点是通过代理映射规则定义的。

当使用代理映射规则定义结点时, 您可以使用具有或不具有 JunctionPrefix 选项的 Proxy 伪指令。

定义不具有 JunctionPrefix 选项的结点

以下是一些有效结点的示例, 它们可按照结点重写例程进行操作:

- Proxy /shop/* http://shopserver.acme.com/*

- Proxy /auth/* http://authserver.acme.com/*

以下是一个有效结点的示例，它将不按照结点重写例程进行操作：

- Proxy /* http://defaultserver.acme.com/*

以下是一些无效结点的示例：

- Proxy /images/*.gif http://imageserver.acme.com/images/*.gif
- Proxy /cgi-bin/* http://cgiserver.acme.com/cgi/perl/*

这些映射规则已为 shopserver、authserver 和 b2bserver 创建了结点。请考虑 shopserver 将返回 HTML 文档和下列 URL（它们包含在适当的 HTML 标记中）：

- /index.html（服务器相对引用）
- /images/shop.gif（服务器相对引用）
- buy/buy.jsp（目录相对引用）
- http://ebay.com（绝对引用）

结点重写例程将使用如下所示的代理映射规则重写服务器相对引用：

- /shop/index.html（已更改）
- /shop/images/shop.gif（已更改）
- buy/buy.jsp（未更改）
- http://ebay.com（未更改）

定义具有 JunctionPrefix 选项的结点（建议的方法）

当 JunctionPrefix 选项和 Proxy 伪指令协同使用时，您可以使用以下格式在 Proxy 规则中声明结点前缀，而不用从 Proxy 规则中的第一个 URL 模式推断 JunctionPrefix：

```
Proxy url_pattern1 url_pattern2 JunctionPrefix: url_prefix
```

当使用 JunctionPrefix 时，不存在对于第一种 URL 模式的格式的限制。为了支持结点重写，当不使用 JunctionPrefix 选项时，代理 URL 必须具有以下格式 Proxy /market/* http://b2bserver/*。然而，当使用 JunctionPrefix 时，以下 Proxy 规则对于结点重写有效：

```
Proxy /market/partners/*.html http://b2bserver.acme.com/*.html
      junctionprefix:/market/partners
```

结点重写例程影响下列标记：

表 3. 受结点重写例程影响的标记

标记	属性
!—	URL
a	href
applet	archive, codebase
area	href
base	href
body	background
del	cite
embed	pluginspage

表 3. 受结点重写例程影响的标记 (续)

标记	属性
form	action
input	src
frame	src, longdesc
iframe	src, longdesc
ilayer	src, background
img	src, usemap, lowsrc, longdesc, dynsrc
layer	src, background
link	href
meta	url
object	data, classid, codebase, codepage
script	src
table	background
td	background
th	background
tr	background

注: 结点重写例程不会影响由 JavaScript 或浏览器中的插件生成的标记。

关联的伪指令

下列伪指令用于启用结点重写例程和插件。

- 第 240 页的『ServerInit - 定制服务器初始化步骤』
- 第 247 页的『Transmogriifier - 定制数据操作步骤』
- 第 202 页的『JunctionRewrite - 启用 URL 重写』
- 第 202 页的『JunctionRewriteSetCookiePath — 当与 JunctionRewrite 插件协同使用时, 重写 Set-Cookie 头中的路径选项』
- 第 201 页的『JunctionReplaceUrlPrefix — 当与 JunctionRewrite 插件协同使用时, 替换 URL 而不是插入前缀』
- 第 202 页的『JunctionSkipUrlPrefix — 当与 JunctionRewrite 插件协同使用时, 跳过已包含前缀的重写 URL』

如需了解更多信息, 请参阅第 13 页的第 4 章, 『手工编辑 ibmproxy.conf 文件』。

“配置和管理” 表单

下列“配置和管理”表单可用于启用结点重写插件:

- 服务器配置 -> 请求处理 -> API 请求处理

注: “配置和管理”不支持 JunctionRewrite 伪指令。

如需了解更多信息, 请参阅第 7 页的第 2 章, 『使用“配置和管理”表单』。

使用 Cookie 替代 JunctionRewrite

您可以按如下所示使用 cookie 存储后端服务器信息：cookie 被发送到客户机浏览器。当浏览器发送 HTML 页面中的资源请求时，它连接一个 cookie 以便高速缓存代理把这些请求转发给正确的后端服务器。

要使用 cookies 替代 JunctionRewrite，对 ibmproxy.conf 文件进行以下修改：

1. 用 **JunctionRewrite on UseCookie** 更改 **JunctionRewrite on**。
2. 注释掉 JunctionRewrite 插件。

以下是 JunctionRewrite 插件和 cookie 实现的比较。

- JunctionRewrite 插件
 - HTML 页面是重写的。
 - 除非使用 transmogrifier 插件，否则不支持脚本语言和 applet 的重写。请参阅『用于扩展 JunctionRewrite 功能的样本 transmogrifier 插件』。
 - 降低性能。
 - 对后端服务器配置没有限制。客户机可以跨越访问会话中的后端服务器。
- Cookie 实现
 - HTML 页面不是重写的。Cookie 发送到该客户机。
 - 客户机浏览器必须启用 cookie 支持。
 - 提高性能。
 - 对后端服务器配置有一些限制。仅当客户机访问会话中的后端服务器时才可使用。

注：当 JunctionRewrite 和 UseCookie 选项协同使用时，存在已知的限制。它将为所有请求错误地转换 URL，即使 cookie 仅应用于主机的一个子目录。以下是两种正确处理在 ROOT 下并且不需要结点的 URL 的方法。

- 在将 JunctionRewrite 伪指令放入 ibmproxy.conf 文件中前，先要放入代理规则。（将不重写所有在 JunctionRewrite 伪指令前放入文件的代理规则。）
- 显式映射每个 URL，而不是使用通配符（*）。例如：

```
Proxy /no-junction.jpg http://login-server/no-junction.jpg
```

用于扩展 JunctionRewrite 功能的样本 transmogrifier 插件

把 HTML 文件中的重写并语法分析 JavaScript™（SCRIPT）和 applet（APPLET）标记块提供给 Customizable 样本代码。JunctionRewrite 插件不能单独处理 JavaScript 中或 Java™ 参数值中的资源链接。

安装高速缓存代理后，您可以编译该同一代码并配置它以与 JunctionRewrite 一起运行。

下列样本文件位于您下载修订包的目录下的 ...samples/cp/ 子目录。

- Makefile（该样本插件的 Makefile）
- junctionRewrite2.h（用于定制解析器处理程序的接口）
- junctionRewrite2.c（上述接口的实现）
- scriptHandler.c（样本 JavaScript 重写处理程序）
- appletHandler.c（样本 Applet 块处理程序）

- junctionRewrite2.def (Windows 插件定义文件)
- junctionRewrite2.exp (Linux 和 UNIX 插件导出文件)

第 11 章 管理本地内容的传递

Pass 和 Exec 映射规则用于将本地内容传递到请求客户机。缺省情况下，一个含有通配符模板的 Pass 规则作为最后一个映射规则。此规则将使所有与先前模板不匹配的请求从目标目录中检索文件，这里的目标目录通常是指文档根目录。

当接收到一个不包含文件名的 URL 时，高速缓存代理将通过搜索一定的目录（如果有指定的目录，则为该指定目录，或者如果没有指定目录，则为文档根目录），查找一个与配置文件中指定的欢迎页面的列表相匹配的文件，以满足请求。如果定义了多个欢迎页面，代理服务器将按定义它们的顺序来搜索页面。将提供第一个找到的欢迎页面。

服务器主页是服务器在接收到只包含服务器的 URL（没有目录或文件名）的请求时，缺省情况下将传递的 Web 页面。正如先前说明的那样，缺省通配符映射规则要求服务器主页存储在文档根目录中，且主页的文件名与定义的欢迎页面匹配。

注：某些 Web 浏览器使用术语**主页**，它指的浏览器启动时首先装入的页面。本文档使用此术语仅指服务器主页。

本章描述了如何定义文档根目录和欢迎页面。

定义文档根目录

缺省文档根目录为：

- 在 Linux 和 UNIX 上：/opt/ibm/edge/cp/server_root/pub/lang/
- 在 Windows 上：drive:\Program Files\IBM\edge\cp\server_root\pub\lang\，或者安装期间您指定作为 HTML 目录的目录

关联的伪指令

下列伪指令定义了文档根目录：

- 第 216 页的『Pass - 指定接受请求的模板』

如需了解更多信息，请参阅第 13 页的第 4 章，『手工编辑 ibmproxy.conf 文件』。

“配置和管理” 表单

要更改“配置和管理”表单中的文档根目录，使用下列过程：

1. 选择**服务器配置** -> **请求处理** -> **请求路由**。
2. 在请求路由表中，在**请求模板**列中查找包含字符串 /*（斜杠星号）的行。这代表文档根目录。在表下面的**索引框**，单击相应于那个行的**索引列**中的数的数。
3. 单击**替换**。
4. 在**操作**下拉列表中，单击 **Pass**。
5. 在“URL 请求模板”字段输入 /*。
6. 在**替换文件路径**字段中输入您新的文档根目录。
7. 单击**提交**。

8. 在您的更改被接受后，单击上部框架中的**重新启动服务器**图标 (|)。

重新启动后，服务器开始使用新的文档根目录。

如需了解更多信息，请参阅第 7 页的第 2 章，『使用“配置和管理”表单』。

定义缺省欢迎页面

服务器在文档根目录中查找主页，但它返回的特定文件由欢迎页面的列表定义。

关于欢迎页面

当服务器接收到没有指定文件名的 URL 请求时，它尝试根据服务器的配置文件中设置的欢迎页面的列表来满足请求。此列表定义了作为缺省主页使用的文件。服务器通过将欢迎页面列表与文档根目录中的文件进行匹配来确定您的主页。第一个找到的匹配将作为您的主页返回。如果没有找到匹配，服务器将显示文档根目录的列表。

要将一个特定文件用作您的服务器的主页，并在请求没有指定目录或文件名时返回该文件，您必须把文件放到文档根目录中，并确保它的名称与欢迎页面列表中列出的一个文件名相匹配。

缺省配置文件按以下顺序定义这些文件名用作为欢迎页面：

1. welcome.html 或 welcome.htm
2. index.html 或 index.htm
3. Frntpage.html

服务器返回第一个它找到的匹配列表中的文件名的文件。在您创建 welcome.html 或 index.html 文件并将其放到文档根目录之前，服务器使用 Frntpage.html 作为您的主页。

例如，如果您使用缺省配置，且您的文档根目录中不包含名为 welcome.html 的文件，但包含名为 index.html 和 FrntPage.html 的文件，则 index.html 文件将用作为您的主页。

如果没有找到主页，文档根目录的内容将显示为目录。

关联的伪指令

下列伪指令定义了欢迎页面：

- 第 251 页的『Welcome - 指定欢迎文件的名称』

如需了解更多信息，请参阅第 13 页的第 4 章，『手工编辑 ibmproxy.conf 文件』。

“配置和管理”表单

下列“配置和管理”表单定义了欢迎页面；

- 服务器配置 -> 目录和欢迎页面 -> 欢迎页面

如需了解更多信息，请参阅第 7 页的第 2 章，『使用“配置和管理”表单』。

第 12 章 管理 FTP 连接

高速缓存代理将对 FTP URL 的请求代理到相应的 FTP 服务器，但无法用于代理来自 FTP 客户机的请求。它只能支持那些接收自 HTTP 客户机的 FTP 请求（使用 ftp:// 协议方案）。

对 FTP 文件的请求只支持 GET、PUT 和 DELETE 方法。对 FTP 目录列表的请求只支持 GET 方法。缺省情况下，PUT 和 DELETE 在高速缓存代理中是禁用的。如需了解更多信息，请参阅第 37 页的『启用 HTTP/FTP 方法』。

本章描述如何保护 FTP 文件和管理 FTP 服务器登录、目录路径和链接。

保护 FTP 文件

如果您已启用 PUT 方法来上载 FTP 文件或启用 DELETE 方法来删除 FTP 文件，您至少需要为 PUT 和 DELETE 请求定义 FTP 代理保护，从而防止未授权的文件在您的 FTP 服务器更新。

要保护代理 FTP 请求，在“配置和管理”表单，选择**服务器配置 -> 文档保护**。要创建 FTP 文件请求的保护设置，在请求模板开始时，包含 ftp://。例如，要保护名为 exams 目录中的文件，使用模板 ftp://exams/*。

要获取有关创建保护设置的更多信息，请参阅第 103 页的第 25 章，『服务器保护设置』。

管理 FTP 服务器登录

如果在请求 URL 中未指定用户标识和密码，高速缓存代理将尝试匿名登录到所请求的 FTP 服务器（使用用户标识 ANONYMOUS）。许多 FTP 服务器需要一个电子邮件地址作为匿名 FTP 的密码。如果 FTP 服务器要求一个匿名登录的密码，高速缓存代理将发送配置文件中由 WebmasterEmail 伪指令指定的电子邮件地址。

要在“配置和管理”表单中设置 Web 管理员电子邮件地址，选择**服务器配置 -> 系统管理 -> SNMP MIB**。电子邮件地址还可用 WebmasterEmail 伪指令设置；有关详细信息，请参阅参考章节：第 250 页的『WebMasterEMail - 设置电子邮件地址以接收选择服务器报告』。

如果请求 URL 中的 FTP 服务器需要特定用户标识和密码来登录，用户可在请求 URL 中输入用户标识和密码，例如：

```
ftp://userid:password@ftpserverhost/
```

如果您不要为 URL 请求中的 FTP 用户标识指定密码，用户能在 URL 中仅输入用户标识：ftp://userid@ftpserverhost。高速缓存代理将首先尝试用指定的用户标识来登录到 FTP 服务器，并且没有密码。如果没有密码登录失败，则浏览器提示与指定用户标识相关联的密码。

对匿名外的其它登录，至少在 URL 中必须指定用户标识。如果未指定用户标识，则尝试匿名登录并且不会提示客户机输入用户标识。

管理 FTP 目录路径

您必须为高速缓存代理指定 FTP URL 中的路径名是被解释为相对于用户的工作目录，还是相对于根目录。例如，如果登录到 FTP 服务器的用户有名为 /export/home/user1 的缺省工作目录，并且用户要从名为 test 的子目录中检索一个名为 test1.exe 的文件，那么代理将使用下列 URL 从 FTP 服务器检索文件（这取决于如何解释 FTP URL）：

- 如果设置了绝对路径名：
ftp://user1:user1pw@FTPhost/export/home/user1/test/test1.exe
- 如果设置了相对路径名：ftp://user1:user1pw@FTPhost/test/test1.exe

如果设置了相对 FTP URL 路径，用户仍能使用转义约定（用 %2F 转义初始斜杠字符（/））来指定绝对路径名，从而表明根目录。例如，如果用户 1，其工作目录为 /export/home/user1，要访问用户 2 工作目录 /export/home/user2 中的文件，请求 ftp://user1:user1pw@FTPhost/%2Fexport/home/user2/file 被正确地解释为相对于根目录 /（即，作为一个绝对路径名）的 URL，即使选择了相对 FTP URL。

要指定如何解释 FTP URL，在“配置和管理”表单，选择代理配置 -> 代理性能。在表单的稍下部分，FTP URL 路径应为：下，选择绝对路径以指定服务器根目录，或相对路径以指定用户工作目录作为路径的开始。

该设置还能在代理配置文件中更改；要获取更多的信息，请参阅第 194 页的『FTPUrlPath - 指定如何解释 FTP URL』。

管理 FTP 链接

如果将多个 Web 代理服务器链接在一起，那么您可指定将包含 FTP URL 的请求发送到相链接的 Web 代理服务器而不是直接发送到 FTP 服务器。要为 FTP 请求指定链接的代理服务器，在“配置和管理”表单中，选择代理配置 -> 代理链接和非代理域。http:// 协议方案用于指定链接代理的 URL，尽管这是对于 ftp:// 协议方案的链接请求。

要通过使用代理配置文件配置 FTP 链接，请参阅第 194 页的『ftp_proxy - 指定 FTP 请求的另一个代理服务器』上的参考章节。

第 13 章 定制服务器处理

本章描述如何使用服务器端包含将信息插入发送到客户机的 CGI 程序和 HTML 文档。同时还讨论定制服务器的错误消息和资源映射。

服务器端包含

当作为源服务器时（即，不是到已代理的或高速缓存的对象），服务器端包含允许您添加信息到 CGI 程序和 HTML 文档（由服务器发送到客户机）。当前日期、文件的大小以及最后更改文件的日期是能发送到客户机的信息的类型的示例。本节描述服务器端包含的命令格式并解释如何在您的 CGI 程序和 HTML 文档中使用服务器端包含命令。您还可使用服务器端包含来定制错误页面。

服务器端包含的注意事项

在您的服务器上使用服务器端包含前，您应该考虑性能、安全性和风险问题：

- 当服务器处理正在发送的文件时，可能会显著地削弱性能。
- 如果您让普通的用户在您的服务器上执行命令，就可能会削弱安全性。当您决定在哪个目录放置服务器端包含，或在哪个目录放置 **exec** 命令时要谨慎。如果您不启用 **exec** 命令，您能将安全性风险最小化。
- 使用服务器端包含会导致一些问题。例如，文件无法递归地引用：如果您正在执行文件 `sleepy.html`，并且程序找到 `<-- !#include file="sleepy.html" -->`，则服务器不检测错误并且会发生故障。（非递归地在其它文件中参考文件不是问题。）

服务器端包含的配置

要启用服务器端包含，在“配置和管理”表单，选择**服务器配置 -> 基本设置**。使用该表单指定下列服务器端包含类型中哪些是可接受的：

- CGI 脚本
- 文件
- 除使用 **exec** 命令的 CGI 脚本外的所有
- 无

还可使用该表单指定是否服务器端包含处理除其它文件类型外的文本或 HTML 文档。

另外，确保您用于包括的文件扩展名是可识别的。在“配置和管理”表单，选择**服务器配置 -> MIME 类型和编码**，并使用 **MIME 类型** 表单。注意，`shtml` 和 `htmls` 扩展名缺省是可识别的。

要通过编辑代理配置文件中的伪指令来为服务器端包含配置您的服务器，请参阅下列伪指令的参考章节：

- 第 162 页的『`AddType` - 为具有特殊后缀的文件指定数据类型』
- 第 199 页的『`imbeds` - 指定是否使用服务器端包含处理』

服务器端包含的格式

包含命令必须作为注释包含在 HTML 文档或 CGI 程序中。命令有下列格式:

```
<!--#directive tag=value ... -->
```

或

```
<!--#directive tag="value" ... -->
```

值周围的引号是可选的，但当要嵌入空格时是必需的。

服务器端包含的伪指令

本节解释对服务器端包含可被服务器接受的伪指令。(不要将这些伪指令和第 155 页的附录 B, 『配置文件伪指令』中的代理配置文件的伪指令相混淆)。

配置 - 控制文件处理

使用此伪指令来控制文件处理的某几个方面。有效的标记是 `cmntmsg`、`errmsg`、`sizefmt` 和 `timefmt`。

cmntmsg

使用此标记指定由其它伪指令添加的注释开始前面的消息。对于包含伪指令规范和 “-->” 之间的文本的任何伪指令，该文本作为一个注释并添加到此文件，服务器把此文件发送到客户机。

示例:

```
<!--#config cmntmsg="[This is a comment]" -->
<!-- #echo var=" " extra text -->
```

结果: <!--[This is a comment] extra text -->

缺省值: [the following was extra in the directive]

errmsg

如果在处理文件时发生错误，使用该标记来指定发送到客户机的消息。消息记录在服务器的错误日志中。

示例:

```
<!-- #config errmsg="[An error occurred]" -->
```

缺省值: "[An error occurred while processing this directive]"

sizefmt

使用此标记来指定已显示的文件大小的格式。在下列示例中，字节是用于显示字节数的值，而 `abbrev` 是用于显示千字节或兆字节数的值。

示例 1:

```
<!--#config sizefmt=bytes -->
<!--#fsize file=foo.html -->
```

结果: 1024

示例 2:

```
<!--#config sizefmt=abbrev -->
<!--#fsize file=foo.html -->
```

结果: 1K

缺省值: "abbrev"

timefmt

使用此标记来指定用于提供日期的格式。

示例:

```
<!--#config timefmt="%D %T" -->  
<!--#flastmod file=foo.html -->
```

结果: "10/18/95 12:05:33"

缺省值: "%a, %d %b %Y %T %Z"

下列 strftime() 格式与 timefmt 标记有效:

说明符	含义
%%	用 % 替换
%a	用缩写的星期几名称替换
%A	用完整的星期几名称替换
%b	用缩写的月份名称替换
%B	用完整的月份名称替换
%c	用日期和时间替换
%C	用世纪数字 (年除以 100 并截尾) 替换
%d	用月份的天数 (01-31) 替换
%D	插入日期为 %m/%d/%y
%e	以十进制数插入月份 (01-12) (仅在 C POSIX 下, 它是 2 位字符、右对齐、以空白填充的字段)
%E[cCxyY]	如果替换日期 / 时间格式不可用, %E 描述符将映射到其未扩展的副本 (例如, %EC 映射到 %C)
%Ec	用替换日期和时间表示法替换
%EC	用在替换表示法中的基本年 (周期) 的名称替换
%Ex	用替换日期表示法替换
%EX	用替换时间表示法替换
%Ey	从替换表示法 %EC (仅年份) 的偏移量替换
%EY	用全部替换年份表示法替换
%h	用缩写的月份名称 (与 %b 一样) 替换
%H	以十进制数 (00-23) 用小时 (23 小时时钟) 替换
%I	以十进制数 (00-12) 用小时 (12 小时时钟) 替换
%j	用年份的天数 (001-336) 替换
%m	用月份 (01-12) 替换
%M	用分钟 (00-59) 替换
%n	用新行替换
%O[deHImMSUwWy]	如果替换的日期 / 时间格式不可用, %E 描述符将映射到其未扩展的副本 (例如, %Od 映射到 %d)
%Od	要用月份的天数替换, 使用替换数字符号, 用前导零按所需填充 (如果有零的任何替换符号), 否则用前导空格

说明符	含义
%Oe	要用月份的天数替换，使用替换数字符号，按所需用前导空格填充
%OH	用替换数字符号，以小时（24 小时时钟）替换
%OI	用替换数字符号，以小时（12 小时时钟）替换
%Om	用替换数字符号，以月份替换
%OM	用替换数字符号，以分钟替换
%OS	用替换数字符号，以秒替换
%OU	用替换数字符号，以年份的周数（星期日作为一星期的第一天，与 %U 相应的规则）替换
%Ow	用替换数字符号，以星期几（Sunday=0）替换
%OW	用替换数字符号以年份的周数（星期一作为一星期的第一天）替换
%Oy	用替换数字符号以替换表示法中的年份（从 %C 偏移）替换
%p	用本地对等的 AM 或 PM 替换
%r	用与字符串对等的 %I:%M:%S%p 替换
%R	用 24 小时记数法（%H:%M）时间替换
%S	用秒钟（00-61）替换
%t	用制表符替换
%T	用与字符串对等的 %H:%M:%S 替换
%u	用星期几以十进制数（1-7）替换，1 代表星期一
%U	用年份的周数（00-53）替换，以星期日作为一星期的第一天
%V	用年份的周数（01-53）替换，以星期一作为一星期的第一天
%w	用星期几（0-6）替换，星期日为 0
%W	用年份的周数（01-53）替换，以星期一作为一星期的第一天
%x	用适当的日期表示法替换
%X	用适当的时间表示法替换
%y	用世纪数字之中的 2 位数字年替换
%Y	用全部 4 位数字年数替换
%Z	如果是未知时区，用时区的名称或无字符替换

操作系统配置确定完整和缩写的月份名称和年份。

echo - 显示变量值

使用此伪指令显示用 `var` 标记指定的环境变量值。如果未找到变量，将显示（无）。同样，`echo` 能显示由 `set` 或 `global` 伪指令设置的值。可显示下列环境变量：

DATE_GMT

格林威治标准时间当前的日期和时间。该变量的格式化使用 `config timefmt` 伪指令来定义。

DATE_LOCAL

当前日期和本地时间。该变量的格式化使用 `config timefmt` 伪指令来定义。

DOCUMENT_NAME

最顶端文档的名称。如果 HTML 已由 CGI 生成，此变量包含 CGI 的名称。

DOCUMENT_URI

客户机请求的完整 URL，不带查询字符串。

LAST_MODIFIED

最后修改当前文档的日期和时间。该变量的格式化使用 **config timefmt** 伪指令来定义。

QUERY_STRING_UNESCAPED

由客户机发送的搜索查询。除非 HTML 由 CGI 生成，否则它是未定义的。

SSI_DIR

当前文件的路径，与 SSI_ROOT 相关。如果当前文件在 SSI_ROOT 中，则值为 “/”。

SSI_FILE

当前文件的文件名。

SSI_INCLUDE

在包含命令中使用的值，检索当前文件。它不是为最顶端文件定义的。

SSI_PARENT

文件的路径和文件名包含检索当前文件的包含命令（与 SSI_ROOT 相关）。

SSI_ROOT

最顶端文件的路径。所有包含请求必须在该目录或该目录的子目录中。

示例:

```
<!--#echo var=SSI_DIR -->
```

exec - 指定 CGI 程序

使用该伪指令来包含 CGI 程序的输出。exec 伪指令废弃任何 CGI 输出的 HTTP 头，除了下列:

Content-type

确定是否为其它包含分析输出的主体

Content-encoding

确定是否需要完成 EBCDIC 到 ASCII 的转换。

Last-modified

替换当前最后修改的头值，除非当前值比指定值更新。

cgi - 指定 CGI 程序 URL

使用该伪指令来指定 CGI 程序的 URL。

在该示例中，程序是要执行的 CGI 程序，并且 **path_info** 和 **query_string** 代表一个或多个传递到程序的参数作为环境变量:

```
<!--#exec cgi="/cgi-bin/program/path_info?query_string" -->
```

该示例显示变量的使用:

```
<!--#exec cgi="&path;&cgiprogram;&pathinfo;&querystring;" -->
```

flastmod - 显示文档最后更改的日期和时间

使用该伪指令来显示文档最后更改的日期和时间。该变量的格式化由 **config timefmt** 伪指令定义。**file** 和 **virtual** 标记与该伪指令一起使用时有效，其意义定义如下。

伪指令格式:

```
<!--#flastmod file="/path/file" -->
<!--#flastmod virtual="/path/file" -->
```

file 使用该标记来指定文件名称。对于 **flastmod**、**FSIZE** 和 **include**，假设 **file** 与 **SSI_ROOT** 相关（如果以 “/” 前导）。否则，它与 **SSI_DIR** 相关。所指定的文件必须在 **SSI_ROOT** 中或其子代中存在。例如:

```
<!--#flastmod file="/path/file" -->
```

virtual 使用该标记来指定到文档的虚拟路径的 URL。对于 **flastmod**、**FSIZE** 和 **include**，**virtual** 永远是通过服务器的映射伪指令的。例如:

```
<!--#flastmod virtual="/path/file" -->
```

示例:

```
<!--#flastmod file="foo.html" -->
```

结果: 12May96

FSIZE - 显示文件大小

使用该伪指令来显示指定文件的大小。该变量的格式化由 **config sizefmt** 伪指令定义。**file** 和 **virtual** 标记与该伪指令一起使用时有效，它们的意义与之前为 **flastmod** 伪指令定义的相同。

示例:

```
<!--#FSIZE file="/path/file" -->
<!--#FSIZE virtual="/path/file" -->
```

结果: 1K

global - 定义全局变量

使用该伪指令来定义稍后可由该文件或任何包含的文件回送的全局变量。

示例:

```
<!--#global var=VariableName value="SomeValue" -->
```

例如，要引用跨虚拟边界的父文档，您需要设置全局变量 **DOCUMENT_URI**。您还需要引用在子文档中的全局变量。该示例显示您要在父文档中插入的 HTML 编码:

```
<!--#global var="PARENT_URI" value=&DOCUMENT_URI; -->
```

该示例显示您要在子文档中插入的 HTML 编码:

```
<!--#flastmod virtual=&PARENT_URI; -->
```

include - 包含输出中的文档

使用该伪指令包含来自输出中文档的文本。**file** 和 **virtual** 标记与该伪指令一起使用时有效，它们的意义与之前为 **flastmod** 伪指令定义的相同。

set - 设置要回送的变量

使用该伪指令来设置稍后能回送的变量，但仅通过该文件。

示例:

```
<!--#set var="Variable 2" value="AnotherValue" -->
```

在定义伪指令时，您能在 value 中回送字符串。例如:

```
<!--#include file="&filename;" -->
```

变量：服务器端设置伪指令通常后跟一个回送伪指令，以便其查找设置变量（在找到变量的地方回送），并继续函数。它可包含多个对变量的引用。服务器端设置还允许您回送已设置好的变量。如果未找到设置变量，则无任何显示。

当服务器端设置在服务器端包含伪指令中遇到变量引用，将尝试在服务器端解决。在下列示例的第二行，服务器端变量 `&index;` 用于和字符串 `var` 一起来构造变量名 `var1`。变量 `&var1;` 然后通过转义 `ê` 中的 `&` 来指定值，这样它不会被识别为一个变量。相反，它作为字符串用于创建值 `frêd`，或 `fred` 用 `e` 上的弯曲。变量 `ê` 是一个客户端变量。

```
<!--#set var="index" value="1" -->
<!--#set var="var&index;" value="fr\&ecirc;d" -->
<!--#echo var="var1" -->
```

可转义的字符（称作转义变量）前有反斜杠（\）并包含下列：

字符	含义
\a	警告（响铃）
\b	退格
\f	换页（新页面）
\n	新行
\r	回车
\t	水平制表
\v	垂直制表
\'	单引号
\"	双引号
\?	问号
\\	反斜杠
\-	连字号
\.	点
\&	& 号

定制错误消息

您可定制高速缓存代理返回的错误消息，并可定义特殊错误条件的特定消息。在“配置和管理”表单中，选择**服务器配置 -> 错误消息定制**。使用此表单选择错误条件并指定特殊 HTML 文件用于此条件。

要通过编辑代理配置文件中的伪指令来定制错误消息，请参阅伪指令第 188 页的『**ErrorPage** - 为特定错误条件指定定制消息』的参考章节。

实时流式协议 (RTSP) 重定向

WebSphere Application Server, 版本 6.0 以 RTSP 重定向器形式引入流式媒体支持。RTSP 使高速缓存代理能够充当与媒体播放器的第一个接触点, 并且能够重定向它们的请求到一个合适的代理服务器, 或一个将提供所请求的媒体内容的内容服务器。

RTSP (实时流式协议) 在 RFC 2326 中定义。它是一个用于控制数据流的因特网标准协议。尽管它不包含发送流的技术, 但它足够灵活可用于控制与视频或音频回放无关的数据流。

关于 RTSP 重定向

RTSP 重定向特性允许高速缓存代理重定向 RTSP 所控制的任何流式媒体会话。这些包括下列类型的媒体:

- RealNetworks 录制的音频
- RealNetworks 录制的视频
- RealNetworks 活动流 (音频和视频)
- Microsoft Media Player 文件
- Apple Quicktime 媒体文件

任何可配置为与代理服务器在其 RTSP 端口 (通常为 554) 上联系的播放器都可使用高速缓存代理中的此框架, 使它的请求由 RTSP 重定向器来处理。

RTSP 重定向器不高速缓存或直接代理媒体表示。RTSP 重定向器必须与第三方流式媒体服务器联合使用, 以提供两个功能之一或所有功能。带有 RTSP 重定向器的高速缓存代理必须拥有对一个或多个 RTSP 代理服务器的网络访问权。

RTSP 限制

此功能受到如下限制:

当前, 仅支持 RealNetworks 技术。这些包括 RealProxy 代理服务器、RealServer 源服务器和 RealPlayer 媒体播放器。

RTSP 增强

以前, RTSP 重定向器存在如下限制: 对任意 URL 的同一源服务器的所有请求以同一方法进行重定向。不能根据文件名或所请求 URL 的其它部分进行重定向。此限制不再适用。RTSP 重定向器现在使用所接收请求中的完整 URL, 以及高速缓存代理配置文件中设置的阈值 (rtsp_proxy_threshold), 以确定将客户机请求重定向到源服务器还是代理服务器。现在, 对同一源服务器的请求将逐一进行处理。

配置 RTSP 重定向

下列配置文件伪指令用于控制 RTSP 重定向。这些伪指令的设置不是由服务器重新启动来刷新的。服务器必须完全停止, 然后在对这些伪指令的更改生效之前再次启动。

- 第 237 页的『RTSPEnable - 启用 RTSP 重定向』
- 第 237 页的『rtsp_proxy_server - 指定用于重定向的服务器』
- 第 237 页的『rtsp_proxy_threshold - 指定在重定向到高速缓存之前的请求数』
- 第 237 页的『rtsp_url_list_size - 指定代理内存中的 URL 数』

第 14 章 配置头选项

当请求文档时，Web 客户机发送头（提供关于浏览器或请求的其它信息）。当请求发送时，头生成自动发生。

高速缓存代理允许几个选项来定制头信息从而让其从目标服务器保持隐藏。尽管为实际的头替换更多的一般头时，有增加客户机匿名的好处，它有禁用写入一些 Web 页面的基于头的页面定制的缺点。

头通常使用此格式：

```
User-Agent: Mozilla 2.02/OS2
Client-IP: 45.37.192.3
Referer: http://www.bigcompany.com/WebTrafficExpress/main.html
```

此头包含下列字段：

- **User-Agent:** 提供关于浏览器和操作系统的信息。
- **Client-IP:** 提供请求 URL 客户机的 IP 地址。
- **Referer:** 用与该页面链接的参考 URL 提供目标服务器。

大多数头都能用适当的代理配置设置进行阻拦。然而，一些头字段是由源服务器请求的，因此，阻拦这些头可能导致显示不正确的 Web 页面（例如，在某些情况下，阻拦“host”头字段可导致用户看到错误的 Web 页面）。要获取关于头字段的进一步信息，请参阅 HTTP V1.1 规范。

关联的伪指令

要通过编辑代理配置文件来更改头选项，请参阅下列伪指令的参考章节：

- 第 215 页的『NoProxyHeader - 将客户机头指定到块』
- 第 230 页的『ProxyFrom - 指定具有 From: 头的客户机』
- 第 230 页的『ProxyIgnoreNoCache - 忽略重新装入的请求』
- 第 231 页的『ProxySendClientAddress - 生成 Client IP Address: 头』
- 第 231 页的『ProxyUserAgent - 修改“User Agent”字符串』
- 第 232 页的『ProxyVia - 指定 HTTP 头的格式』

如需了解更多信息，请参阅第 13 页的第 4 章，『手工编辑 ibmproxy.conf 文件』。

“配置和管理”表单

您可使用两个“配置和管理”表单指定头选项：

- 选择代理配置 -> 隐私设置。在隐私设置表单，设置以下内容：
 - 转发客户机 IP 地址到目标服务器

如果您要将请求客户机的 IP 地址转发到目标（内容）服务器，选取此框。如果您没有选取该框，则目标服务器将接收代理服务器的 IP 地址。保留此框不选取将增加在 Web 冲浪时客户机的匿名性。

– User-agent 字符串

输入字符串来从头中发送到目标服务器以替换浏览器类型和客户机正在使用的操作系统。例如：指定高速缓存代理 4.0 替换以下头的 Mozilla 2.02/OS2:

```
Content-Type:MIME
User-Agent: Mozilla 2.02/OS2
Referer: http://www.ics.raleigh.ibm.com/WebTrafficExpress/main.html
Pragma:no-cache
```

– From:

当它分析 “From:” 头时，输入目标服务器读取的电子邮件地址。您可能希望指定代理管理员的电子邮件地址，因为管理员是需要接收任何问题报告的人。

– 单击**提交**更改配置文件。

• 选择代理配置 -> 代理头过滤。使用此表单列出 HTTP 头以阻拦:

1. 单击**添加**或**除去**，表明阻拦的头的索引位置。
2. 输入要阻拦的客户机 HTTP 头。（请参考 HTTP 1.1 规范以获得头完整的列表和说明。）
3. 单击**提交**更改配置文件。

如需了解更多信息，请参阅第 7 页的第 2 章，『使用 “配置和管理” 表单』。

第 15 章 关于应用程序编程接口

应用程序编程接口 (API) 在 *Programming Guide for Edge Components* 中有详细介绍。配置文件中的 API 伪指令将启用在请求处理 workflow 中的特定步骤期间调用的插件例程。除内置例程外，可替换或运行这些插件例程。

关联的伪指令

下列是 API 伪指令:

- 第 166 页的『Authentication - 定制认证步骤』
- 第 166 页的『Authorization - 定制授权步骤』
- 第 187 页的『Error - 定制 Error 步骤』
- 第 205 页的『Log - 定制 Log 步骤』
- 第 212 页的『Midnight - 指定用于归档日志的 API 插件』
- 第 212 页的『NameTrans - 定制名称转换步骤』
- 第 215 页的『ObjectType - 定制对象类型步骤』
- 第 220 页的『PostAuth - 定制 PostAuth 步骤』
- 第 221 页的『PostExit - 定制 PostExit 步骤』
- 第 221 页的『PreExit - 定制 PreExit 步骤』
- 第 240 页的『ServerInit - 定制服务器初始化步骤』
- 第 241 页的『ServerTerm - 定制服务器终止步骤』
- 第 241 页的『Service - 定制服务步骤』
- 第 247 页的『Transmogriifier - 定制数据操作步骤』
- 第 247 页的『TransmogriifiedWarning - 发送警告消息到客户机』

如需了解更多信息，请参阅第 13 页的第 4 章，『手工编辑 `ibmproxy.conf` 文件』。

“配置和管理” 表单

下列“配置和管理”表单编辑关联伪指令的值:

- 服务器配置 -> 请求处理 -> **API 请求处理**

如需了解更多信息，请参阅第 7 页的第 2 章，『使用“配置和管理”表单』。

第 4 部分 配置代理服务器高速缓存

本节讨论代理高速缓存以及如何配置。可设置高速缓存，以在内存（内存高速缓存）中或在一个或多个存储设备（磁盘高速缓存）上存储文件。可配置一个高速缓存刷新代理程序，以将被频繁请求的文件预装入到高速缓存中，并且各种 URL 过滤器可适用于高速缓存。本节还讨论通过使用远程高速缓存访问或因特网高速缓存协议（ICP）插件共享高速缓存，使用高速缓存垃圾收集除去过时文件，以及高速缓存动态生成的文件。

这一部分包含下列章节：

- 第 65 页的第 16 章，『代理服务器高速缓存概述』
- 第 69 页的第 17 章，『配置基本高速缓存』
- 第 73 页的第 18 章，『控制高速缓存哪些内容』
- 第 77 页的第 19 章，『维护高速缓存内容』
- 第 83 页的第 20 章，『为自动刷新和预装入配置高速缓存代理程序程序』
- 第 89 页的第 21 章，『使用共享高速缓存』
- 第 93 页的第 22 章，『动态地高速缓存生成的内容』
- 第 97 页的第 23 章，『调整代理服务器高速缓存』

第 16 章 代理服务器高速缓存概述

高速缓存是这样一种功能部件，在其中代理服务器保存客户机请求文件的本地副本，这样，当它们被相同或不同的客户机再次请求时，就可以从高速缓存为它们快速服务。

高速缓存代理遵从 HTTP 1.1 并且通常遵守 HTTP 1.1 协议，以高速缓存和确定文档的新鲜度。

本章将讨论代理服务器高速缓存的一些功能部件。对于那些可配置的功能部件，有关如何设置相应值的详细信息将包含在功能部件章节中。

高速缓存存储

代理服务器可在物理存储设备或系统内存中存储高速缓存。有利于您系统的高速缓存存储类型取决于您硬件的能力，和取决于需要更快的高速缓存响应还是需要高速缓存中存储大量的项。内存高速缓存的响应时间通常比磁盘高速缓存的响应时间更快，但内存高速缓存的大小受到代理服务器中 RAM 数量的限制。磁盘高速缓存的大小收到存储设备大小的限制，但通常它比 RAM 的数量大得多。

对于磁盘高速缓存，高速缓存代理使用原始磁盘高速缓存，它意味着代理服务器直接写入到高速缓存设备而不使用操作系统的读和写协议。必须使用 **htcformat** 命令来准备磁盘高速缓存的存储设备。有关 **htcformat** 的详细信息包含在节第 69 页的第 17 章，『配置基本高速缓存』中。

高速缓存索引

在内存高速缓存或磁盘高速缓存中，高速缓存代理都使用系统内存空间来保存高速缓存的索引，这样可减少查找已高速缓存文件的处理时间。

高速缓存代理的高速缓存目录结构和其查找方法不同于其它代理服务器的目录结构和查找方法。高速缓存代理在内存中维护一个索引，包含有关高速缓存中文件的信息。使用 RAM 查找而不是磁盘或其它媒体会导致较快的文件查找和检索。

索引包含高速缓存对象的 URL、高速缓存位置和到期信息。因为这个原因，所以保存索引必需的内存容量与高速缓存中的对象数目是成比例的。

当从客户机接收到一个请求时，代理为该 URL 检查内存中的高速缓存索引。

- 如果文件不在索引中，则请求发送到目标服务器。
 - 然后，URL 被检查以确定是否可以高速缓存已检索的文件。如果允许，代理服务器将高速缓存已检索的文件。
 - 然后，使用新高速缓存对象的 URL、位置和到期信息来更新高速缓存索引。
- 如果文件在索引中：
 - 检查到期信息以确定是否高速缓存的文件是新鲜的。
 - 如果对象已到期，则联系目标服务器并且用新检索的文档来替换该已到期对象。到期信息在高速缓存索引中更新。

- 如果对象还没有到期，则从代理高速缓存服务此文档。

FTP 高速缓存

当代理被配置成高速缓存请求时，除高速缓存 HTTP 文件请求外，它还可以高速缓存 FTP 文件请求。然而，由于 FTP 文件未包含与 HTTP 文件相同类型的头信息，所以计算高速缓存的 FTP 文件的到期日期与计算其它高速缓存的文件的到期日期不同。

当请求发送到 FTP 服务器以检索文件时，代理首先为文件发送一个 LIST 请求到 FTP 服务器，以获取有关该文件的 FTP 目录信息。如果 FTP 服务器响应肯定完成应答的 LIST 请求和该文件的目录信息，则代理创建一个 Last-Modified 头（包含从 FTP 目录信息分析的日期）。然后，代理的高速缓存功能使用此 Last-Modified 头以及配置文件中的 CacheLastModifiedFactor 伪指令的值设置，来确定 FTP 文件在到期之前应该保留在高速缓存中的时间长度。

要获取有关如何使用 Last-Modified 头和 CacheLastModifiedFactor 伪指令来确定文件在高速缓存中的保留时间的更多信息，请参阅第 77 页的第 19 章，『维护高速缓存内容』。

为特定用户标识而非通过匿名登录检索的 FTP 文件被认为是专用文件而不被高速缓存。

DNS 高速缓存

处理高速缓存 Web 内容之外，代理服务器还执行域名服务器（DNS）高速缓存。例如，当客户机请求一个来自 www.myWebsite.com 的 URL 时，代理要求它的 DNS 服务器将主机名 www.myWebsite.com 解析为 IP 地址。然后高速缓存 IP 地址以加速到该主机名的后继请求的响应时间。DNS 高速缓存是自动的，并且不能被重新配置。

高速缓存排除

某些文件和文档从不高速缓存。它们包含下列：

- 使用 HTTP 方法（如 POST 和 PUT）而不是 GET 方法从请求返回的文件。
- 任何需要认证的文档，除非高速缓存源服务器明确允许的文档。
- 任何 CGI 脚本的动态输出（因为每次请求它时，它都是唯一的）。当启用动态高速缓存时，可高速缓存 IBM WebSphere Application Server 执行 servlet 和 JavaServer Pages (JSP) 时动态生成的结果。请参阅第 93 页的第 22 章，『动态地高速缓存生成的内容』以获得详细信息。
- 任何从包含问号 (?) 的 URL 返回的文件，除非明确地允许查询高速缓存。（请参阅第 73 页的第 18 章，『控制高速缓存哪些内容』，以获得有关配置查询结果高速缓存的信息。）

还可通过设置高速缓存过滤器，来进一步限制被高速缓存的项。例如，您可能不希望代理服务器高速缓存来自代理提供本地服务的文件。请参阅第 73 页的第 18 章，『控制高速缓存哪些内容』以获得详细信息。

高速缓存管理

管理高速缓存涉及许多因素。作为服务器管理员，您可以指定下列：

- 要高速缓存哪些文档（请参阅第 73 页的第 18 章，『控制高速缓存哪些内容』以获得详细信息）。
- 可高速缓存多少文档（请参阅第 69 页的第 17 章，『配置基本高速缓存』以获得详细信息）。
- 高速缓存文档被认为是当前的时间长度（请参阅第 77 页的第 19 章，『维护高速缓存内容』以获得详细信息）。
- 清除高速缓存（垃圾收集）的频率和要保存哪类文件（请参阅第 77 页的第 19 章，『维护高速缓存内容』以获得详细信息）。
- 如何索引高速缓存的文档（请参阅第 69 页的第 17 章，『配置基本高速缓存』以获得详细信息）。
- 何时刷新高速缓存（请参阅第 83 页的第 20 章，『为自动刷新和预装入配置高速缓存代理程序程序』以获得详细信息）。
- 远程高速缓存访问（请参阅第 89 页的第 21 章，『使用共享高速缓存』以获得详细信息）。
- 如何保留和归档日志（请参阅第 69 页的第 17 章，『配置基本高速缓存』以获得详细信息）。

另外，还可对高速缓存配置进行调整，从而提高高速缓存代理的整体性能。要获得有关性能调整的详细信息，请参阅第 97 页的第 23 章，『调整代理服务器高速缓存』。

第 17 章 配置基本高速缓存

如果您使用 Edge components 产品安装程序中的缺省设置来安装高速缓存代理，则高速缓存是启用的，并且高速缓存存储在内存中。您可能要调整下列基本高速缓存设置，以定制适合您系统需要的高速缓存。

如果您未使用安装程序，请配置这些设置以启用高速缓存。

配置高速缓存必需的基本步骤如下所示：

1. 启用高速缓存。
2. 配置高速缓存存储。

在配置基本高速缓存设置之后，您可能要添加或更改下列功能部件的设置。

- 定制高速缓存。
- 保存或装入高速缓存内存到磁盘。
- 通过使用 URL 过滤器，限制被高速缓存的内容。
- 通过为查询结果或动态生成的文件启用高速缓存，展开被高速缓存的内容。
- 配置高速缓存的文件到期和垃圾收集。
- 配置自动高速缓存刷新和预装入。
- 配置与远程高速缓存访问（RCA）或因特网高速缓存协议（ICP）共享的高速缓存。
- 配置日志记录。

有关更改这些设置中每个设置的说明，在本章中给出或引用。

1. 启用高速缓存

要启用高速缓存，将 Caching 伪指令设置为开，或在**高速缓存配置** -> **高速缓存设置**配置表单中选取**启用代理高速缓存框**。如果您未指定高速缓存设备，那么高速缓存将存储在内存中。要创建磁盘高速缓存，遵循『2. 配置高速缓存存储』中的步骤执行。

2. 配置高速缓存存储

配置高速缓存存储的任务取决于您是使用内存高速缓存还是使用磁盘高速缓存。

要使用内存高速缓存，定制“高速缓存”设置以使它包含足够的内存来保存高速缓存的内容。请参阅第 70 页的『设置高速缓存』，以获得推荐的高速缓存大小。

要使用磁盘高速缓存，您必须执行下列操作：

1. 准备保存高速缓存的存储设备。

高速缓存需要特别格式化的设备。建议提供整个设备或磁盘分区给高速缓存。高速缓存的最小大小是 16392 KB。

要格式化高速缓存设备：

- a. 选择要保存高速缓存的设备。确保没有其它程序正在使用该存储空间，并且设备可以作为原始（或字符格式化的）设备访问。

- b. 使用 **htcformat** 命令格式化设备。语法按如下所示:

```
htcformat raw_device_path [-blocksize block_size]
          [-blocks number_of_blocks]
```

-blocksize 和 -blocks 自变量是可选的。缺省块大小是 8192 字节。如果未指定块的数目, 将以尽可能多的块来填充磁盘分区。

当指定设备路径时, 确保指定了原始的设备路径。

- 在 AIX 平台上, 定义为 /dev/lv02 的逻辑卷的裸设备路径为 /dev/rlv02
- 在 Linux 平台上, 您必须在运行 **htcformat** 前首先运行 **raw** 命令以连接裸设备路径和实 SCSI 驱动器 sdb1。

```
raw /dev/raw/raw1 dev/sdb1
```

- 在 HP-UX 和 Solaris 平台上, 定义为 /dev/dsk/c0t0d0s0 的分区的裸设备路径为 /dev/rdsk/c0t0d0s0
- 在 Windows 平台上, 定义为 e: 的设备的原设备路径为 \\.\e:

请参阅您文件系统的参考资料, 以获取有关访问裸设备的其它信息。

2. 使用 CacheDev 伪指令或高速缓存设置配置表单指定高速缓存设备。您可指定多个设备。

注意:

在 Windows 系统上, **htcformat** 命令不自动将高速缓存设备标记为不可写。

如果操作系统尝试写到高速缓存设备中, 则高速缓存数据可能会丢失。要避免这种情况, 您可使用 Windows 磁盘管理实用程序以在使用 **htcformat** 命令前准备磁盘。要准备磁盘, 使用磁盘实用程序来删除您需要的设备或分区, 然后重新创建它而不进行格式化。这将导致系统认为设备不可用于系统存储。

可选定制

设置高速缓存

根据下列原则, 设置 CacheMemory 伪指令 (或高速缓存设置配置表单的高速缓存字段) 中的值。此值的内存大小设置用于高速缓存基础结构支持, 包括高速缓存索引, 并且如果配置了内存高速缓存, 也用于存储高速缓存内容。

最小值

要获得磁盘高速缓存的最优性能, 建议高速缓存基础结构支持的最小高速缓存值为 64 MB, 包含高速缓存索引。当增加高速缓存大小时, 高速缓存索引将增加, 所以需要更多的高速缓存来存储索引。64 MB 的高速缓存值已足够大以提供高速缓存基础结构支持, 存储高速缓存索引的磁盘高速缓存大约需要 6.4 GB。要获得较大的磁盘高速缓存, 高速缓存应当为高速缓存大小的 1%。

对于内存高速缓存, 高速缓存值是为高速缓存基础结构支持和高速缓存本身留出的内存量。建议最小高速缓存值为 64 MB。

最大值

如果给内存高速缓存分配太多的物理内存，可能发生不理想的操作错误（如“内存不够”）或代理服务器发生故障。高速缓存的值限制是由于 32 位应用程序的限制。因为高速缓存代理是一个 32 位应用程序，它可使用最大 2 GB 内存。

高速缓存代理分配 CacheMemory 伪指令定义的内存并且把它作为高速缓存使用以存储对象。不管附加内存是内存高速缓存还是原始磁盘高速缓存，都必须分配它以用于高速缓存的数据结构，网络输入 / 输出和连接缓冲区，会话缓冲区和主进程和所有线程的内存。此外，来自一些客户机的请求可能会需要分配一个比缺省内存池块更大的内存池块。因此，如果 CacheMemory 伪指令设置为接近 2-GB GB 的标记，则可能高速缓存代理没有足够的内存来操作（特别是在高请求负载的情况下）。

建议 CacheMemory 伪指令的值小于或等于 1600 MB。设置大于 1600 MB 的值会妨碍高速缓存代理正常操作的内存，并且引起不利的副作用。通常这些副作用包括（但不限于）增加的 CPU 使用（可能达到 100% 使用）、内存不够错误和迟缓的性能。如果需要整体更大的高速缓存大小，使用高速缓存设备或用 RCA 或 ICP 实现共享的高速缓存配置。

保存或装入高速缓存内存到磁盘

您可以导入和导出高速缓存内容至转储文件或从转储文件导入和导出高速缓存内容。当高速缓存内存在重新启动期间丢失或者为多个代理部署相同的高速缓存时，这是有用的。

设置高速缓存过滤器

通过与 URL 请求的格式相匹配，过滤器可限制被高速缓存的内容。请参阅第 73 页的第 18 章，『控制高速缓存哪些内容』，以获得有关设置过滤器的详细信息。

配置用于查询结果和动态生成文件的高速缓存

可选地，您可配置代理服务器以高速缓存查询请求的结果。缺省情况下，包含问号 (?) 的 URL 不被高速缓存。请参阅第 73 页的『高速缓存查询响应』以获得详细信息。

另一个选项是高速缓存 IBM WebSphere Application Server 上执行 servlet 或 JSP 的结果。请参阅第 93 页的第 22 章，『动态地高速缓存生成的内容』以获得详细信息。

配置文件到期和垃圾收集

请参阅第 77 页的第 19 章，『维护高速缓存内容』，以获得有关配置高速缓存中文件到期的时间，以及如何除去过时文件的信息。

配置自动预装入

高速缓存可被配置成在每天最流行的文件被请求前，自动刷新这些文件。请参阅第 83 页的第 20 章，『为自动刷新和预装入配置高速缓存代理程序程序』，以获得相关信息。

配置高速缓存共享

在某些情况下，使用共享高速缓存能增加在高速缓存中找到请求文件的可能性。请参阅第 89 页的第 21 章，『使用共享高速缓存』，以获得相关信息。

配置日志记录

维护简明和详细日志对于管理高速缓存代理相当重要。第 129 页的第 6 部分,『监视高速缓存代理』包含有关配置和使用代理服务器日志的信息。

第 18 章 控制高速缓存哪些内容

高速缓存代理提供几种过滤方法来控制高速缓存哪些文件、文档和其它对象。这些包含下列功能部件：

- 基于 URL 的高速缓存过滤器
- 查询响应高速缓存
- 高速缓存本地提供服务的文件
- 部分基于 URL 的高速缓存
- 高速缓存基于部分请求 URL 的文件
- 高速缓存动态生成的文件 - 请参阅第 93 页的第 22 章，『动态地高速缓存生成的内容』

注：高速缓存配置 -> 高速缓存行为“配置和管理”表单包含一个标签为基于入网 URL 的高速缓存的选项。（相应的配置文件伪指令名为 CacheByIncomingURL。）此伪指令引用高速缓存的文件的文件名。选取此框以拥有基于入网 URL 的高速缓存文件的文件名；如果未选取此框，文件名将基于出网的 URL。

配置基于 URL 的高速缓存过滤器

代理服务器可配置为比较到 URL 模板的请求，以确定是否高速缓存文件。此功能部件是通过设置谁的文件总是高速缓存的请求模板，和谁的文件从不高速缓存的独立请求模板来进行设置的。可使用多个模板。

一个类似系统被用于启用查询响应高速缓存。请参阅『高速缓存查询响应』，以获得相关信息。

要通过编辑 ibmproxy.conf 文件设置 URL 高速缓存过滤器，请参阅第 174 页的『CacheOnly - 仅高速缓存其 URL 与模板匹配的那些文件』和第 213 页的『NoCaching - 指定不高速缓存其 URL 与模板匹配的那些文件』。

要设置“配置和管理”表单中的 URL 高速缓存过滤器，使用高速缓存配置 -> 高速缓存行为：基于 URL 过滤高速缓存字段。使用此部分来指定其文件总是被高速缓存的 URL，或指定其文件从不高速缓存的 URL。要指定两个列表，一个列表中的文件总被高速缓存而另一个列表中的文件从不被高速缓存，先创建一个列表，单击提交后再创建另一个列表。

高速缓存查询响应

从查询（包含一个问号的 URL）返回的响应可通过使用高速缓存过滤器来进行高速缓存。如果许多客户机发出相同的查询请求，此功能在逆向代理（代理）方案中是非常有用的。

查询高速缓存可通过编辑 ibmproxy.conf 配置文件中的 CacheQueries 伪指令来进行配置。CacheQueries 伪指令有下列选项：

- 总是 - 来自和模板匹配的主机的所有查询响应将被高速缓存（如果根据 HTTP 1.1 标准，它们是可被高速缓存的）。
- 公用 - 如果来自和模板匹配的主机的查询响应包含 “Cache-control: public” 头或强制的重新生效头，且根据 HTTP 1.1 标准它们是可被高速缓存的，则将被高速缓存。

其它关于这些选项的信息包含在第 174 页的『CacheQueries - 指定高速缓存对包含问号 (?) 的 URL 的响应』。

要在“配置和管理”表单中配置查询请求高速缓存，使用**高速缓存配置** -> **高速缓存行为**：基于 **URL 过滤高速缓存查询响应** 字段。要指定两个列表，先创建一个列表，然后在创建另一个列表之前单击**提交**。

查询响应高速缓存的附加要求

除配置查询高速缓存设置外，确保已正确配置下列设置以启用高速缓存查询响应。请参阅第 79 页的『配置高速缓存新鲜度』以获取关于使用“配置和管理”表单设置这些选项的信息。

- CacheTimeMargin - 此伪指令指定指定最小到期时间；到期时间小于此最小值的文件将不被高速缓存。因为有时查询响应有非常短的到期时间，因此将此伪指令设置为较低设置将允许高速缓存更多的查询响应。请参阅第 175 页的『CacheTimeMargin - 指定高速缓存文件的最小生命期』，或使用第 79 页的『配置高速缓存新鲜度』中描述的高速缓存到期设置表单。
- CacheDefaultExpiry - 此伪指令指定文件（这些文件没有一个明确的到期日期，或者没有可从中计算出到期日期的最后修改日期）的到期时间。从缺省 0 增加 HTTP 请求的设置将允许高速缓存更多的查询响应。但是，用这种方式更改设置也会增加从高速缓存服务失效内容的危险。请参阅第 169 页的『CacheDefaultExpiry - 指定文件的缺省到期时间』，或使用第 79 页的『配置高速缓存新鲜度』中描述的高速缓存到期设置表单。
- CacheLastModifiedFactor - 此伪指令用于计算文件（这些文件有最后修改的日期但是没有明确的到期日期）的到期日期。把 HTTP 文件的因子设置为较高的值将增加 HTTP 文件驻留在高速缓存中的时间（不需要重新生效）。用这种方式更改设置也会增加从高速缓存服务失效内容的危险。请参阅第 170 页的『CacheLastModifiedFactor - 指定值以确定到期日期』，或使用第 79 页的『配置高速缓存新鲜度』中描述的最后修改的因子表单。
- 可选地，设置 SignificantUrlTerminator 伪指令和 AggressiveCaching 伪指令。请参阅第 242 页的『SignificantURLTerminator - 为 URL 请求指定终止代码』和第 164 页的『AggressiveCaching - 为非高速缓存文件指定高速缓存』。

高速缓存本地提供服务的文件

因为通常高速缓存那些从代理服务器提供服务的文件的效率较差，所以缺省情况下不高速缓存来自服务器本地域中的文件。要高速缓存来自服务器本地域中的对象，请选择**高速缓存配置** -> **高速缓存行为**的“配置和管理”表单上的**高速缓存本地域文件框**。或者，将代理配置文件中的 CacheLocalDomain 伪指令设置为开。

根据部分 URL 高速缓存文件

可以仅根据入网 URL 的指定（关键）部分来高速缓存各项，而不是根据完整 URL。这在事务模型 Web 服务中或对于动态高速缓存而言是十分有用的，因为当入网请求的 URL 的关键部分相同时，通常会返回相同的响应给各种入网请求。

您不能使用“配置和管理”表单来指定基于部分 URL 的高速缓存。相反，可使用代理配置文件中的 `SignificantUrlTerminator` 伪指令来指定 URL 请求的终止代码。当此规范处理请求和确定是否高速缓存请求的文件时，它使高速缓存代理仅评估终止代码之前的字符。当定义了多个终止代码时，高速缓存代理根据在 `ibmproxy.conf` 文件中定义的终止代码的顺序来比较入网的 URL。请参阅第 242 页的『`SignificantURLTerminator` - 为 URL 请求指定终止代码』以获取更多信息。

相关的配置文件伪指令

要通过直接编辑代理配置文件来设置高速缓存过滤器，请参阅参考章节来获取下列伪指令：

- 第 213 页的『`NoCaching` - 指定不高速缓存其 URL 与模板匹配的那些文件』
- 第 174 页的『`CacheOnly` - 仅高速缓存其 URL 与模板匹配的那些文件』
- 第 174 页的『`CacheQueries` - 指定高速缓存对包含问号 (?) 的 URL 的响应』
- 第 171 页的『`CacheLocalDomain` - 指定是否要高速缓存本地域』
- 第 242 页的『`SignificantURLTerminator` - 为 URL 请求指定终止代码』

请参阅第 65 页的第 16 章，『代理服务器高速缓存概述』，以获取有关不能被高速缓存的文档的信息。

第 19 章 维护高速缓存内容

因为高速缓存涉及生成和保存服务文件的副本，所以高速缓存适当的功能需要某些例程维护。必须检查高速缓存文件的新鲜度，且当高速缓存文件不再与源服务器的文件一致时使之无效。此文件到期过程在『文件到期』中进行解释。同时，无效或不使用的文件必须从高速缓存中除去，以容纳新文件。此清除高速缓存的进程在第 80 页的『垃圾收集』中进行描述。

文件到期

保持高速缓存的对象与内容服务器上的原始对象一致，是一种已知的维护高速缓存新鲜度的方法。对于每个其高速缓存的文档或其它对象，高速缓存代理计算对象到期的时间。

对于 HTTP 页面，内容服务器生成的文档头包含到期信息。

因为 FTP 协议不包含等价的到期信息，所以高速缓存代理为 FTP 文件生成它自己的 Last-Modified: 头，这基于每个文件的 FTP 目录信息，并使用此信息计算到期时间。如果代理服务器无法从 FTP 服务器获取该文件的目录信息，则使用匹配 FTP URL 的缺省值。另外，因为 FTP 服务器没有标准的日期格式，所以高速缓存代理可能无法理解某些 FTP 服务器发送的日期和时间。这样，将使用代理服务器的缺省到期时间值。此过程允许代理以相似的方式管理 HTTP 页面和 FTP 文件的高速缓存。

内容服务器可以用几种方法中的一种来指定到期（以首选项的顺序）：

1. 内容服务器指定表示 Cache-control: s-maxage= *n* 的头。它告诉代理接收该对象后，它在 *n* 秒内是新鲜的。
2. 内容服务器指定表示 Cache-control: max-age= *n* 的头。它告诉代理接收该对象后，它在 *n* 秒内是新鲜的。
3. 内容服务器指定表示: Expires: *n* 的头。它告诉代理该对象是新鲜的，直到指定的 *n* 时间为止。
4. 内容服务器使用 Last-Modified: *n* 头来表示文档最后修改的时间。代理服务器计算自文档最后一次修改以来经过的时间，乘以在代理配置文件中设置的高速缓存最后修改的因子，然后假设该文档在此段时间内有效。例如，如果内容服务器表示最后一次修改该文档是在 1 周（7 天）前，而高速缓存最后修改的因子是 0.14，则代理服务将假设该文档的有效期为 1 天。请参阅第 79 页的『配置高速缓存新鲜度』，以获取设置高速缓存最后修改的因子的说明。
5. 如果内容服务器没有指定以上任何一条信息，则高速缓存代理将查找匹配当前 URL 的高速缓存缺省到期设置，并将它用作到期时间。请参阅第 79 页的『配置高速缓存新鲜度』，以获取设置高速缓存缺省到期值的说明。

根据刚刚的描述计算到期时间后，高速缓存代理检查是否存在适用此 URL 的最小保留值。如果存在的话，并且指定的时间大于计算出的到期时间，则最小保留值指定的时间用作该对象的到期时间。即使高速缓存代理计算出的文档到期时间为 0 分钟，它也为真。因此，要避免服务旧文件内容，谨慎使用最小保留设置。（要设置最小保留值，使

用 CacheMinHold 伪指令或使用**高速缓存配置 -> 高速缓存到期设置: URL 到期设置**。请参阅第 79 页的『配置高速缓存新鲜度』，以获得其它信息。）

根据时间极限设置中指定的时间来检查最后的到期时间值。如果到期时间大于时间极限值，则高速缓存该文档，否则不将它添加到高速缓存。（要设置时间极限值，使用 CacheTimeMargin 伪指令或参阅第 79 页的『配置高速缓存新鲜度』中的指示。）

如果在高速缓存中找到该文档，但它已到期，高速缓存代理发出称为*自那时以来如果已修改*请求的特殊请求到内容服务器。此请求致使内容服务器仅当自代理最后一次接收文档以来，已对它进行了修改时才发送该文档。如果未修改文档，则内容服务器发送一条表明它未经修改的消息，并且不重发页面。在这种情况下，代理服务高速缓存的文档。对于 FTP 文件，代理服务器模拟此“自那时以来如果已修改”进程。如果它确定文件在 FTP 服务器上未经更改，则它从高速缓存服务该文件。否则，它从 FTP 服务器上获取更新的版本。

有关高速缓存新鲜度的其它信息

- 几乎所有的静态 Web 文档（与动态生成的文档相反）都包含 Last-Modified 头。这是代理为文档计算到期时间的最通用的方法，也是高速缓存代理为 FTP 文件尝试的首选方法。如果此方法失败，则代理引用缺省到期值。
- 很少有文档使用“Cache-control: s-maxage, Cache-control: max-age”或“Expires:”头。
- 动态生成的页面，其通常时不可高速缓存的，可能包含表明 Expires: 0 或 Cache-control: no-cache 的头，这意味着文档立即到期。要获得有关高速缓存从 IBM WebSphere Application Server 动态生成的文件，请参阅第 93 页的第 22 章，『动态地高速缓存生成的内容』。
- 设置缺省到期值时，警惕要为使用 HTTP: 语法的 URL 设置任何不为 0 分钟的值。许多动态生成的页面不包含到期头，因此它服从缺省到期值。设置缺省到期为大于 0 分钟的值将允许代理高速缓存那些对象，但它可能意味着用户获取的是过时的内容（或 CGI 程序或 servlet 的意外结果）。
- 在下列情况下，无论高速缓存文档是否到期，代理服务器用每个请求的服务器重新生效文档：
 - 文档包含下列头中的一个：
 - Cache-control: s-maxage
 - Cache-control: must-revalidate
 - Cache-control: proxy-revalidate
 - 文档需要用户凭证，但是允许被服务器高速缓存。
 - 此文档包含“Cache-Control: no-cache”头，但总是高速缓存的（由于积极的高速缓存）。

有关 FTP 中的日期

因为 FTP 协议并不如 HTTP 协议那样严格地定义日期和时间，所以某些因素可以导致代理为 FTP 文件生成的 Last-Modified 头与实际文件日期略有不同。这些因素包含下列：

- 不象 HTTP 协议，FTP 协议不指定返回的日期必须是格林威治标准时间（GMT）。FTP 服务器返回的日期很可能是 FTP 服务器的当地时间。因为代理无法确定 FTP 服务器正在运行的时区，所以它用自己的时区来解释该时间。但对于 Windows FTP 服务器

而言这是一个例外，因为它以 GMT 返回日期。如果代理检测到 FTP 服务器正在 Windows 系统上运行，则假设该目录日期是 GMT。

- 有些 FTP 服务器指定仅以月日年的格式返回目录信息中的日期，并且不包含指定日期的实际小时或分钟信息。如果 FTP 服务器不返回文件的小时和分钟信息，代理将假设文件最后的修改是发生在 FTP 服务器返回日期的尽可能近的小时和分钟中。例如，如果 FTP 服务器返回文件的目录信息表明文件最后修改是在 1998 年 10 月 13 日，但不包含小时或分钟信息，代理将假设文件在 1998 年 10 月 13 日的下午 11:59:59 被修改。如果 FTP 服务器不是 Windows FTP 服务器，则代理将此日期从它自己的本地时区转换到相应的 GMT。

当 FTP 文件在高速缓存中已到期，则代理将为该 FTP 文件模拟 HTTP “自那时以来如果已修改”重新生效进程。通过为请求的文件重新发出 FTP LIST 命令来实现它，从 FTP 服务器返回的响应分析文件日期，并将此日期与最初检索文件时代理服务器为 Last-Modified 头生成的日期比较。如果文件日期未更改，则代理服务器标记高速缓存的 FTP 文件为重新生效，为该文件设置新的到期时间，并且从高速缓存服务该文件而不是从 FTP 服务器重新检索它。如果两个文件日期不匹配，则代理从 FTP 服务器重新检索该文件，并用新的文件日期来高速缓存新的副本。

不可能总能从 FTP 服务器获取该文件的目录信息。如果代理无法确定 FTP 文件的文件日期，它将不生成该文件的 Last-Modified 的头。然而，它使用为 CacheDefaultExpiry 伪指令指定的值，其匹配 URL 以确定高速缓存中保留文件的时间长度。当此时间段到期时，代理将总是再次从 FTP 服务器重新检索文件。如果高速缓存中的特定 FTP 文件看似频繁使用 CacheDefaultExpiry 伪指令，并且经常被检索（生成高容量的网络流量），则考虑为那些特定文件指定更粒化的 CacheDefaultExpiry 值。这样做可使它们在高速缓存中保持更长的时间。

要指定“配置和管理”表单中指定的高速缓存到期设置，使用**高速缓存配置 -> 高速缓存到期设置 -> 高速缓存文件的时间限制**表单。要获取有关设置高速缓存文件到期日期的更多详细信息，请参阅第 77 页的『文件到期』。

配置高速缓存新鲜度

要指定高速缓存文件的到期时间，在“配置和管理”表单中，选择**高速缓存配置 -> 高速缓存到期设置**。下列表单是有用的。

基于 URL 的到期

使用此表单可设置文件在高速缓存中保留的最小时间长度（基于它们的 URL）。您可为不同的 URL 请求模板指定不同的高速缓存行为。

要设置基于 URL 的文件到期，编辑代理配置文件，请参阅第 155 页的附录 B，『配置文件伪指令』中参考章节来获取下列伪指令：

- 第 173 页的『CacheMinHold - 指定文件的保留时间』

缺省到期设置

使用**高速缓存到期设置**表单指定用过的或未用过的文件的缺省到期设置。您可以为 HTTP、FTP 和 Gopher 文件设置不同的值，并可以为用过的或未用过的文件设置不同的值。

此表单还包含其它文件到期选项：

- **启用高速缓存文件到期检查。** 缺省情况下此复选框是选中的。通常，理想的是选择此选项，这样服务器不发送失效内容。
- **禁用从远程服务器检索文件。** 如果您不要服务器从远程服务器检索文件，选择此选项。
- **不要高速缓存将在...到期的文件。** 要阻止高速缓存在很短时间内到期的文件，用此选项指定时间周期。缺省情况下，不高速缓存 10 分钟内到期的文件。

要通过编辑代理配置文件来设置缺省到期设置，请参阅参考页面以获取下列伪指令：

- 第 169 页的『CacheDefaultExpiry - 指定文件的缺省到期时间』
- 第 170 页的『CacheExpiryCheck - 指定服务器是否返回到期文件』
- 第 175 页的『CacheTimeMargin - 指定高速缓存文件的最小生命期』
- 第 176 页的『CacheUnused - 指定未使用的高速缓存文件要保留多久』
- 第 174 页的『CacheNoConnect - 指定独立高速缓存方式』

最后修改的因子设置

使用**最后修改的因子**表单设置代理为高速缓存文件（它们的头没有到期日期）计算到期日期的值。您可以为匹配不同请求模板的文件设置不同的值。首个匹配模板用于计算到期日期。

要通过直接编辑代理配置文件来设置最后修改的因子，请参阅第 170 页的『CacheLastModifiedFactor - 指定值以确定到期日期』。

高速缓存时间限制

使用**高速缓存文件的时间限制**配置表单可设置文件可保留在高速缓存中的最大时间。时间限制基于请求模板设置，并且您可以指定当时间限制期满时，文件应该被废弃还是重新生效。这些设置可以用于维护其到期日期无效的文件，或有很长到期时间的文件。

要通过编辑代理配置文件来设置高速缓存文件的最大到期时间限制，请参阅下列：

- 第 172 页的『CacheMaxExpiry - 指定高速缓存文件的最大生命期』
- 第 169 页的『CacheClean - 指定要保留高速缓存文件多久』

垃圾收集

作为努力保持高速缓存最受欢迎的 URL 以及最小化系统资源的使用的一部分，高速缓存代理执行称为**垃圾收集**的清除进程，通过它将旧的或未用过的文件从高速缓存中除去，以腾出空间来存放更当前的文件。

垃圾收集进程测试高速缓存目录中的文件，并尝试除去到期文件以减少高速缓存的大小并为新文件腾出空间。垃圾收集是自动完成的，但需配置一些设置以定制您需要的进程。

配置垃圾收集

要配置垃圾收集，在“配置和管理”表单中，选择**高速缓存配置 -> 垃圾收集设置**。使用此表单设置高水位标志和低水位标志，这确定何时启动和停止垃圾收集。当高速缓存中使用的空间大小到达或超出高水位标志的百分率设置时，垃圾收集开始。垃圾收集继续直到高速缓存中用过的空间等于或低于低水位标志为止。

您可在两个垃圾收集算法中选择。**响应时间**算法通过择优除去高速缓存中的大文件，优化响应用户所必需的时间。通过择优地从高速缓存中除去更小的文件，**带宽**算法优化网络带宽的使用。选择一种或将两种混合起来。

要通过编辑代理配置文件来配置垃圾收集，请参阅参考部分以获得下列伪指令：

- 第 194 页的『Gc - 指定垃圾收集』
- 第 195 页的『GcHighWater - 指定垃圾收集何时开始』
- 第 195 页的『GcLowWater - 指定垃圾收集何时结束』
- 第 168 页的『CacheAlgorithm - 指定高速缓存算法』

第 20 章 为自动刷新和预装入配置高速缓存代理程序程序

多数高速缓存代理服务器仅在用户请求文件之后才高速缓存它。高速缓存代理具有提供自动高速缓存预装入的高速缓存代理程序。您可指定此高速缓存代理程序可以自动检索指定的 URL 或一些最流行的 URL，或两者全部，并在它们被请求之前先将它们放到高速缓存中。

在某些情况下，在预装入高速缓存之前，您将需要设置代理服务器的主机名并标识高速缓存访问日志。要配置高速缓存代理程序，在“配置和管理”表单中，选择**高速缓存配置**并使用**高速缓存预装入**和**高速缓存刷新**表单。注意，代表查询结果的文件（即，仅当启用查询高速缓存时，其 URL 包含问号（?）的文件才高速缓存）。

自动高速缓存刷新和预装入提供下列优点：

- 用户请求页面之前，高速缓存应用于指定的 URL。
- 服务器忙于应付用户的活动之前，植入高速缓存。
- 从高速缓存中将当前文件提供给用户要比首次请求时取得它们要快。

缺点包含下列：

- 代理服务器忙于高速缓存页面，甚至在较少用户活动期间。
- 您必须练习一些控制自动装入内容的技巧。装入高级别页面的链接文件（如 Web 索引和搜索站点），这样可生成数目庞大的页面请求。

为了达到最高效率，设置高速缓存代理程序为当服务器活动较少的时候以及服务器忙于应付客户机请求之前运行。然后，文件将在高速缓存中就绪，以在用户首次请求它时提供快速服务。缺省情况下，高速缓存代理程序在当地时间的每天凌晨 3 点启动。

注：当高速缓存代理程序运行以刷新高速缓存时，您必须取消注释 `ibmproxy.conf` 文件中的“`Proxy http:*`”行。否则，导致错误日志中的“`403 Forbidden By Rule Error`”，并且不完成刷新高速缓存。

设置服务器主机名

在 Linux 和 UNIX 平台上，指定其正在预装入或刷新高速缓存的代理服务器的主机名。在 Windows 平台上，仅当代理服务器在非本地机器上被刷新时，才指定主机名（注意，基于其最频繁访问的文件来刷新远程服务器的高速缓存是不可能的，因为本地高速缓存代理程序没有访问远程服务器的高速缓存访问日志的权限）。

要在“配置和管理”表单中设置代理服务器的主机名，选择**高速缓存配置** -> **高速缓存刷新**：标识高速缓存目标服务器。

用特定文件预装入高速缓存

要用存储在特定 URL 的内容预装入高速缓存，在“配置和管理”表单中使用**高速缓存配置 -> 高速缓存预装入**。在此表单中，您可为要装入的高速缓存代理程序指定 URL。当高速缓存代理程序启动时，代理检索那些页面（无论它们是否已在高速缓存中）（这些 URL 通过 LoadURL 伪指令在代理配置文件中指定）。此表单也可用于定义其内容从不高速缓存的 URL。对于此类型的高速缓存预装入，并不需要访问高速缓存访问日志。

使用**高速缓存预装入**表单配置下列选项：

- **每天刷新高速缓存** - 如果您希望高速缓存代理程序每天晚上刷新高速缓存，则选取此框。如果您不希望启动高速缓存代理程序，确保不选取此框。
- **高速缓存刷新时间** - 如果您希望高速缓存代理程序在不是当地时间的凌晨 3:00 运行，指定需要它启动的时间。
- **高速缓存内容** - 在 **URL 或 IP 地址** 字段，指定要装入的 URL。要排除预装入的 URL，指定 URL 并单击**高速缓存状态**框中的**忽略**。

用频繁高速缓存的文件预装入高速缓存

要自动预装入最频繁访问的页面，使用**配置高速缓存 -> 高速缓存刷新**表单。此功能需要代理服务器的高速缓存访问日志。（日志位置和名称的更改，请参阅第 129 页的第 6 部分，『**监视高速缓存代理**』以获得相关信息。）从高速缓存访问日志自动确定最流行的 URL。管理员还可以指定在高速缓存中要预装入的流行页面数。（此数目在代理配置文件中由 LoadTopCached 伪指令指定。）

使用**高速缓存刷新**表单配置下列选项：

- **每天刷新高速缓存** - 如果您希望高速缓存代理程序每天晚上刷新高速缓存，则选取此框。如果您不希望启动高速缓存代理程序程序，确保此框没有选中。
- **高速缓存刷新时间** - 如果您希望高速缓存代理程序在不是凌晨 3:00 运行，则指定需要它启动的确切时间。
- **标识高速缓存目标服务器** - 如果您要刷新服务器而不是本地机器，则使用此选项。（注意，您不能根据对特定文件的访问频率来刷新远程服务器。）
- **高速缓存最流行 URL** - 从前一晚的高速缓存访问日志指定要高速缓存的 URL 数。
- **装入链接页面** - 使用此设置配置挖掘（请参阅下列部分以获取挖掘的详细信息）。设置挖掘的级别数，以及是挖掘所有页面（**always**）还是不挖掘页面（**never**），或仅管理员指定的页面（**admin**），也可以是仅流行页面（**topn**）。也可以指定是否跨越主机挖掘，是否在请求间挖掘以及是否高速缓存直接插入图像。
- **线程数** - 设置高速缓存刷新使用的最大线程数。
- **最大工作队列深度** - 设置请求 URL 的最大队列。
- **最大请求 URL** - 设置装入的最大页面数。在挖掘页面检索开始之前将检查此数目。
- **最长时间** - 设置运行高速缓存代理程序的最长时间。如果此时间设置为 0 小时 0 分钟，则运行高速缓存代理程序直到完成为止。

挖掘

挖掘是自动高速缓存刷新功能的可选部分。多数 Web 页面具有链接到其它具有相关信息页面的链接，并且用户经常按照路径从一个页面链接到另一个页面，以及从一个站点链接到另一个站点。挖掘是一种高速缓存这些逻辑信息路径的方法。在挖掘中，高速缓存代理程序跟随正在装入的页面上指定级别的超文本（HTML）链接，并且还高速缓存所有那些链接的页面。链接的页面可能驻留在与源页面相同的主机上，或其它主机上。在图 1 中给出了图解。

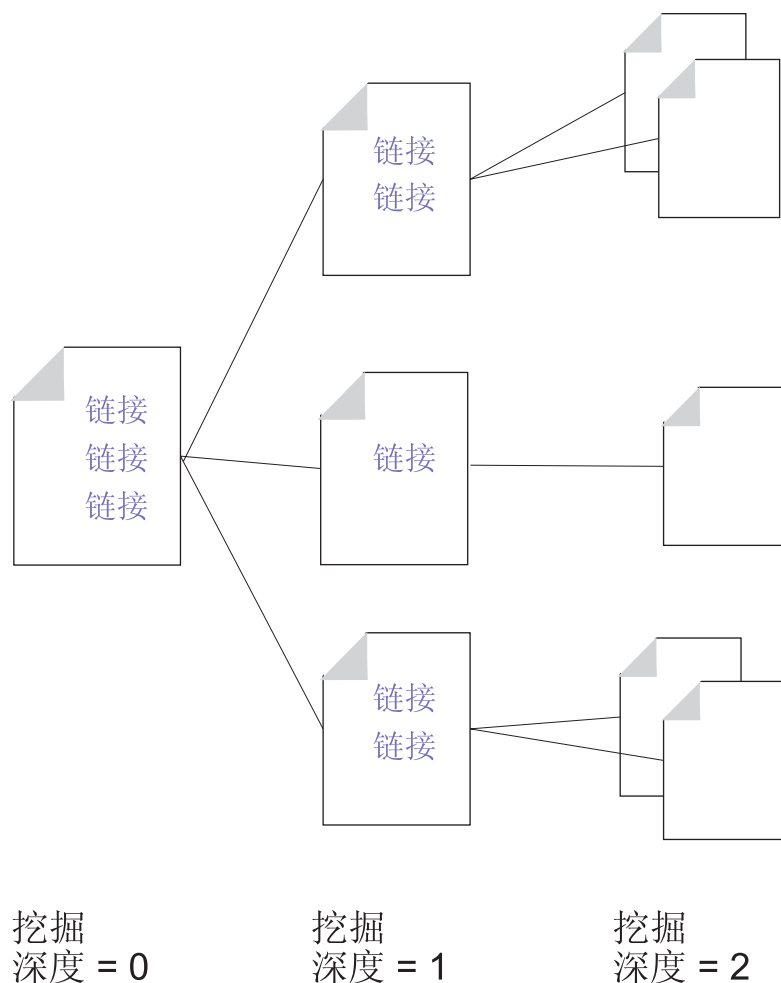


图 1. 挖掘

要控制挖掘进程，管理员指定给高速缓存代理程序其可装入 URL 的最大数目（缺省设置是 2000），可运行的最长时间（缺省设置是 2 小时）以及可使用的最大线程数（缺省设置是 4）。管理员还可以配置其它控制。缺省情况下，为两个级别的层次结构启用挖掘，并且不允许跨越主机。另外，在请求间插入延迟。要更改这些设置，请参阅第 86 页的『相关代理配置文件伪指令』。

高速缓存代理程序以此顺序装入然后刷新高速缓存：

1. 装入管理员指定的特定页面。
2. 装入从高速缓存访问日志取得的流行（频繁访问）页面。

3. 如果在这点上还未到达最大页面数，装入挖掘的其它页面。

注意，直到高速缓存代理程序开始跨越链接挖掘为止，它才检查是否已到达最大页面数。如果最大页面值（在代理配置文件中称为 MaxURLs）小于步骤 1 和步骤 2 中检索的页面数，则不检索链接的页面。

下列示例显示高速缓存代理程序如何处理高速缓存刷新优先级和挖掘，相对于指定的最大 URL 数目（假设为所有这些示例配置了挖掘）。

配置文件设置	结果
LoadURL http://www.getthis.com/main.html LoadURL http://www.getmetoo.com/welcome.htm LoadTopCached 30 MaxURLs 50	如果高速缓存访问日志有多于 30 个的唯一 URL，高速缓存代理程序将检索 main.html、welcome.htm 和基于高速缓存访问日志的前 30 个请求的 URL。因为它没有到达 MaxURLs 值，所以还可从高速缓存的页面检索和装入 18 个链接的 URL。
LoadURL http://ww.joesmith.edu/favorites.html LoadURL http://www.janesmith.edu/dislikes.html LoadTopCached 30 MaxURLs 25	如果高速缓存访问日志有多于 30 个的唯一 URL，高速缓存代理程序将检索 favorites.html、dislikes.html 和高速缓存访问日志的前 30 个请求的 URL。不检索其它文件，因为已超出 MaxURLs 中的值。
LoadURL http://www.hello.com/hi.htm LoadURL http://www.ballyhoo.com/index.html LoadTopCached 20 MaxURLs 25	如果高速缓存访问日志有多于 20 个的唯一 URL，高速缓存代理程序检索 hi.htm、index.html，以及高速缓存访问日志的前 20 个请求的 URL，并且还可从早先的页面中检索 3 个链接的 URL。不检索其它文件，因为已超出 MaxURLs 中的值。

相关代理配置文件伪指令

高速缓存代理程序也可以通过直接编辑代理配置文件中的相应伪指令来配置。要获取与高速缓存代理程序有关的代理配置文件伪指令，请参阅第 155 页的附录 B，『配置文件伪指令』中的下列参考页面：

- 第 167 页的『AutoCacheRefresh - 指定是否要使用高速缓存刷新』
- 第 167 页的『CacheAccessLog - 指定高速缓存访问日志文件的路径』
- 第 175 页的『CacheRefreshTime - 指定何时启动高速缓存代理程序』
- 第 182 页的『DelayPeriod - 指定请求之间的暂停』
- 第 182 页的『DelveAcrossHosts - 指定跨域高速缓存』
- 第 182 页的『DelveDepth - 指定高速缓存时要逐层访问的链接层次』
- 第 182 页的『DelveInto - 指定高速缓存代理程序是否要逐层访问链接』
- 第 199 页的『IgnoreURL - 指定不刷新的 URL』
- 第 205 页的『LoadInlineImages - 控制嵌入图像的刷新』
- 第 205 页的『LoadTopCached - 指定要刷新的受欢迎页面数』
- 第 205 页的『LoadURL - 指定要刷新的 URL』
- 第 211 页的『MaxUrls - 指定要刷新的最大 URL 数』

手工启动高速缓存代理程序

启用自动高速缓存刷新，高速缓存代理程序在指定的时间自动运行刷新操作。然而，您还可随时从命令行运行高速缓存代理程序。

可执行文件按如下所示：

- 在 Linux 和 UNIX 平台上：`usr/sbin/cacheagt`
- 在 Windows 平台上：`server_root\bin\cacheagt.exe`

其中 `server_root` 是安装高速缓存代理的驱动器和目录（例如，`C:\Program Files\IBM\edge\cp`）。

在 Linux 和 UNIX 平台上，您可以使用 **cron** 守护程序随时自动运行高速缓存代理程序。受控于 **cron** 的作业通过向系统 `crontab` 文件添加行来指定。Linux 和 UNIX 上的命令文件的示例条目是：

```
45 16 * * * /usr/sbin/cacheagt
```

此命令示例将在当地时间的每天下午 4:45 启动高速缓存代理程序。如果期望的话，您可使用多个条目来多次运行高速缓存代理程序。要获取更多的信息，请参阅您的有关 **cron** 守护程序的操作系统文档。

当使用 **cron** 守护程序运行高速缓存代理程序时，记得关闭自动刷新选项，您可使用 **高速缓存配置** -> **高速缓存刷新配置** 表单或编辑代理配置文件来关闭此选项。否则，高速缓存代理程序每天都将多次运行。

第 21 章 使用共享高速缓存

Web 上呈现点的流量比单个服务器可处理的流量多是很普遍的。一种解决方案是简单地添加更多服务器。然而，当使用多个高速缓存代理服务器时，一个高速缓存的内容经常与其它高速缓存中的内容重叠。除了存储器中不必要的冗余外，还不能达到最大节省带宽，因为当到代理服务器的请求在其高速缓存中没有那个文件时，需从源服务器重新获取高速缓存文件。尽管重复的高速缓存可以通过使用代理服务器的分层链来最小化，此方案仍会导致通过给定的服务器的附加流量，并且每个链中的附加链接将增加等待时间。

高速缓存共享通过允许每个高速缓存与其它高速缓存共享其内容来解决这些问题。节省带宽的结果是因为下列事实：

- 对象不是多次取的。
- 大的、组合的逻辑高速缓存产生较高的命中率。

为将多个高速缓存作为一个逻辑高速缓存使用提供的两种方法：

- 远程高速缓存访问（RCA）是高速缓存代理的一种功能部件，它定义了成员高速缓存的数组。根据内部逻辑，文件精确地存储在这些高速缓存的其中一个中。
- 提供高速缓存代理插件以允许代理服务器使用因特网高速缓存协议（ICP）。如果您要共享高速缓存代理机器和非高速缓存代理高速缓存间的数据时，您可以使用 ICP 插件，而不使用 RCA。

可同时使用 RCA 和 ICP。

远程高速缓存访问

在计划 RCA 时，考虑下列建议：

- 参与的代理服务器需要彼此靠近，并与高带宽链接连接（例如，FDDI、SP2 总线）。
- RCA 数组中的成员需要是长期的，这样配置才会尽可能的稳定。
- 代理服务器应该具有相似的容量（例如，CPU、内存大小、高速缓存大小）。
- 必须很少发生网络崩溃。
- 所有数组中的成员数都应该小于 100。
- 所有数组中的成员必须使用相同版本的高速缓存代理软件。

注：如果 RCA 数组中的代理使用不同的 Linux 操作系统（例如，SuSE 和 Red Hat），则确保所有其对等代理中的“nobody”用户具有相同的 UID。检查每台计算机上的 /etc/ 目录中的密码和组文件条目，并指定相同的 UID 给“nobody”。

如果违反这些条件中的任何一条，或不同组织所管理的是作为数组的成员的不同服务器，则不适合远程高速缓存访问。

配置远程高速缓存访问

要配置远程高速缓存访问，在“配置和管理”表单中，选择**高速缓存配置 -> 远程高速缓存访问**。此表单中的字段定义将共享一个逻辑高速缓存的命名数组。为每个数组成员输入所需的信息。

要通过编辑代理配置文件来配置远程高速缓存访问，请参阅第 155 页的附录 B，『配置文件伪指令』中的参考章节，以获取下列伪指令：

- 第 165 页的『ArrayName - 命名远程高速缓存数组』
- 第 211 页的『Member - 指定一个数组成员』

配置因特网高速缓存协议插件

因特网高速缓存协议插件使高速缓存代理能在 HTML 页面和其它可高速缓存的资源中查询符合 ICP 的高速缓存。当代理服务器接收到 HTTP 请求时，它将搜索自己的高速缓存以获取此资源。如果在本地高速缓存中未找到此资源，而 ICP 插件已启用，则代理服务器将在 ICP 查询信息包中封装此 URL 请求，然后把此信息包发送到所有识别的 ICP 对等高速缓存。如果对等高速缓存响应有资源，代理服务器将从对等的高速缓存中检索资源。如果有两个或多个对等肯定地响应，则处理第一个响应。如果无对等响应为命中，则源服务器继续根据其工作流处理此请求。例如，代理服务器可调用另一个插件，继续远程高速缓存访问例程（当 RCA 启用时）或检索它自己请求的资源。

配置 ICP 插件

ICP 插件是通过编辑代理配置文件 `ibmproxy.conf` 来激活和配置的。ServerInit 伪指令、PreExit 伪指令或两者都必须添加到配置文件的 API 伪指令部分，这样才能使用 ICP 插件。根据高速缓存代理在 ICP 系统中的角色，决定使用哪些伪指令：

- 对于作为 ICP 服务器功能的高速缓存代理，使用 ServerInit 伪指令来调用 `icpServer` 模块。
- 对于作为 ICP 客户机功能的高速缓存代理，使用 PreExit 伪指令来调用 `icpClient` 模块。
- 对于作为 ICP 客户机和 ICP 服务器功能的高速缓存代理，使用以上两种伪指令。
- 使用伪指令 `icpAddress`、`icpMaxThreads`、`icpPeer`、`icpPort` 和 `icpTimeout` 来配置插件使用的设置。

要创建这些伪指令，手工编辑 `ibmproxy.conf` 文件，或当代理服务器已在运行时，连接到“配置和管理”表单**服务器配置 -> 请求处理 -> API 请求处理**。

注意，prototype 伪指令（以注释形式）已添加到 `ibmproxy.conf` 文件的 API 部分。这些 API 伪指令是按有目的的顺序进行排列的。当添加 API 伪指令以启用新功能和插件模块时，请按配置文件原型部分中显示的顺序排列这些伪指令。或者，取消注释并编辑（如果需要）API 伪指令，以包含对每个所需功能或插件的支持。

ServerInit 和 PreExit 伪指令都具有两个自变量：(1) 共享库的全限定路径 (2) 函数调用。这两个自变量由冒号 (:) 定界。第一个自变量是系统特定的，并取决于插件组件的安装位置。第二个自变量是硬编码到共享库中的，且必须按显示精确输入。

在代理配置文件中，每个伪指令必须出现在一行中。

```
ServerInit path_of_shared_library:icpServer
```


Linux 和 UNIX 示例:

```
ServerInit /opt/ibm/edge/cp/internet/lib/plugins/icp/libicp_plugin.so:icpServer
```

Windows 示例:

```
ServerInit C:\Program Files\IBM\edge\cp\Bin\plugins\icp\icppugin.dll:icpServer  
PreExit path_of_shared_library:icpClient
```

Linux 和 UNIX 示例:

```
PreExit /opt/ibm/edge/cp/internet/lib/plugins/icp/libicp_plugin.so:icpClient
```

Windows 示例:

```
PreExit C:\Program Files\IBM\edge\cp\Bin\plugins\icp\icppugin.dll:icpClient
```

要配置插件的设置，添加或修改在代理配置文件中提供的 ICP* 伪指令。要获得其它信息，请参阅下列伪指令的描述。

- 第 197 页的『ICP_Address - 指定 ICP 查询的 IP 地址』
- 第 198 页的『ICP_MaxThreads - 指定 ICP 查询的最大线程』
- 第 198 页的『Occupier - 指定 Occupier 群集的成员』
- 第 198 页的『ICP_Port - 指定 ICP 查询的端口号』
- 第 199 页的『ICP_Timeout - 指定 ICP 查询的最大等待时间』
- 第 221 页的『PreExit - 定制 PreExit 步骤』
- 第 240 页的『ServerInit - 定制服务器初始化步骤』

第 22 章 动态地高速缓存生成的内容

动态高速缓存功能使高速缓存代理能高速缓存动态生成的内容，其格式为来自 IBM WebSphere Application Server 生成的 JavaServer Pages (JSP) 和 servlet 的响应。高速缓存代理适配器模块在应用程序服务器上使用以修改响应，这样它们能在代理服务器上高速缓存（除在应用程序服务器的动态高速缓存中高速缓存外）。使用此功能，动态生成的内容可在网络的边缘高速缓存，当多个客户机请求相同内容时，使内容主机可不再向应用程序服务器作出重复请求。

注： 动态高速缓存功能部件不会启用代理服务器来高速缓存来自 URL 查询的结果。要高速缓存查询结果，配置高速缓存过滤器，这在第 73 页的第 18 章，『控制高速缓存哪些内容』中和第 174 页的『CacheQueries - 指定高速缓存对包含问号 (?) 的 URL 的响应』的伪指令查考文档中有描述。可高速缓存来自源服务器（此服务器并非 IBM WebSphere Application Server）的查询结果。

有时必需启用查询高速缓存以使用动态高速缓存功能部件，例如，当 servlet 使用查询格式的 URL 时。代理服务器将任何包含问号 (?) 的 URL 认作是查询。

对动态生成内容进行高速缓存的益处在于：

- 减少内容主机上的负载。
- 减少应用程序服务器上的负载。
- 加快所请求资源传递给最终用户的速度。
- 减少服务器之间的带宽使用。
- 改进创建或服务动态地生成内容的 Web 站点的可伸缩性。

应用程序服务器为代理高速缓存仅导出完全组成的公用页面。代理不对专用页面进行高速缓存。例如，从公用站点动态生成并列出现当前天气预报的页面可由 IBM WebSphere Application Server 导出，由高速缓存代理高速缓存。然而，列出用户购物手推车中内容的动态生成的页面则不能由代理服务器高速缓存。而且，为了使动态生成的页面能被高速缓存，此页面的所有子组件也必须是可高速缓存的。

高速缓存动态文件和常规文件到期的方式不同；它们可由生成它们的应用程序服务器来使之成为无效。

动态高速缓存的条目在下列情况下是无效的：

- 动态高速缓存垃圾收集器在高速缓存堵塞时除去条目。
- servlet 条目 (servletcache.xml) 或代理的 ExternalCacheManager 伪指令中超时设置到期。
- 一个外部代理程序或应用程序调用动态高速缓存 API 以使高速缓存条目无效。

无效动态高速缓存条目由生成一条高速缓存代理动态高速缓存插件的特定实例的无效消息造成。高速缓存代理接收无效消息，同时登记到 /WES_External_Adapter 资源定位器。然后，高速缓存代理从高速缓存中清除无效的条目。

动态高速缓存需要下列配置步骤。

- IBM WebSphere Application Server 配置：

- 配置每个应用程序服务器以完成本地动态高速缓存。
- 配置每个应用程序服务器以使用外部高速缓存适配器。
- 指定哪些外部高速缓存可用于每个可高速缓存的 servlet 和 JSP 文件。
- 高速缓存代理配置:
 - 启用高速缓存代理以使用动态高速缓存插件。
 - 指定源，从其中动态内容将被高速缓存。

为代理高速缓存配置 IBM WebSphere Application Server

在应用程序服务器配置动态高速缓存

按照 IBM WebSphere Application Server 文档的关于配置应用程序服务器的说明来使用它的本地动态高速缓存（也称为动态分段高速缓存）。动态分段高速缓存在 Application Server 高速缓存代理和外部高速缓存交互。

配置应用程序服务器适配器

使用称为外部高速缓存适配器（与 Application Server 一起安装）的软件模块，IBM WebSphere Application Server 和高速缓存代理进行通信。

注：请参阅 IBM WebSphere Application Server 支持 Web 站点，以获得 TechNote 有关配置动态高速缓存的信息。

配置动态高速缓存的高速缓存代理

要使代理服务器能高速缓存动态生成的内容（来自 servlet 和 JSP 的结果），您必须在代理配置文件 `ibmproxy.conf` 中做两个更改。第一个更改启用动态高速缓存插件模块，第二个更改配置它以识别可高速缓存动态内容的源。

设置 Service 伪指令启用动态高速缓存插件

“服务”步骤的 API 伪指令用于启用动态高速缓存插件。要创建此伪指令，手工编辑 `ibmproxy.conf` 文件，或当代理服务器已运行时，使用“配置和管理”表单选择**服务器配置** -> **请求处理** -> **API 请求处理**。伪指令内容显示在本节后面出现的示例中。

启用动态高速缓存的原型 Service 伪指令作为 `ibmproxy.conf` 文件的 API 部分的注释存在。它有标题 JSP 插件。注意，原型 API 伪指令以有目的的顺序排列。当添加 API 伪指令以启用新功能和插件模块时，请按配置文件原型部分中显示的顺序排列这些伪指令。可选地，您可从原型 API 伪指令除去注释字符，如有必要编辑它们以包括每个期望的功能或插件的支持。

根据下列示例设置 Service 伪指令。（注意，每个伪指令必须出现在代理配置文件的一行上；这些示例有时为了可读性而有换行符。）

- 对于 AIX:

```
Service /WES_External_Adapter /opt/ibm/edge/cp/lib/plugins/  
dynacache/libdyna_plugin.o:exec_dynacmd
```

- 对于 Solaris:

```
Service /WES_External_Adapter /opt/ibm/edge/cp/lib/plugins/  
dynacache/libdyna_plugin.so:exec_dynacmd
```

- 对于 Linux:
Service /WES_External_Adapter /usr/lib/libdyna_plugin.so:exec_dynacmd
- 对于 Windows:
Service /WES_External_Adapter C:\Program Files\IBM\edge\cp\bin\plugins\
dynacache\dyna_plugin.dll:exec_dynacmd

如果高速缓存代理软件安装在非缺省目录中，则用安装路径替换这些示例中的路径。

设置 ExternalCacheManager 伪指令以指定文件源

还必须配置每个高速缓存代理以识别动态生成的文件的源。为将在此代理服务器上高速缓存动态生成的内容的每个应用程序服务器，将 ExternalCacheManager 伪指令添加到 ibmproxy.conf 文件。此伪指令指定 WebSphere Application Server 将在代理处高速缓存结果，为服务器的内容设置最长到期时间。更多的详细信息出现在第 191 页的『ExternalCacheManager - 对 IBM WebSphere Application Server 中的动态高速缓存配置高速缓存代理』。

ExternalCacheManager 伪指令中使用的服务器标识必须匹配应用程序服务器的 dynacache.xml 文件的外部高速缓存组节中使用的组标识。

对于上述示例，添加下列条目到每个代理的 ibmproxy.conf 文件。

```
ExternalCacheManager IBM-edge-cp-XYZ-1 20 seconds
```

高速缓存代理仅高速缓存来自组标识匹配 ibmproxy.conf 文件的 ExternalCacheManager 条目的 IBM WebSphere Application Server 的内容。

第 23 章 调整代理服务器高速缓存

当启用高速缓存时，高速缓存存储设备的速度对于高速缓存代理性能是十分关键的。本节给出了关于选择一种高速缓存存储类型和配置您的高速缓存存储设备以获得最佳性能的建议。

选择高速缓存存储介质

高速缓存代理可使用两种不同类型的高速缓存存储介质：

- 内存
- 原始磁盘分区

内存高速缓存提供了最快的文件检索，但是内存高速缓存的大小受到代理服务器上可用内存数量的限制。磁盘高速缓存（由一个或多个原始磁盘分区组成）低于内存高速缓存，但是在大多数情况下允许更大的高速缓存大小。

优化磁盘高速缓存性能

用于磁盘高速缓存的设备分区需专用于高速缓存；即，不要使用这些物理磁盘包含任何其它文件系统，并且不要将它们用于除存储代理高速缓存之外的任何目的。另外，不要在任何用于代理高速缓存的磁盘上使用数据压缩，因为它会减少性能。

每个高速缓存存储设备（无论是磁盘还是文件）将导致代理服务器上的内存开销。一般而言，使用整个物理磁盘作为单个高速缓存将产生最佳性能。使用 RAID 或其它机制将多个物理磁盘组合到单个逻辑磁盘可能达不到预期目的。如果您想要使用多个磁盘，则通过使用**高速缓存设置**配置表单或编辑代理配置文件中的 `CacheDev` 伪指令来将它们指定为多个高速缓存设备。此方法允许代理服务器控制读和写到多个磁盘的并行性，并且不依赖操作系统或磁盘子系统的性能。

高速缓存垃圾收集

代理服务器的高速缓存垃圾收集将废弃高速缓存中的到期文件，并释放空间以高速缓存用于新请求的文件。当高速缓存中已使用的空间量达到称为高水位标志的管理员指定的限制时将自动触发垃圾收集，并继续直至已使用的空间量达到低水位标志。

因为垃圾收集例程使用最小 CPU 资源并且不会影响未到期的高速缓存资料的可用性，所以不需要配置垃圾收集以在特定时间运行。

要改进垃圾收集的性能，您可以设置高水位标志和低水位标志。您还可以配置用于垃圾收集的算法类型。请参阅第 80 页的『垃圾收集』以获得关于修改垃圾收集的更多信息。

平台特定的最优化

下列附加建议用于在每个平台上优化高速缓存性能。

AIX

最好使用所有可用的物理分区 (PP) 在一个磁盘上创建单个逻辑卷。例如, 给定一个 9-GB 的磁盘, 创建一个称为 `cpcache1` 的 9-GB 逻辑卷。使用其原始逻辑卷 `/dev/rcpcache1` 格式化它并将其指定为代理高速缓存设备。

HP-UX 和 Solaris

在您的高速缓存设备上, 创建一个使用磁盘的整个大小的单个分区 (或片)。例如, 在一个 9-GB 的磁盘上, 创建一个称为 `c1t3d0s0` 的 9-GB 分区。使用其裸设备 `/dev/rdisk/c1t3d0s0` 格式化它并将其指定为代理高速缓存设备。

Windows

使用磁盘的整个大小, 创建单个分区。例如, 在一个 9-GB 的磁盘上, 创建一个称为 `i:` 的 9-GB 分区。使用其裸设备 `\\.\i:` 格式化它并将其指定为代理高速缓存设备。

关于配置代理服务器的高速缓存以及格式化并指定高速缓存设备的信息包括在第 63 页的第 4 部分, 『配置代理服务器高速缓存』中。

第 5 部分 配置高速缓存代理安全性

这一部分提供有关基本安全性、高速缓存代理中使用 SSL、启用加密硬件以及使用 IBM Tivoli® Access Manager（以前是 Tivoli Policy Director）插件和 PAC-LDAP 授权模块方面的信息。

这一部分包含下列章节：

第 101 页的第 24 章，『关于代理服务器安全性』

第 103 页的第 25 章，『服务器保护设置』

第 107 页的第 26 章，『安全套接字层（SSL）』

第 117 页的第 27 章，『启用加密硬件的支持』

第 119 页的第 28 章，『使用 Tivoli Access Manager 插件』

第 121 页的第 29 章，『使用 PAC-LDAP 授权模块』

第 24 章 关于代理服务器安全性

任何可从因特网访问的服务器都会因吸引不必要的关注而给运行它的系统带来风险。未授权的人可能尝试猜测密码、更新文件、执行文件或读机密数据。万维网吸引人的一部分在于它的开放性。但是，Web 对于积极使用和滥用是同时开放的。

下面部分讨论如何控制访问您的高速缓存代理服务器上的文件的访问者。

高速缓存代理支持安全套接字层（SSL）连接，此处建立客户机浏览器和目标服务器（内容服务器或代理服务器）之间涉及加密和解密的安全传输。

当高速缓存代理配置为代理时，它能建立与客户机或内容服务器或两者的安全连接。要启用 SSL 连接，在“配置和管理”表单中选择**代理配置 -> SSL 设置**。在此表单上，选择**启用 SSL** 复选框并指定密钥环数据库和密钥环数据库密码文件。

您可采取一些基本防范措施来保护您的系统：

- 在与您的本地或内部网络分隔的网络中放置一个服务器以用于公共访问。
- 禁用允许远程用户访问服务器的内部进程的实用程序。特别地，考虑在运行服务器的系统上禁用 **telnet**、**TN3270**、**rlogin** 和 **finger** 客户机。
- 使用包过滤和防火墙。

包过滤允许您定义数据的来源和去处。您可配置您的系统以拒绝某些资源 / 目标组合。

防火墙将内部网络与可公共访问的网络（如，因特网）分隔开。防火墙可以是一组计算机或单个计算机，担当两个方向的网关，调节和跟踪通过它传递的流量。IBM Firewall 是防火墙软件的示例。

- 控制 CGI 脚本。在 Web 服务器上使用 CGI 脚本会带来安全性风险，因为这些脚本可能会显示包括诸如用户标识和密码等敏感数据的环境变量。在服务器上执行 CGI 程序之前，确保知道它确切的目的，并控制可访问您服务器上 CGI 脚本的访问者。

注：如果使用配置向导来配置代理服务器，然后启用 SSL，那么必须为通过端口 443 检索到的代理请求创建一个映射规则。如需了解更多信息，请参阅第 39 页的『定义映射规则』。

示例：

```
Proxy /* http://content server :443
```

或者

```
Proxy /* https://content server :443
```

第 25 章 服务器保护设置

本章描述如何通过使用保护设置保护您服务器上的数据和文件。保护设置是根据服务器接收的请求（特别是该请求指向的特定目录、文件或文件类型）来触发的。在保护设置中，子伪指令根据要保护的目录或文件的特征，控制应该如何授权或拒绝访问。

使用“配置和管理”表单设置保护

要定义保护设置和如何应用它，在“配置和管理”表单，选择**服务器配置 -> 文档保护**。为下列步骤使用此表单：

1. 设置此保护规则的顺序。

保护规则以配置表单上表中列出的顺序应用。一般，从特殊到一般列出规则。

使用下拉菜单和按钮指定保护规则的位置。

2. 定义请求模板。

保护是根据与客户机发送给您代理服务器的请求内容相比较的请求模板激活的。

请求是完整 URL 中紧跟服务器主机名的一部分。例如，如果服务器名为 `fine.feathers.com`，并且浏览器用户输入 URL `http://fine.feathers.com/waterfowl/schedule.html`，则服务器接收请求 `/waterfowl/schedule.html`。请求模板指定要进行保护的目录和 / 或文件名。例如，根据刚刚描述请求模板 (`/waterfowl/schedule.html`) 激活保护的一些请求包括 `/waterfowl/*` 和 `/*schedule.html`。

在 **URL 请求模板** 字段输入请求模板。

3. 定义保护设置。

保护设置会告知高速缓存代理如何处理与请求模板匹配的请求。您可使用命名保护设置或在**文档保护**表单定义新的设置。

要使用命名设置，单击**命名保护**单选按钮，并在提供的字段中输入名称。要定义新设置，单击**直接插入**单选按钮并遵循提供的说明（请参阅步骤 6）。

4. 选择请求者地址（可选）。

不同的规则可应用于来自不同服务器地址的请求。例如，当对日志文件的请求是从指定给您公司的 IP 地址接收的，则您可能要将不同的保护设置应用到那些请求。

注：对于屏蔽的请求者地址，必须启用 DNS 查找。请参阅第 186 页的『DNS-Lookup - 指定服务器是否查找客户机主机名』。

如果您要将请求者的地址包括在规则中，在**服务器 IP 地址或主机名**字段输入它。

5. 单击**提交**。

如果您使用了命名保护设置，则无需另外输入。如果您选择了直接插入保护设置或指定了不存在的命名设置，则系统将打开附加表单。

6. 设置保护详细信息。

如果您不指定现有的命名保护设置，则会打开附加表单，在此表单上您可指定哪些用户可访问与请求模板匹配的文档或目录，以及允许那些用户执行哪些操作。

- **密码认证设置** - 指定用于用户认证的密码文件和 / 或组文件。当提示您输入请求者名称和密码时，还要指定用于标识服务器的名称。

注：一些浏览器将用户标识和密码高速缓存并将它们与服务器标识相关联。如果您始终将同一个服务器标识与同一个密码文件一起使用，则您的用户可能会发现更方便。

- **许可权** - 指定授权哪些用户或组读、写或删除受保护文件。

7. 单击提交。

8. 重新启动服务器。

使用配置文件伪指令设置保护

要通过直接编辑高速缓存代理的配置文件来设置保护，您必须首先了解下列问题：

- **Protect、defProt 和 Protection 伪指令之间的差异。**
 - **Protect** 伪指令通过链接请求模板到保护设置来设置保护。请参阅第 221 页的『**Protect - 为匹配模板的请求激活保护设置**』以获取更多信息。
 - **defProt** 伪指令设置特定请求模板的缺省保护设置。请参阅第 180 页的『**DefProt - 为与模板匹配的请求指定缺省保护设置**』以获取更多信息。
 - **Protection** 伪指令用于定义命名保护设置。请参阅第 225 页的『**Protection - 在配置文件中定义已命名的保护设置**』以获取更多信息。
- **保护如何与请求路由交互**

请求路由伪指令，如 **Map**、**Exec**、**Pass** 和 **Proxy**，用于控制您的服务器接受哪些请求，以及它如何将请求重定向到实际的文件位置。请求路由伪指令与保护伪指令使用相同的请求模板类型。因为对于每个请求会执行与第一个匹配模板关联的指令，所以在配置文件中保护伪指令必须列出在路由伪指令的前面，以使保护正确地工作。要获取更多的信息，请参阅第 221 页的『**Protect - 为匹配模板的请求激活保护设置**』。

- **直接插入保护设置和命名保护设置之间的差异**

Protect 伪指令既可用于指定直接插入保护设置，也可引用现有的命名设置。两种语句类型的语法略微有所不同。要获取更多信息，请参阅第 221 页的『**Protect - 为匹配模板的请求激活保护设置**』。

- **如何写保护设置**

保护设置是使用保护子伪指令的一系列语句。关于写保护设置的语法和参考信息包含在第 155 页的附录 B，『**配置文件伪指令**』中；请参阅下列参考章节：

- 第 221 页的『**Protect - 为匹配模板的请求激活保护设置**』
- 第 225 页的『**Protection - 在配置文件中定义已命名的保护设置**』
- 第 226 页的『**Protection 子伪指令 - 指定一组资源如何受保护**』

缺省保护设置

缺省代理配置文件包括需要管理员标识和密码以访问 `/admin-bin/` 目录中文件的保护设置。此设置限制对“配置和管理”表单的访问。

第 26 章 安全套接字层 (SSL)

安全套接字层 (SSL) 是在将信息发送到因特网上之前自动对信息加密, 并在使用信息之前在另一端对信息进行解密的系统。当敏感信息 (如信用卡号) 在因特网上发送时, 这会对它进行保护。

高速缓存代理使用 SSL 保护代理服务器并提供安全的远程管理, 如下面的部分所描述的。SSL 还可用于保护到后端服务器 (例如, 内容或应用程序服务器) 的连接, 并保护代理服务器及其客户机之间的通信。

SSL 握手

当安全连接请求从一个机器发送到另一个机器时 (例如, 当浏览器发送请求到代用品代理服务器时), 会启动 SSL 保护。请求语法 `https://` 而不是 `http://` 会告知浏览器在服务器侦听安全连接请求的端口 443 (而不是用于一般请求的端口 80) 上发送请求。要建立浏览器和服务器之间的安全会话, 两个机器执行称为 *SSL 握手* 的交换, 以同意密码规范和选择用于对信息进行加密和解密的密钥。密钥会自动生成, 并且它们会在会话到期时到期。典型方案 (假设为 SSL V3) 如下:

1. Client hello

通过发送描述客户机加密能力的 Client Hello 消息, 客户机将启动与高速缓存代理的 SSL 会话。

2. Server hello

服务器将它的证书发送到客户机并选择用于数据加密的密码套件。

3. Client finish

客户机发送用于创建加密数据的对称加密密钥的密码密钥信息。此密钥资料称为 *预先控制加密*, 并且它是用服务器的公用密钥 (从服务器的证书获得; 请参阅第 108 页的『密钥和证书管理』) 加密的。服务器和客户机都可从预先控制的秘密获得读和写对称加密密钥。

4. Server finish

服务器发送最终确认和整个握手协议的消息认证代码 (MAC)。

5. Client validation

客户机发送消息以验证服务器完成消息。

6. Secure data flow

如果客户机验证服务器完成消息, 则加密数据流开始。

使用高速缓存代理作为安全连接的端点, 可减轻您的内容或应用程序服务器上的负载。当高速缓存代理维护安全连接时, 它执行全部 CPU 密集的操作: 加密、解密和密钥创建。高速缓存代理还允许您配置 SSL 会话超时, 以最大化每个密钥的使用。

SSL 的限制

下列限制适用于 WebSphere Application Server 的高速缓存代理中的 SSL:

- 高速缓存代理本身不能用作认证中心（请参阅『密钥和证书管理』）。
- 一些浏览器可能不支持用于高速缓存代理的所有加密技术。

配置安全远程管理

使用安全套接字层（SSL）提供的安全性功能和密码认证，可实现您的高速缓存代理的远程管理。这样显著地降低了未经授权人员访问代理服务器的可能性。

要在远程管理您的服务器期间应用 SSL，使用 https:// 请求而不是 http:// 请求打开“配置和管理”表单。例如：

```
https://your.server.name/yourFrontPage.html
```

密钥和证书管理

如上所述，在配置 SSL 之前，您必须设置密钥数据库并获取或创建证书。证书用于认证服务器身份。使用 IBM 密钥管理实用程序（有时称为 iKeyman）以设置您的证书文件。此实用程序是 GSKit 软件的部件，此软件包含在 Application Server 中。GSKit 还包括用于打开证书文件的基于 Java 的图形界面。

下列是设置您的 SSL 密钥和证书的基本步骤。

1. 确保 GSKit 已安装。在大多数平台上，它是与高速缓存代理组件一起自动安装的。该软件包的名称是 gsk7ikm（在 i386 的 Linux 系统上是 gsk7ikm_gcc295）。GSKit 通常安装在 ibm/gsk7/ 目录（在 AIX 系统上安装在 ibm/gskit/ 目录）。在 Windows 平台上，还可以从**开始**菜单访问它。

注：在 Windows 上，当使用 InstallShield 时，如果未安装 GSKit，则请检查以确保到安装介质目录的路径不包含空格。

2. 使用密钥管理器创建用于安全网络通信的密钥并从认证中心接收证书。当等待从认证中心接收证书时，您可能决定创建自签署证书。
3. 创建密钥数据库并指定密钥数据库密码

注：无论何时卸载 Application Server 的高速缓存代理组件，都会卸载密钥和密钥存储文件。要避免必须从认证中心请求新证书，在卸载代理软件之前，将这两个文件的备用副本保存在另一个目录中。

认证中心

您的公用密钥必须与来自认证中心（CA）（指定为您服务器上的可信根 CA）的数字签署证书关联。您可通过向认证中心（CA）供应商提交证书请求，来购买已签署证书。高速缓存代理支持下列外部的 CA：

- VeriSign
- Thawte

缺省情况下，下列 CA 指定为可信 CA：

- Verisign Class 1 Individual Subscriber CA - 角色未验证
- Verisign Class 2 Individual Subscriber CA - 角色未验证

- Verisign Class 3 Individual Subscriber CA - 角色未验证
- Verisign Class 3 International Server CA
- Verisign Class 2 OnSite Individual CA
- Verisign Class 1 Public Primary CA
- Verisign Class 2 Public Primary CA
- Verisign Class 3 Public Primary CA
- Verisign Class 1 Public Primary CA - G2
- Verisign Class 2 Public Primary CA - G2
- RSA Secure Server CA (来自 Verisign)
- Thawte Personal Basic CA
- Thawte Personal Freemail CA
- Thawte Personal Premium CA
- Thawte Premium Server CA
- Thawte Server CA

使用 IBM 密钥管理器实用程序

此部分提供使用 IBM 密钥管理实用程序 (iKeyman) 的快速参考。使用密钥管理器创建您的 SSL 密钥数据库文件、公用 / 专用密钥对和证书请求。接收 CA 签署的证书之后, 使用密钥管理器将证书放置在您创建了原始证书请求的密钥数据库中。

有关 IBM 密钥管理器和 GSKit 的更详细的文档与 GSKit 软件封装在一起。

设置您的系统以运行密钥管理器

在启动 IKeyman GUI 前, 执行下列操作:

1. 安装 IBM 或 IBM 相当的 32 位 Java 2 技术, V1.4.2
2. 设置 JAVA_HOME 为 Java 目录位置。例如:
 - Windows: set JAVA_HOME=C:\Program Files\IBM\Java142
 - Linux 和 UNIX: export JAVA_HOME=/usr/opt/IBMJava2-142
3. 从 JAVA_HOME/jre/lib/ext 目录除去 ibmjss.jar 和 gskikm.jar (如果存在) 和 ibmjcaprovider.jar 文件。

注: 对于 Sun, 用 JAVA_HOME/lib/ext/ 目录替换 JAVA_HOME/jre/lib/ext 目录。

4. 所有下列 JAR 文件当前都在 *GSKit_Installation_path/classes/jre/lib/ext/* 中。
 - 将指定的 JAR 文件复制到 JAVA_HOME/jre/lib/ 中
 - ibmjcefw.jar
 - ibmpkcs11.jar
 - 将指定的 JAR 文件复制到 JAVA_HOME/jre/lib/ext 中
 - ibmjceprovider.jar
 - ibmpkcs.jar
 - 将指定的 JAR 文件复制到 JAVA_HOME/jre/lib/security 中
 - local_policy.jar
 - US_export_policy.jar
5. 注册 IBM JCE、IBM CMS 和 / 或 IBMJCEFIPS 服务提供者:

更新 `JAVA_HOME/jre/lib/security/java.security` 文件以在 Sun 提供程序后添加 IBM CMS 和 IBM JCE 提供程序。例如:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.ibm.spi.IBMCMSProvider
security.provider.3=com.ibm.crypto.provider.IBMJCE
```

在 `GSKit_Installation_path/classes/gsk_java.security` 中可以找到样本 `java.security` 文件。

- 要启用 FIPS 操作, 更新 `JAVA_HOME/jre/lib/security/java.security` 文件以在 Sun 提供者后添加 `IBMJCEFIPS` 提供者。确保 `IBMJCEFIPS` 提供程序比 `IBMJCE` 注册优先级高。例如:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.ibm.spi.IBMCMSProvider
security.provider.3=com.ibm.crypto.fips.provider.IBMJCEFIPS
security.provider.4=com.ibm.crypto.provider.IBMJCE
```

6. (可选) 如果您是 JSSE 用户并且使用 JSSE 访问 crypto 硬件, 则在 `JAVA_HOME/jre/lib` 目录中安装 `ibmpkcs11.jar`, 并且按照 `GSKit_Installation_path/classes/native/native-support.zip` 中的说明设置 crypto 硬件共享库。

注: 您还应该查找 2002 年 8 月 5 日后发行的 JSSE 软件包中的 `ibmpkcs11.jar`。要注册 `IBMPKCS11` 服务提供者, 如下是更新 `JAVA_HOME/jre/lib/security/java.security` 文件的一个示例:

```
security.provider.1=sun.security.provider.Sun
security.provider.2=com.ibm.crypto.provider.IBMJCE
security.provider.3=com.ibm.crypto.pkcs11.provider.IBMPKCS11
```

启动密钥管理器

启动密钥管理器图形用户界面, 按如下所示:

- 在 Linux 和 UNIX 平台上, 在命令提示符下输入 `gsk7ikm`。
- 在 Windows 平台上, 单击开始 -> 程序 -> **IBM WebSphere -> Edge Components -> 高速缓存代理 -> 启动密钥管理实用程序**。

注意, 如果您在此会话期间创建新密钥数据库文件, 则此文件将存储在您启动密钥管理器的目录中。

创建新的密钥数据库、密码和存储文件

密钥数据库是服务器用于存储一个或多个密钥对和证书的文件。您可将一个密钥数据库用于您所有的密钥对和证书或创建多个数据库。密钥管理实用程序用于创建新的密钥数据库并指定它们的密码和存储文件。

要创建密钥数据库和存储文件:

1. 启动密钥管理实用程序。
2. 从主菜单, 选择**密钥数据库文件 -> 新建**。
3. 在**新建**对话框中, 确保文件类型 **CMS 密钥数据库**是选中的。输入您的密钥数据库名称和文件位置或接受缺省值, **key.kdb**。单击**确定**。
4. 在**密码提示**对话框, 输入并确认此数据库的密码。单击**确定**。
5. 选择存储密码文件的复选框。当出现提示时, 输入并确认密码以进行验证。将显示以下消息: `DB-Type: CMS key database file keyfile_database_name`

注：如果您不存储密码文件，则服务器将启动但不在端口 443 上侦听。

当您创建新的密钥数据库时指定的密码会保护专用密钥。此专用密钥是可签署文档或使用公用密钥加密的消息进行解密的唯一密钥。

当指定密码时使用下列准则：

- 密码必须由美国英语字符集组成。
- 密码长度应该至少为六个字符并包含至少两个非连续数字。确保密码不包含有关您的公开可获得信息的组合，如您的姓名或您的家人的姓名、姓和名的头一个字母或出生日期。
- 存储密码。

经常更改密钥数据库密码是一个很好的做法。然而，如果您指定密码的到期日期，则记录何时更改它。在您更改密码之前如果密码到期，有一条消息会写到错误日志中并且服务器将启动，但无法进行安全网络连接。

按照这些步骤更改密钥数据库密码：

1. 从主菜单，单击**密钥数据库文件** -> **打开**。
2. 在**打开**对话框，输入密钥数据库名称或接受缺省值，**key.kdb**。单击**确定**。
3. 在**密码提示**对话框，输入已建立的密码并单击**确定**。
4. 从主菜单，单击**密钥数据库文件** -> **更改密码**。
5. 从**更改密码**对话框，输入并确认新密码。单击**确定**。

对于代理和 LDAP 服务器之间的 SSL 连接，将密钥数据库密码放入 pac_keyring.pwd 文件中。（注意，pac_keyring.pwd 文件不是由 IKeyMan 生成的储存文件。）

创建新的密钥对和证书请求

密钥数据库存储密钥对和证书请求。要创建公用 / 专用密钥对和证书请求，按照这些步骤执行：

1. 如果您未创建密钥数据库，则遵循第 110 页的『创建新的密钥数据库、密码和存储文件』中的说明。
2. 在密钥管理实用程序中，从主菜单，单击**密钥数据库** -> **文件** -> **打开**。
3. 在**打开**对话框，输入密钥数据库名称（或如果使用缺省值，单击 **key.kdb**）。单击**确定**。
4. 在**密码提示**对话框，输入您的密码并单击**确定**。
5. 从主菜单，单击**创建** -> **新证书请求**。
6. 在**新建密钥和证书请求**对话框，指定下列内容：
 - **密钥标签：**输入用于标识数据库中的密钥和证书的一个名称（标签）：例如，my self-signed certificate 或 www.companyA.com。
 - **Keysize：**密钥的大小，例如，1024。（为了利用 128 位加密，建议使用密钥大小为 1024。）
 - **组织名称：**与密钥关联的组织的名称，例如，Company A。
 - **组织单元**（可选）
 - **地区**（可选）
 - **省 / 自治区 / 直辖市**（可选）

- 邮编（可选）
 - 国家或地区：您的国家或地区代码。您必须至少指定两个字符，例如 US。
 - 证书请求文件名：请求文件的名称。可选地，可使用缺省名称。
7. 单击**确定**。将显示确认消息：

已成功地在文件 *keyfile_database_name* 中创建新的证书请求。
 8. 单击**确定**。您输入的标签名将显示在**个人证书请求**标题下面。
 9. 在**信息**对话框中，单击**确定**。将提醒您把文件发送到认证中心。
 10. 除非您已创建自签署证书（请参阅以下部分“创建自签署证书”以获取详细信息），否则将证书请求发送到 CA：
 - 使密钥管理器保持运行状态。
 - 启动 Web 浏览器并输入您要从中获得证书的 CA 的 URL。
 - 按照 CA 提供的说明发送您的证书请求。

完成证书请求可能需要两到三周。当您等待 CA 处理您的证书请求时，您可担当您自己的 CA 并使用 iKeyman 创建自签署服务器证书，以启用客户机和您的高速缓存代理服务器之间的 SSL 会话。

创建自签署证书

在等待发出证书时，使用密钥管理实用程序创建自签署服务器证书，以启用客户机和您的代理服务器之间的 SSL 会话。您还可使用用于测试目的的自签署证书。

按照此过程创建自签署证书：

1. 如果您未创建密钥数据库，则遵循第 110 页的『创建新的密钥数据库、密码和存储文件』中的说明。
2. 在密钥管理实用程序中，从主菜单，单击**密钥数据库 -> 文件 -> 打开**。
3. 在**打开**对话框，输入您的密钥数据库名称（或接受缺省值，**key.kdb**）。单击**确定**。
4. 在**密码提示**对话框，输入您的密码并单击**确定**。
5. 在**密钥数据库内容**框架中，选择**个人证书**并单击**创建新建自签署证书**。
6. 在**创建新建自签署证书**窗口，指定下列内容：
 - **密钥标签**：用于标识数据库中的密钥和证书的一个名称（标签）：例如，**my self-signed certificate**
 - **密钥大小**：密钥的大小，例如，**512**。
 - **公共名**：服务器的全主机名，例如，**www.myserver.com**
 - **组织名**：与密钥关联的组织的名称，例如 **Company A**
 - **组织单元**（可选）
 - **地区**（可选）
 - **省 / 自治区 / 直辖市**（可选）
 - **邮编**（可选）
 - **国家或地区**：您的国家或地区代码。您必须至少指定两个字符，例如，**US**。
 - **有效期**：证书有效的时期。
7. 单击**确定**。
8. 通过将密钥文件和存储文件添加到配置设置，在服务器中注册密钥数据库（请参阅第 110 页的『创建新的密钥数据库、密码和存储文件』）。

导出密钥

使用此过程导出密钥到另一个密钥数据库:

1. 启动密钥管理实用程序。
2. 从主菜单, 单击**密钥数据库文件** -> 打开。
3. 在打开对话框, 输入您的密钥数据库名称 (或接受缺省值, **key.kdb**)。单击**确定**。
4. 在**密码提示**对话框, 输入您的密码并单击**确定**。
5. 在**密钥数据库**内容框架中, 选择**个人证书**, 然后单击标签上的**导出 / 导入**按钮。
6. 在**导出 / 导入密钥**窗口中:
 - 选择**导出密钥**。
 - 选择目标数据库类型 (例如, **PKCS12**)。
 - 输入文件名或单击**浏览**以选择它。
 - 输入正确位置。
7. 单击**确定**。
8. 在**密码提示**对话框中, 输入正确密码, 再次输入密码以确认, 然后单击**确定**以导出所选的密钥到另一个密钥数据库。

导入密钥

要从另一个密钥数据库导入密钥:

1. 启动密钥管理实用程序。
2. 从主菜单, 选择**密钥数据库文件** -> 打开。
3. 在打开对话框, 输入您的密钥数据库名称 (或接受缺省值, **key.kdb**)。单击**确定**。
4. 在**密码提示**对话框, 输入正确密码并单击**确定**。
5. 在**密钥数据库**内容框架中, 选择**个人证书**, 然后单击标签上的**导出 / 导入**按钮。
6. 在**导出 / 导入密钥**窗口中:
 - 选择**导入密钥**。
 - 选择密钥数据库文件类型 (例如, **PKCS12**)。
 - 输入文件名或单击**浏览**以选择它。
 - 选择正确的位置。
7. 单击**确定**。
8. 在**密码提示**对话框, 输入正确密码并单击**确定**。
9. 在**从密钥标签选择**列表中, 选择正确标签名并单击**确定**。

列出认证中心

要显示密钥数据库中可信认证中心 (CA) 的列表:

1. 启动密钥管理实用程序。
2. 从主菜单, 单击**密钥数据库文件** -> 打开。
3. 在打开对话框, 输入您的密钥数据库名称 (或接受缺省值, **key.kdb**)。单击**确定**。
4. 在**密码提示**对话框, 输入正确密码并单击**确定**。
5. 在**密钥数据库**内容框架, 选择**签署者证书**。
6. 单击**签署者证书**、**个人证书**或**证书请求**以查看密钥信息窗口的 CA 列表。

接收 CA 证书

缺省情况下，使用此过程从指定为可信 CA 的认证中心（CA）中，接收通过电子邮件发送的证书（请参阅第 108 页的『认证中心』中的列表）。如果发出您 CA 签署的证书的 CA 不是密钥数据库中的可信 CA，则您必须首先存储此 CA 的证书并将此 CA 指定为可信 CA。然后将您的 CA 签署的证书接收到数据库中。您无法接收来自不是可信 CA 的 CA 签署的证书（请参阅『存储 CA 证书』）。

要将 CA 签署的证书接收到密钥数据库：

1. 启动密钥管理实用程序。
2. 从主菜单，选择**密钥数据库文件 -> 打开**。
3. 在打开对话框，输入您的密钥数据库名称（或接受缺省值，**key.kdb**）。单击**确定**。
4. 在**密码提示**对话框，输入您的密码并单击**确定**。
5. 确保 **DB-Type** 列表中的文件名是正确的。
6. 在**密钥数据库**窗口中，选择**个人证书**，然后单击**接收**。
7. 在**从文件接收证书**对话框中，在**证书文件名**文本字段中输入有效的基本 64 编码的文件名。单击**确定**。
8. 要关闭密钥管理实用程序，从主菜单，单击**密钥数据库文件 -> 退出**。

存储 CA 证书

仅可信 CA 签署的证书可接受用于建立安全连接。要将 CA 添加到可信认证中心的列表，您必须获取它的证书并存储为可信的。按照此过程，在将来自新的 CA 的证书接收到数据库之前存储它：

1. 启动密钥管理实用程序。
2. 从主菜单，单击**密钥数据库文件 -> 打开**。
3. 在打开对话框，输入您的密钥数据库名称（或接受缺省值，**key.kdb**）。单击**确定**。
4. 在**密码提示**对话框，输入您的密码并单击**确定**。
5. 在**密钥数据库**内容框架，选择**签署者证书**，然后单击**添加**。
6. 在**从文件添加 CA 证书**对话框中，选择基本 64 编码的 ASCII 数据证书文件名，或使用**浏览**选项。单击**确定**。
7. 在**标签**对话框中，输入标签名并单击**确定**。
8. 使用此复选框以指定证书为可信的（缺省值）。

注：使用“查看 / 编辑”按钮创建证书后，查看此复选框。此复选框在面板上列出，但是在添加证书期间不显示。

显示密钥数据库中的缺省密钥

显示缺省密钥条目如下所示：

1. 启动密钥管理实用程序。
2. 从主菜单，单击**密钥数据库文件 -> 打开**。
3. 在打开对话框，输入您的密钥数据库名称（或接受缺省值，**key.kdb**）。单击**确定**。
4. 在**密码提示**对话框，输入您的密码并单击**确定**。
5. 在**密钥数据库**内容框架，选择**个人证书**和选择 CA 证书标签名。
6. 在**密钥信息**窗口，单击**查看 / 编辑**以显示证书缺省密钥信息。

支持的密码规范

下表列出了用于 SSL V2 和 V3 的加密算法和散列。

密钥对生成: RSA 512–1024 专用密钥大小

SSL V2

美国版本	出口版本
RC4 US	RC4 Export
RC2 US	RC2 Export
DES 56 位	不可用
Triple DES US	不可用
RC4 Export	不可用
RC2 Export	不可用

SSL V3

美国版本	出口版本
Triple DES SHA US	DES SHA Export
DES SHA Export	RC2 MD5 Export
RC2 MD5 Export	RC4 MD5 Export
RC4 SHA US	NULL SHA
RC4 MD5 US	NULL MD5
RC4 MD5 Export	NULL NULL
RC4 SHA 56 位	不可用
DES CBC SHA	不可用
NULL SHA	不可用
NULL MD5	不可用
NULL NULL	不可用

还可通过直接编辑代理配置文件配置这些 SSL 规范。要获取详细信息，请参阅第 155 页的附录 B，『配置文件伪指令』中的参考章节，以获取关于下列伪指令的信息：

- 第 249 页的『V2CipherSpecs - 列出 SSL V2 的支持密码规范』
- 第 249 页的『V3CipherSpecs - 列出 SSL V3 的支持密码规范』

用于高速缓存代理的 128 位加密

现仅提供 128 位加密版本的高速缓存代理。56 位版本不再可用。如果您正在更新先前的版本，则可以将高速缓存代理直接安装到当前已安装的 128 位或 56 位版本上。如果您先前使用的是 56 位（出口）浏览器，则您必须升级到 128 位浏览器，这样才能利用代理中的 128 位加密。

在将高速缓存代理的 56 位版本升级到 128 位版本后，如果用于加密证书的密钥大小设置为 1024，则无需更改配置。然而，如果密钥大小设置为 512，则为了利用代理的 128 位加密，您必须新建密钥大小为 1024 的证书。使用 IBM 密钥管理器实用程序（iKeyman）创建新密钥。

1. 启动密钥管理器。
 - 在 Linux 和 UNIX 平台上，在命令提示符下输入 `gsk7ikm`。
 - 在 Windows 系统上，单击 **开始 -> 程序 -> IBM WebSphere -> Edge Components -> 启动密钥管理实用程序**。
2. 从主菜单，单击 **密钥数据库文件 -> 打开**。
3. 在打开对话框，输入您的密钥数据库名（或如果您正在使用缺省值，单击 **key.kdb**）并单击 **确定**。
4. 如果 **密码提示** 对话框打开，输入您的密码并单击 **确定**。
5. 从主菜单，单击 **创建 -> 新证书请求**。
6. 在 **新建密钥和证书请求** 窗口中，指定下列内容：
 - **密钥标签**：输入一个名称，用于标识数据库中的密钥和证书。
 - **密钥大小**：选择 **1024**。
 - **组织名称**：输入与密钥关联的组织名称。
 - **国家或地区**：输入您的国家或地区代码。您必须至少指定两个字符，例如 **US**。
 - **证书请求文件名**：输入请求文件的名称，或可选地使用缺省名称。
7. 单击 **确定**。

请参阅第 108 页的『密钥和证书管理』以获得 IBM 密钥管理器实用程序的详细讨论。

注意，此产品的该版本不支持 SuSE Linux 上的加密。

第 27 章 启用加密硬件的支持

按照此过程启用要下载的 SSL 握手例程以加密硬件卡:

1. 根据制造商说明安装加密硬件卡。
2. 为高速缓存代理启用 SSL。如需了解更多信息, 请参阅第 107 页的第 26 章, 『安全套接字层 (SSL)』。
3. 手工编辑 `ibmproxy.conf` 配置文件中的 `SSLCryptoCard` 伪指令。没有此伪指令的条目出现在“配置和管理”表单。要获取更多的信息, 请参阅 `SSLCryptoCard` 伪指令引用, 第 244 页的『`SSLCryptoCard` - 指定已安装的加密卡』。

第 28 章 使用 Tivoli Access Manager 插件

与 Tivoli Access Manager (以前是 Tivoli Policy Director) 一起提供的高速缓存代理插件使高速缓存代理能使用 Access Manager 进行认证和授权。此插件使一个企业能够使用 Access Manager 进行 Web 访问控制, 以添加 Edge Server 技术而无需通过为代理服务器设置独立的授权方案而进行重复的工作。

要获得关于 Tivoli Access Manager 的其它信息, 请查看该产品 Web 站点: <http://www.ibm.com/software/tivoli/products/>。要获得有关软件和硬件需求以及安装 Access Manager 插件的信息, 请参阅与 Tivoli AccessManager 一起提供的文档。

注: Tivoli Access Manager 插件在 Red Hat Linux 上可能不支持。联系 Tivoli 以获得关于 Linux 平台的当前支持信息。

配置

高速缓存代理的设置脚本与 Access Manager 插件一起提供。

使用配置脚本前要执行的步骤

在运行脚本前, 执行下列操作:

- 安装所有必需的软件。
- 确保代理服务器设置为使用 80 (这是缺省值)。
- 配置您的 LDAP 和 Access Manager 组件, 并确保您在配置 Access Manager 插件时它们正在运行。
- 确保您具有 Access Manager 管理员标识, 并且 LDAP 管理员名是可用的。这些值是设置代理服务器必需的。

使用配置脚本

设置名为 **wslconfig.sh** 的脚本, 它提供在 `/opt/pdweb-lite/bin/` 目录中。提示时, 输入 Access Manager 管理员标识和 LDAP 管理员名称。

配置脚本自动执行下列步骤:

- 把高速缓存代理用户标识设置为 `root`, 并将组标识设置为 `other`
- 把 `noLog` 伪指令设置为 `*`, 这样任何项都不会写入高速缓存代理访问日志中
- 用下列信息创建 `ServerInit` 伪指令:

```
ServerInit /opt/pdweb-lite/lib/wesauth.so:WTESeal_Init  
/opt/pdweb-lite/etc/ibmwesas.conf
```

- 用下列信息创建 `PreExit` 伪指令:

```
PreExit /opt/pdweb-lite/lib/wesauth.so:WTESeal_PreExit
```

- 用下列信息创建 `Authorization` 伪指令:

```
Authorization * /opt/pdweb-lite/lib/wesauth.so:WTESeal_Authorize
```

- 用下列信息创建 `ServerTerm` 伪指令:

```
ServerTerm /opt/pdweb-lite/lib/wesauth.so:WTESeal_Term
```

创建 Protect 语句和 Protection 设置，把所有请求转发到 Access Manager 认证过程，如下所示：

```
Protection PROXY-PROT {
    ServerId WebSEAL-Lite
    Mask      All@(*)
    AuthType  Basic
}
Protect * PROXY-PROT
```

启动高速缓存代理和 Access Manager 插件

在配置代理服务器和 Access Manager 插件后，使用命令 **wslstartwte**（而不是 **ibmproxy start**）来启动代理服务器。**wslstartwte** 命令自动装入环境变量，这些变量是 Access Manager 插件初始化时需要的。如果启动代理服务器时不使用 **wslstartwte**，那么会显示一条有关 Access Manager 插件的错误消息。当使用该插件时，相应的停止命令 **ibmproxy stop** 仍然有效。

第 29 章 使用 PAC-LDAP 授权模块

概述

当执行授权或认证例程时，PAC-LDAP 授权模块使高速缓存代理能访问轻量级目录访问协议（LDAP）服务器。模块由两个组件集组成：一对共享库（它们将 LDAP 功能添加到高速缓存代理 API）和策略认证控制（PAC）守护程序。当启动高速缓存代理时，ibmproxy.conf 文件中的 ServerInit 伪指令指示共享库初始化一个或多个 PAC 守护程序。共享库读 paccp.conf 文件以确定 PAC 守护程序数和特征。在初始化期间，守护程序为配置伪指令读 pac.conf 文件，并为策略信息读 pacpolicy.conf 文件。然后，ibmproxy.conf 文件中的 Authentication 伪指令指示代理服务器在需要认证时调用共享库，或者 Authentication 伪指令在标准 HTTP 请求处理期间，篡夺高速缓存代理工作流。

认证

认证过程确定提供的凭证集（用户名和密码）是否有效。此过程包括验证用户是否在注册表并且验证提供的密码是否匹配注册表中存储的密码。这些是通过在认证步骤期间使用 PAC-LDAP 模块而执行的操作。

当启用 PAC-LDAP 授权模块来认证时，它成为缺省库，用户标识、密码和组将从中检索。由于 HTTP 请求通过高速缓存代理工作流传递，所以每个 Protect 伪指令将请求的 URL 与其请求模板相比较。如果匹配发生，Protect 伪指令调用保护模式，该保护模式包含服务器标识、使用的认证类型、适用于请求客户机的掩码规则以及密码和组文件的位置。如果未定义密码文件，则用户标识和密码将通过 PAC-LDAP 授权模块检索。类型 0、1、2 和 3 策略定义认证方案。如果认证通过，则为请求提供服务；否则，高速缓存代理将 401 错误返回到客户机。

授权

授权过程确定用户是否具有访问受保护资源必需的许可权。使用 PAC-LDAP 模块时，包括应用驻留在 HTTP 请求的 pacpolicy.conf 文件中的授权规则。

当为授权启用 PAC-LDAP 授权模块时，pacpolicy.conf 文件中的授权规则将应用于 HTTP 请求。由于 HTTP 请求通过高速缓存代理工作流传递，所以每个 Protect 伪指令将请求的 URL 与其请求模板相比较。如果匹配发生，Protect 伪指令将调用保护模式。在这种情况下，保护模式是由 PAC-LDAP 授权模块篡夺的授权例程。Authorization 伪指令将请求的 URL 与其请求模板相比较，并且当发生匹配时，调用 PAC-LDAP 授权模块。定义在 pacpolicy.conf 文件中的类型 4 策略进一步细化了各种 URL 请求必需的认证。

轻量级目录访问协议（LDAP）

LDAP 以最小的系统资源消耗，提供对 X.500 目录的交互式访问。IANA 已把 TCP 端口 389 和 UDP 端口 389 指定给 LDAP。要获取更多的信息，请参阅 RFC 1777，其中定义了 LDAP。

安装

当安装 WebSphere Application Server, 版本 6.0 的高速缓存代理系统时, 将自动安装 PAC-LDAP 授权模块的所有组件。在 Linux 和 UNIX 系统上, 高速缓存代理 library (./lib/) 目录、PAC-LDAP 授权模块库 (./lib/plugins/pac/) 目录、二进制 (./bin/) 目录和配置 (./etc/) 目录在 /opt/ibm/edge/cp/ 目录中创建。然后, 将符号链接从 /usr/lib/、/usr/sbin/ 和 /etc 目录创建到这些产品的特定目录中。

目录结构

Linux 和 UNIX 目录	Windows 目录	内容
/opt/ibm/edge/cp/	\Program Files\IBM\edge\cp\	高速缓存代理基本目录 (cp_root)
cp_root/sbin/	\Program Files\IBM\edge\cp\Bin\	高速缓存代理二进制和脚本
/usr/sbin/		到 cp_root/sbin/ 的符号链接
cp_root/etc/	\Program Files\IBM\edge\cp\etc\	高速缓存代理配置文件
/etc/		到 cp_root/etc/ 的符号链接
cp_root/lib/	\Program Files\IBM\edge\cp\lib\plugins\	高速缓存代理库
cp_root/lib/ plugins/pac/	\Program Files\IBM\edge\cp\lib\plugins\pac\	PAC-LDAP 授权模块库
/usr/lib/		到 cp_root/lib/ 和 cp_root/lib/plugins/pac/ 的符号链接
cp_root/server_root/pac/data/	\Program Files\IBM\edge\cp\server_root\pac\data\	PAC-LDAP 授权模块数据存储
cp_root/server_root/ pac/creds/	\Program Files\IBM\edge\cp\server_root\pac\creds\	PAC-LDAP 授权模块凭证

LDAP 插件文件

Linux 和 UNIX 文件名	Windows 文件名	描述
libpacwte.so	pacwte.dll	共享库
libpacman.so	pacman.dll	共享库
pacd_restart.sh	pacd_restart.bat	PAC 守护程序重新启动脚本
paccp.conf, pac.conf, pacpolicy.conf	paccp.conf, pac.conf, pacpolicy.conf	配置和策略文件

安全 PACD-LDAP 服务器连接的附加要求

LDAP 客户机包需要 GSKit

要启用 PACD 守护程序和 LDAP 服务器间的安全套接字层 (SSL) 连接, 您应该安装 LDAP 客户机包需要的 GSKit 包。缺省情况下, 高速缓存代理机器上需要并提供 GSKit 7, 但它可能不是机器上的 LDAP 客户机需要的版本。可对不同进程在同一机器上使用不同 GSKit 版本。

将 GSKit 密钥文件放到 `$pacd_creds_dir/pac_keyring.kdb`, 而将密码放到 `$pacd_creds_dir/pac_keyring.pwd`。

注: 要获取有关 LDAP 服务器的 GSKit 要求信息, 请参阅以下 Web 站点的 IBM Directory Server 文档: <http://www.ibm.com/software/tivoli/products/directory-server/>

必须为 Linux 系统设置 LD_PRELOAD 环境变量

在 Linux 系统上, LD_PRELOAD 环境变量必须按如下所示进行配置, 以启用 PACD 守护程序和 LDAP 服务器间的 SSL 连接。将变量设置为下列值:

```
LD_PRELOAD=/usr/lib/libstdc++-libc6.1-1.so.2
```

在本节中先前引用过的 GSKit 要求也适用于 Linux 系统。

编辑 ibmproxy.conf 文件以启用 PAC-LDAP 授权模块

三个伪指令 (ServerInit、Authorization 或 Authentication) 和 ServerTerm 必须添加到 `ibmproxy.conf` 文件的 API 伪指令部分, 这样才能初始化 PAC-LDAP 授权模块。要创建这些伪指令, 手工编辑 `ibmproxy.conf` 文件, 或者如果代理服务器已在运行, 则使用因特网浏览器连接到“配置和管理”表单, 并打开“API 请求处理”表单 (单击**服务器配置** -> **请求处理** -> **API 请求处理**)。在代理配置文件中, 每个伪指令必须出现在一行中, 无论在此部分中给出的这些示例是否包含换行符以便清楚显示。

注意, `prototype` 伪指令 (以注释格式) 在 `ibmproxy.conf` 文件的 API 部分中给出。这些 API 伪指令是按有目的的顺序进行排列的。当添加 API 伪指令以启用新功能和插件模块时, 请按配置文件原型部分中显示的顺序排列这些伪指令。或者, 取消注释并编辑 (如果需要) API 伪指令, 以包含对每个所需功能或插件的支持。

`ServerInit` 伪指令有三个参数: (1) 共享库的全限定路径, (2) 函数调用, 和 (3) `paccp.conf` 文件的全限定路径。第一和第二个自变量由冒号 (:) 定界。第二和第三个自变量由空格定界。第一和第三个自变量是系统特定的, 并取决于插件组件的安装位置。第二个自变量是硬编码到共享库中的, 且必须按显示精确输入。当使用“API 请求处理”表单创建 `ServerInit` 伪指令时, 必须在**函数名字段**中输入第二和第三个自变量。第三个自变量显示在 **IP 模板列**中。

`Authorization` 伪指令有三个自变量: (1) 请求模板、(2) 共享库的全限定路径, 和 (3) 函数名。将 HTTP 请求与请求模板比较, 以确定是否调用应用程序函数。请求模板可包含协议、域和主机; 请求模板前可有斜杠 (/); 可使用星号 (*) 作为通配符。例如, `/front_page.html`、`http://www.ics.raleigh.ibm.com`、`/pub*`、`/*` 以及 `*` 均为有效。函数名是在程序中给定您的应用程序函数的名称。它是硬编码的, 且必须按显示精确输入。前两个自变量由空格定界。后二个自变量由冒号 (:) 定界。

Authentication 伪指令有两个自变量: (1) 共享库的全限定路径和 (2) 函数名。这两个自变量由冒号 (:) 定界。第一个自变量是系统特定的, 并取决于共享库的安装位置。当将高速缓存代理用作反向代理时, 第一个自变量的 URL 模板必须以文档根 (/) 开头。第二个自变量是硬编码到共享库中的, 且必须按显示精确输入。

ServerTerm 伪指令有两个自变量: (1) 共享库的全限定路径和 (2) 函数名。这两个自变量由冒号 (:) 定界。第一个自变量是系统特定的, 并取决于共享库的安装位置。第二个自变量是硬编码到共享库中的, 且必须按显示精确输入。当代理服务关闭时, 此伪指令终止 PAC 授权守护程序。如果守护程序的所有者与代理服务器的所有者不同, 则代理服务器可能无法停止该守护程序, 在这种情况下, 管理员必须手工停止此守护程序。

```
ServerInit path_of_shared_library:pacwte_auth_init path_of_conf_policy_file
```

Linux 和 UNIX 示例:

```
ServerInit /usr/lib/libpacwte.so:pacwte_auth_init /etc/pac.conf
```

Windows 示例:

```
ServerInit C:\Progra~1\IBM\edge\cp\lib\plugins\  
pac\pacwte.dll:pacwte_auth_init C:\Progra~1\IBM\edge\cp  
Authorization request-template path_of_shared_library:pacwte_auth_policy
```

Linux 和 UNIX 示例:

```
Authorization http://* /usr/lib/libpacwte.so:pacwte_auth_policy
```

Windows 示例:

```
Authorization http://* C:\Program Files\IBM\edge\cp\lib\plugins\  
pac\pacwte.dll:pacwte_auth_policy  
Authentication BASIC path_of_shared_library:pacwte_auth_policy
```

Linux 和 UNIX 示例:

```
Authentication BASIC /usr/lib/plugins/pac/libpacwte.so:pacwte_auth_policy
```

Windows 示例:

```
Authentication BASIC C:\Program Files\IBM\edge\cp\lib\plugins\  
pac\pacwte.dll:pacwte_auth_policy  
ServerTerm path_of_shared_library:pacwte_shutdown
```

Linux 和 UNIX 示例:

```
ServerTerm /usr/lib/libpacwte.so:pacwte_shutdown
```

Windows 示例:

```
ServerTerm BASIC C:\Program Files\IBM\edge\cp\lib\plugins\  
pac\bin\pacwte.dll:pacwte_shutdown
```

编辑 PAC-LDAP 授权模块配置文件

PAC-LDAP 授权模块 配置和策略文件必须使用文本编辑器手工编辑。伪指令名与其第一个自变量之间用冒号 (:) 分隔。多个自变量间用逗号 (,) 定界。包含在配置和策略文件中的注释可用于帮助编辑此文件。密钥策略伪指令显示如下。

paccp.conf

在高速缓存代理初始化期间, `paccp.conf` 由共享库读取, 并包含要启动的每个 PAC 守护程序的定义 (`[PAC_MAN_SERVER]` 节)。每个 PAC 守护程序必须拥有其自己的 `[PAC_MAN_SERVER]` 节。

```
[PAC_MAN_SERVER]
hostname:                # name of PAC daemon
port:                    # port pacd is listening on

[PACWTE_PLUGIN]
hostname_check:[true|false] # enables DNS lookup. Must have
                             # DNS lookup turned on for ibmpoxy to work.
```

pac.conf

`pac.conf` 文件指定 PAC 守护程序尝试连接的 LDAP 服务器。

```
[PAC_MAN_SERVER]
hostname:                # name of PAC daemon
port:                    # port pacd is listening on
conn_type:ssl            # comment out if you do not use SSL
authentication_sequence: [primary|secondary|none]
authorization_sequence: [primary|secondary|none]

[LDAP_SERVER]
hostname:                # LDAP Server hostname
port:389                 # Port LDAP is listening on
ssl_port:636             # SSL port used by the LDAP server
admin_dn:                # User with permission to access the LDAP server
                             # specify admin_dn=NULL to enable anonymous binding
search_base:             # Portion of LDAP tree to search for policy info
                             # If not required, specify search_base=NULL
search_key:              # ID field to search

[CACHE]
cred_cache_enabled [TRUE|FALSE] # turn credentials cache on
cred_cache_min_size:100        # minimum number of credentials to cache in pacd
cred_cache_max_size:64000      # maximum number of credentials to cache in pacd
cred_cache_expiration:86400    # when a credential expires
policy_cache_enabled:[TRUE|FALSE] # turns policy cache on/off
policy_cache_min_size:100      # min. number of policy related items to cache
policy_cache_max_size:64000    # max. number of policy related items to cache
policy_cache_expiration:86400  # when a policy related item expires
```

pacpolicy.conf

每个 LDAP 策略在配置和策略文件中都使用下列模板。每个策略必须以用方括号括住的大写关键字 `POLICY` 开头。

```
[POLICY]
default_policy:[grant|deny] # describes the default policy for users
                             # that are not described in the POLICY section
pac_client_hotname:         # the instances of Caching Proxy that are allowed
                             # to use a policy list
id:                          # the id for the LDAP entry or ip/hostname
                             # (wildcard supported, such as *.ibm.com)
grant:[true|false]          # true means to grant access, false means
                             # to deny access
type:[0|1|2|3|4]            # 0 LDAP entry that is a group,
                             # 1 LDAP entry that is not a group,
                             # 2 IP address
                             # 3 hostname
                             # 4 URL
propagate:[true|false]     # true means that the access rights (grant
                             # or deny) will be propagated to all
```

```

stop_entry:[entry|NULL] # descendants or members
                        # Propagation of the access right stops
                        # at this entry. If the id is a group,
                        # stop_entry must be set to NULL.
                        # stop_entry may be applied to an IP
                        # address or hostname. Each stop_entry
                        # must be on its own line
exception_entry:[entry|NULL] # Assignment of the access right skips
                              # these entries, but continues through their
                              # subtrees. This may be a list of entries.
                              # exception_entry may be applied to a group,
                              # IP address, or hostname. Each
                              # exception_entry must be on its own line.

Exception_type:
Exception:

```

在 `id` 和 `stop_entry` 伪指令中，仅支持在 IP 地址的最后一位或主机名的第一位使用通配符（*）。`exception_entry` 中不支持通配符。在所有字段中，任何 LDAP 条目都不支持通配符。

支持多个策略，如果策略间发生冲突，值 `false` 始终具有优先级。换句话说，任何策略中只要有一个拒绝，就会阻拦访问。配置和策略文件中所列的策略顺序是不相关的，因此不要建立优先级。

要获得一组策略示例，请参阅配置文件目录中的 `pacpolicy.conf` 文件。

注：嵌套的组不能从父组继承策略。只有那些在某个组上执行的策略，该组才是此策略的显式成员。

创建 `pac_ldap.cred`

在 `/cp_root/server_root/pac/creds` 中创建名为 `pac_ldap.cred` 的纯文本文件。此文件包含与 `admin_dn` 伪指令中的用户名相应的密码，此密码在 `pac.conf` 文件中。

注：要启用匿名绑定，将 `pac.conf` 中的 `admin_dn` 伪指令更改为 `admin_dn:NULL`，并将虚拟字符串放入 `pac_ldap.cred` 文件中。

在第一次读此文件时，PAC 守护程序加密此密码。

要在 Linux 和 UNIX 平台上创建文件 `pac_ldap.cred`，发出下列命令：

```

cd cp_root/server_root/pac/creds
echo "password" > pac_ldap.cred
chown nobody pac_ldap.cred
chgrp nobody pac_ldap.cred
(在 SuSE Linux 上，使用 chgrp nogroup pac_ldap.cred。)

```

要在 Windows 平台上创建此文件，在文本文件中输入密码并在 `server_root\pac\creds\ds\` 目录中存储此文件。

启动和停止 `pacd`

LDAP 授权守护程序作为 `pacd` 进程运行。您可以通过使用提供的脚本，在不中断高速缓存代理的情况下重新启动 LDAP 授权守护程序。按如下所示运行 `pacd` 脚本：

- 在 Linux 和 UNIX 平台上：


```
/usr/sbin/pacd_restart.sh pacd_user_id
```

- 在 Windows 平台上:

```
C:\Program Files\IBM\edge\cp\Bin\pacd_restart.bat CP_install_root
```

注: 通过在 AIX 系统上使用 **stopsrc -ibmproxy** 命令或在 HP-UX、Linux 和 Solaris 系统上使用 **ibmproxy -stop** 命令, pacd 进程可以在高速缓存代理服务器关闭后继续运行。使用 **kill** 命令可以安全关闭 pacd 进程, 按如下所示:

```
kill -15 pacd_process_ID
```

在 HP-UX 上: PAC-LDAP 插件和 pacd 在运行时可能不装入其所有的从属共享库。使用它们之前, 确保系统变量按如下所示设置

```
SHLIB_PATH=/usr/lib:/usr/IBMldap/lib
```

```
PATH=/usr/IBMldap/bin:$PATH
```

```
PATH=/usr/IBMldap/bin
```

/usr/IBMldap/ 是 HP-UX 上的 LDAP 客户机的缺省安装路径。如果 LDAP 客户机安装在不同的位置, 请相应调整 PATH 和 SHLIB_PATH。如果不设置这些变量, 可能会发生以下错误:

- 启用 PAC-LDAP 插件后, 在错误日志中将出现以下消息
"Serverinit Error: server did not load functions
from DLL module /opt/ibm/edge/cp/lib/plugins/pac/libpacwte.sl"
- 尝试启动 /usr/sbin/pacd 时, 将显示以下链接错误
"/usr/lib/dld.sl: 无法查找共享库 libibmldap.sl 的路径
/usr/lib/dld.sl: 没有如此文件或目录
被异常终止"

在 AIX 上: 当用 IBM Tivoli Directory Server 5.2 启动 pacd 时, PAC-LDAP 模块可能无法装入, 导致下列错误:

```
exec(): 0509-036 由于下列错误, 无法装入程序 /usr/sbin/pacd :  
0509-022 无法装入模块 /usr/lib/libpacman.a。  
0509-150 无法装入从属模块 libldap.a。  
0509-022 无法装入模块 libldap.a。
```

要解决此问题, 创建以下符号: `ln -s /usr/lib/libibmldap.a /usr/lib/libldap.a`

第 6 部分 监视高速缓存代理

这一部分提供有关使用日志和服务器活动监视器监视高速缓存代理的说明。

这一部分包含下列章节：

第 131 页的第 30 章，『配置日志记录』

第 137 页的第 31 章，『使用服务器活动监视器』

第 30 章 配置日志记录

要定制日志记录，您可使用“配置和管理”表单，或编辑代理配置文件中的伪指令。您可以设置下列选项：

- 存储日志文件的路径和文件名
- 从访问日志文件中包括和排除信息的过滤器
- 归档或删除日志的维护选项

关于日志

除了事件日志和错误日志之外，高速缓存代理还可创建三种类型的访问日志：

- 访问日志：
 - 访问日志 - 跟踪对高速缓存代理本身的本地管理请求。
 - 高速缓存访问日志 - 跟踪位于高速缓存中的对象的请求。
 - 代理访问日志 - 跟踪来自源服务器代理的请求。
- 事件日志 - 跟踪高速缓存信息消息。
- 错误日志 - 跟踪与高速缓存代理相关的错误消息。

高速缓存代理在每天午夜创建新的日志文件。如果代理在午夜时未在运行，则将在当天第一次启动它时创建新日志。您可以为每个日志文件指定目录和文件名前缀；创建的每个日志文件还包含一个日期后缀，格式为 *.Mmmddyyyy*（例如，*.Apr142000*）。

因为日志可能会使用大量空间，请考虑将您的日志文件存储在一个与操作系统和高速缓存分离的存储设备上，以防止错误。另外，按第 134 页的『维护和归档日志』中的指定配置日志维护例程。

日志文件名和基本选项

要指定代理服务器日志的基本配置，在“配置和管理”表单中，选择**服务器配置 -> 日志记录 -> 日志文件**。指定您要使用的每个日志文件的路径与文件名。每个日志的当前文件名显示在相应的文本框中；如果您尚未指定路径，则将显示缺省路径。

记录到代理日志中的信息不会自动写入系统日志，但是您可配置高速缓存代理以（除了写入它自己的日志或不写入它自己的日志而）写入系统日志。在**日志文件**表单上，选择**将信息记录到系统日志**复选框。请注意，系统日志必须在此选项被选择之前已创建。

要指定将代理服务器日志信息仅写入系统日志，您必须编辑代理配置文件；请参阅第 207 页的『LogToSyslog — 指定是否将访问信息发送到系统日志（仅 Linux 和 UNIX）』的参考章节。

相关的配置文件伪指令

要使用代理配置文件设置日志，请参阅第 155 页的附录 B，『配置文件伪指令』中的参考章节以获取关于下列伪指令的信息：

- 第 157 页的『AccessLog - 命名访问日志文件的路径』
- 第 167 页的『CacheAccessLog - 指定高速缓存访问日志文件的路径』
- 第 188 页的『ErrorLog - 指定记录服务器错误的文件』
- 第 190 页的『EventLog - 指定事件日志文件的路径』
- 第 207 页的『LogToSyslog - 指定是否将访问信息发送到系统日志（仅 Linux 和 UNIX）』
- 第 229 页的『ProxyAccessLog - 命名代理访问日志文件的路径』

访问日志过滤器

访问日志记录主机、代理和高速缓存的活动。对于您的代理接收的每个访问请求，在相应的访问日志中的一个条目包含下列信息：

- 请求什么
- 何时请求它
- 谁请求它
- 请求的方法
- 您的服务器为响应请求发送的文件类型
- 返回码，它表明请求是否成功
- 发送的数据大小

访问错误被记录到服务器的错误日志中。

控制所记录的内容的原因

限制所记录内容的原因如下：

- 为了减少日志大小

一个繁忙的服务器的日志文件可能变得太大以至于填满服务器的磁盘空间。缺省情况下，将记录所有访问请求，这意味着不仅要为 HTML 页面建立日志条目，而且要为该页面包含的每个图像建立日志条目。仅包括有意义的访问请求可以显著减少日志中的条目数。例如，您可能需要配置您的访问日志以便它们包含 HTML 页面访问请求而非 GIF 图像请求的日志条目。

- 为了收集目标信息

例如，如果您对谁从公司外部访问您的服务器感兴趣，则可以过滤掉从公司内部的 IP 地址发出的访问请求。如果您对查明一个特定 Web 站点的观众的数量感兴趣，则您可以创建一个仅显示对该 URL 的访问请求数的访问日志。

从访问日志中排除的信息不会记录到任何访问报告中，并且不可在将来使用。因此，如果您不确定您需要跟踪多少访问信息，请保守地应用排除过滤器，直至您获得关于监视您的服务器的经验。

配置访问日志过滤器

访问日志条目可基于下列任何属性来过滤：

- URL（文件或目录）
- IP 地址或主机名

- 用户代理程序
- 方法
- MIME 类型
- 返回码

要指定您的过滤器，在“配置和管理”表单中选择**服务器配置** -> **日志记录** -> **访问日志排除**。仅指定您要的排除。您不需要使用所有类别。

- 在标题为**不要记录对访问日志中下列目录或文件的请求**的部分中，列出您要为其排除日志条目的 URL 路径。
- 在标题为**不要记录来自下列用户代理程序的请求**的部分中，列出您要为其排除日志条目的代用品代理程序。
- 在标题为**不要记录来自下列主机名或 IP 地址的请求**的部分中，列出您要为其排除日志条目的主机名或 IP 地址。
- 在标题为**不要按下列方法记录请求**的部分中，选取您要为其排除日志条目的任何方法的框。
- 在标题为**不要记录对下列 MIME 类型的文件的请求**的部分中，选取您要为其排除日志条目的任何 MIME 类型的框。

注：此伪指令仅影响代理访问日志。它不可能过滤列出这些按其 MIME 类型高速缓存的对象的日志。使用 `AccessLogExcludeURL` 来执行此操作。

- 在标题为**不要记录带有下列返回码的请求**的部分中，选取您要为其排除日志条目的请求返回码的框。

单击**提交**。

相关的配置文件伪指令

要使用代理配置文件设置访问日志过滤器，请参阅第 155 页的附录 B，『配置文件伪指令』中的参考章节以获得关于下列伪指令的信息：

- 第 158 页的『`AccessLogExcludeMethod` - 抑制通过指定方法请求的文件或目录的日志条目』
- 第 158 页的『`AccessLogExcludeMimeType` - 禁止特定 MIME 类型的代理访问日志条目』
- 第 159 页的『`AccessLogExcludeReturnCode` - 抑制特定返回码的日志条目』
- 第 159 页的『`AccessLogExcludeURL` - 抑制指定文件或目录的日志条目』
- 第 159 页的『`AccessLogExcludeUserAgent` - 抑制来自特定浏览器的日志条目』
- 第 214 页的『`NoLog` - 抑制匹配模板的特定的主机或域的日志条目』

缺省日志设置

- **缺省路径**

在高速缓存代理缺省配置中启用所有日志。它们存储在安装目录的 `logs/` 子目录。缺省路径如下：

- 本地（管理）访问日志：
 - Linux 和 UNIX: `/opt/ibm/edge/cp/server_root/logs/local`

- Windows: *drive:\Program Files\IBM\edge\cp\logs\local*
- 高速缓存访问日志:
 - Linux 和 UNIX: */opt/ibm/edge/cp/server_root/logs/cache*
 - Windows: *drive:\Program Files\IBM\edge\cp\logs\cache*
- 代理访问日志:
 - Linux 和 UNIX: */opt/ibm/edge/cp/server_root/logs/proxy*
 - Windows: *drive:\Program Files\IBM\edge\cp\logs\proxy*
- 错误日志:
 - Linux 和 UNIX: */opt/ibm/edge/cp/server_root/logs/error*
 - Windows: *drive:\Program Files\IBM\edge\cp\logs\error*
- 事件日志:
 - Linux 和 UNIX: */opt/ibm/edge/cp/server_root/logs/event*
 - Windows: *drive:\Program Files\IBM\edge\cp\logs\event*

每个日志文件名都是基名和以 *.Mmmddyyyy* 格式的日期后缀的组合，例如，*proxy.Feb292000*。

- **缺省格式**

缺省情况下，日志以公共文件格式存储。组合的日志格式也可用，并可通过将下列行添加到代理配置文件（*ibmproxy.conf*）中来设置。

```
LogFileFormat combined
```

组合的日志格式与公共格式相似，但是已添加显示引用者、用户代理和 *cookie* 信息的字段。本地时间格式是缺省时间格式。

- **缺省内容**

缺省情况下，所有的访问请求都记录在适当的访问日志中，并且不在系统日志中记录任何访问信息。错误日志信息仅写入错误日志，而事件日志信息仅写入事件日志。

- **缺省维护**

在缺省配置中，日志不被归档或删除。

维护和归档日志

高速缓存代理使用一个插件来管理日志。要获取进一步的信息，请参阅第 155 页的附录 B，『配置文件伪指令』中的参考页面以获得配置文件伪指令第 212 页的『Midnight - 指定用于归档日志的 API 插件』。

您可以指定如何归档和除去每日日志。基本选项为：

- 压缩和除去比指定时效长的日志。
- 在日志达到某一指定的时效或在该日志类别达到某一指定的总的大小后除去日志。
- 每晚午夜运行您自己的程序以维护和归档日志。

缺省情况下，当天和前一天的日志不会被任何维护代理删除。当天的所有日志和前一天的高速缓存访问日志不会被任何维护代理压缩。

要配置日志维护，在“配置和管理”表单中选择**服务器配置** -> **日志记录** -> **日志归档**。在此表单中，使用下拉框指定维护方法。

- 如果您选择**清除**，则设置时效、文件大小或两者都设置，以用于确定删除哪些日志。当根据时效和大小来清除文件时，大于最大时效的文件即被清除，而不必等到文件大于最大大小。当根据大小清除文件时，首先删除更老的日志。
- 如果您选择**压缩**，则设置要压缩的日志时效，以及用于压缩日志文件的命令（包括所有参数）。同样要设置日志的最大时效。在压缩日志后，维护代理删除比最大时效长的压缩日志。

相关的配置文件伪指令

要使用代理配置文件配置日志归档，请参阅第 155 页的附录 B，『配置文件伪指令』中的参考页面以获得下列伪指令：

- 第 177 页的『CompressAge - 指定何时压缩日志』
- 第 178 页的『CompressDeleteAge - 指定何时删除日志』
- 第 177 页的『CompressCommand - 指定压缩命令和参数』
- 第 206 页的『LogArchive - 指定日志归档的行为』
- 第 212 页的『Midnight - 指定用于归档日志的 API 插件』
- 第 233 页的『PurgeAge - 指定日志的时效』
- 第 233 页的『PurgeSize - 指定日志压缩文档的大小限制』。

日志文件方案

下例显示您可如何定制日志记录以满足您的需要。假设您刚购买并安装了高速缓存代理。您想要设置您的服务器以按下列要求记录访问和错误信息：

- 日志必须带有本地时间戳记，且必须使用公共日志文件格式。
- 当访问日志超过 30 天或当日志达到 25 MB 的总的大小时，必须清除它们。
- 下列请求类型不得记录到访问日志中：
 - 对 GIF 图像的请求
 - 来自其 IP 地址匹配模式 130.128.*.* 的主机的请求
 - 重定向请求（这些请求产生一个介于 300 和 399 之间的返回码）

要配置高速缓存代理以根据这些标准保留日志，在“配置和管理”表单中选择**服务器配置** -> **日志记录**。

1. 可选地选择“日志文件”表单以指定访问日志文件的路径。（提供缺省路径。）
2. 使用日志归档表单指定如何归档文件：
 - 指定**清除**作为归档方法。
 - 在当使用清除下，按如下所示填充字段：
 - 删除超过 **30** 天的日志
 - 删除大于 **25 MB** 的日志
3. 使用访问日志排除表单按如下所示过滤日志条目：
 - 在不要记录来自下列主机名或 IP 地址的请求列表中，在排除的主机字段中添加 130.128.*.*。
 - 在不要记录对下列 **MIME** 类型的文件的请求下，选择 **image/gif** 复选框。

- 在不要记录带有下列返回码的请求下，选择（**3xx**）重定向复选框。

按照这些指导将在代理配置文件中产生下列行：

```
LogArchive purge
PurgeAge 30
PurgeSize 25
AccessLogExcludeURL *.gif
NoLog 130.128.*.*
AccessLogExcludeReturnCode 300
```

第 31 章 使用服务器活动监视器

高速缓存代理的服务器活动监视器显示服务器和网络性能统计信息、服务器和网络状态，以及访问日志条目。监视器可远程使用，并且不需要位于运行代理服务器的同一台机器上。服务器活动监视器在缺省情况下启用，并且无需配置。

有两种方法可以打开服务器活动监视器：

- 在任何连接的 Web 浏览器中，在指明的地方使用您的服务器的全名输入下列 URL。
`http://your.server.name/Usage/Initial`
- 在“配置和管理”表单中，选择**服务器活动监视器**。

和配置客户机的其它表单不同的是，此类别中的表单不设置服务器的配置，但会显示关于服务器使用情况的数据。这些表单提供的信息比单个控制台窗口中显示的多很多。

下列部分显示了**服务器活动监视器**提供的信息类型，并建议了如何使用这些信息来调整性能。

一些**服务器活动监视器**页面是可用的：

- 活动统计信息
- 网络统计信息
- 访问统计信息
- 代理访问统计信息
- 高速缓存统计信息
- 高速缓存刷新摘要

每个页面都具有一个**刷新**按钮，它可用于更新信息。

活动统计信息

表 4 显示了**活动统计信息**页面的一个示例。

表 4. 活动统计信息

活动统计信息	
连接	1 活动, 431 最大值
响应时间	不可用
吞吐量	0 连接/秒
今天处理的请求数	0
处理的总请求数	114
请求错误数	3

这些服务器活动统计信息可用于根据访问请求数、响应时间、吞吐量、今天处理的请求数、处理的总请求数和错误数监视服务器流量。下列配置更改影响活动页面上的统计信息。

- **活动线程数** - 这指定了多少线程用于服务器请求。您可以根据您拥有多少内存来增加或减少可用线程数。要更改活动线程数，在“配置和管理”表单中选择**服务器配置 -> 服务器管理 -> 性能**，或编辑配置文件中的 `MaxActiveThreads` 伪指令。（请参阅第 209 页的『`MaxActiveThreads` - 指定有效线程的最大数』。）
- **持久连接** - 代理是否允许与客户机持久连接可能会影响网络吞吐量。要在“配置和管理”表单中更改此设置，选择**代理配置 -> 代理性能**启用或禁用持久连接，并选择**服务器配置 -> 服务器管理**来定义如何维护连接。要使用配置文件来改变这些设置，请参阅下列伪指令的参考章节：
 - 第 210 页的『`MaxPersistRequest` - 指定一个持久连接上接收的最大请求数』
 - 第 218 页的『`PersistTimeout` - 指定等待客户机发送另一个请求的时间』
 - 第 231 页的『`ProxyPersistence` - 允许持久连接』

网络统计信息

表 5 显示**网络统计信息**页面的一个示例。

表 5. 网络统计信息

网络统计信息	
出网数据	1 千字节/秒
入网数据:	1 千字节/秒
节省的带宽:	3 千字节 (0 千字节/秒)
今天节省的带宽:	0 千字节 (0 千字节/秒)

网络统计信息表单提供了关于代理当前运行所在的网络的信息，包括发送和接收的字节的数据速度。

访问统计信息

访问统计信息页面显示了访问日志中的 20 个最新条目。此页面显示了代理访问日志（以黑字体）以及高速缓存访问日志（以蓝字体）中的最新条目。您可通过定制记录的内容来定制显示的内容。要获取关于访问日志统计信息的更多信息，请参阅第 132 页的『**访问日志过滤器**』。

代理访问统计信息

代理访问统计信息表单提供了关于代理活动的信息，如请求哪些 URL 以及它们是否从高速缓存服务。URL 后面是给予客户机的返回码和文件大小（以字节计）。下列设置可以改进代理访问统计信息：

- 使用自动高速缓存刷新来增加请求的文档位于高速缓存中的可能性。要获得详细信息，请参阅第 83 页的第 20 章，『**为自动刷新和预装入配置高速缓存代理程序**』。
- 增加高速缓存文件的最小持续时间。要获得详细信息，请参阅第 79 页的『**配置高速缓存新鲜度**』。
- 不要高速缓存从您的本地域服务的文件。虽然此设置趋向减少从高速缓存服务的请求数，如果从本地内部网服务文件与从高速缓存服务文件一样快（在某些情况下从本地内部网服务更快），则不会损失任何性能。要获得详细信息，请参阅第 73 页的第 18 章，『**控制高速缓存哪些内容**』。

高速缓存统计信息

如果启用高速缓存，则**高速缓存统计信息**页面显示最近的高速缓存访问信息。它提供了关于高速缓存和索引的信息，包括下列内容：

- 高速缓存当前可操作还是正在服务器启动时重新索引
- 垃圾收集是否正在运行
- 高速缓存命中率

许多高速缓存配置选项更改高速缓存统计信息结果（请参阅第 63 页的第 4 部分，『配置代理服务器高速缓存』）。

高速缓存刷新摘要

如果配置高速缓存代理程序以在高速缓存中预装入文件，则**高速缓存刷新摘要**页面显示关于高速缓存代理程序的最新运行的信息。高速缓存代理程序必须至少运行了一次以显示任何信息。要更改高速缓存刷新代理程序工作的方式，考虑下列内容：

- 如果您的内部网上的大部分流量不是到本地 Web 站点的，请考虑禁用本地域高速缓存。要获得详细信息，请参阅第 73 页的第 18 章，『控制高速缓存哪些内容』。
- 如果许多客户机请求一个不在高速缓存访问日志中出现的页面，则您可以手工配置装入 URL。请参阅第 86 页的『相关代理配置文件伪指令』以获得说明。
- 调整要预装入的流行 URL 数。请参阅第 86 页的『相关代理配置文件伪指令』以获得说明。
- 指定高速缓存代理程序可运行的最大时间。请参阅第 86 页的『相关代理配置文件伪指令』以获得说明。

附录 A. 使用高速缓存代理命令

本附录提供了代理服务器命令的参考。

cgiparse 命令

目的

使用 **cgiparse** 命令来语法分析 CGI 脚本的 `QUERY_STRING` 环境变量。如果未设置 `QUERY_STRING` 环境变量，则该命令会从它的标准输入读 `CONTENT_LENGTH` 字符。返回的所有输出被写至它的标准输出。

格式

```
cgiparse -Flag [Modifier]
```

参数

标志，它们的等价一位字符 (`-k -f -v -r -i -s -p -c -q -P`) 和它们的功能，包括：

-keywords | -k

为关键字语法分析 `QUERY_STRING`。关键字解码并写至标准输出，每行一个关键字。

-form | -f

将 `QUERY_STRING` 语法分析为表单请求。返回一个字符串，当外壳程序评估它时，该字符串将使用后跟字段名称的前缀 `FORM_` 来设置外壳变量。字段值为变量的内容。

-value field-name | -v field-name

将 `QUERY_STRING` 语法分析为表单请求。仅返回 `field-name` 的值。

-read | -r

从标准输入读 `CONTENT_LENGTH` 字符，并将它们写入标准输出。

-init | -i

如果未设置 `QUERY_STRING`，则读标准输入的值并返回一个将 `QUERY_STRING` 设置为该值的 `SET` 语句。这可以与 `GET` 和 `POST` 方法一起使用。典型使用为：

```
eval 'cgiparse -init'
```

这将设置 `QUERY_STRING` 环境变量，无论是使用 `GET` 方法还是使用 `POST` 方法。

当使用 `GET` 方法时，可在同一脚本中多次调用 **cgiparse**，但是如果使用 `POST` 方法，则它应该只能被调用一次。使用 `POST` 方法，在读标准输入之后，下一个 **cgiparse** 将发现标准输入为空，并将不确定地等待。

-sep separator | -s separator

指定用于分隔多个值的字符串。如果您使用 **-value** 标志，则缺省分隔符为换行。如果您使用 **-form** 标志，则缺省分隔符为逗号 (,)。

-prefix prefix | -p prefix

与 **-POST** 和 **-form** 一起使用，指定在创建环境变量名称时所使用的后缀。缺省值为 `"FORM_"`。

-count | -c

与 **-keywords**、**-form** 和 **-value** 一起使用，返回与这些标志相关的项目计数。

-keywords | -k

返回关键字的个数。

-form | -f

返回唯一字段的个数（多个值计为一个）。

-value field-name | -v field-name

返回 *field-name* 的值的个数（如果没有名为 *field-name* 的字段，则输出为 0）。

-number

与 **-keywords**、**-form** 和 **-value** 一起使用，返回与这些标志相关的指定发生。

-keywords

返回第 *n* 个关键字。（例如，**-2 -keywords** 输出第二个关键字。）

-form

返回第 *n* 个字段的所有值。（例如 **-2 -form** 输出第二个的所有值。）

-value field-name

返回字段 *field-name* 的多个值的第 *n* 个值。（例如，**-2 -value -whatsit** 输出 **whatsit** 字段的第二个值）。

-quiet | -q

抑制所有错误消息。（非零退出状态仍表明错误。）

-POST | -P

来自标准输入（或如果意图使用一个文件名，则为标准输入文件）的信息直接解码，并语法分析至外壳变量中，不使用 `QUERY_STRING`。**-POST** 等价于 **-init** 和 **-form** 选项的连续使用。

示例

下列示例忽略一个事实，就是实际情况下，`QUERY_STRING` 已由服务器设置。在下列示例中，`$` 是 Bourne shell 提示符。

- 关键字搜索

```
$ QUERY_STRING="is+2%2B2+really+four%3F"
$ export QUERY_STRING
$ cgifparse -keywords
is
2+2
really
four?
$
```

- 语法分析所有表单字段

```
$ export QUERY_STRING="name1=Value1&name2=Value2%3f+That%27s+right%21";
$ cgifparse -form
FORM_name1='Value1'; FORM_name2='Value2? That'\s right!'
$ eval `cgifparse -form`
$ set | grep FORM
FORM_name1="Value1"
FORM_name2="Value2? That's right!"
$
```

- 仅抽取一个字段值

```
$ QUERY_STRING="name1=value1&name2=Second+value%3F+That'\s%27s
$ cgifparse -value name1
value1
$ cgifparse -value name2
Second value? That's right!
$
```

结果

- 0 成功
- 1 非法命令行
- 2 未正确设置环境变量
- 3 无法获取所请求的信息（例如，没有这样的字段或者当请求表单字段值时，`QUERY_STRING` 包含关键字）

注：当您接收这些错误代码之一时，您可能还会收到其它信息性消息。取决于发出的命令，这些消息会有所改变。

cgiutils 命令

目的

在非语法分析头程序中使用 **cgiutils** 命令以产生完全的 HTTP 1.0 响应。

注: 如果您要提供您自己的非语法分析头 (nph) 程序以特别地返回您自己的返回值, 则该程序的名称必须以 **nph-** 开头。这阻止了服务器头使用标准服务器返回值覆盖您的返回值。

格式

```
cgiutils -Flag [Modifier]
```

如果 *Modifier* 包含空格, 则用引号 (") 把它括起来。

参数

-version

返回版本信息。

-nodate

不返回 **Date:** 头。

-noel

不在头后返回空白行。如果您在初始头行之后想要其它 MIME 头, 则这是有用的。

-status nnn

返回带有状态码 *nnn* 的完全 HTTP 响应, 而不仅是一组 HTTP 头。如果您仅想要 **Expires:** 头, 则不要使用此标志。

-reason explanation

指定 HTTP 响应的原因行。您仅可将此标志与 **-status nnn** 标志一起使用。

-ct [type/subtype]

指定 MIME Content-Type 头。此示例指定一个 text/html 的 MIME 内容类型:

```
cgiutils -ct text/html
```

如果您省略 *type/subtype*, 则 MIME 内容类型设置为缺省值 text/plain。此示例设置 MIME 内容类型为 text/plain。

```
cgiutils -ct
```

-ce encoding

指定 MIME Content-Encoding 头。例如:

```
cgiutils -ce x-compress
```

-cl language-code

指定 MIME Content-Language 头。例如:

```
cgiutils -cl en_UK
```

-length nnn

指定 MIME Content-Length 头。

-expires *Time-Spec*

指定 MIME **Expires:** 头。此标志以 days、hours、minutes 和 seconds 的任意组合指定存活时间（文档的到期日期）。这是认为文档有效的时间长度。例如：

```
cgiutils -expires 2 days 12 hours
```

cgiutils 命令将您指定的时间加上当前格林威治标准时间以确定到期日期。到期日期以 HTTP 格式放在 **Expires:** 头中。

-expires now

产生一个与 **Date:** 头匹配的 **Expires:** 头。

-uri *URI*

为返回的文档指定统一资源定位器（URI）。可认为 URI 与 URL 相同。

-extra *xxx: yyy*

指定一个额外的头，它无法另外为 **cgiutils** 指定。

示例

- 示例计算 **Expires:** 头的到期日期。

```
cgiutils -expires "1 year 3 months 2 weeks 4 days 12 hours 30 mins"
```

- 下列示例指定了一个状态代码和原因，并将 **Expires:** 头设置为与 **Date:** 头相等。

```
cgiutils -status 200 -reason "Virtual doc follows" -expires now
```

这可能会产生与这些相似的头：

```
HTTP/1.0 200 Virtual doc follows
MIME-Version: 1.0
Server: IBM-ICS
Date: Tue, 05 Jan 1996 03:43:46 GMT
Expires: Tue, 05 Jan 1996 03:43:46 GM
```

cgiutils 命令会自动产生 **Server:** 头，因为它在 CGI 环境中可用。**Date:** 字段也会自动生成，除非指定了 **-nodate** 标志。

这会在输出后包括一个空白行以标记 MIME 头部分的末尾。如果您想自己在后面再加一些头，则使用 **-noel**（非空白行）标志，如下一示例所示。

- 如果您不想要在头行后有空白行，则使用 **-noel** 标志：

```
cgiutils -noel -expires "2 days" -nodate
```

```
HTTP/1.0 200 Virtual doc follows
MIME-Version: 1.0
Server: IBM-ICS
Expires: Tue, 07 Jan 1996 03:43:46 GMT
```

htadm 命令

目的

使用 **htadm** 命令以控制您的服务器密码文件。您的服务器使用密码文件以控制对您的文件的访问。您可以将一个用户名添加到密码文件，从密码文件删除一个用户，验证用户的密码，以及创建一个空的密码文件。您还可以通过先删除用户的密码然后创建一个新密码来更改用户的密码。

注：当您使用 **htadm** 来添加用户、更改密码或检查密码时，您必须在命令行上输入密码。因为命令会尽可能快地销毁来自命令行的密码，所以您不太可能通过查看在机器上列出的进程（例如，使用 **ps** 命令）来看见用户的密码。

格式

```
htadm -Flag [Modifier]
```

参数

-adduser *password-file* *user-name* [*password* [*real-name*]]

将用户和密码添加到密码文件中。如果您输入仅带 *password-file* 的命令，则会提示您输入另一个参数。

password-file

您要添加用户的密码文件的路径和名称。

user-name

您要添加的用户的名称。

对用户名仅使用字母和数字字符；不要使用特殊字符。

如果在密码文件中已有一个同名用户，则该命令将失败。

password

您要为用户名定义的密码。

密码最长可为 32 个字符。对密码仅使用字母和数字字符；不要使用特殊字符。

注：一些浏览器无法读和发送长于 8 个字符的密码。由于此限制，如果您定义一个长于 8 个字符的密码，则服务器会识别完整密码或仅识别密码的前 8 位作为有效字符。

real-name

您要用来标识您当前添加的用户名的注释或名称。您输入的任何内容都会被写入密码文件中。

-deluser *password-file* [*user-name*]

从密码文件中删除用户。如果您输入仅带 *password-file* 的命令，则会提示您输入 *user-name* 参数。

password-file

您要从中删除用户的密码文件的路径和名称。

user-name

您要删除的用户的名称。如果您指定的用户名在密码文件中不存在，则命令失败。

-passwd *password-file* [*user-name* [*password*]]

更改已在密码文件中定义的用户名的密码。如果您输入仅带 *password-file* 的命令，则会提示您输入另一个参数。

password-file

包含您要更改密码的用户名的密码文件的路径和名称。

user-name

您要更改密码的用户名。如果您指定的用户名在密码文件中不存在，则命令失败。

password

您要为用户名定义的新密码。

密码最长可为 32 个字符。对密码仅使用字母和数字字符；不要使用特殊字符。

注：一些浏览器无法读和发送长于 8 个字符的密码。由于此限制，如果您定义一个长于 8 个字符的密码，则服务器会识别完整密码或仅识别密码的前 8 位作为有效字符。

-check *password-file* [*user-name* [*password*]]

为已在密码文件中定义的用户名验证密码并让您知道它是否正确。如果您输入仅带 *password-file* 的命令，则会提示您输入另一个参数。

password-file

包含您要验证其密码的用户名的密码文件的路径和名称。

user-name

您想要验证其密码的用户名。如果您指定的用户名在密码文件中不存在，则命令失败。

password

您要验证的密码。如果您输入的密码是一个已为用户名定义的密码，则命令会将 `Correct` 写入标准输出并完成后返回一个 0 返回码。如果您输入的密码不是为用户名定义的密码，则命令会将 `Incorrect` 写入标准输出。

-create *password-file*

创建一个空密码文件。

password-file

您想要创建的密码文件的路径和名称。

示例

- 要将用户添加到密码文件:

– Linux 和 UNIX 系统:

```
htadm -adduser /opt/ibm/edge/cp/server_root/protect/heroes.pwd  
clark superman "Clark Kent"
```

– Windows 系统:

```
htadm -adduser "C:\Program Files\IBM\edge\cp\server_root\protect\  
heroes.pwd" clark superman "Clark Kent"
```

注：`htadm` 命令必须在一行上。出于可读性，此处它显示在多行上。在一行上输入实际的命令，且在 `clark` 和 `superman` 之间至少有一个空格。

- 要从密码文件删除用户:

- Linux 和 UNIX 系统:

```
htadm -deluser /opt/ibm/edge/cp/server_root/protect/  
heroes.pwd clark superman "Clark Kent"
```

- Windows 系统:

```
htadm -deluser "C:\Program Files\IBM\edge\cp\server_root\protect\  
heroes.pwd" clark superman "Clark Kent"
```

htcformat 命令

目的

使用 **htcformat** 命令以准备一个裸设备或文件来保存代理高速缓存。此格式化命令必须在指定与代理高速缓存一起使用的设备之前使用。

设备路径必须指定裸设备。请参阅您的文件系统文档，以获得关于如何访问裸设备的详细信息。示例可在第 63 页的第 4 部分，『配置代理服务器高速缓存』中找到。

注：Linux 2.2 内核不支持到裸设备的高速缓存。在 Linux 平台上，仅文件和内存可用于高速缓存存储。

高速缓存代理高速缓存的最小大小为 16392 KB，它为 2049 块。

格式

```
htcformat device [-blocksize <block size>] [-blocks number of blocks]
htcformat -file filepath [-blocksize block size] -blocks number of blocks
```

参数

-blocksize

这设置高速缓存设备介质中的块大小。块大小以字节计。缺省值为 8192 且应该用于所有情况。

-blocks

在设备上或在文件中创建的块数。当格式化文件时，需要此自变量来指定文件大小。此自变量也可用于限制用于高速缓存存储的特定设备或分区数量。如果未指定块自变量，将会在分区上创建尽可能多的块。

-file

格式化文件而非存储设备。

用法

高速缓存系统另外将高速缓存文件或设备分离到容器中以用于索引和垃圾收集。容器大小设置为特定数量的块；容器大小不能配置。为使垃圾收集运行，最少需要两个容器；最小高速缓存大小为 16392 KB。

htcformat 命令拒绝产生少于两个容器的高速缓存设备的格式化请求。

示例

以下示例在 Solaris 上格式化一个称为 c0t0d0s0 的磁盘分区。

```
htcformat /dev/rdisk/c0t0d0s0
```

以下示例在 AIX 上格式化一个称为 lv02 的磁盘分区。

```
htcformat /dev/r1v02
```

以下示例在 Windows 上格式化一个称为 d: 的磁盘分区。

```
htcformat \\.\d:
```

以下示例将一个名为 filecache 的文件格式化为约 1 GB 大小。

```
htcformat -file /opt/ibm/edge/cp/filecache -blocks 131072
```

ibmproxy 命令

目的

使用 **ibmproxy** 命令以启动服务器。

您可使用服务器配置文件中的伪指令来设置所有这些标志（除了 **-r** 之外）。

通常要创建一个名为 **README** 的文件，此文件包含对此目录不熟悉的任何人可阅读的说明和注意事项。缺省情况下，**ibmproxy** 命令在目录的超文本版本中嵌入任何自述文件。自述文件说明也可使用 **DirReadme** 配置伪指令来设置。

格式

```
ibmproxy [-Flag [-Flag [-Flag..]]]
```

参数

-nobg

将服务器作为前台进程（而非后台进程）运行。缺省是作为后台进程运行。

-nosnmp

关闭 SNMP 支持。

-p *port-number*

在此端口号上侦听。缺省端口号为 80。此标志覆盖在配置文件中指定的端口伪指令。要使用缺省值或在配置文件中指定的值，则省略此标志。

-r *configuration-file*

指定要用作配置文件的文件。如果您想用不同于缺省配置文件的配置文件来启动服务器，则您必须使用此标志。这允许使用多个配置文件。

-restart

重新启动当前正在运行的服务器。**ibmproxy** 命令从 **PidFile** 获取运行的服务器的进程号，并将此进程号发送给 **HangUP** 信号（**HUP**）。然后它重新装入它的配置文件并重新打开它的日志文件。要避免崩溃，不要同时运行两个使用同一 **PidFile**、日志文件和代理高速缓存的服务器实例。

因为 **http** 守护程序必须读服务器当前使用的配置文件以便访问 **PidFile**，您在重新启动时必须指定同一配置文件。如果您在启动服务器时使用 **-r** 标志和一个特定的配置文件，则您必须指定此标志和同一文件以及 **-restart**。

-snmp

打开 SNMP 支持。

-unload

在 AIX 上，这将卸载透明代理内核扩展。在 Linux 上，这将除去关联的防火墙规则。

信号处理选项也仅存在于 Linux 和 UNIX 平台上。在 Linux 和 UNIX 平台上，下列选项是可用的。

SIGTERM

ibmproxy 命令在完成时停止并退出。您可以使用 **SIGKILL** 或 **CANCEL** 以立即终止。

SIGHUP

如果正在运行，则 **ibmproxy** 命令重新启动，重新装入配置文件，并继续处理。

示例

- 要使用 `/usr/etc/ibmproxy.conf` 配置文件而非缺省的 `/etc/ibmproxy.conf` 在端口 8080 上启动服务器，输入：

```
ibmproxy -p 8080 -r /usr/etc/ibmproxy.conf
```

- 在 AIX 上，要使用系统资源控制器来用缺省配置文件启动，输入：

```
startsrc -s ibmproxy
```

如果缺省配置文件不存在，则 **ibmproxy** 命令将导出 `/Public` 目录树。此树可包含到其它目录树的软链接。

附录 B. 配置文件伪指令

本附录描述包含在 `ibmproxy.conf` 配置文件中的伪指令。

- 在 **Linux** 和 **UNIX** 系统上。这些伪指令位于 `/etc/` 目录中的 `ibmproxy.conf` 配置文件。
- 在 **Windows** 系统。这些伪指令通常位于 `C:\Program Files\IBM\edge\cp\`。

如果通过编辑 `ibmproxy.conf` 文件来配置服务器，则将此信息用作参考。如果您使用“配置和管理”表单，则不需要参阅本章。

伪指令按字母顺序列出。

重新启动时不更改的伪指令

有些伪指令在重新启动服务器时不会刷新。如果在服务器正在运行时更改了下列伪指令，则您必须手工停止，然后再次手工启动服务器。（请参阅第 15 页的第 5 章，『启动和停止高速缓存代理』）。

表 6. 重新启动时不刷新伪指令

伪指令组	伪指令
CGI	DisinheritEnv, InheritEnv
Caching	Caching
记录	AccessLog, CacheAccessLog, ErrorLog, ProxyAccessLog, ServerRoot
网络访问	BindSpecific, Hostname, ListenBacklog, Port
性能	MaxActiveThreads
RTSP	所有 RTSP 伪指令
SSL	所有 SSL 伪指令
Linux 和 UNIX 进程控制	GroupId, UserId
杂项	TransparentProxy

伪指令概述

此附录提供下列关于每个伪指令的信息：

- 带有伪指令名称和简要描述的标题
- 使用说明
- 伪指令的格式按照一般语法：

DirectiveName value

- 在适当的地方，伪指令可能的设置示例在配置文件中

注：带有 Windows 特定路径的伪指令的示例有时包含 `server_root`，它是安装时所选的服务器的根目录。

- 伪指令的缺省值或值

这些是缺省配置文件中编码的原始值。仅更改配置文件中您想要与缺省不同的部分。没有已初始编码的缺省值的伪指令显示在注释标记（#）之后的文件中。如果您要指定伪指令的值，则除去注释标记并将值添加到配置文件中的行。

可接受的值

下表包含配置文件中所接受的值：

- 在一些伪指令的参考信息中，值部分包含请求、路径名或主机名的模板。除非另有所释，您可在模板中使用星号字符（*）。对于要匹配的模板，星号可由任何字符串或单个字符替换。
- 允许您输入肯定字符串的配置伪指令接受这些值：
 - Yes
 - On
 - OK
 - Enable
- 允许您输入否定字符串的配置伪指令接受这些值：
 - No
 - Off
 - None
 - Disable
- 配置伪指令允许您指定接受下列任何组合的时间量：
 - *hh* - 小时
 - *hh:mm* - 小时和分钟
 - *hh:mm:ss* - 小时、分钟和秒
 - *n* 年 - 以 365 天为一年的年数
 - *n* 月 - 以 30 天为一月的月数
 - *n* 周 - 以 7 天为一周的周数
 - *n* 天 - 以 24 小时为一天的天数
 - *n* 小时 - 以 60 分钟为一小时的小时数
 - *n* 分钟 - 以 60 秒为一分钟的分钟数
 - *n* 秒 - 秒数
 - *n* 两星期 - 以 14 天为一时间间隔的数

所有条目转换为秒并添加在一起。

- 在配置文件中指定的文件名不允许有空白符。空格视为定界符。

配置文件记录的语法

编辑配置文件时，请记住下列要求：

- 每个伪指令必须在新行上开始。
- 值由一个或多个空格分隔。时间间隔字符和制表符之间没有区别。
- 注释的开始由 # 符号表明。忽略从 # 符号到行结束的所有字符。
- 如果需要为伪指令指定数字符号或空白，则将反斜杠字符（\）用作为它之前的转义字符。转义字符表明下一字符将解释为字符，而不是命令；例如，如果在行上查找

到 \#，则服务器将其解释为 # 符号（而不作为注释的开头）并继续读取字符。如果在行上查找到字符 \，则服务器将其解释为空白（而不作为值定界符）并继续读取字符以构建值。

高速缓存代理伪指令

下列高速缓存代理伪指令。

AcceptAnything - 服务所有文件

即使文件的 MIME 类型与客户机发送的 ACCEPT: 头不匹配，也使用此伪指令将文件供应到客户机。如果伪指令设置为 OFF，则不显示 MIME 类型与客户机可接受类型不同的文件。相反，将显示错误页面。

格式

```
AcceptAnything {off | on}
```

示例

```
AcceptAnything off
```

缺省值

```
AcceptAnything on
```

AccessLog - 命名访问日志文件的路径

使用此伪指令指定您要服务器记录访问统计信息的目录与文件。缺省情况下，每次客户机为存储在本地服务器的数据向服务器发送请求时，服务器就会向此日志写一个条目。通常，这些条目包含仅从配置客户机发出的请求，或将高速缓存代理机器用作源服务器时的访问。此日志不包含代理或高速缓存访问信息。

使用 NoLog 伪指令以指定您不要记录其请求的客户机。要获取 NoLog 伪指令的描述，请参阅第 214 页的『NoLog - 抑制匹配模板的特定的主机或域的日志条目』。

如果服务器运行，则每天服务器在午夜启动新的日志文件。否则，服务器在任何一天第一次启动它时启动新的日志文件。创建文件时，服务器使用您指定的文件名并附加日期后缀。日期后缀的格式是 *Mmmddyyyy*，其中 *Mmm* 是月份的前三个字母；*dd* 是日期；而 *yyyy* 是年。

注：如果您更改用户标识、组标识或日志目录路径的服务器缺省值，则创建新的目录并更新目录的许可和所有权。要使服务器能将信息写入用户定义的日志目录，请将此目录的许可设置为 755，并将用户定义的服务器用户标识设置为所有者。例如，如果您将服务器的用户标识从缺省值更改为 *jdoue*，从缺省日志目录更改到 *server_root/account*，则 *server_root/account* 目录必须具有许可 755 并从属于 *jdoue*。

除去旧的日志文件是个好主意，因为它们在您的硬盘驱动器上会占用很大空间。

格式

```
AccessLog /directory_path/logfile_name
```

示例

```
AccessLog /logs/accesslog
```

缺省值

- **Linux** 和 **UNIX** 系统: `AccessLog /opt/ibm/edge/cp/server_root/logs/local`
- **Windows** 系统: `AccessLog drive:\Program Files\IBM\edge\cp\logs\local`

AccessLogExcludeMethod – 抑制通过指定方法请求的文件或目录的日志条目

使用此伪指令防止记录由特定方法所做的请求以访问文件或目录。例如，您可能不想对文件或目录记录 `DELETE` 请求。

在配置文件中，能多次出现该伪指令。如果用一个或多个空格将其分隔，您还可在同一个伪指令上放置多个方法。

格式

```
AccessLogExcludeMethod method [...]
```

示例

```
AccessLogExcludeMethod GET
AccessLogExcludeMethod PUT
AccessLogExcludeMethod POST
AccessLogExcludeMethod DELETE
AccessLogExcludeMethod GET PUT
```

缺省值

无。服务器在访问日志中包含通过所有类型的方法请求的文件和目录。

AccessLogExcludeMimeType – 禁止特定 MIME 类型的代理访问日志条目

使用此伪指令指定不要在代理访问日志中记录指定 `MIME` 类型的目录或文件的访问请求。（`MIME` 类型的示例如 `text/html`、`image/gif` 和 `image/jpeg`。）例如，您可能不想记录对 `GIF` 图像的访问请求。

在配置文件中，能多次出现该伪指令。您还可在同一个伪指令上放置多个 `MIME` 类型，只要以一个或多个空格分隔它们即可。

注：此伪指令仅影响代理访问日志。它不可能过滤列出这些按其 `MIME` 类型高速缓存的对象的日志。使用 `AccessLogExcludeURL` 来执行此操作。

格式

```
AccessLogExcludeMimeType MIME_type [...]
```

示例

```
AccessLogExcludeMimeType image/gif
AccessLogExcludeMimeType text/html
AccessLogExcludeMimeType image/gif text/html
```

缺省值

无。访问日志包含为所有 `MIME` 类型的文件和目录而对服务器所作的请求。

AccessLogExcludeReturnCode – 抑制特定返回码的日志条目

使用此伪指令指定您不要记录的错误代码数指定范围以内的访问请求。这些错误代码号是代理服务器状态代码。您不能指定个别代码。指定 300 就表明您要排除带有重定向返回码（301、302、303 和 304）的访问请求。

在配置文件中，能多次出现该伪指令。您还可在同一个伪指令上放置多个返回码，只要以一个或多个空格分隔它们即可。

格式

```
AccessLogExcludeReturnCode range
```

示例

```
AccessLogExcludeReturnCode 300
```

缺省值

无。无论代码是多少，访问日志包含对服务器的所有请求。

AccessLogExcludeURL – 抑制指定文件或目录的日志条目

使用此伪指令指定您不要记录的与指定 URL 模板相匹配的特定文件或目录访问的请求。例如，您可能不要记录对于 GIF 文件的访问请求，或您可能不要记录对于您服务器上的特殊文件或目录的访问请求。

在配置文件中，能多次出现该伪指令。您还可在同一个伪指令上放置多个条目，只要以一个或多个空格分隔它们即可。

格式

```
AccessLogExcludeURL file_or_type [...]
```

示例

```
AccessLogExcludeURL *.gif  
AccessLogExcludeURL /Freebies/*  
AccessLogExcludeURL *.gif /Freebies/*
```

缺省值

无。服务器记录对于所有文件和目录的访问请求。

AccessLogExcludeUserAgent – 抑制来自特定浏览器的日志条目

使用此伪指令指定您不要记录特定用户代理程序（例如 Internet Explorer 5.0）所作的访问请求。

在配置文件中，能多次出现该伪指令。您还可在同一个伪指令上放置多个条目，只要以一个或多个空格分隔它们即可。

格式

```
AccessLogExcludeUserAgent user_agent [...]
```

示例

```
AccessLogExcludeUserAgent *Mozilla/2.0
AccessLogExcludeUserAgent *MSIE 5*
```

缺省值

缺省情况下，ibmproxy.conf 文件包含下列定义，用于 AccessLogExcludeUserAgent 伪指令：

```
AccessLogExcludeUserAgent IBM_Network_Dispatcher_HTTP_Advisor
AccessLogExcludeUserAgent IBM_Network_Dispatcher_WTE_Advisor
```

上述列出的用户代理是那些为某些 Load Balancer 顾问程序（通常位于高速缓存代理服务服务器前）定义的代理。要通过最小化写入日志的次数来改进性能，这些用户代理不被记录。缺省情况下，服务器日志访问请求由所有其它用户代理完成。

AddBlankIcon – 指定用于对齐目录列表标题的图标的 URL

使用此伪指令指定用于对齐目录列表上标题的图标，这些列表在服务器作为 FTP 请求的代理时返回。图标出现在关联的文件边上以帮助用户区别文件。

图标可以是空白图标或另一图标，您指定它以出现在您目录列表的标题上。为了正确对齐，您使用的图标必须与目录列表上使用的其余图标的大小相同。

格式

```
AddBlankIcon icon_URL alternative_text
```

icon_URL

指定图标 URL 的最近部分。服务器将此值添加到 /icons/，以形成完整的 URL 请求。如果请求是针对本地文件的，服务器会通过映射伪指令转换请求。对于要检索的图标，映射伪指令必须允许要传递的请求。

如果您将服务器作为代理使用，完整的请求必须是指向服务器的全限定 URL。

alternative_text

如果请求的浏览器不显示图形，则指定用于图标的备用文本。

示例

```
AddBlankIcon logo.gif logo
```

缺省值

- **Linux 和 UNIX:** AddBlankIcon blank.m.pm.gif
- **Windows:** AddBlankIcon blank.gif

因为图标是空白的，所以缺省不指定备用文本。

AddDirIcon – 指定目录列表上目录的图标 URL

使用此伪指令指定表示目录列表上目录的图标。

格式

```
AddDirIcon icon_URL alternative_text
```

icon_URL

指定图标 URL 的最近部分。服务器将此值添加到 /icons/，以形成完整的 URL 请求。如果请求是针对本地文件的，服务器会通过映射伪指令转换请求。对于要检索的图标，映射伪指令必须允许要传递的请求。

如果您将服务器作为代理使用，完整的请求必须是指向服务器的全限定 URL。您必须将 URL 映射到本地文件并确保映射伪指令允许传递 URL。

alternative_text

如果请求的浏览器不显示图形，则指定用于图标的备用文本。

示例

```
AddDirIcon direct.gif DIR
```

缺省值

- **Linux** 和 **UNIX**: AddDirIcon dir.m.pm.gif DIR
- **Windows**: AddDirIcon dir.gif DIR

AddEncoding - 为具有特殊后缀的文件指定 MIME 内容编码

使用此伪指令将带有特殊后缀的文件绑定为 MIME 编码类型。很少使用此伪指令。

格式

```
AddEncoding .extension encoding
```

.extension

指定文件后缀模式。

encoding

指定您要绑定到文件的 MIME 编码类型，它与相应的后缀模式相匹配。

示例

```
AddEncoding .qp quoted_printable
```

缺省值

```
AddEncoding .Z x-compress
```

AddIcon - 将图标绑定到 MIME 内容类型或编码类型

使用此伪指令指定代表具有特定 MIME 内容类型或编码类型的图标。服务器在目录列表上使用图标，包括 FTP 目录列表。

格式

```
AddIcon icon_URL alternative_text MIME_type_template
```

icon_URL

指定图标 URL 的最近部分。服务器将此值添加到 /icons/，以形成完整的 URL 请求。如果请求是针对本地文件的，服务器会通过映射伪指令转换请求。对于要检索的图标，映射伪指令必须允许要传递的请求。

如果您将服务器作为代理使用，完整的请求必须是指向服务器的全限定 URL。您必须将 URL 映射到本地文件并确保映射伪指令允许传递 URL。

alternative_text

如果请求的浏览器不显示图形，则指定用于图标的备用文本。

type_template

指定 MIME 内容类型或编码类型模板。内容类型模板总是包含斜杠 (/)。编码类型模板没有斜杠。

示例

```
AddIcon video_file.m.pm.gif MOV video/*
```

缺省值

在 `ibmproxy.conf` 配置文件中为 `AddIcon` 伪指令设置了许多缺省值。

AddParentIcon - 指定代表目录列表上父目录的图标 URL

使用此伪指令指定代表目录列表上父目录的图标。

格式

```
AddParentIcon icon_URL alternative_text
```

icon-URL

指定图标 URL 的最近部分。服务器将此值添加到 `/icons/`，以形成完整的 URL 请求。如果请求是针对本地文件的，服务器会通过映射伪指令转换请求。对于要检索的图标，映射伪指令必须允许要传递的请求。

如果您将服务器作为代理使用，完整的请求必须是指向服务器的全限定 URL。您必须将 URL 映射到本地文件并确保映射伪指令允许传递 URL。

alternative_text

如果请求的浏览器不显示图形，则指定用于图标的备用文本。

示例

```
AddParentIcon parent.gif UP
```

缺省值

```
AddParentIcon dir-up.gif UP
```

AddType - 为具有特殊后缀的文件指定数据类型

使用此伪指令将带有特殊后缀的文件绑定为 MIME 类型和子类型。在配置文件中，能多次出现该伪指令。服务器为最常用的后缀提供缺省值。

格式

```
AddType .extension type/subtype encoding [quality[ character_set]]
```

.extension

文件后缀模式。您仅可在下列两个特殊的后缀模式上使用通配符 (*)：

- *.* 匹配所有包含点字符 (.) 的文件名并且不受其它规则匹配。
- * 匹配所有不包含点字符 (.) 的文件名并且不受其它规则匹配。

type/subtype

您要绑定到文件的 MIME 类型和子类型，这些文件与相应的后缀模式匹配。

encoding

已将数据转换到的 MIME 内容编码。FTP 代理服务器也使用编码以确定文件是否以二进制方式检索。在大多数情况中，适当的编码是 7bit、8bit 或 binary，并由以下所示确定为：

7bit 所有数据都表示为 8859-1 ASCII 数据的短（少于 1000 个字符）行。源代码或纯文本文件通常属于此类别。包含绘线型字符或强调字符的文件除外。

8bit 数据表示为短行，但可包含使用高位设置的字符（例如，绘线型字符或强调字符）。PostScript 文件和来自欧洲站点的文本文件通常属于此类别。

binary 此编码可用于所有数据类型。数据不仅可包含非 ASCII 字符，还可包含长（大于 1000 字符）行。几乎每个类型为 image/*、audio/* 和 video/* 的文件均归为此类别，正如类型为 application/* 的二进制数据文件一样。

编码的任何其它值都接受与二进制相同的处理，并作为内容编码 MIME 头在 MIME 头中传递。规范 7bit 和 8bit 不在 MIME 头中发送。

quality

指定内容类型相对值（从比例 0.0 到 1.0）的可选指示符。如果请求匹配文件的多个表示法，就使用质量值。服务器选择与最高质量值关联的文件。例如，如果请求了文件 internet.ps，并且服务器具有下列 AddType 伪指令集，则服务器使用 application/postscript 行，因为它的质量数比较高。

```
AddType .ps application/postscript 8bit 1.0
AddType *.* application/binary binary 0.3
```

character_set

您要与文本文件相关联的字符集的可选指示符。对于您指定字符集的文件，服务器告知客户机浏览器在显示文件时使用哪个字符集。如果您为 *character_set* 字段设置值，则您必须也包含 *quality* 字段的值。

示例

```
AddType .bin application/octet-stream binary 0.8
```

缺省值

许多 AddType 伪指令的缺省设置包含于配置文件（ibmproxy.conf）。

AddUnknownIcon – 指定目录列表上未知文件类型的图标 URL

使用此伪指令指定表示目录列表上未知文件类型的文件的图标。

格式

```
AddUnknownIcon icon_URL alternative_text
```

icon_URL

指定图标 URL 的最近部分。服务器将此值添加到 /icons/ 以形成完整的 URL 请求。如果请求是针对本地文件的，服务器会通过映射伪指令转换请求。对于要检索的图标，映射伪指令必须允许要传递的请求。

如果您将服务器作为代理使用，完整的请求必须是指向服务器的全限定 URL。您必须将 URL 映射到本地文件并确保映射伪指令允许传递 URL。

alternative_text

如果请求的浏览器不显示图形，则指定用于图标的备用文本。

示例

```
AddUnknownIcon saywhat.gif unknown
```

缺省值

- **Linux 和 UNIX:** AddUnknownIcon unknown.gif ???
- **Windows:** AddUnknownIcon unknown.gif ???

AdminPort - 指定请求管理页面或表单的端口

使用此伪指令指定管理员可用于访问服务器状态页面或配置表单的端口。对于此端口的请求和所有对于使用 Port 伪指令定义的标准端口或端口的其它入网请求不列在一起。然而，对于 AdminPort 的请求通过相同的普通访问控制和请求映射规则（例如，Pass、Exec、Protect）。

注：管理端口必须和使用 Port 伪指令定义的端口或标准端口不相同。

格式

```
AdminPort port_number
```

示例

```
AdminPort 2001
```

缺省值

无

AggressiveCaching - 为非高速缓存文件指定高速缓存

使用此伪指令指定是否对由源服务器返回且标记为非高速缓存的文件也进行高速缓存。将高速缓存的非高速缓存文件根据此伪指令标记为必须重新生效。每次请求文件时，代理服务器将 If-Modified-Since 请求发送到源服务器从而在高速缓存发出响应前重新生效响应。当前此伪指令影响的唯一一种非高速缓存文件是来自源服务器并包含 cache-control: no-cache 头的响应。可多次指定此伪指令。

格式

```
AggressiveCaching url_pattern
```

示例

```
AggressiveCaching http://www.hosta.com/*  
AggressiveCaching http://www.hostb.com/*
```

对于向下兼容，此伪指令的前一语法（AggressiveCaching {on | off}）现在按如下所示：

AggressiveCaching on 视作为 AggressiveCaching *。

AggressiveCaching off 被忽略。

注：如果同时指定 AggressiveCaching off 和 AggressiveCaching url_pattern，则忽略 AggressiveCaching off，并显示警告消息。

缺省值

无

AlwaysWelcome - 指定是否要为欢迎文件搜索所请求的目录

对于包含目录名而不是文件名的请求，AlwaysWelcome 伪指令控制服务器是否查看要返回的欢迎文件的目录。缺省情况下，AlwaysWelcome 设置为 on 值。这意味着服务器总是查看与 Welcome 伪指令上指定的名称相匹配文件的请求目录。如果查找到匹配，则文件返回到请求者。如果服务器在目录中的文件和 Welcome 伪指令上的文件名之间查找到一个以上匹配，则 Welcome 伪指令的顺序确定返回哪个文件。服务器使用最靠近配置文件顶部的 Welcome 伪指令。

格式

```
AlwaysWelcome on | off
```

缺省值

```
AlwaysWelcome on
```

相关伪指令

- 第 251 页的『Welcome - 指定欢迎文件的名称』

appendCRLFtoPost - 附加 CRLF 到 POST 请求

使用此伪指令指定 URL，其中高速缓存代理将回车和换行字符附加到 POST 请求的主体结束处。可多次指定此伪指令。

注：仅对在处理 POST 请求方面有已知问题的 URL 指定此伪指令。

格式

```
appendCRLFtoPost url_pattern
```

示例

```
appendCRLFtoPost http://www.hosta.com/
```

缺省值

无

ArrayName - 命名远程高速缓存数组

使用此伪指令指定服务器共享的远程高速缓存数组。

注：设置数组时，在此数组的所有成员上同一配置 Hostname 伪指令。

格式

```
ArrayName array_name
```

缺省值

无

Authentication – 定制认证步骤

使用此伪指令指定您要服务器在服务器请求过程的认证步骤期间调用的定制应用程序功能。按照认证方案执行此代码。仅支持基本认证。

注：认证是授权过程的一部分；它仅在需要授权时才发生。

格式

Authentication type /path/file:function_name

type

指定认证方案以进一步确定是否调用了您的应用程序功能。星号 (*) 和 BASIC 都是可接受的值。

/path/file

指定编译程序的全限定文件名，包含扩展名。

function_name

指定在程序中您给应用程序功能起的名字。

示例

```
Authentication BASIC /ics/api/bin/icsextpgm.so:basic_authentication
```

缺省值

无

Authorization – 定制授权步骤

使用此伪指令指定服务器在服务器请求过程的授权步骤期间调用的定制应用程序功能。此代码验证所请求的对象能对客户机服务。

格式

Authorization request_template /path/file:function_name

request_template

指定进一步确定是否调用应用程序功能的请求模板。规范可包含协议、域和主机；它可放在斜杠字符 (/) 之后，并可使用星号 (*) 作为通配符。例如，/front_page.html、http://www.ics.raleigh.ibm.com、/pub*、/* 和 * 均有效。当将高速缓存代理用作反向代理时，请求模板必须以文档根 (/) 开头。

/path/file

指定编译程序的全限定文件名，包含扩展名。

function_name

指定在程序中您给应用程序功能起的名字。

示例

```
Authorization /index.html /api/bin/icsextpgm.so:auth_url
```

缺省值

无

AutoCacheRefresh – 指定是否要使用高速缓存刷新

用此伪指令打开或关闭高速缓存刷新。如果打开了刷新，则会自动刷新高速缓存内容。如果关闭了刷新，则不调用高速缓存代理程序并且忽略所有它的设置。如果您正用另一方法启动高速缓存代理程序，例如，通过使用 Linux 和 UNIX 系统上的 **cron** 作业，将此伪指令设置为 **off**。

格式

```
AutoCacheRefresh {on | off}
```

缺省值

```
AutoCacheRefresh On
```

BindSpecific – 指定服务器是否绑定到一个或所有 IP 地址

在多宿主系统上使用此伪指令指定服务器是否在单个网络地址上侦听。如果您将值设置为 **On**，则服务器绑定到在 **Hostname** 伪指令中指定的 IP 地址，而不是绑定到所有本地 IP 地址。

如果未指定此伪指令，则服务器绑定到缺省主机名。

如果更改 **Caching** 伪指令，必须手工停止服务器，然后启动它。如果您仅重新启动它，服务器不做更改。（请参阅第 15 页的第 5 章，『启动和停止高速缓存代理』）。

格式

```
BindSpecific {on | off} [OutgoingSrcIp ip_addr | host_name]
```

[OutgoingSrcIp ip_addr | host_name]

当进行出局连接时，**OutgoingSrcIp** 选项允许高速缓存代理使用特定源 IP 地址。当特殊的防火墙规则需要此功能时，这对于 DMZ 中的高速缓存代理设置有用。

缺省值

```
BindSpecific Off
```

BlockSize – 指定在高速缓存中的块大小

此伪指令指定在高速缓存设备的介质中块的大小（以字节表示）。缺省情况下，值为 8192。因为它是唯一支持的大小，所以不要更改此值。要获取更多的信息，请参阅第 150 页的『**htcformat** 命令』的参考章节。

格式

```
BlockSize 大小
```

缺省值

缺省情况下，在配置文件中没有 **BlockSize** 的设置。（缺省值是 8192。）

CacheAccessLog – 指定高速缓存访问日志文件的路径

使用此伪指令指定您要服务器将访问日志存储到代理高速缓存位置的路径和文件名。仅当服务器作为代理运行时此伪指令才有效。要获取更多的信息，请参阅第 175 页的『**CacheRefreshTime** – 指定何时启动高速缓存代理程序』。

要将请求的记录启用到代理高速缓存，`Caching` 伪指令必须设置为 `ON`，并且必须为 `CacheMemory` 和 `CacheAccessLog` 伪指令设置值。可选地，通过使用 `CacheDev` 伪指令可定义一个或多个高速缓存设备。

`CacheAccessLog` 的值可以是绝对路径或相对于 `ServerRoot` 的路径。（每个显示一个示例。）

格式

```
CacheAccessLog path/file
```

示例

```
CacheAccessLog /absolute/path/logfile
CacheAccessLog /logs/logfile
```

缺省值

- **Linux** 和 **UNIX** 系统: `CacheAccessLog /opt/ibm/edge/cp/server_root/logs/cache`
- **Windows** 系统: `CacheAccessLog drive:\Program Files\IBM\edge\cp\logs\cache`

CacheAlgorithm - 指定高速缓存算法

使用此伪指令指定在垃圾收集期间服务器使用的高速缓存算法。

格式

```
CacheAlgorithm {bandwidth | responsetime | blend}
```

bandwidth

尝试最大化保存网络带宽。

responsetime

尝试最小化用户响应时间。

blend

使用带宽和响应时间的均衡组合。

缺省值

```
CacheAlgorithm bandwidth
```

CacheByIncomingUrl - 指定生成高速缓存文件名的基础

使用此伪指令指定生成的高速缓存文件名是否基于请求的入网 URL。

如果将此伪指令设置为 `on`，则基于入网 URL 生成高速缓存文件名。如果此伪指令设置为 `off`，则进入 URL 首先通过所有可应用的名称转换插件、MAP 规则和 PROXY 规则，并且生成的高速缓存文件名基于结果 URL。

注：当定义高速缓存过滤器时，在基于 URL 的高速缓存过滤器的反向代理方案中，使用以 `/`（正斜杠）的文档根开头的格式。例如：`/test/index.html`。此格式不应该包括协议，例如 `http://`。

格式

```
CacheByIncomingUrl {on | off}
```

缺省值

CacheByIncomingURL off

CacheClean – 指定要保留高速缓存文件多久

使用此伪指令指定您要服务器保留高速缓存文件多久。当垃圾收集运行时，服务器将删除那些已超出此时间的高速缓存文件，而不论文件的到期日期。任何时候都要求高速缓存中的文件比指定的时间长，服务器将重新验证此文件以确保在保存之前它是有效的。

格式

CacheClean *time_specification*

示例

CacheClean 2 weeks

缺省值

CacheClean 1 month

CacheDefaultExpiry – 指定文件的缺省到期时间

使用此伪指令以设置文件的缺省到期时间，其中服务器不提供 Expires 或 Last-Modified 头。使用与模板相匹配的 URL 指定 URL 模板和文件的到期时间。此伪指令的多个出现可包含于配置文件。每个模板包含一个单独的伪指令。URL 模板必须包含协议。以月份、周、天和小时的任意组合指定时间值。

格式

CacheDefaultExpiry *URL_template expiration_time*

缺省值

```
CacheDefaultExpiry ftp:* 1 day
CacheDefaultExpiry gopher:* 2 days
CacheDefaultExpiry http:* 0 days
```

注：HTTP 协议的缺省到期是 0 天。由于许多脚本程序不给予到期日期，所以建议保留此值，但它们的输出立即到期。0 以外的值可使客户机查看过时内容。

CacheDev – 指定高速缓存的存储设备

使用此伪指令指定高速缓存存储设备。可以指定文件或原始磁盘分区。在 AIX 平台上，可指定原始逻辑卷。（如果不使用内存高速缓存，则原始磁盘高速缓存会生成最佳性能。）

注意，指定高速缓存设备前必须将它们准备好。要准备高速缓存设备，则使用 **htcformat** 命令格式化它。要获取更多的信息，请参阅第 150 页的『htcformat 命令』。

可指定多个高速缓存设备。每个设备与相同的 CacheMemory 和 BlockSize 值相关联。但每个高速缓存设备在代理服务器中会导致大约 8 MB 的内存开销。少数大设备比更多小设备更有效率。为获得最高效率，可使用整个磁盘作为一个大分区，在磁盘上不冗余地。关于高速缓存存储的更多信息包含在第 97 页的『优化磁盘高速缓存性能』中。

格式

CacheDev {*raw_disk_partition* | *file*}

示例

AIX: CacheDev /dev/r1v02

HP-UX: CacheDev /dev/rdsk/clt15d0

Linux: CacheDev /opt/IBMWTE/filecache1

Solaris: CacheDev /dev/rdsk/clt3d0s0

Windows: CacheDev \\.\E:

缺省值

无

CacheExpiryCheck - 指定服务器是否返回到期文件

使用此伪指令指定服务器是否返回已到期的高速缓存文件。如果您要服务器能返回到期文件，则将值设置为 `Off`。如果您要代理在客户机请求到期文件时检查源服务器以获取最新的版本，则使用缺省值为 `On`。通常，管理员不要服务器返回到期文件，除非当他们演示服务器并且不特别关注返回内容时。

格式

CacheExpiryCheck {`on` | `off`}

缺省值

CacheExpiryCheck `On`

CacheFileSizeLimit - 指定要高速缓存文件的最大大小

使用此伪指令指定要高速缓存文件的最大大小。不高速缓存比此大小更大的文件。可以以字节（`B`）、千字节（`K`）、兆字节（`M`）或千兆字节（`G`）指定该值。规范是否在数和计量单位（`B`、`K`、`M`、`G`）之间包含空格是无关紧要的。

格式

CacheFileSizeLimit *maximum* {`B` | `K` | `M` | `G`}

缺省值

CacheFileSizeLimit `4000 K`

CacheLastModifiedFactor - 指定值以确定到期日期

使用此伪指令指定用于计算特定 `URL` 或所有与模板匹配的 `URL` 到期日期的值。

HTTP 服务器频繁地给予文件最近修改的时间，但不是到期日期。同样地，FTP 文件可具有最近修改的时间戳记，但不是到期日期。高速缓存代理基于最近修改时间计算这些文件的到期日期。它使用最近修改时间以确定文件修改后时间的长度并且通过 `CacheLastModifiedFactor` 伪指令上的值增加。此计算的结果是文件的生命期，或者文件前的时间长度失效。

您也可指定 `off` 或 `-1` 以关闭伪指令并且不计算到期日期。代理服务器按它们在配置文件中出现的顺序读取 `CacheLastModifiedFactor` 伪指令。它使用第一个可应用于高速缓存文件的伪指令。

格式

`CacheLastModifiedFactor url factor`

url

指定正在高速缓存的文件的完整 URL（包含协议）。您可使用星号（*）用作为通配符的 URL 模板以应用掩码。

factor

指定要在计算中使用的因子。也可指定值 `off` 或 `-1`。

示例

```
CacheLastModifiedFactor */hosta/* off
CacheLastModifiedFactor ftp://hostb/* 0.30
CacheLastModifiedFactor ftp://* 0.25
CacheLastModifiedFactor http://* 0.10
CacheLastModifiedFactor * 0.50
```

缺省值

```
CacheLastModifiedFactor http://*/ 0.10
CacheLastModifiedFactor http://*.htm* 0.20
CacheLastModifiedFactor http://*.gif 1.00
CacheLastModifiedFactor http://*.jpg 1.00
CacheLastModifiedFactor http://*.jpeg 1.00
CacheLastModifiedFactor http://*.png 1.00
CacheLastModifiedFactor http://*.tar 1.00
CacheLastModifiedFactor http://*.zip 1.00
CacheLastModifiedFactor http:* 0.15
CacheLastModifiedFactor ftp:* 0.50
CacheLastModifiedFactor * 0.10
```

缺省值 0.14 使一周前修改的文件在一天内到期。

CacheLocalDomain - 指定是否要高速缓存本地域

使用此伪指令指定是否高速缓存来自与代理相同域的主机的 URL。内部网上的本地站点通常不需要高速缓存，因为内部带宽已经足够快速装入 URL。不高速缓存本地站点可以节约要用较长时间检索的 URL 的高速缓存空间。

格式

`CacheLocalDomain {on | off}`

缺省值

`CacheLocalDomain on`

CacheMatchLanguage — 指定返回的高速缓存内容的语言首选项

如果后端服务器有能力为同一 URL 的客户返回语言变量，则使用此伪指令支持高速缓存同一 URL 的不同语言。此伪指令允许高速缓存代理使用高速缓存的响应的语言验证请求中的语言首选项。

当启用 `CacheMatchLanguage` 时，在高速缓存代理装入高速缓存的内容前，高速缓存代理将请求的 `Accept-Language` 头中的语言首选项与高速缓存的内容的语言进行比较。高速缓存代理还比较首选项间的差异。如果首选项间的差异小于指定的限制，则它返回高速缓存的副本；否则，代理将请求转发给后端服务器以获取您请求的语言的最新副本。

格式

```
CacheMatchLanguage {on | off} lang-prefer-distance-limit special-id-for-all-lang  
lang-prefer-distance-limit
```

指定 0.001– 0.9999 范围内的值。

```
special-id-for-all-lang
```

指定从 `Content-Language` 头中的服务器返回的语言字符串，以通知代理响应可用于所有语言首选项。

示例

以下是伪指令、高速缓存对象和请求的一个配置示例。

```
CacheMatchLanguage On 0.2
```

如果高速缓存对象是简体中文 (`zh_cn`)，则请求是：

```
GET / HTTP/1.1  
...  
Accept-Language: en_US;q=1.0、zh_cn;q=0.7、ja;q=0.3  
....
```

对于此请求，客户先以英语（代码和质量为 `en_US/1.0`）请求页面，然后以简体中文（代码和质量为 `zh_cn/0.7`）请求页面，再以日文（代码和质量为 `ja/0.3`）请求页面。高速缓存的对象是以简体中文请求的。最期待的质量和匹配的语言质量间的首选项间的差异是 $1.0 - 0.7 = 0.3$ 。由于此限制由 `CacheMatchLanguage` 伪指令设置为 0.2，而 0.3 大于此限制，所以代理向服务器请求一个新的该 URL 的副本，而不是返回高速缓存的对象。

当返回响应时，如果服务器不指定语言或不指定 `Content-Language` 头中的 `special-id-for-all-lang`，则当下一次请求到达时，代理不会匹配语言首选项，而是返回高速缓存的对象。

缺省值

```
CacheMatchLanguage off
```

CacheMaxExpiry – 指定高速缓存文件的最大生命期

使用此伪指令以定义可在高速缓存中停留的时间文件的最大长度。高速缓存文件的生命期定义可从高速缓存中服务而不检查更新起始地址的时间长度。在一些情况中，高速缓存文件计算的生命期可以比您要保留文件的时间更长。文件的生命期（由起始地址指定或由高速缓存代理计算）不能超出 `CacheMaxExpiry` 伪指令指定的限度。

在配置文件中可以多次出现此伪指令。每个模板包含一个单独的伪指令。

格式

```
CacheMaxExpiry URL lifetime
```

URL

指定高速缓存文件的完整指定的 URL（包含协议）。您可使用星号（*）用作为通配符的 URL 模板以应用掩码。

lifetime

指定与 URL 模板相匹配的高速缓存文件的最大生命期。可以以月份、周、天、小时、分钟或秒的任意组合指定到期时间。

示例

```
CacheMaxExpiry ftp:* 1 month
CacheMaxExpiry http://www.santaclaus.np/* 2 days 12 hours
```

缺省值

```
CacheMaxExpiry 1 month
```

CacheMemory – 指定高速缓存 RAM

使用此伪指令指定与高速缓存关联的内存数量。要获得磁盘高速缓存的最优性能，建议高速缓存基础结构支持的最小高速缓存值为 64 MB，包含高速缓存索引。当增加高速缓存大小时，高速缓存索引将增加，所以需要更多的高速缓存来存储索引。64 MB 的高速缓存值已足够大以提供高速缓存基础结构支持，存储高速缓存索引的磁盘高速缓存大约需要 6.4 GB。要获得较大的磁盘高速缓存，高速缓存应当为高速缓存大小的 1%。

如果使用内存高速缓存，则设置此伪指令以包含高速缓存索引所需的高速缓存和内存量。

对于该伪指令建议所用的最大值为 1600 MB。该限定是由高速缓存代理的实际确定的，作为 32 位的应用程序，能使用 2 GB 的最大内存。如果高速缓存所需的内存数量和用于日常处理方法所需的内存数量超出 2 GB 的话，则高速缓存代理不会正常操作。

可以下列单位之一指定数量：字节（B）、千字节（K）、兆字节（M）或千兆字节（G）。

格式

```
CacheMemory amount {B | K | M | G}
```

缺省值

```
CacheMemory 64 M
```

CacheMinHold – 指定文件的保留时间

使用此伪指令指定文件的 URL，它们的到期将要被重设。一些站点设置文件以在它们生命期结束前到期，这要求服务器更频繁地请求文件。CacheMinHold 伪指令使到期文件在再次请求以前以指定的时间数量保持在高速缓存中。可多次指定此伪指令。

注：如果重设到期日期，则在高速缓存中的文件可成为废弃或过时的文件。

示例

```
CacheMinHold http://www.cachebusters.com/* 1 hour
```

缺省值

无

CacheNoConnect – 指定独立高速缓存方式

使用此伪指令指定代理服务器是否检索来自远程服务器的文件。缺省值 (Off) 使服务器能检索来自远程服务器的文件。值 On 设置服务器以运行于独立高速缓存方式。这意味着服务器可以仅返回已经存储在高速缓存中的文件。通常, 以此方式运行服务器时, 您也可将 CacheExpiryCheck 伪指令设置为 Off。

如果您将服务器用于演示, 以独立高速缓存方式运行服务器是很有用的。如果您知道要用于演示的所有文件都存储在高速缓存中, 则您不需要网络连接。

格式

```
CacheNoConnect {on | off}
```

缺省值

```
CacheNoConnect Off
```

CacheOnly – 仅高速缓存其 URL 与模板匹配的那些文件

使用此伪指令指定仅高速缓存带有与指定模板匹配的 URL 的文件。您可以在配置文件中多次使用此伪指令。每个模板包含一个单独的伪指令。URL 模板必须包含协议。如果没有为此伪指令设置任何值, 则可高速缓存任何与 NoCaching 伪指令不匹配的 URL。如果 CacheOnly 或 NoCaching 伪指令均未包含于配置文件, 则可高速缓存任何 URL。

格式

```
CacheOnly url_pattern
```

示例

```
CacheOnly http://realstuff/*
```

缺省值

无

CacheQueries – 指定高速缓存对包含问号 (?) 的 URL 的响应

使用此伪指令指定 URL, 对它们查询请求的响应高速缓存。如果使用值 PUBLIC *url_pattern*, URL 中包含问号的 GET 请求响应将被高速缓存, 前提是源服务器包含 cache-control: public 头且响应可高速缓存。如果指定了值 ALWAYS *url_pattern*, 当响应可以其它方法高速缓存时, URL 中包含问号的 GET 请求响应就将被高速缓存。

可多次指定此伪指令。

```
CacheQueries {ALWAYS | PUBLIC} url_pattern
```

示例

```
CacheQueries ALWAYS http://www.hosta.com/*  
CacheQueries PUBLIC http://www.hostb.com/*
```

注: 为了反向兼容, CacheQueries {ALWAYS | PUBLIC | NEVER} 的前一语法将按如下所示处理:

- CacheQueries ALWAYS 和 CacheQueries PUBLIC 视作为 CacheQueries ALWAYS * 和 CacheQueries PUBLIC *。
- CacheQueries NEVER 被忽略。
- 如果指定了 CacheQueries NEVER 和 CacheQueries *url_pattern*，则忽略 CacheQueries NEVER，但发出警告消息。

缺省值

无

CacheRefreshInterval – 指定重新生效高速缓存对象的时间间隔

使用此伪指令指定何时检查源服务器以确定是否已更改高速缓存文件。

尽管 CacheClean 伪指令看起来与此伪指令相似，但还是有差异的。CacheRefreshInterval 仅指定此代理在使用它之前重新生效文件，而 CacheClean 伪指令在一段指定时间后会从高速缓存中除去文件。

格式

- 下列格式指定任何匹配 URL 模式文件的刷新时间间隔。
`CacheRefreshInterval URL_pattern time_period`
- 下列格式指定任何不匹配 URL 模式文件的刷新时间间隔。仅指定刷新时间间隔。
`CacheRefreshInterval time_period`

示例

```
CacheRefreshInterval *.gif 8 hours
CacheRefreshInterval 1 week
```

缺省值

CacheRefreshInterval 2 weeks

CacheRefreshTime – 指定何时启动高速缓存代理程序

使用此伪指令指定何时启动高速缓存代理程序。您可在特定时间启动高速缓存代理程序。

格式

CacheRefreshTime *HH:MM*

缺省值

CacheRefreshTime 03:00

CacheTimeMargin – 指定高速缓存文件的最小生命期

CacheTimeMargin 伪指令指定文件要高速缓存时需要的最小生命期。

高速缓存代理计算每个文件的到期时间。如果在文件到期前可能不会接收文件的另一请求，则高速缓存代理会考虑文件的生命期对于要高速缓存的文件来说太短。缺省情况下，高速缓存代理不高速缓存生命期小于 10 分钟的文件。如果您的缓存未接近其最大容量，请将此伪指令设置为初始值。如果您的缓存已经快要填充到容量，则考虑增加最小生命期的值。

格式

`CacheTimeMargin minimum_lifetime`

缺省值

`CacheTimeMargin 10 minutes`

注：将此伪指令设置为大于四小时将大大降低高速缓存的效率。

CacheUnused – 指定未使用的高速缓存文件要保留多久

使用此伪指令指定您要服务器保留带有与指定模板匹配的 URL 的未使用高速缓存文件的最大时间长度。在指定的高速缓存时间过后，无论文件的到期日期为何时，服务器都会删除带有与模板匹配的 URL 的未使用文件。您可以在配置文件中包含多个此伪指令。每个模板包含一个单独的伪指令。URL 模板必须包含协议。以月份、周、天和小时的任意组合指定时间值。

格式

`CacheUnused url_template time_length`

示例

```
CacheUnused ftp:* 3 weeks
CacheUnused gopher:* 3 days 12 hours
CacheUnused * 4 weeks
```

缺省值

```
CacheUnused ftp:* 3 days
CacheUnused gopher:* 12 hours
CacheUnused http:* 2 days
```

Caching – 启用代理高速缓存

使用此伪指令启用文件的高速缓存。打开高速缓存时，代理服务器在本地高速缓存中存储它从其它服务器检索的文件。则代理服务器无需从其它服务器检索就能响应对相同文件的后继请求。

格式

`Caching {on | off}`

缺省值

`Caching On`

注：如果更改此 Caching 伪指令，必须手工停止服务器，然后启动它。（请参阅第 15 页的第 5 章，『启动和停止高速缓存代理』）。

CdfAware – 指定高速缓存代理的实例作为内容分发框架的部件

使用此伪指令指定高速缓存代理是否是内容分发框架的部件。

格式

`CdfAware {yes | no}`

缺省值

CdfAware no

CdfRestartFile - 指定文件将文件名存储到 url 映射

使用该伪指令来指定将文件名存储到 URL 映射数据的文件名，以便于它可持久于内容分发框架中的 ibmproxy 的多个实例。高速缓存代理维护映射表，此表在 Web 服务器上把请求的 URL 及其文件名相关联。此文件在重新启动后提供此表的持久存储。仅当 CdfAware 伪指令设置为 yes 时才使用此伪指令。

格式

CdfRestartFile *path/filename*

示例

- **Linux 和 UNIX:** CdfRestartFile /opt/ibm/edge/cd/cdfRestartFile
- **Windows:** CdfRestartFile C:\progra~1\ibm\edge\cd\cdfRestartFile.txt

缺省值

无

CompressAge - 指定何时压缩日志

使用此伪指令指定其后压缩日志的时效。当日志比为 CompressAge 设置的值旧时，就将被压缩。如果 CompressAge 设置为 0，则不压缩日志。从不压缩当前和几天前的日志。

格式

CompressAge *number_of_days*

缺省值

CompressAge 1

相关伪指令

- 第 178 页的『CompressDeleteAge - 指定何时删除日志』
- 『CompressCommand - 指定压缩命令和参数』
- 第 206 页的『LogArchive - 指定日志归档的行为』
- 第 212 页的『Midnight - 指定用于归档日志的 API 插件』
- 第 233 页的『PurgeAge - 指定日志的时效』
- 第 233 页的『PurgeSize - 指定日志压缩文档的大小限制』

CompressCommand - 指定压缩命令和参数

使用此伪指令构建一个命令，标识用于压缩日志的压缩实用程序并将参数传递到该实用程序。包含归档日志的路径。

压缩实用程序必须安装在该机器路径中列出的目录里。

格式

CompressCommand *command*

command

包含您要使用的命令和参数，在一行上输入。通常参数包含 `%%LOGFILES%%` 和 `%%DATE%%`。

%%LOGFILES%%

指定可用于特定 `%%DATE%%` 的日志文件列表。

%%DATE%%

指定日志文件的日期标记。

示例

• Linux 和 UNIX:

```
CompressCommand tar -cf /logarchs/log%%DATE%%.tar %%LOGFILES%% ;
                    gzip /logarchs/log%%DATE%%.tar
CompressCommand tar -cf /logarchs/log%%DATE%%.tar %%LOGFILES%% ;
                    compress /logarchs/log%%DATE%%.tar
CompressCommand zip -q /logarchs/log%%DATE%%.zip %%LOGFILES%%
```

注：命令和所有参数必须在一行上输入。在前面的示例中，分开前两个命令示例以便于阅读。

• Windows:

```
CompressCommand pkzip -q d:\logarchs\log%%DATE%%.tar %%LOGFILES%%
```

缺省值

无

相关伪指令

- 第 177 页的『CompressAge - 指定何时压缩日志』
- 『CompressDeleteAge - 指定何时删除日志』
- 第 206 页的『LogArchive - 指定日志归档的行为』
- 第 212 页的『Midnight - 指定用于归档日志的 API 插件』
- 第 233 页的『PurgeAge - 指定日志的时效』
- 第 233 页的『PurgeSize - 指定日志压缩文档的大小限制』

CompressDeleteAge - 指定何时删除日志

使用此伪指令指定何时删除压缩过的日志。当日志比为 `CompressDeleteAge` 值设置的天数旧时，就会被删除。如果 `CompressDeleteAge` 设置为 0，或者如果它的值小于为 `CompressAge` 伪指令设置的值，则不删除日志。

注：压缩插件从不删除当日或前一日的日志。

格式

```
CompressDeleteAge number_of_days
```

缺省值

```
CompressDeleteAge 7
```


相关伪指令

- 第 177 页的『CompressAge - 指定何时压缩日志』
- 第 177 页的『CompressCommand - 指定压缩命令和参数』
- 第 206 页的『LogArchive - 指定日志归档的行为』
- 第 212 页的『Midnight - 指定用于归档日志的 API 插件』
- 第 233 页的『PurgeAge - 指定日志的时效』
- 第 233 页的『PurgeSize - 指定日志压缩文档的大小限制』

ConfigFile — 指定其它配置文件的名称

使用此伪指令指定附加配置文件的名称和位置。指定配置文件中找到的伪指令在当前配置文件之后处理。

注：确保附加配置文件对用户 nobody 的许可权设置为读以允许高速缓存代理程序读取此文件。

示例

- **Linux 和 UNIX:** ConfigFile /etc/rca.conf
- **Windows:** ConfigFile c:\WINNT\rca.conf

缺省值

无

ConnThreads — 指定要用于连接管理的连接线程数

使用此伪指令定义要用于连接管理的连接线程数。

格式

ConnThreads *number*

缺省值

ConnThreads 5

相关伪指令

- 第 209 页的『MaxActiveThreads - 指定有效线程的最大数』

ContinueCaching - 指定需要高速缓存一个文件的百分率

使用此伪指令指定即使在客户机连接终止时，为了完成创建高速缓存文件，必须为高速缓存代理传送一个请求文件的百分率。此变量的有效值是 0 - 100 范围中的整数。

例如，如果指定了 ContinueCaching 75 时，高速缓存代理检测到客户机连接终止前文件已经传送了 75% 以上，则高速缓存代理会继续从内容服务器传送文件并生成高速缓存文件。

格式

ContinueCaching *percentage*

缺省值

ContinueCaching 75

DefinePicsRule – 提供内容过滤规则

使用此伪指令将必需信息提供给代理以过滤包含评级服务信息的内容的 URL。您可多次指定此伪指令。

格式

```
DefinePicsRule "filter_name" {
```

缺省值

```
DefinePicsRule "RSAC Example" {
```

DefProt – 为与模板匹配的请求指定缺省保护设置

使用此伪指令将缺省保护设置与和模板匹配的请求关联起来。

注: 为了保护正常工作, 在配置文件中 DefProt 和 Protect 伪指令必须放在任何 Pass 或 Exec 伪指令之前。

格式

```
DefProt request_template setup_name [FOR server_IP_address | host_name]
```

request_template

指定您要将缺省保护设置与请求相关联的模板。服务器会比较入网客户机请求与模板, 如果匹配就关联一个保护设置。

实际上不会为匹配模板的请求激活保护, 除非请求还与后继 Protect 伪指令上的模板相匹配。请参阅第 221 页的『Protect – 为匹配模板的请求激活保护设置』以获取 Protect 伪指令如何与 DefProt 一起使用的说明。

setup

在配置文件中定义的、要与匹配 *request_template* 的请求关联的命名保护设置。用 protection 子伪指令定义保护设置。此参数可使用三种格式之一:

- 指定一个单独文件的全路径与文件名, 该文件包含 protection 子伪指令。
- 保护设置标号名称, 它与先前在 Protection 伪指令上定义的名称相匹配。Protection 伪指令包含 protection 子伪指令。
- 实际的 protection 子伪指令。这些子伪指令必须加大括号 ({}). 左大括号字符必须是 DefProt 伪指令同一行上的最后一个字符。后面每个子伪指令各占一行。右大括号字符必须在最后一行子伪指令后面独占一行。大括号中不能出现任何注释行。要获取 protection 子伪指令的描述, 请参阅:
 - 第 226 页的『AuthType – 指定认证类型』
 - 第 226 页的『DeleteMask – 指定允许以删除文件的用户名、组和地址』
 - 第 226 页的『GetMask – 指定允许以获取文件的用户名、组和地址』
 - 第 227 页的『GroupFile – 指定关联的组文件的位置』
 - 第 227 页的『Mask – 指定允许发出 HTTP 请求的用户名、组和地址』
 - 第 227 页的『PasswdFile – 指定关联的密码文件的位置』
 - 第 227 页的『PostMask – 指定允许以粘贴文件的用户名、组和地址』

- 第 228 页的『PutMask - 指定允许以放置文件的用户名、组和地址』
- 第 228 页的『ServerID - 指定与密码文件相关联的名称』

[FOR Server_IP_address | host_name]

如果您正在使用多个 IP 地址或虚拟主机，则使用此参数以指定 IP 地址或主机名。服务器仅对于来到此 IP 地址上服务器的请求或此主机使用伪指令。对于 IP 地址，这是服务器的网络连接的地址，而不是请求客户机的地址。

您可以指定 IP 地址（例如，FOR 240.146.167.72），或者您可以指定主机名（例如，FOR hostA.bcd.com）。

此参数是可选的。如果没有该参数，服务器使用该伪指令响应所有请求，无论是请求到达的 IP 地址还是 URL 中的主机名。

注:

1. 要使用此参数，*setup* 参数的格式必须指定于路径与文件名的表单或保护设置标号。您不能将此参数用于 *setup* 参数，它指定于大括号中围绕的实际保护子伪指令的表单。
2. 要使用此参数，您必须将 FOR 或其它字符串（不带空格）放在 *setup* 参数与 *IP_address* 或 *host_name* 之间。

不能为服务器的 IP 地址指定通配符。

注: 伪指令必须输入在一行。

示例

- 以下示例标识一个单独文件，该文件包含 protection 子伪指令。

```
DefProt /secret/* /server/protect/setup1.acc
```

- 以下示例使用标号名称指向 protection 子伪指令。标号名称必须与 Protection 伪指令上的标号名称相匹配。Protection 伪指令必须在 DefProt 伪指令之前。

```
DefProt /secret/* SECRET-PROT
```

- 以下示例包含 protection 子伪指令作为 DefProt 伪指令的一部分。

```
DefProt {
  AuthType Basic
  ServerID restricted
  PasswdFile /docs/etc/WWW/restrict.password
  GroupFile /docs/etc/WWW/restrict.group
  GetMask authors
  PutMask authors
}
```

- 下例使用可选的 IP 地址参数。如果您的服务器接收到以 /secret/ 开头的请求，它会基于请求到达的网络连接的 IP 地址将不同的缺省保护设置与请求关联。对于到达 0.67.106.79 的请求，服务器会将请求与带有 CustomerA-PROT 标号的 Protection 伪指令上定义的缺省保护关联。对于到达 0.83.100.45 的请求，服务器会将请求与带有 CustomerB-PROT 标号的 Protection 伪指令上定义的缺省保护关联。

```
DefProt /secret/* CustomerA-PROT 0.67.106.79
DefProt /secret/* CustomerB-PROT 0.83.100.45
```

- 下例使用可选的主机名参数。如果您的服务器接收到以 /secret/ 开头的请求，它会基于 URL 中的主机名将不同的缺省保护设置与请求关联。对于到达 hostA 的请求，服务器会将请求与带有 CustomerA-PROT 标号的 Protection 伪指令上定义的缺省保护

关联。对于到达 hostB 的请求，服务器会将请求与带有 CustomerB-PROT 标号的 Protection 伪指令上定义的缺省保护关联。

```
DefProt /secret/* CustomerA-PROT hostA.bcd.com
DefProt /secret/* CustomerB-PROT hostB.bcd.com
```

缺省值

无

DelayPeriod – 指定请求之间的暂停

使用此伪指令指定高速缓存代理程序是否在两次向目标服务器发送请求之间等待。指定两次请求之间的延迟将减少代理机器和网络链接上以及在目标服务器上的负载。指定无延迟将使高速缓存代理程序以最大速度运行。对于缓慢的因特网连接，考虑指定无延迟周期以完成您网络的最大使用。

注：如果您到因特网的连接快于 128 kbps，将 DelayPeriod 设置为 0n 以避免过多请求过快发送到正在刷新的站点。

格式

```
DelayPeriod {on | off}
```

缺省值

```
DelayPeriod 0n
```

DelveAcrossHosts – 指定跨域高速缓存

使用此伪指令指定高速缓存代理程序是否要逐层访问跨主机的超文本链接。如果高速缓存的 URL 包含到其它服务器的链接，服务器可以忽略或按照该链接。如果将伪指令 DelveInto 设置为 never，则不应用此伪指令。

格式

```
DelveAcrossHosts {on | off}
```

缺省值

```
DelveAcrossHosts Off
```

DelveDepth – 指定高速缓存时要逐层访问的链接层次

使用此伪指令指定搜索装入高速缓存的页面时要逐层访问的链接层次。如果伪指令 DelveInto 设置为 never，则不应用此伪指令。

格式

```
DelveDepth number_of_levels
```

缺省值

```
DelveDepth 2
```

DelveInto – 指定高速缓存代理程序是否要逐层访问链接

使用此伪指令指定高速缓存代理程序是否装入来自高速缓存 URL 链接的页面。

格式

`DelveInto {always | never | admin | topn}`

always

高速缓存代理程序按照所有先前高速缓存的 URL 的连接。

never

高速缓存代理程序忽略 URL 上的所有链接。

admin

高速缓存代理程序仅按照 `LoadURL` 伪指令中指定的 URL 上的链接

topn

高速缓存代理程序仅按照来自高速缓存最频繁检索文件的链接。

缺省值

`DelveInto always`

DirBackgroundImage - 为目录列表指定背景图像

使用此伪指令将背景图像应用到代理服务器生成的目录列表。当代理服务器用于浏览 FTP 站点时生成目录列表。

为背景图像指定绝对路径。如果图像位于另一个服务器，必须以完整 URL 的形式指定背景图像。如果未指定背景图像，则使用普通的白色背景。

格式

`DirBackgroundImage /path/file`

示例

`DirBackgroundImage /images/corplogo.png`

`DirBackgroundimage http://www.somehost.com/graphics/embossed.gif`

缺省值

无

DirShowBytes - 在目录列表上显示小文件的字节计数

使用此伪指令指定目录列表是否包含小于 1 KB 的文件的精确字节计数。值 `Off` 表示目录列表将所有 1 KB 或更小的文件都显示为 1 KB 大小。

格式

`DirShowBytes {on | off}`

缺省值

`DirShowBytes Off`

DirShowCase - 对目录列表上的文件排序时使用大小写

使用此伪指令指定目录列表在对文件名排序时是否分辨大写和小写字母。

值 `On` 表示大写字母在文件列表的小写字母之前。

格式

`DirShowCase {on | off}`

缺省值

`DirShowCase On`

DirShowDate – 在目录列表上显示最后修改日期

使用此伪指令指定目录列表是否包含最近修改的每个文件的日期。

格式

`DirShowDate {on | off}`

缺省值

`DirShowDate On`

DirShowDescription – 在目录列表上显示文件的描述

使用此伪指令指定目录列表是否包含 HTML 文件的描述。从文件的 HTML `<title>` 标记获取描述。

FTP 目录列表的描述在可以确定的情况下会显示文件的 MIME 类型。

格式

`DirShowDescription {on | off}`

缺省值

`DirShowDescription On`

DirShowHidden – 在目录列表上显示隐藏文件

使用此伪指令指定目录列表是否包含目录中任何隐藏文件。服务器将任何名称以句点 (.) 开头的文件视作为隐藏文件。

格式

`DirShowHidden {on | off}`

缺省值

`DirShowHidden On`

DirShowIcons – 在目录列表上显示图标

使用此伪指令指定服务器是否在目录列表中包含图标。图标可用于提供列表中文件的内容类型的图形表示法。图标本身由

`AddBlankIcon`、`AddDirIcon`、`AddIcon`、`AddParentIcon` 和 `AddUnknownIcon` 伪指令定义。

格式

`DirShowIcons {on | off}`

缺省值

`DirShowIcons On`

DirShowMaxDescrLength – 指定目录列表上描述的最大长度

使用此伪指令指定目录列表上描述字段中显示的最大字符数。

格式

```
DirShowMaxDescrLength number_of_characters
```

缺省值

```
DirShowMaxDescrLength 25
```

DirShowMaxLength – 指定目录列表上文件名的最大长度

使用此伪指令指定用于目录列表上文件名的最大字符数。

格式

```
DirShowMaxDescrLength number_of_characters
```

缺省值

```
DirShowMaxLength 25
```

DirShowMinLength – 指定目录列表上文件名的最小长度

使用此伪指令指定总是为目录列表上文件名保留的最小字符数。目录中的文件名可超出此数。但是，文件名不能长于 `DirShowMaxLength` 伪指令上指定的数字。

格式

```
DirShowMinLength number_of_characters
```

缺省值

```
DirShowMinLength 15
```

DirShowSize – 在目录列表上显示文件大小

使用此伪指令指定目录列表是否包含每个文件的大小。

格式

```
DirShowSize {on | off}
```

缺省值

```
DirShowSize On
```

Disable – 禁用 HTTP 方法

使用此伪指令指定哪些 HTTP 方法是服务器不接受的。对于每个服务器要拒绝的方法，输入单个 `Disable` 伪指令。

在缺省配置文件中，启用 GET、HEAD、OPTIONS、POST 和 TRACE 方法，禁用其它所有支持的 HTTP 方法。要禁用当前启用的方法，从 `Enable` 伪指令删除它并将其添加到 `Disable` 伪指令。

格式

`Disable method`

注：“配置和管理”表单使用 POST 方法更新服务器配置。如果您禁用 POST 方法，将不能使用“配置和管理”表单。

缺省值

```
Disable PUT
Disable DELETE
Disable CONNECT
```

DisInheritEnv - 指定 CGI 程序不继承的环境变量

使用此伪指令指定您要 CGI 程序不继承的环境变量（除了特定于 CGI 处理的 CGI 环境变量）。

缺省情况下，CGI 程序继承所有环境变量。使用此伪指令以禁止继承个别环境变量。

格式

`DisInheritEnv environment_variable`

示例

```
DisInheritEnv PATH
DisInheritEnv LANG
```

在此示例中，除了 PATH 和 LANG，CGI 程序继承所有环境变量。

缺省值

无

DNS-Lookup - 指定服务器是否查找客户机主机名

使用此伪指令指定服务器是否查找请求客户机的主机名。

格式

`DNS-Lookup {on | off}`

您使用的值会影响下列与服务器如何工作有关的项：

- 服务器的性能。使用缺省值 Off 会提高服务器的性能和响应时间，因为它不使用资源来执行主机名查找。
- 写日志文件时服务器记录的关于客户机的信息。
 - Off - 客户机由 IP 地址标识。
 - On - 客户机由主机名标识。
- 是否可以在保护设置、服务器组文件和访问控制表（ACL）文件中的地址模板上使用主机名。
 - Off - 不能使用地址模板的主机名；您必须使用 IP 地址。
 - On - 您可使用地址模板的主机名；您不能使用 IP 地址。

注：要在保护规则中使用域名，您必须设置 DNS-Lookup 伪指令为 On。

缺省值

DNS-Lookup Off

Enable - 启用 HTTP 方法

使用此伪指令指定哪些 HTTP 方法是服务器接受的。

您需要多少 HTTP 方法就可以启用多少。对于每个服务器要接受的方法，输入单个 Enable 伪指令。

格式

Enable *method*

如果特定 URL 的 Service 伪指令不存在，您可以使用 Enable 伪指令执行任何 HTTP 方法的定制编程。您在此伪指令上指定的程序覆盖该方法的标准处理。

Enable *method /path/fileDLL:function_name*

缺省值

Enable GET
Enable HEAD
Enable POST
Enable TRACE
Enable OPTIONS

EnableTcpNodelay — 启用 TCP NODELAY 套接字选项

使用此伪指令启用 TCP NODELAY 套接字选项。

当小的 IP 数据包（如 SSL 握手或简短 HTTP 响应）在高速缓存代理和客户机间传输时，EnableTcpNodelay 伪指令将改进性能。缺省情况下，启用所有套接字的 TCP NODELAY 选项。

格式

EnableTcpNodelay {All | HTTP | HTTPS | None}

缺省值

EnableTcpNodelay All

Error - 定制 Error 步骤

使用此伪指令指定 Error 步骤期间您要服务器调用的定制应用程序功能。当遇到错误时执行此代码，以提供定制错误例程。

格式

Error *request_template /path/file:function_name*

request_template

指定进一步确定是否调用应用程序功能的请求模板。规范可包含协议、域和主机；可放在斜杠（/）之后，并可使用星号（*）作为通配符。例如，/front_page.html、http://www.ics.raleigh.ibm.com、/pub*、/* 以及 * 均有效。

/path/file

指定编译程序的全限定文件名，包含扩展名。

function_name

指定程序内应用程序功能的名称。

示例

```
Error /index.html /ics/api/bin/icsext05.so:error_rtms
```

缺省值

无

ErrorLog - 指定记录服务器错误的文件

使用此伪指令指定您希望服务器记录内部错误处的路径与文件名。

注：如果您更改用户标识、组标识或日志目录路径的服务器缺省值，则创建新的目录并更新目录的许可和所有权。要使服务器能将信息写入用户定义的日志目录，请将此目录的许可设置为 755，并将用户定义的服务器用户标识设置为所有者。例如，如果您将服务器的用户标识从缺省值更改为 `jdoe`，您可将缺省日志目录更改到 `server_root/account`，则 `server_root/account` 目录必须具有许可 755 并从属于 `jdoe`。

如果它正在运行，则每天服务器在午夜启动新的日志文件。否则，服务器在您任何一天第一次启动它时启动新的日志文件。创建文件时，服务器使用您指定的文件名并附加日期后缀。日期后缀的格式是 `Mmddy`，其中 `Mmm` 表示月份的前三个字母；`dd` 表示日子；而 `yyyy` 表示年。

格式

```
ErrorLog /path/logs_directory/file_name
```

缺省值

- **Linux 和 UNIX 系统：** `ErrorLog /opt/ibm/edge/cp/server_root/logs/error`
- **Windows 系统：** `ErrorLog drive:\Program Files\IBM\edge\cp\logs\error`

ErrorPage - 为特定错误条件指定定制消息

使用此伪指令指定在服务器遇到特定错误条件时，发送到请求客户机的文件名称。配置文件 `ibmproxy.conf` 提供 `ErrorPage` 伪指令，它将错误关键字与错误消息文件关联。

要定制错误消息，您可以修改 `ErrorPage` 伪指令，将错误关键字与不同的文件相关联，也可修改提供的错误消息文件。例如您可以更改消息，在其中包含更多关于问题起因的信息和建议可能的解决方案。对于内部网络，您可以为用户提供可呼叫的联系人。

`ErrorPage` 伪指令可放置在配置文件中的任何地方。当发生错误时，根据配置文件中定义的映射规则处理文件。因此，您要发送的文件所在位置必须可以通过 `Fail`、`Map`、`NameTrans`、`Pass`、`Redirect` 和 `Service` 伪指令定义的映射规则到达。至少，您需要 `Pass` 伪指令，它允许服务器传递错误消息文件。

格式

```
ErrorPage keyword /path/filename.html
```

keyword

指定与错误条件关联的关键字之一。关键字列出在文件 `ibmproxy.conf` 中的 `ErrorPage` 伪指令。您不能更改关键字。

/path/filename.html

指定 Web 上的客户机查看到的错误文件的全限定 Web 名称。缺省错误消息文件在 `/HTML/errorpages/` 中。

示例

```
ErrorPage scriptstart /HTML/errorpages/scriptstart.htmls
```

在此示例中，当遇到 `scriptstart` 条件时，服务器向客户机发送在 `/HTML/errorpages/` 目录中找到的 `scriptstart.htmls` 文件。

下列 HTML 文本是文件可能包含的示例：

```
<HTML>
<HEAD>
<TITLE>SCRIPTSTART 条件的消息</TITLE>
</HEAD>
<BODY>
无法启动 CGI 程序。
<P>
<A HREF="mailto:admin@websvr.com">通知管理员</A>
此问题。
</BODY>
</HTML>
```

如果在服务器的配置文件中匹配以上路径的伪指令是 `PASS /* /wwwhome/*`，则此消息文件的全路径将是 `/wwwhome/HTML/errorpages/scriptstart.htmls`。

定制服务器返回的错误消息

每个错误条件都由一个关键字标识。要确定哪些错误消息是您要定制的，首先复查高速缓存代理提供的错误消息文件（可在 `/HTML/errorpages` 中找到）。错误页面包含错误号码、缺省消息、原因的说明和相应的恢复操作。

然后，执行下列其中一种操作以更改错误消息：

- 修改现有 HTML 或 HTMLS 文件（先要创建备份），或者用期望的文本新建 HTML 或 HTMLS 文件。您可使用 HTML 编辑器或 ASCII 编辑器。如果您要使用服务器端包含，必须使用 HTMLS 文件。
- 如果您使用不同的名称（或在不同的路径中）创建错误消息文件，请修改该关键字的 `ErrorPage` 伪指令以指向该文件。

错误条件、原因和缺省消息

所有错误关键字和缺省错误消息文件都在文件 `ibmproxy.conf` 的 `ErrorPage` 伪指令一节中列出。错误消息文件包含错误消息号码、关键字、缺省消息、说明和用户响应（操作）。

缺省值

许多缺省值包含于文件 `ibmproxy.conf`

如果您不修改对应一个错误条件的 `ErrorPage` 伪指令，则发送该条件的服务器缺省错误页面。

EventLog – 指定事件日志文件的路径

使用此伪指令指定事件日志路径与文件名。事件日志捕捉关于高速缓存自身的信息性消息。

注：如果您更改用户标识、组标识或日志目录路径的服务器缺省值，则创建新的目录并更新目录的许可和所有权。要使服务器能将信息写入用户定义的日志目录，请将此目录的许可设置为 755，并将用户定义的服务器用户标识设置为所有者。例如，如果您将服务器的用户标识从缺省值更改为 `jdoue`，您可将缺省日志目录更改到 `server_root/account`，则 `server_root/account` 目录必须具有许可 755 并从属于 `jdoue`。

如果它正在运行，则每天服务器在午夜启动新的日志文件。否则，服务器在您任何一天第一次启动它时启动新的日志文件。创建文件时，服务器使用您指定的文件名并附加日期后缀。日期后缀的格式是 `Mmddyyyy`，其中 `Mmm` 表示月份的前三个字母；`dd` 表示日子；而 `yyyy` 表示年。

格式

```
EventLog /path/logs_directory/file_name
```

缺省值

- **Linux 和 UNIX 系统：** `EventLog /opt/ibm/edge/cp/server_root/logs/event`
- **Windows 系统：** `EventLog drive:\Program Files\IBM\edge\cp\logs\event`

Exec – 运行 CGI 程序以匹配请求

使用此伪指令，为通过运行 CGI 程序接受和响应的请求指定模板。请求与 Exec 伪指令上的模板匹配之后，则请求就不与任何后继伪指令上的请求模板比较。

格式

```
Exec request_template program_path [Server_IP_address | host_name]
```

request_template

指定服务器通过运行 CGI 程序接受和响应请求的模板。

在 `request_template` 和 `program_path` 中，您可将星号 (*) 用作为通配符。与 `request_template` 通配符匹配的请求部件必须以包含 CGI 程序的文件名开始。

请求也可包含传递到 `PATH_INFO` 环境变量中 CGI 程序的其它数据。其它数据按照请求上 CGI 程序文件名开始的第一个斜体字符 (/)。根据 CGI 规范传递数据。

program_path

指定到包含服务器为请求执行的 CGI 程序的文件的完整路径。`program_path` 也必须包含通配符。通配符由包含 CGI 程序的文件名替换。

Exec 伪指令是循环的并适用于所有子目录。您不需要 `cgi-bin` 和 `admin-bin` 下每个目录的分隔 Exec 伪指令。

[Server_IP_address | host_name]

如果您使用多个 IP 地址或虚拟主机，则使用该参数指定 IP 地址或主机名。服务器仅对于来到此 IP 地址上服务器的请求或此主机使用伪指令。对于 IP 地址，这是服务器的网络连接的地址，而不是请求客户机的地址。

您可以指定 IP 地址（例如，240.146.167.72），或者您可以指定主机名（例如，`hostA.bcd.com`）。

此参数是可选的。没有此参数，则服务器对于所有请求使用伪指令，无论请求到达的 IP 地址或 URL 中的主机名。

通配符无法用于指定服务器 IP 地址。

示例

在以下示例中，如果您的服务器接受 `/idd/depts/plan/c92` 的请求，则它运行 `/depts/bin/plan.exe` 中的 CGI 程序并且 `c92` 作为输入传递到程序。

下例使用可选的 IP 地址参数。如果您的服务器接收 `/cgi-bin/` 开始的请求，它服务来自不同目录的请求，此目录基于请求到达的网络连接的 IP 地址。对于到达 `130.146.167.72` 的请求，服务器使用 `/CGI-BIN/customerA` 目录。对于到达任何与 `0.83.100.45` 地址连接的请求，服务器使用 `/CGI-BIN/customerB` 目录。

```
Exec /cgi-bin/* /CGI-BIN/customerA/* 130.129.167.72
Exec /cgi-bin/* /CGI-BIN/customerB/* 0.83.100.45
```

以下示例使用可选的主机名参数。如果您的服务器接收 `/cgi-bin` 开始的请求，它服务来自不同目录的请求，此目录基于 URL 中的主机名。对于到达 `hostA.bcd.com` 的请求，服务器使用 `/CGI-BIN/customerA` 目录。对于到达 `hostB.bcd.com` 的请求，服务器使用 `/CGI-BIN/customerB` 目录。

```
Exec /cgi-bin/* /CGI-BIN/customerA/* hostA.bcd.com
Exec /cgi-bin/* /CGI-BIN/customerB/* hostB.bcd.com
```

缺省值

- **Linux 和 UNIX 系统**

```
Exec /cgi-bin/* /opt/ibm/edge/cp/server_root/cgi-bin/*
Exec /admin-bin/* /opt/ibm/edge/cp/server_root/admin-bin/*
```

- **Windows 系统**

```
Exec server_root/cgi-bin/*
Exec server_root/admin-bin/*
Exec server_root/DOCS/admin-bin/*
```

ExportCacheImageTo - 将高速缓存内存导出到磁盘

使用此伪指令将高速缓存内容导出到转储文件中。当重新启动期间丢失内存高速缓存，或者当为多个代理部署同一高速缓存时，这是有用的。

格式

```
ExportCacheImageTo export_file_name
```

缺省值

无

ExternalCacheManager - 对 IBM WebSphere Application Server 中的动态高速缓存配置高速缓存代理

使用该伪指令配置高速缓存代理来识别能高速缓存动态创建资源的 IBM WebSphere Application Server (用高速缓存代理适配器模块配置)。高速缓存代理保存 JSP 结果的副本，这同样是存储在应用程序服务器的动态高速缓存中。高速缓存代理仅高速缓存来自 IBM WebSphere Application Server (其组标识与 ExternalCacheManager 相匹配) 的内容。

注意还需将 `Service` 伪指令添加到高速缓存代理配置文件以启用此特性。其它配置步骤对应用程序服务器同样是必需的。请参阅第 93 页的第 22 章，『动态地高速缓存生成的内容』以获取完整的信息。

格式

```
ExternalCacheManager External_Cache_Manager_ID Maximum_Expiry_Time
```

External_Cache_Manager_ID

指定给向代理提供服务的 IBM WebSphere Application Server 的标识。此标识必须与在应用程序服务器上 `dynacache.xml` 文件中的 `externalCacheGroup: 组标识` 属性中设置的标识匹配。

Maximum_Expiry_Time

设置经高速缓存资源的缺省到期时间代表外部高速缓存管理器。如果外部高速缓存管理器不使经高速缓存的资源在指定时间内无效，则此资源会在该指定时间内到期。该时间可按分或秒为单位进行指定。

示例

以下条目定义了一个位于 `www.xyz.com` 域内的外部高速缓存管理器（IBM WebSphere Application Server），其资源在 20 秒或更早便到期。

```
ExternalCacheManager IBM-CP-XYZ-1 20 seconds
```

缺省值

无

Fail – 拒绝匹配请求

使用此伪指令，为服务器不处理的请求指定模板。当请求与 `Fail` 伪指令上的模板匹配之后，则请求就不与任何后继伪指令上的请求模板相比较。

格式

```
Fail request_template [Server_IP_address | host_name]
```

request_template

为服务器要拒绝的请求指定模板。如果请求与模板匹配，则服务器向请求者发送错误消息。

在模板中，您可将星号用作通配符。必须明确匹配紧跟在 (/) 后面的否定号字符 (~)；不能使用通配符来匹配它。

[Server_IP_address | host_name]

如果您使用多个 IP 地址或虚拟主机，则使用该参数指定 IP 地址或主机名。服务器仅对于来到此 IP 地址上服务器的请求或此主机使用伪指令。对于 IP 地址，这是服务器的网络连接的地址，而不是请求客户机的地址。

您可以指定 IP 地址（例如，240.146.167.72），或者您可以指定主机名（例如，hostA.bcd.com）。

此参数是可选的。没有此参数，则服务器对于所有请求使用伪指令，无论请求到达的 IP 地址或 URL 中的主机名。

不能为服务器的 IP 地址指定通配符。

示例

在以下示例中，服务器拒绝 `/usr/local/private/` 开始的任何请求。

```
Fail /usr/local/private/*
```

下例使用可选的 IP 地址参数。如果请求到达 IP 地址为 240.146.167.7 的网络连接，服务器拒绝 `/customerB/` 开始的任何请求。如果请求到达 IP 地址为 0.83.100.45 的网络连接，服务器拒绝 `/customerA/` 开始的任何请求。

```
Fail /customerB/* 240.146.167.7
Fail /customerA/* 0.83.100.45
```

下例使用可选的主机名参数。如果请求到达 `hostA.bcd.com`，服务器拒绝 `/customerB/` 开始的任何请求。如果请求到达 `hostB.bcd.com`，服务器拒绝 `/customerA/` 开始的任何请求。

```
Fail /customerB/* hostA.bcd.com
Fail /customerA/* hostB.bcd.com
```

缺省值

无

flexibleSocks - 启用灵活的 SOCKS 实现

使用此伪指令指示代理使用 SOCKS 配置文件以确定要做的连接类型。

格式

```
flexibleSocks {on | off}
```

缺省值

```
flexibleSocks on
```

FTPDirInfo - 生成目录的欢迎或描述消息

使用此伪指令以使 FTP 服务器能生成目录的欢迎或描述性消息。此消息可作为 FTP 列表的部件可选地显示。FTPDirInfo 伪指令允许您控制显示消息的地方。

格式

```
FTPDirInfo {top | bottom | off}
```

top

在目录中文件列表之前，在页面顶部显示欢迎消息。

bottom

在目录中文件列表之后，在页面底部显示欢迎消息。

off

不显示欢迎页面。

缺省值

```
FTPDirInfo top
```

ftp_proxy - 指定 FTP 请求的另一个代理服务器

如果您的代理服务器是代理链的部件，则使用此伪指令指定另一个代理的名称，因而此服务器应联系 FTP 请求。您必须指定完整 URL，包含收尾斜杠字符 (/)。要获得有关使用可选域名或模板的信息，请参阅第 214 页的『no_proxy - 指定直接连接到域的模板』。

格式

```
ftp_proxy full_URL [domain_name_or_template]
```

示例

```
ftp_proxy http://outer.proxy.server/
```

缺省值

无

FTPUrlPath - 指定如何解释 FTP URL

使用此伪指令指定在 FTP URL 中的路径信息是被解释为相对于登录用户的工作目录，还是相对于根目录。

格式

```
FTPUrlPath {relative | absolute}
```

如果 FTPUrlPath 伪指令设置为 absolute，则登录用户的 FTP 工作目录必须包含于 FTP URL 路径。如果指定了 FTPUrlPath Relative，则登录用户的 FTP 工作目录必须从 FTP URL 路径省略。例如，要访问包含于登录用户工作目录 /export/home/user1 的文件 test1.html，取决于 FTPUrlPath 伪指令的设置，需要下列 URL 路径：

- 如果设置是 FTPUrlPath absolute，则需要的 URL 路径是 ftp://ftphost/export/home/user1/test1.html。
- 如果设置是 FTPUrlPath relative，则需要的 URL 路径是 ftp://ftphost/test1.html。

缺省值

无

Gc - 指定垃圾收集

使用此伪指令指定是否使用垃圾收集。如果启用高速缓存，则服务器使用垃圾收集进程以删除不再高速缓存的文件。基于文件的到期日期和其它代理伪指令值删除文件。通常，如果启用了高速缓存，就使用垃圾收集。如果不使用垃圾收集，则会没有效率地使用代理高速缓存。

格式

```
Gc {on | off}
```

缺省值

Gc On

GCAdvisor – 定制垃圾收集过程

使用此伪指令指定您要服务器用于垃圾收集的定制的应用程序。

格式

```
GCAdvisor /path/file:function_name
```

```
/path/file
```

指定编译程序的全限定文件名，包含扩展名。

```
function_name
```

指定程序内应用程序功能的名称。

示例

```
GCAdvisor /api/bin/customadvise.so:gcadv
```

GcHighWater – 指定垃圾收集何时开始

使用此伪指令指定必须填充到触发器垃圾收集的总计高速缓存容量的百分率。此百分率称为高水位标记。将高水位标记指定为高速缓存容量总计的百分率。继续垃圾收集，直到达到低水位标记 – 请参阅『GcLowWater – 指定垃圾收集何时结束』以获取关于设置的信息。可在 50 和 95 之间设置高水位标记的百分率。

格式

```
GcHighWater percentage
```

缺省值

```
GcHighWater 90
```

GcLowWater – 指定垃圾收集何时结束

使用此伪指令指定触发器垃圾收集结束的总计高速缓存容量的百分率。此百分率称为低水位标记。将低水位标记指定为高速缓存容量总计的百分率。它必须设置为低于高水位标记设置的值；请参阅『GcHighWater – 指定垃圾收集何时开始』以获取关于设置高水位标记的信息。

格式

```
GcLowWater percentage
```

缺省值

```
GcLowWater 60
```

gopher_proxy – 指定 Gopher 请求的另一个代理服务器

如果您的代理服务器是代理链的部件，则使用此伪指令指定另一个代理的名称，因此服务器应联系 Gopher 请求。您必须指定完整 URL，包含收尾斜杠 (/)。要获得有关使用可选域名或模板的信息，请参阅第 214 页的『no_proxy – 指定直接连接到域的模板』。

格式

```
gopher_proxy full_URL[domain_name_or_template]
```

示例

```
gopher_proxy http://outer.proxy.server/
```

缺省值

无

GroupId - 指定组标识

使用此伪指令指定服务器在访问文件之前更改到的组名和用户数。

如果您更改此伪指令，您必须手工停止服务器，然后再次启动它以使更改生效。如果您仅重新启动服务器，则更改不生效。（请参阅第 15 页的第 5 章，『启动和停止高速缓存代理』）。

注：如果您更改用户标识、组标识或日志目录路径的服务器缺省值，则创建新的目录并更新目录的许可和所有权。要使服务器能将信息写入用户定义的日志目录，请将此目录的许可设置为 755，并将用户定义的服务器用户标识设置为所有者。例如，如果您将服务器的用户标识从缺省值更改为 `jdoe`，从缺省日志目录更改到 `server_root/account`，则 `server_root/account` 目录必须具有许可 755 并从属于 `jdoe`。

格式

```
GroupId { group_name | group_number }
```

缺省值

AIX: GroupId nobody

HP-UX: GroupId other

Linux:

Red Hat: GroupId nobody

SuSE: GroupId nogroup

Solaris: GroupId nobody

HeaderServerName - 指定 HTTP 头中返回的代理服务器名称

使用此伪指令指定 HTTP 头中返回的代理服务器的名称

格式

```
HeaderServerName name
```

缺省值

无

Hostname - 指定服务器的全限定域名或 IP 地址

使用此伪指令指定返回到来自文件请求客户机的域名或 IP 地址。如果您指定域名，则域名服务器必须可将名称解析到 IP 地址。如果您指定 IP 地址，则不需要或访问域名服务器。

注：当设置一个数组时，必须在该数组的所有成员上同等地配置 Hostname 伪指令。

格式

Hostname {*name* | *IP address*}

缺省值

缺省情况下，此伪指令不会在初始配置文件中指定。如果您未在配置文件中指定此伪指令，则该值缺省为在域名服务器中定义的主机名。

http_proxy - 为 HTTP 请求指定另一个代理服务器

如果您的代理服务器是代理链的部件，则使用此伪指令指定另一个代理的名称，因而此服务器应联系 HTTP 请求。您必须指定完整 URL，包含收尾斜杠 (/)。要获得有关使用可选域名或模板的信息，请参阅第 214 页的『no_proxy - 指定直接连接到域的模板』。

格式

http_proxy *full_URL*[*domain_name_or_template*]

示例

```
http://outer.proxy.server/
```

缺省值

无

HTTPSCheckRoot - 过滤 HTTPS 请求

使用此伪指令指定高速缓存代理是否检索 URL 不安全的主页并尝试查找其中的标号。如果查找到标号，则将它们应用到安全请求。例如，如果您请求 `https://www.ibm.com/`，高速缓存代理检索 `http://www.ibm.com/`，搜索它的标号，并使用任何查找到的标号以过滤 `https://www.ibm.com/`。

如果 HTTPSCheckRoot 设置为 off，高速缓存代理不检索不安全的主页并搜索标号。

格式

HTTPSCheckRoot {on | off}

缺省值

HTTPSCheckRoot on

ICP_Address - 指定 ICP 查询的 IP 地址

使用此子伪指令以指定用于发送和接收 ICP 查询的 IP 地址。它必须包含在 <MODULEBEGIN> ICP 和 <MODULEEND> 伪指令之间。

格式

ICP_Address *IP_address*

缺省值

缺省情况下，此伪指令不会在初始配置文件中指定。如果您未在配置文件中指定此伪指令，则该值缺省为接受和发送所有接口上的 ICP 查询。

ICP_MaxThreads - 指定 ICP 查询的最大线程

使用此子伪指令以指定产生的用于侦听 ICP 查询的线程数。它必须包含在 **<MODULEBEGIN> ICP** 和 **<MODULEEND>** 伪指令之间。

注: 在 Redhat Linux 6.2 和更低版本中, 此数必须是低的, 因为每个进程可创建的线程的最大数较小。指定用于 ICP 的大量线程可以限制在服务请求中使用的可用线程数。

格式

`ICP_MaxThreads number_of_threads`

缺省值

`ICP_MaxThreads 5`

Occupier - 指定 Occupier 群集的成员

如果代理服务器是某个 ICP 群集的一部分, 则使用此子伪指令指定 ICP 对等。它必须包含在 **<MODULEBEGIN> ICP** 和 **<MODULEEND>** 伪指令之间。

当 ICP 群集中添加新的对等时, 必须把 ICP 对等信息添加到所有现有对等的配置文件中。一行一个对等。注意, 当前主机也可以包含在对等列表中。当 ICP 初始化时, 它会忽略当前主机的条目。这样就能够具有一个配置文件, 该文件可以复制到其它对等机器上, 而无需进行编辑以除去当前主机。

格式

`ICP_Peer hostname http_port icp_port`

hostname

对等的名称

http_port

对等的代理端口

icp_port

对等的 ICP 服务器端口

示例

以下行添加主机 `abc.xcompany.com`, 其代理端口是 80, ICP 端口是 3128 (作为一个对等主机)。

```
ICP_Peer abc.xcompany.com 80 3128
```

缺省值

无

ICP_Port - 指定 ICP 查询的端口号

使用此子伪指令以指定 ICP 服务器用于侦听 ICP 查询的端口号。它必须包含在 **<MODULEBEGIN> ICP** 和 **<MODULEEND>** 伪指令之间。

格式

`ICP_Port port_number`

缺省值

ICP_Port 3128

ICP_Timeout - 指定 ICP 查询的最大等待时间

使用此子伪指令以指定高速缓存代理等待 ICP 查询响应的最长时间。时间是以毫秒为单位指定的。它必须包含在 `<MODULEBEGIN> ICP` 和 `<MODULEEND>` 伪指令之间。

格式

ICP_Timeout *timeout_in_milliseconds*

缺省值

ICP_Timeout 2000

IgnoreURL - 指定不刷新的 URL

使用该伪指令以指定将不由高速缓存代理程序装入的 URL。当高速缓存代理程序从高速缓存的 URL 卸载链接的页面时，该伪指令有用。您可使用多个 IgnoreURL 伪指令的出现以指定不同的 URL 或 URL 掩码。该伪指令的值可包含星号 (*) 作为通配符，以应用掩码。

格式

IgnoreURL *URL*

示例

```
IgnoreURL http://www.yahoo.com/  
IgnoreURL http://*.ibm.com/*
```

缺省值

IgnoreURL */cgi-bin/*

imbeds - 指定是否使用服务器端包含处理

使用该伪指令以指定您是否要为文件系统、CGI 程序或其两者都提供的文件执行服务器端包含处理。服务器端包含处理是对 `ext/x-ssi-html` 内容类型的文件执行的。可选地，您可指定对 `text/html` 内容类型的文件执行服务器端包含处理。要获取更多关于内容类型的信息，请参阅第 162 页的『AddType - 为具有特殊后缀的文件指定数据类型』。

您可使用服务器端包含处理，以动态地将信息插入到正在返回的文件中。这种信息可包含日期、文件大小、文件的最后更改日期、CGI 或服务器端包含环境变量，或文本文件。服务器端包含处理仅对本地产生的文件执行。高速缓存代理不对代理的或者高速缓存的对象执行服务器端包含处理。

服务器端包含处理使服务器搜索您的文件中每次提供的特殊命令。这可以影响服务器性能并减慢到客户机的响应时间。

格式

imbeds {on | off | files | cgi | noexec} {SSIOnly | html}

on 为来自文件系统和 CGI 程序的文件执行服务器端包含处理。

off

不为任何文件执行服务器端包含处理。

files

仅为来自文件系统的文件执行服务器端包含处理。

cgi

仅为由 CGI 程序返回的文件执行服务器端包含处理。

noexec**SSIOnly**

为 text/x-ssi-html 内容类型的文件执行服务器端包含处理。

html

为 text/html 内容类型及 text/x-ssi-html 内容类型的文件执行服务器端包含处理。

服务器检查其检索的每个文件的内容类型，以及其处理的每个 CGI 程序的输出。

通常仅为具有 text/x-ssi/html 内容类型的文件执行服务器端包含处理。然而，您可以指定 text/html 内容类型的文件来执行服务器端包含处理。

注：服务器将 html、.html 和 .htm 作为 html 来处理。任何其它的则作为 SSIOnly 处理。

每个后缀具有一个以正确内容定义的 AddType 伪指令。如果您使用不同于 .htm 或 .html 的后缀，则确保 AddType 伪指令以 text/x-ssi/html 内容类型定义。

缺省值

imbeds on SSIOnly

ImportCacheImageFrom - 从文件导入高速缓存内存

使用此伪指令从转储文件导入高速缓存内容。当重新启动期间丢失内存高速缓存，或者当为多个代理部署同一高速缓存时，这是有用的。

格式

ImportCacheImageFrom *import_file_name*

缺省值

无

InheritEnv - 指定 CGI 程序继承的环境变量

使用该伪指令以指定您要让 CGI 程序继承的环境变量（除了特定于 CGI 处理的 CGI 环境变量）。

如果您不包含 InheritEnv 伪指令，则由 CGI 程序继承所有环境变量。如果您包含任何 InheritEnv 伪指令，则仅继承指定的关于 InheritEnv 伪指令的环境变量以及特定于 CGI 的环境变量。伪指令允许您可选地初始化继承的变量值。

格式

InheritEnv *environment_variable*

示例

```
InheritEnv PATH
InheritEnv LANG=ENUS
```

在该示例中，CGI 程序仅继承 PATH 和 LANG 环境变量，并且 LANG 环境变量以 ENUS 值初始化。

缺省值

无。缺省情况下，CGI 程序继承所有环境变量。

InputTimeout - 指定输入超时

使用该伪指令以设置客户机在连接到服务器后发送请求所允许的时间。客户机首先连接到服务器然后再发送请求。如果客户机不在该伪指令指定的时间内发送请求，则服务器关闭连接。以 hours、minutes（或 mins）和 seconds（或 secs）的任何组合来指定时间值。

格式

```
InputTimeout time
```

示例

```
InputTimeout 3 mins 30 secs
```

缺省值

```
InputTimeout 2 minutes
```

JunctionReplaceUrlPrefix — 当与 JunctionRewrite 插件协同使用时，替换 URL 而不是插入前缀

此伪指令将重设 JunctionRewrite 插件的缺省操作，允许代理改正 html 页面中的某些 URL 链接。此伪指令与 JunctionRewrite 伪指令协同使用。

JunctionReplaceUrlPrefix 伪指令将指示 JunctionRewrite 插件将 *url_pattern_1* 中的 URL 替换为 *url_pattern_2*，而不是在 URL 的开头插入前缀。

格式

```
JunctionReplaceUrlPrefix url_pattern_1 url_pattern_2
```

示例

```
JunctionReplaceUrlPrefix /server1.internaldomain.com/* /server1/*
```

在此示例中，假设 URL 是 /server1.internaldomain.com/notes.nsf，并假设前缀是 /server1。而不是插入前缀以重写 URL 到 /server1/server1.internaldomain.com/notes.nsf，JunctionRewrite 插件将把 URL 更改为 /server1/notes.nsf。

缺省值

无

JunctionRewrite - 启用 URL 重写

此伪指令允许高速缓存代理中的结点重写例程重新写来自源服务器的响应，以确保当使用结点时服务器相对 URL 获得适当源服务器的映射。如果当您设置 **JunctionRewrite on**，而未使用 UseCookie 选项时，结点重写插件也必须已启用。结点是通过代理映射规则定义的。

请参阅第 44 页的『使用 Cookie 替代 JunctionRewrite』和第 44 页的『用于扩展 JunctionRewrite 功能的样本 transmogriifier 插件』，以获得有关 JunctionRewrite 的其它信息。

格式

```
JunctionRewrite {on | on UseCookie | off}
```

缺省值

```
JunctionRewrite off
```

JunctionRewriteSetCookiePath — 当与 JunctionRewrite 插件协同使用时，重写 Set-Cookie 头中的路径选项

当 cookie 名匹配时，此伪指令将允许代理重写 Set-Cookie 头中的路径选项。如果响应需要结点，并且结点前缀已定义，则将在每条路径前插入前缀。此伪指令可与 JunctionRewrite 插件协同使用，或它可与 RewriteSetCookieDomain 伪指令协同使用。

格式

```
JunctionRewriteSetCookiePath cookie-name1 cookie-name2...
```

cookie-name

cookie 名在 Set-Cookie 头中。

缺省值

无

JunctionSkipUrlPrefix — 当与 JunctionRewrite 插件协同使用时，跳过已包含前缀的重写 URL

此伪指令将重设 JunctionRewrite 插件的缺省操作，跳过 URL 重写（如果 URL 模式已匹配）。此伪指令与 JunctionRewrite 插件协同工作，提供改正 html 页面中的某些 URL 链接的方法。通常，此伪指令用于跳过已包括前缀的 URL。

格式

```
JunctionSkipUrlPrefix url_pattern
```

示例

```
JunctionSkipUrlPrefix /server1/*
```

在此示例中，假设 URL 是 /server1/notes.nsf，并假设结点前缀是 /server1/。而不是重写 URL 到 /server1/server1/notes.nsf，JunctionRewrite 插件将跳过重写 URL，并且 URL 保持不变，为 /server1/notes.nsf。

缺省值

无

KeepExpired – 修改代理高速缓存的垃圾收集

使用此伪指令指定代理高速缓存的垃圾收集期间到期文件的处理。有效值为 `on` 和 `off`。当此伪指令为 `on` 时，到期的文件符合删除条件，并且是垃圾收集期间首先考虑要删除的文件。如果高速缓存是满的，则将首先删除到期的文件。当此伪指令为 `off` 时，到期的文件在垃圾收集期间删除，即使代理高速缓存不满。

格式

```
KeepExpired {on | off}
```

缺省值

```
KeepExpired off
```

KeyRing – 指定到密钥环数据库的文件路径

使用该伪指令以指定到密钥环数据库（服务器将用于 SSL 请求）的文件路径。密钥环文件通过 `iKeyman` 密钥管理器实用程序生成。

注：SSL 伪指令在 SuSE Linux 上不支持。

格式

```
KeyRing filename
```

示例

Windows: `KeyRing c:\Program Files\IBM\edge\cp\key.kdb`

Linux 和 UNIX: `KeyRing /etc/key.kdb`

缺省值

无

KeyRingStash – 指定到密钥环数据库的密码文件的文件路径

使用此伪指令指定到密钥环数据库密码文件的文件路径。密码文件是通过 `iKeyman` 密钥管理器实用程序在构建密钥环数据库文件时生成的。

注：SSL 伪指令在 SuSE Linux 上不支持。

格式

```
KeyRingStash file_path
```

示例

Windows: `KeyRingStash c:\Program Files\IBM\edge\cp\key.sth`

Linux 和 UNIX: `KeyRingStash /etc/key.sth`

缺省值

无

LimitRequestBody — 指定 PUT 或 POST 请求中最大主体大小

使用此伪指令控制 PUT 或 POST 请求中最大主体大小。LimitRequest 伪指令用于保护代理不受到攻击。

可以以千字节 (K)、兆字节 (M) 或千兆字节 (G) 指定该值。

格式

```
LimitRequestBody max_body_size {K | M | G}
```

缺省值

```
LimitRequestBody 10 M
```

LimitRequestFields — 指定客户机请求中的最大标题数

使用此伪指令指定可在客户机请求中发送的最大标题数。LimitRequest 伪指令用于保护代理不受到攻击。

格式

```
LimitRequestFields number_headers
```

缺省值

```
LimitRequestFields 32
```

LimitRequestFieldSize — 指定最大标题长度和请求行

使用此伪指令指定最大请求行长度，以及请求中每个标题的最大长度。LimitRequest 伪指令用于保护代理不受到攻击。

可以以字节 (B) 或千字节 (K) 指定该值。

格式

```
LimitRequestFieldSize max_hdr_length {B | K}
```

缺省值

```
LimitRequestFieldSize 4096 B
```

ListenBacklog — 指定服务器可执行的侦听备用客户机连接数

在发送连接拒绝的消息给客户机之前，使用该伪指令以指定服务器执行的侦听备用客户机连接数。此数取决于您的服务器可在几秒钟内处理的请求数。在客户机超时并异常终止连接之前，不要将它设置为高于服务器能处理的数。

注：如果 ListenBacklog 值大于由 TCP/IP 支持的 SOMAXCONN 值，则使用 SOMAXCONN 值。

格式

```
ListenBacklog number_of_requests
```

缺省值

```
ListenBacklog 128
```

LoadInlineImages - 控制嵌入图像的刷新

使用此伪指令指定是否由高速缓存代理程序检索内联图像。如果 `LoadInlineImages` 设置为 `on`，则也将高速缓存嵌入页面（该页面正在高速缓存）的图像。如果它设置为 `off`，则不高速缓存嵌入的图像。

格式

```
LoadInlineImages {on | off}
```

缺省值

```
LoadInlineImages on
```

LoadTopCached - 指定要刷新的受欢迎页面数

使用该伪指令以指示高速缓存代理程序访问前一晚的高速缓存访问日志，并装入请求最多的 URL。

此 `Caching` 伪指令必须设置为 `On`，并且在为 `LoadTopCached` 伪指令设置一个值时必须为 `CacheAccessLog` 伪指令设置一个值。

格式

```
LoadTopCached number_of_pages
```

缺省值

```
LoadTopCached 100
```

LoadURL - 指定要刷新的 URL

使用该伪指令以指定要由高速缓存代理程序装入到高速缓存的 URL。此配置文件中可包含多个 `LoadURL` 伪指令，但不能使用通配符。

格式

```
LoadURL url
```

示例

```
LoadURL http://www.ibm.com/
```

缺省值

无

Log - 定制 Log 步骤

使用该伪指令以指定 `Log` 步骤期间服务器调用的定制应用程序功能。此代码提供关闭连接后执行的日志及其它处理。

格式

```
Log request_template /path/file:function_name
```

request_template

指定进一步确定是否调用应用程序功能的请求模板。规范可包含协议、域和主机；

可放在斜杠 (/) 之后, 并可使用星号 (*) 作为通配符。例如, /front_page.html、http://www.ics.raleigh.ibm.com、/pub*、/* 以及 * 均有效。

/path/file

指定编译程序的全限定文件名, 包含扩展名。

function_name

指定程序内应用程序功能的名称。您必须提供打开、写及关闭功能的名称。

示例

```
Log /index.html /api/bin/icsextpgm.so:log_url
```

缺省值

无

LogArchive - 指定日志归档的行为

使用此伪指令指定归档例程的行为。此伪指令影响所有具有全局设置的日志。它用于指定日志是压缩还是清除还是什么都不执行。

如果您指定 `Compress`, 则使用 `CompressAge` 和 `CompressDeleteAge` 伪指令以指定何时压缩或删除日志。使用 `CompressCommand` 伪指令以指定要使用哪一命令及其参数。

如果您指定 `Purge`, 则使用 `PurgeAge` 和 `PurgeSize` 伪指令以指定何时清除日志。

格式

```
LogArchive {Compress | Purge | none}
```

Compress

指定归档例程压缩日志。

Purge

指定归档例程擦除日志。

none

指定归档例程不做任何事。

缺省值

```
LogArchive Purge
```

相关伪指令

- 第 177 页的『`CompressAge` - 指定何时压缩日志』
- 第 178 页的『`CompressDeleteAge` - 指定何时删除日志』
- 第 177 页的『`CompressCommand` - 指定压缩命令和参数』
- 第 212 页的『`Midnight` - 指定用于归档日志的 API 插件』
- 第 233 页的『`PurgeAge` - 指定日志的时效』
- 第 233 页的『`PurgeSize` - 指定日志压缩文档的大小限制』

LogFileFormat - 指定访问日志格式

使用此伪指令指定访问日志文件的文件格式。

格式

LogFileFormat {common | combined}

缺省情况下，日志以 NCSA 公共日志格式显示。相反，指定 combined，以 NCSA 组合日志格式显示日志。组合格式添加用于引用 URL、用户代理程序以及 Cookie 的字段（如果在当前请求中显示）。

缺省值

LogFileFormat common

LogToGUI（仅 Windows） – 在服务器窗口中显示日志条目

仅适用于 **Windows** 系统。当通过命令行运行代理时，使用此伪指令输出到访问日志。要优化服务器性能，缺省情况下，将此伪指令设置为 off（禁用）。

注：当代理作为服务运行时，此伪指令不起作用。

格式

LogToGUI {on | off}

缺省值

LogToGUI off

LogToSyslog — 指定是否将访问信息发送到系统日志（仅 Linux 和 UNIX）

仅 **Linux** 和 **UNIX** 系统。使用此伪指令指定服务器是否将访问请求及错误除了记录到访问和错误日志文件外还记录到系统日志。

格式

LogToSyslog {on | off}

在您指定要将错误日志信息写入系统日志文件之前，该系统日志文件必须显示在您的服务器上。您可以选择是记录访问信息还是记录错误信息，或者两者都记录。

要仅发送错误信息到系统日志，则将下列行添加到您的 /etc/syslog.conf 文件：

```
user.err syslog_output_file_for_error_information
```

要仅发送访问信息到系统日志，则将下列行添加到您的 /etc/syslog.conf 文件：

```
user.info syslog_info_file_for_access_information
```

要将错误及访问信息都发送到系统日志，则将下列两行都添加到您的 /etc/syslog.conf 文件：

以下列格式指定 *syslog_output_file* 和 *syslog_info_file*：

- **AIX:** /var/adm/name_of_syslog_file
- **HP-UX:** /var/adm/syslog/syslog.log
- **Linux:** /var/adm/messages
- **Solaris:** /var/adm/messages

在您创建系统日志文件后，您可以按命令重新启动系统日志：

```
kill -HUP 'cat /etc/syslog.pid'
```

缺省值

LogToSyslog Off

Map – 将匹配请求更改为新的请求字符串

使用该伪指令以指定您要更改为新的请求字符串的请求模板。在服务器更改请求后，它将获得新的请求字符串，并将其与后继伪指令上的请求模板相比较。

格式

```
Map request_template new_request [server_IP_address | host_name]
```

request_template

指定服务器更改然后继续将新的请求字符串与其它模板相比较的请求模板。

在模板中，您可将 (*) 作为通配符。必须明确匹配紧跟在 (/) 后面的否定号字符 (~)；不能使用通配符来匹配它。

new_request

指定服务器用于继续与后继伪指令上的请求模板相比较的新的请求字符串。如果 *request_template* 具有一个通配符，则以 *new_request* 指定的字符串可包含一个通配符。匹配 *request_template* 通配符的请求部件将插入到 *new_request* 中的通配符位置中。

[*server_IP_address* | *host_name*]

如果您使用多个 IP 地址或虚拟主机，则使用该参数指定 IP 地址或主机名。服务器仅对于来到此 IP 地址上服务器的请求或此主机使用伪指令。对于 IP 地址，这是服务器的网络连接的地址，而不是请求客户机的地址。

您可以指定一个 IP 地址（例如，240.146.167.72）或者可以指定一个主机名（例如，hostA.raleigh.ibm.com）。

此参数是可选的。没有此参数，则服务器对于所有请求使用伪指令，无论请求到达的 IP 地址或 URL 中的主机名。

不能为服务器的 IP 地址指定通配符。

示例

- 在下例中，服务器获得以 /stuff/ 开始的任何请求，并且将请求的 /stuff/ 部分更改为 /good/stuff/。原始请求上任何跟着 /stuff/ 的内容也包含在新的请求字符串内。因此，/stuff/whatsup/ 更改为 /good/stuff/whatsup/。服务器获得新的请求字符串，并继续把它与后继伪指令上的请求模板相比较。

```
Map /stuff/* /good/stuff/*
```

- 下例使用可选的 IP 地址参数。如果您的服务器接收以 /stuff/ 开始的请求，则服务器基于请求到达的网络连接 IP 地址将该请求更改为一个不同的请求字符串。对于入网 240.146.167.72 的请求，服务器将请求的 /stuff/ 部分更改为 /customerA/good/stuff/。对于进入任何与 0.83.100.45 地址连接的请求，服务器将请求的 /stuff/ 部分更改为 /customerB/good/stuff/。

```
Map /stuff/* /customerA/good/stuff/* 240.146.167.72
Map /stuff/* /customerB/good/stuff/* 0.83.104.45
```

- 下例使用可选的主机名参数。如果服务器接收以 `/stuff/` 开始的请求，则服务器基于 URL 中的主机名将该请求更改为一个不同的请求字符串。对于进入 `hostA` 的请求，服务器将请求的 `/stuff/` 部分更改为 `/customerA/good/stuff/`。对于进入 `hostB` 的请求，服务器将请求的 `/stuff/` 部分更改为 `/customerB/good/stuff/`。

```
Map /stuff/* /customerA/good/stuff/* hostA.bcd.com
Map /stuff/* /customerB/good/stuff/* hostB.bcd.com
```

缺省值

无

MaxActiveThreads – 指定有效线程的最大数

使用此伪指令以设置一次有效的线程最大数。如果到达该最大值，则服务器将保持新的请求直到另一个请求完成并且线程可用。通常，机器的功率越高，为此伪指令设置的值就越高。如果机器开始花费太多的时间在开销任务上（如，交换内存），则尝试减少该值。

格式

```
MaxActiveThreads number_of_threads
```

缺省值

```
MaxActiveThreads 100
```

MaxContentLengthBuffer – 指定动态数据的缓冲区大小

使用该伪指令以设置由服务器生成的动态数据的缓冲区大小。动态数据是从 CGI 程序、服务器端包含及 API 程序输出的。

可以以字节（B）、千字节（K）、兆字节（M）或千兆字节（G）指定该值。无论在数和值（B、K、M 和 G）之间有没有空格都不要紧。

格式

```
MaxContentLengthBuffer size
```

缺省值

```
MaxContentLengthBuffer 100 K
```

MaxLogFileSize – 指定每个日志文件的最大大小

使用此伪指令指定每个日志文件的最大大小。每个日志文件不能超出此伪指令定义的大小。一旦日志文件达到最大定义大小后，当前日志文件将关闭，并将创建同名的新日志文件，追加下一个增量的整数值。

注:

1. 高速缓存代理是 32 位应用程序，并打开其带有 32 位功能的日志文件。因为此约束，不要把 `MaxLogFileSize` 指定为大于 2 GB。如果当高速缓存代理在保持处理请求活动时尝试写入日志文件，而日志文件的大小超过 2 GB，则高速缓存代理可能挂起。
2. 在 Linux 和 UNIX 平台上，如果日志文件所驻留目录的许可权没有对至少 `ibmproxy` 守护程序在其下运行的组的写许可权，将不创建日志文件。换句话说，`ibmproxy.conf` 文件中用于记录伪指令的日志文件位置必须具有至少对 `ibmproxy.conf`

文件中 `GroupId` 伪指令定义的组的写许可权。当日志文件的缺省位置已更改，或者 `ibmproxy.conf` 文件中缺省 `UserId` 或 `GroupId` 伪指令已更改时，这是唯一的问题。

可以下列单位之一指定最大大小：字节（B）、千字节（K）、兆字节（M）或千兆字节（G）。

格式

```
MaxLogFileSize maximum {B | K | M | G}
```

缺省值

```
MaxLogfileSize 128 M
```

MaxPersistRequest – 指定一个持久连接上接收的最大请求数

使用此伪指令指定服务器接收持久连接的最大请求数。当确定该数时，请考虑用于您页面的图像数。每个图像需要一个单独的请求。

格式

```
MaxPersistRequest number
```

缺省值

```
MaxPersistRequest 5
```

MaxQueueDepth – 指定队列中最大 URL 数

使用此伪指令指定未完成页面检索请求的高速缓存代理程序的队列的最大深度。如果您具有一个带有大量内存的巨大系统，则您可以在未消耗掉所有可用内存的情况下定义一个巨大的页面检索请求队列。

到高速缓存的 URL 队列是在每个高速缓存代理程序运行开始时确定的。如果您指示高速缓存代理程序以按照到其它 URL 的超文本链接，则这些其它 URL 不计在高速缓存队列内。达到 `MaxURLs` 伪指令中指定的值后，即使队列中有更多的 URL，高速缓存代理程序也会停止。

格式

```
MaxQueueDepth maximum_depth
```

缺省值

```
MaxQueueDepth 250
```

MaxRuntime – 指定高速缓存代理程序运行的最大时间

使用此伪指令指定高速缓存代理程序在某一运行期间检索 URL 的最大时间。值 0 意味着高速缓存代理程序将运行直至完成。

格式

```
MaxRuntime {0 | maximum_time}
```

示例

```
MaxRuntime 2 hours 10 minutes
```


缺省值

MaxRuntime 2 hours

MaxSocketPerServer - 指定服务器的最大打开套接字数

使用此伪指令设置打开套接字的最大数，以维护任何源服务器。仅当 ServerConnPool 伪指令设置为 on 时，才使用此伪指令。

格式

MaxSocketPerServer *num*

示例

MaxSocketPerServer 10

缺省值

MaxSocketPerServer 5

MaxUrls - 指定要刷新的最大 URL 数

使用此伪指令指定高速缓存代理程序将在某一运行期间检索的 URL 的最大数。值 0 意味着无限制。当使用高速缓存代理程序的自动方式时，LoadURL 和 LoadTopCached 伪指令优先于 MaxURLs。

格式

MaxURLs *maximum_number*

缺省值

MaxURLs 2000

Member - 指定一个数组成员

使用此伪指令指定由服务器（使用远程高速缓存访问）共享的数组成员。

注：当设置一个数组时，在该数组的所有成员上同等地配置 Hostname 伪指令。

格式

```
Member name {  
  subdirective  
  subdirective  
  .  
  .  
}
```

包含下列子伪指令：

RCAAddr

此必需的子伪指令标识 RCA 通信的 IP 地址或主机名。

RCAPort

该必需的子伪指令标识 RCA 通信端口。此端口号必须大于 1024 且小于 65535。

CacheSize {*n bytes* | *n Kbytes* | *n Mbytes* | *n Gbytes*}

此必需的子指令标识此成员的高速缓存的大小，它必须为一个正值。

[Timeout *n* milliseconds | *n* seconds | *n* hours | *n* days | *n* months | *n* years | forever]

标识等待该成员的时间。*n* 必须为正整数。Timeout 是可选的；缺省值为 1000 毫秒。超时值通常以秒或毫秒为单位设置。

[BindSpecific {On | Off}]

允许通信在专用子网上发生，并提供了安全性测量。BindSpecific 是可选的；缺省值为 On。

[ReuseAddr {On | Off}]

允许数组更快地重新接合；将其设置为 On 允许其它进程占用端口，这样会导致未定义的行为。ReuseAddr 是可选的；缺省值为 Off。

示例

```
Member bittersweet.chocolate.ibm.com {
  RCAAddr      127.0.0.1
  RCAPort      6294
  CacheSize    25G
  Timeout      500 milliseconds
  BindSpecific On
  ReuseAddr    Off
}
```

缺省值

无

Midnight – 指定用于归档日志的 API 插件

使用此伪指令指定在午夜运行的应用程序插件，以归档日志。该伪指令在安装期间初始化。如果您未在此配置文件中包含此伪指令，则不执行归档。

格式

Midnight */path/file:function_name*

/path/file

指定编译程序的全限定文件名，包含扩展名。

function_name

指定程序内应用程序功能的名称。

缺省值

- **Linux 和 UNIX:** Midnight /usr/lib/archive.so:begin
- **Windows:** Midnight C:\Program Files\IBM\edge\cp\bin\archive.dll:begin

NameTrans – 定制名称转换步骤

使用此伪指令指定服务器在名称转换步骤期间调用的定制应用程序功能。该代码为转换请求中的虚拟路径到服务器上的物理路径以及映射 URL 到特定对象提供机制。

注: 它不是一个终端映象规则。变换的 URL 仍必须匹配其中一个终端映象规则伪指令，例如，Exec、Fail、Map、Pass、Redirect 和 Service。

格式

```
NameTrans request_template /path/file:function_name  
[Server_IP_address | host_name]
```

request_template

指定进一步确定是否调用应用程序功能的请求模板。规范可包含协议、域和主机；可放在斜杠（/）之后，并可使用星号（*）作为通配符。例如，/front_page.html、http://www.ics.raleigh.ibm.com、/pub*、/* 以及 * 均有效。

/path/file

指定编译程序的全限定文件名，包含扩展名。

function_name

指定程序内应用程序功能的名称。

[*Server_IP_address* | *host_name*]

如果您正在使用多个 IP 地址或虚拟主机，则确定是否要仅为进入特定 IP 地址或特定主机的请求调用应用程序功能。

不能为服务器的 IP 地址指定通配符。

注：该伪指令必须在一行上输入，即使为便于读取它显示为两行。

示例

```
NameTrans /index.html /api/bin/icsextpgm.so:trans_url
```

缺省值

无

NoBG – 在前台运行高速缓存代理进程

在 Linux 和 UNIX 平台上，使用此伪指令可防止高速缓存代理服务器进程在后台自动运行。缺省情况下此伪指令设置为 off，其格式为：

```
NoBG [on | off]
```

注：ibmproxy 命令的 -nobg 选项在 Windows 系统中不可用。

示例

```
NoBG on
```

缺省值

```
NoBG off
```

NoCaching – 指定不高速缓存其 URL 与模板匹配的那些文件

使用此伪指令指定服务器不高速缓存带有匹配指定的模板的 URL 的文件。您可以在配置文件中包含多个此伪指令。每个模板包含一个单独的伪指令。URL 模板必须包含协议。

如果既没有设置 CacheOnly 又没有设置 NoCaching 伪指令，则任何一个 URL 都是高速缓存的候选者。

格式

NoCaching *URL_pattern*

示例

NoCaching http://joke/*

缺省值

无

NoLog – 抑制匹配模板的特定的主机或域的日志条目

使用此伪指令指定您不记录从匹配指定模板的特定主机或域发出的访问请求。例如，您可能不想访问来自本地主机的访问请求。

您可以在您的配置文件中出现多个此伪指令。如果您以一个或多个空格分隔多个模板，则您也可以将它们放置在相同的伪指令上。您可以在这些模板上使用主机名或 IP 数地址。

注：要使用主机名模板，您必须设置 DNS-Lookup 伪指令为 On。如果 DNS-Lookup 伪指令设置为 Off（缺省值），则您仅可使用 IP 地址模板。

格式

NoLog {*host_name* | *IP_address*} [...]

示例

NoLog 128.0.* *.edu localhost.*

缺省值

无

no_proxy – 指定直接连接到域的模板

如果您在使用伪指令 http_proxy、ftp_proxy 或 gopher_proxy 用于代理链接，则您可使用此伪指令指定服务器直接连接到的（而非通过代理）域。

指定一个作为域名字符串或域名模板字符串的值。用逗号（,）分隔字符串中的每个条目。不要在字符串中使用任何空格。

在该伪指令上输入的模板与在其它伪指令上输入的不同。最重要的是，您无法使用通配符（*）。您可以通过仅包含域名的最后部分来指定模板。服务器直接连接到任何以匹配您指定的模板的字符串结尾的域。此伪指令仅应用到应用代理链接，并且等价于 SOCKS 配置文件中的直接的 @/= 行。

格式

no_proxy *domain_name_or_template*[,...]

示例

no_proxy www.someco.com,.raleigh.ibm.com,.some.host.org:8080

在此示例中，服务器不为下列请求而通过代理：

- 到以 `www.someco.com` 结尾的域的任何请求
- 到以 `.raleigh.ibm.com` 结尾的域的任何请求，例如，`blugrass.raleigh.ibm.com` 或 `keystone.raleigh.ibm.com`
- 到以 `.some.host.org` 结尾的域端口 `8080` 的任何请求，例如 `myname.some.host.org:8080`。（它不包含到相同的域的任何其它端口的请求，例如 `myname.some.host.org`，其假设了缺省端口 `80`。）

缺省值

无

NoProxyHeader - 将客户机头指定到块

使用此伪指令以将客户机 URL 头指定到块。可以阻塞由客户机发送的任何 HTTP 头（包含必需的头）。当阻塞头时，请务必小心。公共头包含：

- **Pragma:** - 通常用于指示带有高速缓存的浏览器和服务器以从每次请求文件的源服务器获取文件。
- **Referer:** - 可以从其获得 Request-URI 的文件的 URL。

请参阅 HTTP 协议规范以获得关于这些头及其它头的详细信息。您可多次指定此伪指令。

格式

`NoProxyHeader header`

示例

`NoProxyHeader Referer:`

缺省值

无

NumClients - 指定要使用的高速缓存代理程序线程数

使用该伪指令以指定高速缓存代理程序用于检索队列中的页面的线程数。基于内部网络速度及到因特网的连接速度的线程数。可允许的范围是 1 到 100。

注：使用多于六个的线程可能会导致内容服务器上的请求过快。

格式

`NumClients number`

缺省值

`NumClients 4`

ObjectType - 定制对象类型步骤

使用此伪指令指定对象类型步骤期间服务器调用的定制应用程序功能。此代码将请求的对象定位在文件系统中，并指定其 MIME 类型。

格式

`ObjectType request_template /path/file:function_name`

request_template

指定进一步确定是否调用应用程序功能的请求模板。规范可包含协议、域和主机；可放在斜杠（/）之后，并可使用星号（*）作为通配符。例如，/front_page.html、http://www.ics.raleigh.ibm.com、/pub*、/* 以及 * 均有效。

/path/file

指定编译程序的全限定文件名，包含扩展名。

function_name

指定程序内应用程序功能的名称。

示例

```
ObjectType /index.html /api/bin/icsextpgm.so:obj_type
```

缺省值

无

OutputTimeout – 指定输出超时

使用此伪指令设置服务器允许的最大时间，以将输出发送到客户机。时间限制应用于本地文件的请求以及作为代理服务器的请求。时间限制不应用于启动本地 CGI 程序的请求。

如果服务器不在该伪指令指定的时间限制内发送完整的响应，则服务器断开连接。以 hours、minutes（或 mins）和 seconds（或 secs）的任何组合来指定时间值。

格式

```
OutputTimeout time
```

缺省值

```
OutputTimeout 30 minutes
```

PacFilePath – 指定包含 PAC 文件的目录

使用此伪指令指定包含以远程配置 PAC 文件格式生成的代理自动配置文件的目录。

格式

```
PacFilePath directory_path
```

缺省值

- **Windows:** PacFilePath c:\Program Files\IBM\edge\cp\HTML\pacfiles
- **Linux 和 UNIX:** PacFilePath /opt/ibm/edge/cp/server_root/pub/pacfiles

Pass – 指定接受请求的模板

使用此伪指令为您要接收和响应来自服务器文件的请求指定模板。当请求与 Pass 伪指令上的模板匹配之后，则请求就不与任何后继伪指令上的请求模板相比较。

格式

```
Pass request_template [file_path [server_IP_address | host_name]]
```

request_template

为您要接收和响应来自服务器文件的请求指定模板。

在模板中，您可将 (*) 作为通配符。必须明确匹配紧跟在 (/) 后面的否定号字符 (~)；不能使用通配符来匹配它。

[file_path]

指定服务器要返回的文件路径。如果 *request_template* 具有一个通配符，则 *file_path* 可包含一个通配符。匹配 *request_template* 通配符的请求部件将插入到 *file_path* 中的通配符位置中。

此参数是可选的。如果您不指定路径，则请求本身作为路径来使用。

[server_IP_address | host_name]

如果您使用多个 IP 地址或虚拟主机，则使用该参数指定 IP 地址或主机名。服务器仅对于来到此 IP 地址上服务器的请求或此主机使用伪指令。对于 IP 地址，这是服务器的网络连接的地址，而不是请求客户机的地址。

您可以指定一个 IP 地址（例如，240.146.167.72）或者可以指定一个主机名（例如，hostA.raleigh.ibm.com）。

此参数是可选的。没有此参数，则服务器对于所有请求使用伪指令，无论请求到达的 IP 地址或 URL 中的主机名。

不能为服务器的 IP 地址指定通配符。

示例

- 在下例中，服务器响应以 /updates/parts/ 开头的响应，其文件来自所列的路径，并取决于操作系统。任何紧跟在 /updates/parts/ 后的内容也用于指定此文件。

Linux 和 UNIX 系统: Pass /updates/parts/*
/opt/ibm/edge/cp/server_root/pub/*

Windows 系统: Pass /updates/parts/* c:\Program Files\IBM\edge\cp\pub*

- 在下例中，服务器响应以 /gooddoc/ 开头的请求，其文件来自目录 /gooddoc。因此服务器响应请求 /gooddoc/volume1/issue2/newsletter4.html，其中文档在文件 /gooddoc/volume1/issue2/newsletter4.html 中。

Pass /gooddoc/*

- 下例使用可选的 IP 地址参数。如果服务器接收以 /parts/ 开始的请求，则服务器基于请求到达的网络连接 IP 地址，从一个不同的目录返回文件。对于进入 240.146.167.72 的请求，服务器从 /customerA/catalog/ 返回文件。对于进入任何与地址 0.83.100.45 连接的请求，服务器将从 /customerB/catalog/ 返回文件。

Pass /parts/* /customerA/catalog/* 240.146.167.72
Pass /parts/* /customerB/catalog/* 0.83.100.45

- 下例使用可选的主机名参数。如果服务器接收以 /parts/ 开始的请求，则服务器基于 URL 中的主机名，从一个不同的目录返回一个文件。对于进入 hostA 的请求，服务器从 /customerA/catalog/ 返回文件。对于进入 hostB 的请求，服务器从 /customerB/catalog/ 返回文件。

AIX 系统

Pass /Admin/* /usr/lpp/internet/server_root/Admin/*
Pass /Docs/* /usr/lpp/internet/server_root/Docs/*
Pass /errorpages/* /usr/lpp/internet/server_root/pub/errorpages/*
Pass /* /usr/lpp/internet/server_root/pub/*

Solaris、HP-UX 和 Linux 系统

```

Pass /Admin/* /opt/ibm/edge/cp/server_root/Admin/*
Pass /Docs/* /opt/ibm/edge/cp/server_root/Docs/*
Pass /errorpages/* /opt/ibm/edge/cp/server_root/pub/errorpages/*
Pass /* /opt/ibm/edge/cp/server_root/pub/*

```

缺省值

AIX 系统

```

Pass /Admin/* /usr/lpp/internet/server_root/Admin/*
Pass /Docs/* /usr/lpp/internet/server_root/Docs/*
Pass /errorpages/* /usr/lpp/internet/server_root/pub/errporpages/*
Pass /* /usr/lpp/internet/server_root/pub/*

```

HP-UX、Linux 和 Solaris 系统

```

Pass /Admin/* /opt/ibm/edge/cp/server_root/Admin/*
Pass /Docs/* /opt/ibm/edge/cp/server_root/Docs/*
Pass /errorpages/* /opt/ibm/edge/cp/server_root/pub/errorpages/*
Pass /* /opt/ibm/edge/cp/server_root/pub/*

```

Windows 系统

```

Pass /icons/* C:\Program Files\IBM\edge\cp\icons\*
Pass /Admin/* C:\Program Files\IBM\edge\cp\Admin\*
Pass /Docs/* C:\Program Files\IBM\edge\cp\Docs\*
Pass /erropages/* C:\Program Files\IBM\edge\cp\pub\errorpages\*
Pass /* C:\Program Files\IBM\edge\cp\pub\*

```

PersistTimeout - 指定等待客户机发送另一个请求的时间

使用此伪指令指定取消持久连接之前服务器在客户机请求间的等待的时间。此时间可指定为任何有效时间增量，但通常指定为秒或分钟。

服务器使用一个不同的超时伪指令（InputTimeout），以确定在建立连接后等待客户机发送第一个请求的时间。要获取有关输入超时的更多信息，请参阅第 201 页的『InputTimeout - 指定输入超时』。

服务器发送其第一个响应后，它使用为 PersistTimeout 伪指令设置的值以确定在取消持久连接前等待每个后继请求的时间。

格式

```
PersistTimeout time
```

缺省值

```
PersistTimeout 4 seconds
```

PICSDBLookup - 定制 PICS 标号检索步骤

使用该伪指令以指定服务器调用的定制应用程序功能，以检索指定的 URL 的 PICS 标号。功能可以为请求的文件动态地创建 PICS 标号，也可以在一个备用文件或数据库中搜索 PICS 标号。

格式

```
PICSDBLookup /path/file:function_name
```

```
/path/file
```

指定编译程序的全限定文件名，包含扩展名。

function_name

指定程序内应用程序功能的名称。

示例

```
PICSDBLookup /api/bin/icsext05.so:get_pics
```

缺省值

无

PidFile (仅 Linux 和 UNIX) — 指定在其存储高速缓存代理的进程标识的文件

仅 **Linux** 和 **UNIX**。使用此伪指令指定包含高速缓存代理的进程标识的文件的位置。当服务器进程开始时，它将在文件中记录其进程标识 (PID)。如果服务器的多个实例正在单个系统上运行，则每个实例必须具有其自己的 `PidFile` 伪指令。

格式

```
PidFile path_to_pid_file_info
```

示例

```
PidFile /usr/pidinfo
```

缺省值

- 如果指定 `ServerRoot` 伪指令: `PidFile server_root /ibmproxy-pid`
- 如果未指定 `ServerRoot` 伪指令: `PidFile /tmp/ibmproxy-pid`

插件模块伪指令

下面所列的伪指令已添加到高速缓存代理 `ibmproxy.conf` 文件中，以启用新的功能部件和插件。“配置和管理”表单不能用于编辑多数这些伪指令。必须使用标准文本编辑器（如 `vi` 或 `emacs`）来手工编辑它们。关于每个伪指令的进一步信息以字母次序出现在本章中。

- 第 191 页的『`ExternalCacheManager` - 对 IBM WebSphere Application Server 中的动态高速缓存配置高速缓存代理』
- 第 197 页的『`ICP_Address` - 指定 ICP 查询的 IP 地址』
- 第 198 页的『`ICP_Port` - 指定 ICP 查询的端口号』
- 第 199 页的『`ICP_Timeout` - 指定 ICP 查询的最大等待时间』
- 第 198 页的『`Occupier` - 指定 `Occupier` 群集的成员』
- 第 198 页的『`ICP_MaxThreads` - 指定 ICP 查询的最大线程』
- 第 242 页的『`SignificantURLTerminator` - 为 URL 请求指定终止代码』
- 第 243 页的『`SSLCertificate` - 为证书指定密钥标号』
- 第 245 页的『`SSLOnly` - 对 HTTP 请求禁用侦听器线程』

在 `ibmproxy.conf` 文件中，用于配置高速缓存代理插件模块的伪指令必须用以下格式输入：

```
<MODULEBEGIN> plugin name
subdirective1
subdirective2

<MODULEEND>
```

每个插件程序分析 `ibmproxy.conf` 文件，并只读其拥有的子伪指令块。该高速缓存代理语法分析器忽略 `<MODULEBEGIN>` 和 `<MODULEEND>` 间的每个内容。

高速缓存代理插件模块和一些新的功能部件需要将 API 伪指令添加到 `ibmproxy.conf` 文件。因为代理服务器按插件模块列出的顺序与其进行交互，所以当您在代理配置文件中对伪指令排序时，请务必小心。Prototype 伪指令（以注释形式）已添加到 `ibmproxy.conf` 文件的 API 部分。这些 API 伪指令是按有目的的顺序进行排列的。当添加 API 伪指令以启用新功能和插件模块时，请按配置文件原型部分中显示的顺序排列这些伪指令。或者，取消注释并编辑（如果需要）API 伪指令，以包含对每个所需功能或插件的支持。当这些与产品一起提供后，添加用户生成的插件模块。

Port – 指定服务器侦听请求的端口

使用此伪指令指定服务器侦听请求所在的端口号。标准的 HTTP 端口号为 80。为其它 TCP/IP 应用程序保留低于 1024 的其它端口号，并且不准使用它们。用于代理 Web 服务器的公共端口是 8080 和 8008。

当使用除了 80 以外的端口时，客户机必需包含关于到服务器的请求的特定端口号。端口号放置在 URL 的主机名后，并紧跟在一个冒号之后。例如，从浏览器，URL `http://www.turfco.com:8008/` 从在 8008 上侦听的 `www.turfco.com` 主机请求缺省欢迎页面。

当启动此服务器时，您可使用 `ibmproxy` 命令上的 `-p` 选项来覆盖此设置。

格式

Port *number*

如果您更改此伪指令，您必须手工停止服务器，然后再次启动它以使更改生效。如果您仅重新启动服务器，则服务器将不识别更改。（请参阅第 15 页的第 5 章，『启动和停止高速缓存代理』）。

缺省值

Port 80

PostAuth – 定制 PostAuth 步骤

使用此伪指令指定 PostAuth 步骤期间服务器调用的定制应用程序功能。无论从先前步骤或从其它 PostAuth 处理程序得到的返回码怎么样，都执行此代码。它允许您清除分配用于处理请求的任何资源。

格式

PostAuth */path/file:function_name*

/path/file

指定编译程序的全限定文件名，包含扩展名。

function_name

指定程序内应用程序功能的名称。

示例

```
AuthExit /ics/api/bin/icsext05.so:post_exit
```

缺省值

无

PostExit – 定制 PostExit 步骤

使用此伪指令指定 PostExit 步骤期间服务器调用的定制应用程序功能。不管从先前步骤或其它 PostExit 处理程序得到的返回码怎么样，都执行此代码。它允许您清除分配用于处理请求的任何资源。

格式

```
PostExit /path/file:function_name
```

/path/file

指定编译程序的全限定文件名，包含扩展名。

function_name

指定程序内应用程序功能的名称。

示例

```
PostExit /ics/api/bin/icsext05.so:post_exit
```

缺省值

无

PreExit – 定制 PreExit 步骤

使用此伪指令指定 PreExit 步骤期间服务器调用的定制应用程序功能。在读客户机请求后但在其它处理发生前，此代码将执行。您可以在该步骤期间调用 GoServe 模块。

格式

```
PreExit /path/file:function_name
```

/path/file

指定编译 DLL 的全限定文件名，包含扩展名。

function_name

指定程序内应用程序功能的名称。

示例

```
PreExit /ics/api/bin/icsext05.so:pre_exit
```

缺省值

无

Protect – 为匹配模板的请求激活保护设置

使用该伪指令为匹配模板的请求激活保护设置规则。

注：为了保护正常工作，在配置文件中 `DefProt` 和 `Protect` 伪指令必须放置在任何 `Pass`、`Exec` 或 `Proxy` 伪指令之前。

用 `protection` 子伪指令定义保护设置。`Protect` 伪指令的格式取决于您是要包含 `protection` 子伪指令的标号或文件，还是要包含作为 `Protect` 伪指令一部分的 `protection` 子伪指令。

格式

此参数可使用下列格式：

- `Protect` 伪指令可指定为一个包含 `protection` 子伪指令的单独文件的全路径与文件名。它也可由保护设置标号名称（此名称与先前在 `Protection` 伪指令上定义的名称相匹配）指定；此 `Protection` 伪指令包含 `protection` 子伪指令。使用此格式：

```
Protect request_template [setup_file | label[
    [FOR Server_IP_address | host_name]
```

注：该伪指令必须输入在一行上，虽然在此显示为两行。

- 您可在 `Protect` 伪指令上指定实际的 `protection` 子伪指令内联。这些子伪指令必须加大括号（{}）。左大括号字符必须是 `Protect` 一行上的最后一个字符。后面每个子伪指令各占一行。右大括号字符必须在最后一行子伪指令后面独占一行。大括号中不能出现任何注释行。要包含 `protection` 子伪指令内联作为 `Protect` 伪指令的一部分，格式按如下所示：

```
Protect request_template [FOR Server_IP_address | hhost_name]
    subdirective value
    subdirective value
    .
    .
    .
}
```

使用下列参数：

`request_template`

指定激活保护的请求模板。服务器将入网客户机请求与模板相比较，如果匹配，则激活保护。

`[setup_file | label]`

如果您正在指向包含 `protection` 子伪指令的标号或文件，则此参数指定要激活的与 `request_template` 相匹配的请求的保护设置。

此参数是可选的。如果省略，则通过包含匹配模板的最近的 `DefProt` 伪指令来定义保护设置。

`[FOR server_IP_address | host_name]`

如果您使用多个 IP 地址或虚拟主机，则使用该参数指定 IP 地址或主机名。服务器仅对于来到此 IP 地址上服务器的请求或此主机使用伪指令。对于 IP 地址，这是服务器的网络连接的地址，而不是请求客户机的地址。如果保护 IP 地址，那么 IP 地址和全限定主机名都将受到保护。然而，如果服务器是在使用非全限定主机名的名称（如，使用主机名文件中的条目）的网络中调用的，那么它不受保护。

示例：

```
Protect http://x.x.x.x PROT-ADMIN
```

在 Web 浏览器中：

- `http://x.x.x.x` 是受保护的

- `http://hostname.example.com` 是受保护的
- `http://hostname` 是不受保护的

示例:

```
Protect http://hostname.example.com PROT-ADMIN
```

在 Web 浏览器中:

- `http://x.x.x.x` 是不受保护的
- `http://hostname.example.com` 是受保护的
- `http://hostname` 是不受保护的

您可指定 IP 地址 (例如, FOR 240.146.167.72), 或者您可指定主机名 (例如, FOR hostA.bcd.com)。

通配符无法用于指定服务器 IP 地址。

此参数是可选的。没有此参数, 则服务器对于所有请求使用伪指令, 无论请求到达的 IP 地址或 URL 中的主机名。

注: 此 `[server_IP_address | host_name]` 参数与 `[setup_file | label]` 参数或 `subdirective value` 参数一起使用。

- 要将 `[server_IP_address | host_name]` 与 `[setup_file | label]` 一起使用, 您必须将 FOR 或一些其它字符串 (不带空格) 放在 `setup_file | label]` 参数与 `server_IP_address | host_name]` 参数之间。
- 要将 `server_IP_address | host_name]` 与 `subdirective value` 参数一起使用, 请不要在 `IP_address` 或 `host_name` 之前包含 FOR。

subdirective value

要包含 `protection` 子伪指令作为 `Protect` 伪指令的一部分, 则使用此参数。要获取 `protection` 子伪指令的描述, 请参阅下列:

- 第 226 页的『AuthType - 指定认证类型』
- 第 226 页的『DeleteMask - 指定允许以删除文件的用户名、组和地址』
- 第 226 页的『GetMask - 指定允许以获取文件的用户名、组和地址』
- 第 227 页的『GroupFile - 指定关联的组文件的位置』
- 第 227 页的『Mask - 指定允许发出 HTTP 请求的用户名、组和地址』
- 第 227 页的『PasswdFile - 指定关联的密码文件的位置』
- 第 227 页的『PostMask - 指定允许以粘贴文件的用户名、组和地址』
- 第 228 页的『PutMask - 指定允许以放置文件的用户名、组和地址』
- 第 228 页的『ServerID - 指定与密码文件相关联的名称』

示例

- 在下例中, 服务器按如下所示激活保护:
 - 由 `/secret/scoop/` 激活保护开始的请求。保护设置定义于 `/server/protect/setup1.acc` 保护设置文件。由于 `Protect` 伪指令不指定保护设置, 所以使用先前匹配 `DefProt` 伪指令的保护设置。
 - 由 `/secret/business/` 激活保护开始的请求。保护设置定义于具有 `BUS-PROT` 标号的 `Protection` 伪指令。

- 由 /topsecret/ 激活保护开始的请求。保护设置直接包含于 Protect 伪指令。

这些示例使用 IP 地址。如果您的服务器接收以 /secret/ 或 /topsecret/ 开始的请求，则服务器基于请求到达的网络连接 IP 地址以激活请求的不同保护设置。

- 对于到达 0.67.106.79 的 /secret/ 请求，服务器激活定义于带有 CustomerA-PROT 标号的 Protection 伪指令上的保护设置。对于到达 0.67.106.79 的 /topsecret/ 请求，服务器激活内联定义于 /topsecret/ 的 Protect 伪指令上的保护设置。
- 对于到达 0.83.100.45 的 /secret/ 请求，服务器激活定义于带有 CustomerB-PROT 标号的 Protection 伪指令上的保护设置。对于到达 0.83.100.45 的 /topsecret/ 请求，服务器激活内联定义于 /topsecret/ 的 Protect 伪指令上的保护设置。

```
Protection BUS-PROT {
  UserID    busybody
  GroupID   webgroup
  AuthType  Basic
  ServerID  restricted
  PasswdFile /docs/WWW/restrict.pwd
  GroupFile /docs/WWW/restrict.grp
  GetMask   authors
  PutMask   authors
}
DefProt /secret/* /server/protect/setup1.acc
Protect /secret/scoop/*
Protect /secret/business/*  BUS-PROT
Protect /topsecret/* {
  AuthType  Basic
  ServerID  restricted
  PasswdFile /docs/WWW/restrict.pwd
  GroupFile /docs/WWW/restrict.grp
  GetMask   topbrass
  PutMask   topbrass
}
Pass /secret/scoop/*      /WWW/restricted/*
Pass /secret/business/*  /WWW/confidential/*
Pass /topsecret/*        /WWW/topsecret/*

Protect /secret/* CustomerA-PROT FOR 0.67.106.79
Protect /secret/* CustomerB-PROT FOR 0.83.100.45
Protect /topsecret/* 0.67.106.79 {
  AuthType  Basic
  ServerID  restricted
  PasswdFile /docs/WWW/customer-A.pwd
  GroupFile /docs/WWW/customer-A.grp
  GetMask   A-brass
  PutMask   A-brass
}
Protect /topsecret/* 0.83.100.45 {
  AuthType  Basic
  ServerID  restricted
  PasswdFile /docs/WWW/customer-B.pwd
  GroupFile /docs/WWW/customer-B.grp
  GetMask   B-brass
  PutMask   B-brass
}
}
```

- 下例使用虚拟主机。如果服务器接收以 /secret/ 或 /topsecret/ 开始的请求，则服务器基于 URL 中的主机名以激活请求的不同保护设置。
 - 对于到达 hostA.bcd.com 的 /secret/ 请求，服务器激活定义于带有 CustomerA-PROT 标号的 Protection 伪指令上的保护设置。对于到达 hostA.bcd.com 的 /topsecret/ 请求，服务器激活内联定义于 /topsecret/ 的 Protect 伪指令上的保护设置。

- 对于到达 hostB.bcd.com 的 /secret/ 请求，服务器激活定义于带有 CustomerB-PROT 标号的 Protection 伪指令上的保护设置。对于到达 hostB.bcd.com 的 /topsecret/ 请求，服务器激活内联定义于 /topsecret/ 的 Protect 伪指令上的保护设置。
- 对于代理的请求，服务器激活定义于带有 proxy-prot 标号的 Protection 伪指令上的保护设置。例如：

```
Protect http://host1/* proxy-prot

Protect /secret/* CustomerA-PROT FOR hostA.bcd.com
Protect /secret/* CustomerB-PROT FOR hostB.bcd.com
Protect /topsecret/* hostA.bcd.com {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/customer-A.pwd
    GroupFile /docs/WWW/customer-A.grp
    GetMask A-brass
    PutMask A-brass
}
Protect /topsecret/* hostB.bcd.com {
    AuthType Basic
    ServerID restricted
    PasswdFile /docs/WWW/customer-B.pwd
    GroupFile /docs/WWW/customer-B.grp
    GetMask B-brass
    PutMask B-brass
}
```

缺省值

缺省情况下，使用 /admin-bin/* 请求模板，Protect 伪指令向“配置和管理”表单提供保护。

Protection – 在配置文件中定义已命名的保护设置

使用此伪指令在配置文件中定义保护设置。您赋予保护设置名称并使用 protection 子伪指令定义保护类型。

注：

1. 在配置文件中，将 Protection 伪指令放置于指向它们的任何 DefProt 或 Protect 伪指令之前。
2. 要在您的保护规则中使用域名，将 DNS-Lookup 伪指令设置为 on。

格式

```
Protection label_name {
    subdirective value
    subdirective value
    .
    .
}
```

label_name

指定与此保护设置相关联的名称。则后继 DefProt 和 Protect 伪指令可使用名称以指向此保护设置。

subdirective value

子伪指令包含在大括号 ({ }) 内。左大括号字符必须是 *label_name* 一行上的最后

一个字符。后面每个子伪指令各占一行。右大括号字符必须在最后一行子伪指令后面独占一行。大括号中不能出现任何注释行。

请参阅『Protection 子伪指令 - 指定一组资源如何受保护』以获取 protection 子伪指令的描述。

示例

```
Protection NAME-ME {
  AuthType      Basic
  ServerID      restricted
  PasswdFile    /WWW/password.pwd
  GroupFile     /WWW/group.grp
  GetMask       groupname
  PutMask       groupname
}
```

缺省值

```
Protect /admin-bin/* {
  ServerId      Private_Authorization
  AuthType      Basic
  GetMask       All@(*)
  PutMask       All@(*)
  PostMask      All@(*)
  Mask          All@(*)
  PasswdFile    /opt/ibm/edge/cp/server_root/protect/webadmin.passwd
}
```

Protection 子伪指令 - 指定一组资源如何受保护

下列是可用于保护设置的 protection 子伪指令的描述。子伪指令按字母次序。

保护设置可在分隔文件或配置文件中作为 DefProt、Protect 或 Protection 伪指令的一部分。

AuthType - 指定认证类型

当基于用户名和密码限制访问时使用此 Protection 子伪指令。当客户机将密码发送到服务器时，指定要使用的认证类型。使用基本认证 (AuthType Basic)，密码作为纯文本发送到服务器。它们是编码的，但未加密。

缺省值:

AuthType Basic

DeleteMask - 指定允许以删除文件的用户名、组和地址

使用此 Protection 子伪指令以指定已授权的用户名、组和地址模板，它们将 DELETE 请求执行到受保护目录。

示例:

```
DeleteMask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*
```

GetMask - 指定允许以获取文件的用户名、组和地址

使用此 Protection 子伪指令以指定已授权的用户名、组和地址模板，它们将 GET 请求执行到受保护目录。

示例:

```
GetMask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*
```


缺省值:

```
GetMask All@(*)
```

GroupFile - 指定关联的组文件的位置

使用此 Protection 子伪指令以指定此保护设置使用的服务器组文件的路径与文件名。则在服务器组文件中定义的组可由下列使用:

- 作为保护设置部件的任何掩码子伪指令。(掩码子伪指令是 DeleteMask、GetMask、Mask、PostMask 和 PutMask。)
- 由保护设置保护的目录上的任何 ACL 文件。

示例:

```
GroupFile /docs/etc/WWW/restrict.group
```

Mask - 指定允许发出 HTTP 请求的用户名、组和地址

使用此 Protection 子伪指令以指定已授权的用户名、组和地址模板,它们发出其它掩码子伪指令不包括的 HTTP 请求。

示例:

```
Mask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*
```

注: 当您使用 Mask 伪指令时,重要的是要注意掩码是区分大小写的。下列是指定在用户标识上的 Mask 保护的示例:

```
MASK WEBADM,webadm
```

PasswdFile - 指定关联的密码文件的位置

当基于用户名和密码限制访问时使用此 Protection 子伪指令。指定此保护设置要使用的密码文件的路径和名称。

由于一些浏览器在主机中通过安全性域 (ServerID) 高速缓存用户标识和密码,所以在指定 ServerID 和密码文件时按照这些准则:

- 对于使用相同密码文件的保护设置,则使用相同的 ServerID。
- 对于使用不同密码文件的保护设置,则使用不同的 ServerID。

示例:

```
PasswdFile /docs/etc/WWW/restrict.password
```

注: 如果密码文件的路径或文件名包含嵌入的空格,则整个路径和文件名必须以引号 (") 括起。

```
PasswdFile "c:\test this\admin.pwd"
```

PostMask - 指定允许以粘贴文件的用户名、组和地址

对于安全服务器,使用此 Protection 子伪指令以指定已授权的用户、组和地址模板,它们将 POST 请求执行到受保护目录。

示例:

```
PostMask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*
```

PutMask - 指定允许以放置文件的用户名、组和地址

使用此 Protection 子伪指令以指定已授权的用户、组和地址模板，它们将 PUT 请求执行到受保护目录。

示例:

```
PutMask authors,(niceguy,goodie)@45.96.3.1,128.0.*.*
```

ServerID - 指定与密码文件相关联的名称

当基于用户名和密码限制访问时使用此 Protection 子伪指令。指定您要与正在使用的密码文件相关联的名称。名称不需要是实机器名。

名称用作为请求者的标识。由于不同的保护设置可使用不同的密码文件，所以与保护设置相关联的名称可帮助客户机确定要发送的密码。当提示用户名和密码时，大多数客户机显示此名称。

由于一些浏览器在主机中通过安全性域 (ServerID) 高速缓存用户标识和密码，所以在指定 ServerID 和密码文件时按照这些准则:

- 对于使用相同密码文件的保护设置，则使用相同的 ServerID。
- 对于使用不同密码文件的保护设置，则使用不同的 ServerID。

示例:

```
ServerID restricted
```

Proxy - 指定代理协议或反向代理

使用此伪指令以表明高速缓存代理要处理哪些协议，并将请求映射到服务器。有效的协议是 http、ftp 和 gopher。

如果将来自客户机的请求转发给同一服务器，并且这些来自客户机的请求来自同一客户机端 TCP 通道，则 Proxy 伪指令的 UseSession 参数选项将帮助维护服务器端上的持久通道。UseSession 重设伪指令 ServerConnPool，这将允许来自不同客户机的请求共享到后端服务器的持续连接。

Proxy 伪指令的 JunctionPrefix 参数选项与结点重写插件协同使用。要获取更多信息，请参阅第 41 页的『启用结点重写 (可选)』和第 42 页的『定义具有 JunctionPrefix 选项的结点 (建议的方法)』。

格式

```
Proxy request_template target_server_path [[ip]:port]  
[UseSession] [JunctionPrefix:url_prefix]
```

此伪指令将该请求传递到远程服务器。例如，下列伪指令导致所有请求转发到指定的 URL:

```
Proxy /* http://proxy.server.name/*
```

对于安全反向代理服务器，使用下列伪指令:

```
Proxy /* https://proxy.server.name/*
```

如果您要代理服务器受到最小的限制，从您的配置文件取消注释下列伪指令。(然而，当代理配置为反向代理时，这些伪指令可能导致安全性问题。)

```
Proxy http:*
Proxy ftp:*
Proxy gopher:*
```

以下是 Proxy 伪指令的 UseSession 选项的示例:

```
Proxy /abc/* http://server1/default/abc/* :80 UseSession
```

当入局客户机请求来自端口 80 时, 并且如果客户机请求上的 URL 匹配模式 /abc/*, 则将此 URL 映射到 http://server1/default/abc/*。

以下是 Proxy 伪指令的 JunctionPrefix 选项的格式:

```
Proxy url_pattern1 url_pattern2JunctionPrefix:url_prefix
```

要获取更多有关使用 JunctionPrefix 参数选项的信息, 请参阅第 41 页的『启用结点重写 (可选)』和第 42 页的『定义具有 JunctionPrefix 选项的结点 (建议的方法)』。

缺省值

无。

ProxyAccessLog - 命名代理访问日志文件的路径

使用此伪指令指定您要服务器记录代理请求的访问统计信息处的路径与文件名。缺省情况下, 每次服务器作为客户机请求的代理时, 它就会向日志写一个条目。如果您不想记录来自某些客户机的请求, 可以使用 NoLog 伪指令。

如果服务器运行, 则每天服务器在午夜启动新的日志文件。否则, 服务器在任何一天第一次启动它时启动新的日志文件。创建文件时, 服务器使用您指定的文件名, 并附加日期后缀或扩展名。日期后缀或扩展名的格式是 *Mmddyyyy*, 其中 *Mmm* 是月份的前三个字母; *dd* 是日子; 而 *yyyy* 是年。

除去旧的日志文件是个好主意, 因为它们会在您的驱动器上占用很大空间。

格式

```
ProxyAccessLog path/file
```

缺省值

- **Linux** 和 **UNIX** 系统: ProxyAccessLog /opt/ibm/edge/cp/server_root/logs/proxy
- **Windows** 系统: ProxyAccessLog drive:\Program Files\IBM\edge\cp\logs\proxy

ProxyAdvisor - 定制代理请求的服务

使用此伪指令指定您希望服务器在代理顾问程序执行步骤期间调用的定制应用程序。此代码将为请求服务。

格式

```
ProxyAdvisor /path/file:function_name
```

```
/path/file
```

指定已编译程序的全限定文件名。

```
function_name
```

指定程序内应用程序功能的名称。

示例:

```
ProxyAdvisor /api/bin/customadvise.so:proxyadv
```

缺省值

无

ProxyForwardLabels - 指定 PICS 过滤

使用 ProxyForwardLabels 伪指令以在代理服务器和客户机上或在代理层次结构中的两个代理上指定 PICS 过滤。

如果 ProxyForwardLabels 设置为 on, 则代理服务器为所有找到的 PICS 标签 (包含来自源服务器、标签局、高速缓存代理的标签高速缓存, 以及标签提供程序插件的标签) 生成 PICS-Label: HTTP 报头。

如果 ProxyForwardLabels 设置为 Off, PICS-Label: 不生成 HTTP 头。

格式

```
ProxyForwardLabels {On | Off}
```

缺省值

```
ProxyForwardLabels Off
```

ProxyFrom - 指定具有 From: 头的客户机

使用此伪指令以生成 From: 头。通常它用于给出代理管理员的电子邮件地址。

格式

```
ProxyFrom e-mail_address
```

示例

此设置 ProxyFrom webmaster@proxy.ibm.com 导致下列头更改:

原始头

```
Location: http://www.ibm.com/  
Last Modified: Tue 5 Nov 1997 10:05:39  
GMT  
Pragma: no-cache
```

修改后的头

```
Location: http://www.ibm.com/  
Last Modified: Tue 5 Nov 1997 10:05:39  
GMT  
From: webmaster@proxy.ibm.com  
Pragma: no-cache
```

缺省值

无

ProxyIgnoreNoCache - 忽略重新装入的请求

使用此伪指令指定当在浏览器上单击重新装入时服务器如何反应。如果 ProxyIgnoreNoCache 伪指令设置为 on, 在高装入期间, 服务器不请求来自目标服务器的页面, 并提供文件的高速缓存副本 (如果可用)。服务器本质上忽视从浏览器发送的 Pragma: no-cache 头。

格式

ProxyIgnoreNoCache {on | off}

缺省值

ProxyIgnoreNoCache off

ProxyPersistence - 允许持久连接

使用此伪指令指定客户机是否维护持久连接。持久连接减少用户的等待时间，并减少装入代理服务器的 CPU，但它需要更多的资源。因此，持久连接需要代理服务器上的更多线程和更多内存。

如果任何代理都不是 HTTP 1.1 兼容的，则不能在多级代理服务器设置上使用持久连接。

格式

ProxyPersistence {on | off}

缺省值

ProxyPersistence on

ProxySendClientAddress - 生成 Client IP Address: 头

使用此伪指令指定代理是否将客户机的 IP 地址转发到目标服务器。

格式

ProxySendClientAddress {*Client_IP*: | OFF}

示例

此伪指令 ProxySendClientAddress *Client-IP*: 导致下列头更改:

原始头

Location: http://www.ibm.com/
Last Modified: Tue 5 Nov 1997 10:05:39
GMT
Pragma: no-cache

修改后的头

Location: http://www.ibm.com
Last Modified: Tue 5 Nov 1997 10:05:39
GMT
Client-IP: 0.67.199.5
Pragma: no-cache

缺省值

无

ProxyUserAgent - 修改“User Agent”字符串

使用此伪指令指定 User Agent 字符串，此字符串替换客户机所发送的字符串。这允许访问 Web 站点时更大的匿名。然而，一些站点已基于 User Agent 字符串定制了页面。使用 ProxyUserAgent 伪指令防止显示这些页面。

格式

ProxyUserAgent *product_name/version*

示例

此伪指令 ProxyUserAgent Caching Proxy/6.0 导致下列头更改:

原始头	修改后的头
Location: http://www.ibm.com/	Location: http://www.ibm.com
Last Modified: Tue 5 Nov 1997 10:05:39 GMT	Last Modified: Tue 5 Nov 1997 10:05:39 GMT
User Agent: Mozilla/ 2.02 OS2	User Agent: Caching Proxy/6.0
Pragma: no-cache	Pragma: no-cache

缺省值

无

ProxyVia - 指定 HTTP 头的格式

使用伪指令控制 HTTP 头的格式。此伪指令可能有四个值: 如果 ProxyVia 设置为 Full, 则高速缓存代理将把 Via 头添加到请求或应答中; 如果 Via 头已在流中, 高速缓存代理将在最后添加主机信息。如果设置为 Set, 则高速缓存代理将设置 Via 头为主机信息; 如果 Via 头已在流中, 则高速缓存代理将除去它。如果设置为 Pass, 则高速缓存代理将把任何头信息照原样转发。如果设置为 Block, 则高速缓存代理不转发任何 Via 头。

格式

```
ProxyVia {Full | Set | Pass | Block}
```

示例

```
ProxyVia Pass
```

缺省值

```
ProxyVia Full
```

ProxyWAS - 指定请求是发送到 WebSphere Application Server

ProxyWAS 映射伪指令工作和 Proxy 伪指令功能等同, 但是它还对高速缓存代理表明匹配的请求是导向 WebSphere Application Server 的。要获取有关使用此伪指令的示例, 请参阅第 228 页的『Proxy - 指定代理协议或逆向代理』。

格式

```
ProxyWAS request_template target_server_path [UseSession]  
[JunctionPrefix:url_prefix]
```

缺省值

无

PureProxy - 禁用专用的代理

使用此伪指令指定服务器是作为代理服务器还是作为代理, 及指定内容服务器。建议您将高速缓存代理仅用作代理。

格式

PureProxy {on | off}

缺省值

PureProxy on

PurgeAge - 指定日志的时效

使用此伪指令指定清除日志之前的日志时效（以天数计算）。如果 `PurgeAge` 设置为 0，则从不删除日志。

注：插件从不删除当日或前一日的日志。

格式

PurgeAge *number*

缺省值

PurgeAge 7

相关伪指令

- 第 177 页的『CompressAge - 指定何时压缩日志』
- 第 178 页的『CompressDeleteAge - 指定何时删除日志』
- 第 177 页的『CompressCommand - 指定压缩命令和参数』
- 第 212 页的『Midnight - 指定用于归档日志的 API 插件』
- 第 206 页的『LogArchive - 指定日志归档的行为』
- 『PurgeSize - 指定日志压缩文档的大小限制』

PurgeSize - 指定日志压缩文档的大小限制

使用此伪指令指定清除日志压缩文档之前日志文件可发展成多大（以兆字节表示）。如果 `PurgeSize` 伪指令设置为 0，则无大小限制，且不删除文件。

`PurgeSize` 的设置指的是一种日志类型的所有日志。例如，如果您正在记录错误（即，如果已在配置文件中具有 `ErrorLog` 条目），并且 `PurgeSize` 定义为 10 MB，则高速缓存代理计算所有错误日志的大小，将它们添加到一起，然后删除日志直至总计大小少于 10 MB。

注：插件从不删除当日或前一日的日志。删除日志文件时，首先删除最旧的日志直至每个日志类型的日志文件的大小小于或等于 `PurgeSize`（以兆字节表示）定义的值。

格式

PurgeSize *number_of_MB*

缺省值

PurgeSize 0

相关伪指令

- 第 177 页的『CompressAge - 指定何时压缩日志』
- 第 178 页的『CompressDeleteAge - 指定何时删除日志』
- 第 177 页的『CompressCommand - 指定压缩命令和参数』
- 第 206 页的『LogArchive - 指定日志归档的行为』
- 第 212 页的『Midnight - 指定用于归档日志的 API 插件』
- 第 233 页的『PurgeAge - 指定日志的时效』

RCAConfigFile - 指定 ConfigFile 的别名

使用此伪指令指定远程高速缓存访问配置文件的名称和位置。

注: RCA 配置文件已合并为文件 `ibmproxy.conf`。要反向兼容, RCAConfigFile 作为 ConfigFile 的别名来支持。

格式

```
RCAConfigFile /etc/file_name
```

示例

```
RCAConfigFile /etc/user2rca.conf
```

缺省值

```
RCAConfigFile /etc/rca.conf
```

RCAThreads - 指定每个端口的线程数

使用此伪指令指定作用与 RCA 端口的线程数。

格式

```
RCAThreads number_of_threads
```

示例

```
RCAThreads 50
```

缺省值

```
MaxActiveThreads x [(ArraySize -1) / (2 x ArraySize -1)]
```

ReadTimeout - 指定连接的时间限制

使用此伪指令指定取消网络连接前网络不活动所允许的时间限制。

格式

```
ReadTimeout time
```

缺省值

```
ReadTimeout 5 minutes
```


Redirect – 为发送到另一个服务器的请求指定模板

使用此伪指令，为您要接受并发送到另一个服务器的请求指定模板。当请求与 `Redirect` 伪指令上的模板匹配后，它不与配置文件中任何其它伪指令上的模板相比较。

格式

```
Redirect request_template URL [server_IP_address | host_name]
```

request_template

指定您要您的服务器发送到另一个服务器的请求模板。

在模板中，您可将 (*) 作为通配符。必须明确匹配紧跟在 (/) 后面的否定号字符 (~)；不能使用通配符来匹配它。

URL

指定此服务器发送到另一个服务器的 URL 请求。没有任何它不来自您的服务器的指示，此请求的响应转至原始请求者。

URL 必须包含协议规范和请求发送到的服务器的名称。它也可包含路径或文件名。如果 *request_template* 使用通配符，则 URL 上的路径或文件名也可使用通配符。与 *request_template* 上通配符匹配的原始请求部件插入 URL 上通配符的位置。

[*server_IP_address* | *host_name*]

如果您使用多个 IP 地址或虚拟主机，则使用该参数指定 IP 地址或主机名。服务器仅对于来到此 IP 地址上服务器的请求或此主机使用伪指令。对于 IP 地址，这是服务器的网络连接的地址，而不是请求客户机的地址。

您可以指定 IP 地址（例如，240.146.167.72），或者您可以指定主机名（例如，hostA.bcd.com）。

此参数是可选的。没有此参数，则服务器对于所有请求使用伪指令，无论请求到达的 IP 地址或 URL 中的主机名。

不能为服务器的 IP 地址指定通配符。

示例

- 在下例中，服务器将以 /chief/stuff/ 开始的任何请求发送到 www.other.org 服务器的 wahoo 目录。

```
Redirect /chief/stuff/* http://www.other.org/wahoo/*
```

- 下例使用可选的 IP 地址参数。如果您的服务器接收以 /stuff/ 开始的请求，则服务器基于请求到达的网络连接 IP 地址将此请求重定向到不同的服务器。对于到达 240.146.167.72 的请求，服务器将请求发送到 www.chief.org 服务器的 wahoo 目录。对于到达 0.83.100.45 的请求，服务器将请求发送到 www.dawg.com 服务器的 pound 目录。

```
Redirect /stuff/* http://www.chief.org/wahoo/* 240.146.167.72
Redirect /stuff/* http://www.dawg.com/pound/* 0.83.100.45
```

- 下例使用可选的 IP 地址参数。如果您的服务器接收以 /stuff/ 开始的请求，则服务器基于 URL 中的主机名将该请求重定向到不同的服务器。对于到达 hostA 的请求，服务器将请求发送到 www.chief.org 服务器的 wahoo 目录。对于到达 hostB 的请求，服务器将请求发送到 www.dawg.org 服务器的 pound 目录。

```
Redirect /stuff/* http://www.chief.org/wahoo/* hostA.bcd.com
Redirect /stuff/* http://www.dawg.com/pound/* hostB.bcd.com
```

缺省值

无

ReversePass – 拦截自动重定向请求

ReversePass 映射伪指令检查服务器响应流，以检测重写为自动重定向结果的请求。通常，当服务器将 HTTP 代码返回入 3 *xx* 类（例如，301，永久地移动，或 303，请参阅其它），服务器发送消息，应答指示请求客户机将未来的请求导向正确的 IP 地址。在逆向代理设置的情况下，来自源服务器的重定向消息能导致客户机浏览器绕过代理服务器以获得后继请求。要避免客户机直接联系源服务器，使用 ReversePass 伪指令拦截特别为源服务器所做的请求。

与其它处理请求流的映射伪指令不同，ReversePass 将匹配到响应流到模板。响应流是代理服务器从源服务器获得的并将其发送给客户机的应答。

格式

```
ReversePass rewritten_URL proxy_URL [host:port]
```

host:port 选项允许代理应用基于后端服务器名和端口的不同 ReversePass 规则。

示例

- 下列语句示例表示到源服务器的直接请求：

```
ReversePass http://backend.company.com:9080/* http://edge.company.com/*
```

端口 9080 是 Application Service at the Edge 的缺省端口。如果源应用程序服务器将 3 *xx* 代码返回到客户机，则可能生成此类请求。

- 下列语句示例从边缘应用程序：服务器捕捉由 301 代码重定向的请求。

```
ReversePass http://edge.company.com:9080/* http://edge.company.com/*
```

注： *proxy_URL* 模式的内容，就算是通配符（*），都必须完全匹配后端服务器发送在位置头的内容，否则，此伪指令失败。

缺省值

无

RewriteSetCookieDomain — 指定需要重写的域模式

使用此伪指令指定需要重写的域模式。此伪指令将把域从 *domain_pattern1* 转换为 *domain_pattern2*。

格式

```
RewriteSetCookieDomain domain_pattern1 domain_pattern2
```

示例

```
RewriteSetCookieDomain .internal.com .external.com
```

缺省值

无

相关伪指令

- 第 202 页的『JunctionRewriteSetCookiePath — 当与 JunctionRewrite 插件协同使用时, 重写 Set-Cookie 头中的路径选项』

RTSPEnable - 启用 RTSP 重定向

此伪指令启用或禁用 RTSP 重定向。选项是 on 或 off。

格式

```
RTSPEnable {on | off}
```

示例

```
RTSPEnable on
```

缺省值

无

rtsp_proxy_server - 指定用于重定向的服务器

此伪指令用于指定 RTSP 代理服务器以接收重定向的请求。可为不同类型的流指定不同的服务器。该伪指令的格式是:

```
rtsp_proxy_server server dns address[:port] default rank [list of mime types]
```

示例

```
rtsp_proxy_server rproxy.mycompany.com:554 1
rtsp_proxy_server fw1.mycompany.com:554 2
rtsp_proxy_server fw1.mycompany.com:555 3
rtsp_proxy_server fw2.mycompany.com:557 4
```

缺省值

无

rtsp_proxy_threshold - 指定在重定向到高速缓存之前的请求数

此伪指令指定在将 RTSP 请求重定向至代理服务器而非源服务器之前应接收多少个请求。RealNetworks 代理高速缓存第一个请求上的流, 并且初始高速缓存导致接收流的带宽加倍。指定一个大于 1 的阈值将阻止仅发出一次的请求被高速缓存。该伪指令的格式是:

```
rtsp_proxy_threshold number_of_hits
```

示例

```
rtsp_proxy_threshold 5
```

缺省值

无

rtsp_url_list_size - 指定代理内存中的 URL 数

此伪指令指定在内存中保留用于重定向的唯一的 URL 数。代理引用此列表以确定以前是否遇到给定的 URL。较大的列表大小改进代理服务器发送后继请求到接收前一请求的同一代理服务器的能力, 但是每个列表条目都消耗大约 16 字节内存。

格式

`rtsp_url_list_size size_of_list`

示例

`rtsp_url_list_size 8192`

缺省值

无

ScriptTimeout – 指定脚本的超时设置

使用此伪指令以设置服务器开始的 CGI 程序完成允许的时间。当时间到期时，服务器结束程序。在 Linux 和 UNIX 平台上，这是使用 KILL 信号完成的。

使用 hours、minutes（或 mins）和 seconds（或 secs）的任何组合输入时间值。

格式

`ScriptTimeout timeout`

缺省值

`ScriptTimeout 5 minutes`

SendHTTP10Outbound – 指定代理请求的协议版本

使用此伪指令指定：从高速缓存代理发送到下游服务器的请求要使用 HTTP V1.0 协议。（下游服务器是一系列代理中的另一个代理服务器，或者是处理请求的源服务器。）

如果使用此伪指令，则高速缓存代理将 HTTP 1.0 标识为请求行中协议。只有 HTTP 1.0 特定功能和某些 HTTP 1.1 功能（如大多数 HTTP 1.0 服务器支持的高速缓存控制头）将发送到下游服务器。如果您具有不正确处理 HTTP 1.1 请求的下游服务器，则使用此伪指令。

如果未指定 `SendHTTP10Outbound` 伪指令，高速缓存代理将 HTTP 1.1 标识为请求行中的协议。HTTP 1.1 功能（如持久连接）也可用于请求。

格式

`SendHTTP10Outbound url_pattern`

示例

可多次指定此伪指令，例如：

```
SendHTTP10Outbound http://www.hosta.com/*
SendHTTP10Outbound http://www.hostb.com/*
```

为了反向兼容，`SendHTTP10Outbound` 的前一个语法将按如下所示处理：

- `SendHTTP10Outbound on` 看作是指定 `SendHTTP10Outbound *`。
- `SendHTTP10Outbound off` 被忽略。

注：如果同时指定了 `SendHTTP10Outbound off` 和 `SendHTTP10Outbound url_pattern`，则将忽略 `SendHTTP10Outbound off`，但会发出警告消息。

缺省值

无

SendRevProxyName - 在 HOST 头指定高速缓存代理主机名

当作为逆向代理起作用时，高速缓存代理从客户机接收 HTTP 请求并将请求发送到源服务器。缺省情况下，高速缓存代理将源服务器的主机名写入其发送到源服务器的请求的 HOST 头。如果 SendRevProxyName 伪指令是设置为是，则高速缓存代理将其自己的主机名写入 HOST 头。此伪指令能用于为后端服务器启用特殊的配置，因为它允许到源服务器的请求总是显得来自代理服务器，甚至在请求从一个后端服务器重定向到另一个后端服务器的情况下亦是如此。

此伪指令与下列所示的 ReversePass 映射伪指令有所不同：ReversePass 伪指令拦截带有指定语法的请求以及替换您指定的不同请求内容。SendRevProxyName 伪指令能设置为仅对源服务器主机名替换高速缓存代理主机名。在配置 Application Service at the Edge 时该伪指令是无用的。

格式

```
SendRevProxyName {yes | no}
```

ServerConnGCRun - 指定运行垃圾收集线程的时间间隔

此伪指令设置垃圾收集线程检查服务器连接的时间间隔是否超时（与 ServerConnTimeout 伪指令一起设置）。仅当 ServerConnPool 伪指令设置为 on 时，才使用此伪指令。

格式

```
ServerConnGCRun time_interval
```

示例

```
ServerConnGCRun 2 minutes
```

缺省值

```
ServerConnGCRun 2 minutes
```

ServerConnPool - 将连接合用指定到源服务器

此伪指令允许代理一起合用其与源服务器的输出连接。将此伪指令设置为 on 可增强性能，并更好地利用允许持续连接的源服务器。您还可以指定要通过 ServerConnTimeout 伪指令维护未使用的连接的时间。

注：在受控环境中，启用该伪指令是最好的；它能降低使用转发代理的情形或源服务器不是 HTTP 1.1 兼容时的性能。

格式

```
ServerConnPool {on | off}
```

缺省值

```
ServerConnPool off
```

ServerConnTimeout – 指定最大的非活动的周期

使用此伪指令限制取消连接前没有网络活动所允许的时间。仅当 `ServerConnPool` 伪指令设置为 `on` 时，才使用此伪指令。

格式

```
ServerConnTimeout time-spec
```

示例

```
ServerConnTimeout 30 seconds
```

缺省值

```
ServerConnTimeout 10 seconds
```

ServerInit – 定制服务器初始化步骤

使用此伪指令指定初始化例程期间服务器调用的定制应用程序功能。在读取任何客户机请求之前以及无论何时重新启动服务器，都执行此代码。

如果您在 `PreExit` 或 `Service` 步骤中正在使用 `GoServe` 模块，您需要在此调用 `gosclone` 模块。

格式

```
ServerInit /path/file:function_name [initialization_string]
```

/path/file

指定编译程序的全限定文件名，包含扩展名。

function_name

指定程序内应用程序功能的名称。

initialization_string

可选的；指定传递到应用程序功能的文本字符串。

示例

```
ServerInit /ics/api/bin/icsext05.so:svr_init
```

缺省值

无

ServerRoot – 指定服务器程序安装的目录

使用此伪指令指定服务器程序安装的目录（服务器的当前工作目录）。使用相对路径名时，日志伪指令将此当前工作目录用作缺省根。

在 Windows 系统上，此目录在安装期间标识。

格式

```
ServerRoot directory_path
```

缺省值

- **Linux 和 UNIX 系统:** ServerRoot /opt/ibm/edge/cp/server_root/
- **Windows 系统:** C:\Program Files\IBM\edge\cp\bin\

注: 您可更改缺省值, 但它对于服务器如何处理请求没有影响。

注: PASS 和 EXEC 规则可独立于此目录。

ServerTerm - 定制服务器终止步骤

使用此伪指令指定服务器终止步骤期间服务器调用的定制应用程序功能。当发生依次关机以及每次重新启动服务器时, 将执行此代码。它允许您发布 PreExit 应用程序功能分配的资源。

格式

ServerTerm */path/file:function_name*

/path/file

指定编译程序的全限定文件名, 包含扩展名。

function_name

指定程序内应用程序功能的名称。

示例

```
ServerTerm /ics/api/bin/icsext05.so:shut_down
```

缺省值

无

Service - 定制服务步骤

使用此伪指令指定 Service 步骤期间服务器调用的定制应用程序功能。此代码服务客户机请求。例如, 它发送文件或运行 CGI 程序。

没有此伪指令的缺省值。如果请求匹配 Service 规则 (即, 如果执行 Service 伪指令上指定的应用程序功能) 但此功能返回 HTTP_NOACTION, 则服务器生成错误并且请求失败。

格式

Service *request_template/path/file:function_name*
[*server_IP_address* | *host_name*]

request_template

指定进一步确定是否调用应用程序功能的请求模板。规范可包含协议、域和主机; 可放在斜杠 (/) 之后, 并可使用星号 (*) 作为通配符。例如, /front_page.html、http://www.ics.raleigh.ibm.com、/pub*、/* 以及 * 均有效。

/path/file

指定编译程序的全限定文件名, 包含扩展名。

function_name

指定程序内应用程序功能的名称。

[Server_IP_address | host_name]

如果您使用多个 IP 地址或虚拟主机，则此参数确定是否仅为到达特定 IP 地址或特定主机的请求调用您的应用程序功能。

不能为服务器的 IP 地址指定通配符。

示例

```
Service /index.html /ics/api/bin/icsext05.so:serve_req
Service /cgi-bin/hexcalc* /ics/api/calculator:HEXcalc*
```

注：如果您要全部路径转换（包含 *query_string*），您必须如第二个示例所显示那样，在 *request_template* 和 *path/file:function_name* 中都有星号（*）。

缺省值

无

SignificantURLTerminator - 为 URL 请求指定终止代码

使用此伪指令以为 URL 请求指定终止代码。在请求中使用终止代码可使高速缓存代理在处理请求时，以及评估结果是否已高速缓存时，仅仅评估终止代码前的字符。当定义多个终止代码时，高速缓存代理根据 *ibmproxy.conf* 文件中终止代码的定义顺序，比较终止代码与到达的 URL。

格式

SignificantURLTerminator *terminating_string*

示例

```
SignificantURLTerminator &.
```

在此示例中，两个请求被视为是等同的。

```
http://www.exampleURL.com/tx.asp?id=0200&. ;x=004;y=001
http://www.exampleURL.com/tx.asp?id=0200&. ;x=127;y=034
```

缺省值

无

SMTPServer (仅 Windows) - 设置发送邮件例程的 SMTP 服务器

使用此伪指令以设置 Windows 的高速缓存代理中内部发送邮件例程使用的 SMTP 服务器。还必须为此例程设置以下两个伪指令：第 250 页的『WebMasterEMail - 设置电子邮件地址以接收选择服务器报告』和第 250 页的『WebMasterSocksServer (仅 Windows) - 设置发送邮件例程的 socks 服务器』。

格式

SMTPServer *IP address or hostname of SMTP server*

示例

```
SMTPServer mybox.com
```


缺省值

无

SNMP – 启用和禁用 SNMP 支持

使用此伪指令以启用或禁用 SNMP 支持。

格式

```
SNMP {on | off}
```

缺省值

```
SNMP off
```

SNMPCommunity – 为 SNMP 提供安全性密码

使用此伪指令以定义 Web 服务器分布式协议接口 (DPI) 子代理程序与 SNMP 代理程序间的密码。SNMP 公用名授权用户查看 SNMP 为指定的服务器公用而监视的性能变量。系统管理员定义密码输入时可查看来自哪些服务器的哪些变量。如果您更改 SNMP 公用名，确保也更改文件 `/etc/snmpd.conf` 中指定的公用名。

格式

```
SNMPCommunity name
```

缺省值

```
SNMPCommunity public
```

SSLCaching – 为安全请求启用高速缓存

使用此伪指令以高速缓存使用逆向代理的安全请求上的内容。此伪指令配置所有到代理服务器连接（客户机连接和后端内容服务器连接）的高速缓存。

注：SSL 伪指令在 SuSE Linux 上不支持。

格式

```
SSLCaching {on | off}
```

缺省值

```
SSLCaching off
```

SSLCertificate – 为证书指定密钥标号

当高速缓存代理作为多个提供自身 SSL 证书的域的一个逆向代理，并指示代理服务器检索或不检索客户机认证的客户端 PKI 证书时，使用此伪指令可指定密钥标号，以允许代理确定将哪个证书发送给客户机。

格式

```
SSLCertificate serverIP/hostname CertificateLabel  
[NoClientAuth | ClientAuthRequired]
```

serverIP/hostname

您可以指定 IP 地址（例如，204.146.167.72），或者您可以指定 SSL 请求导向至的服务器的主机名（例如，hostA.raleigh.ibm.com）。

CertificateLabel

如果导向至指定 IP 地址或主机名的 SSL 请求需要客户机认证时，要使用的证书名。

[NoClientAuth | ClientAuthRequired]

告诉代理服务器检索还是不检索客户机端 PKI 证书的说明。

示例

```
SSLCertificate www.abc.com ABCCert
SSLCertificate 204.146.167.72 intABCCert
SSLCertificate www.xyz.com XYZCert
SSLCertificate www.xyz.com XYZCert ClientAuthRequired
```

缺省值

无

SSLCryptoCard – 指定已安装的加密卡

使用此伪指令告诉代理服务器安装了加密卡，并且指定该卡。

格式

```
SSLCryptoCard {rainbowcs | nciphernfast} {on | off}
```

示例

```
SSLCryptoCard rainbowcs on
```

缺省值

无

SSLEnable – 为安全请求指定在端口 443 上侦听

使用此伪指令为安全请求指定高速缓存代理在端口 443 上侦听。

注：SSL 伪指令在 SuSE Linux 上不支持。

格式

```
SSLEnable {on | off}
```

缺省值

```
SSLEnable off
```

SSLForwardPort – 指定 HTTP SSL 升级的寻址端口

使用此伪指令为 HTTP 请求指定要寻址的端口，通过实现 SSL 将高速缓存代理升级到 HTTP 请求。指定一个端口，而不是主 HTTP 端口 80 或主 SSL 端口 443。

格式

```
SSLForwardPort port number
```

示例

```
SSLForwardPort 8888
```

缺省值

无

SSLOnly - 对 HTTP 请求禁用侦听器线程

当 SSL（通常端口为 443）启用时，使用此伪指令以禁用标准 HTTP 请求（通常端口为 80 和 8080）的侦听器线程。

格式

```
SSLOnly {on | off}
```

缺省值

```
SSLOnly off
```

SSLTunneling - 启用 SSL 隧道

使用此伪指令允许 SSL 隧道到目标主机上的任何端口。将此伪指令设置为打开以允许 SSL 隧道到目标主机上的任何端口。将此伪指令设置为关闭以允许 SSL 隧道仅到 Proxy 规则给定的端口。如果没有 SSL 隧道的任何 Proxy 规则并且 SSLTunneling 伪指令设置为 off，则不允许 SSL 隧道。如果 SSLTunneling 伪指令设置为 on，则您必须使用伪指令 Enable，启用方法 CONNECT。

当把高速缓存代理用作反向代理时，禁用此伪指令（缺省值）以预防 SSL 隧道传送弱点攻击。

注：使用 Proxy 伪指令以启用 SSL 隧道到目标主机上的特定端口。

格式

```
SSLTunneling {on | off}
```

缺省值

```
SSLTunneling off
```

SSLVersion - 指定 SSL 的版本

使用此伪指令指定要使用的 SSL 的版本：V2、V3 或所有版本。如果您正在使用无法支持 SSL V3 的服务器，则将此伪指令设置为 V2。

注：SSL 伪指令在 SuSE Linux 上不支持。

格式

```
SSLVersion {SSLV2 | SSLV3 | all}
```

缺省值

```
SSLVersion SSLV3
```

SSLV2Timeout - 指定 SSLV2 会话到期之前要等待的时间

使用此伪指令以秒的形式指定会话到期之前 SSL V2 会话在无任何活动的情况下要等待的时间。

注: SSL 伪指令在 SuSE Linux 上不支持。

格式

```
SSLV2Timeout seconds
```

此处的 *seconds* 代表 0 和 100 之间的值。

缺省值

```
SSLV2Timeout 100
```

SSLV3Timeout – 指定 SSLV3 会话到期之前要等待的时间

使用此伪指令以秒的形式指定 SSL V3 会话到期之前在无任何活动的情况下要等待的时间。

注: SSL 伪指令在 SuSE Linux 上不支持。

格式

```
SSLV3Timeout seconds
```

此处的 *seconds* 代表 1 和 86400 秒之间的值 (以秒表示 1 天)。

缺省值

```
SSLV3Timeout 100
```

SuffixCaseSense – 指定后缀定义是否区分大小写

使用此伪指令指定您是否要服务器在将文件后缀与 AddClient、AddCharSet、AddType、AddEncoding 和 AddLanguage 伪指令上的后缀模式比较时分辨大写与小写字母。缺省情况下, 服务器不分辨大小写。

格式

```
SuffixCaseSense {on | Off}
```

缺省值

```
SuffixCaseSense Off
```

TLSV1Enable – 启用传送层安全协议

使用此伪指令在 SSL 连接中启用 TLS V1 协议。在打开此伪指令后, SSL 连接将首先检查 TLS 协议, 然后是 SSLv3 协议, 最后是 SSLv2 协议。

注: 此伪指令使用 Internet Explorer 和其它浏览器, 但不使用 Netscape。(Netscape 不是建议与高速缓存代理协同使用的浏览器。)

格式

```
TLSV1Enable {on | off}
```

示例

```
TLSV1Enable on
```

初始配置设置

无

Transmogriifier - 定制数据操作步骤

使用此伪指令指定数据操作步骤期间服务器调用的定制应用程序功能。此代码将提供三个应用程序功能:

- 打开功能执行任何处理数据之前的初始化
- 写功能处理数据
- 关闭功能执行任何清除活动
- 错误功能通知发生的问题

您可具有服务器每个实例的多个 Transmogriifiers 活动。

格式

```
Transmogriifier /path/file:function_name:function_name:function_name  
/path/file
```

指定编译程序的全限定文件名, 包含扩展名。

function_name

指定程序内应用程序功能的名称。您必须提供打开、写及关闭功能的名称。

示例

```
Transmogriifier /ics/bin/icsext05.so:open_data:write_data:close_data
```

缺省值

无

TransmogriifiedWarning - 发送警告消息到客户机

使用此伪指令将消息发送到通知其数据的客户机:

格式

```
transmogriifiedwaning {yes|no}
```

缺省值

Yes

TransparentProxy - 在 Linux 或 AIX 上启用透明代理

仅适用于 Linux 和 AIX。使用此伪指令指定服务器是否可作为透明代理服务器运行。

注: 当 TransparentProxy 伪指令设置为 On 时, 忽略 BindSpecific 伪指令, 并缺省为 Off。由于大多数 HTTP 流量在端口 80 上流动, 所以强烈建议它是配置的端口之一。

格式

```
TransparentProxy {On | Off}  
Port 80
```

缺省值

TransparentProxy Off

注：开始透明代理之后，如果您要停止高速缓存代理服务器，您还必须将下列命令作为根来发出：

```
ibmproxy -unload
```

在 Linux 系统上，此命令除去重定向防火墙规则，而在 AIX 系统上，它卸装透明代理内核扩展名。如果停止服务器后您不发出此命令，则您的机器将接受未对它指定的请求。

UpdateProxy - 指定高速缓存目标

使用此伪指令指定高速缓存代理程序更新哪个代理服务器。当高速缓存代理程序需要更新它运行的本地代理服务器以外的代理服务器时，这是必需的。可选地，您可指定端口。

注：在 Linux 和 UNIX 平台上，使用高速缓存代理程序需要此伪指令。如果您仅使用代理的一个机器，则指定主机名。

尽管高速缓存代理程序可在另一个服务器上更新高速缓存，它无法检索来自此机器的高速缓存访问日志。因此，如果 UpdateProxy 伪指令指定本地主机以外的主机，则忽略 LoadTopCached 伪指令。

格式

```
UpdateProxy fully_qualified_host_name_of_proxy_server
```

示例

```
UpdateProxy proxy15.ibm.com:1080
```

缺省值

无

UserId - 指定缺省用户标识

使用此伪指令指定服务器在访问文件之前更改到的用户名和用户数。

如果您更改此伪指令，您必须手工停止服务器，然后再次启动它以使更改生效。如果您仅重新启动服务器，则服务器将不识别更改。（请参阅第 15 页的第 5 章，『启动和停止高速缓存代理』）。

注：如果您更改用户标识、组标识或日志目录路径的服务器缺省值，则创建新的目录并更新目录的许可和所有权。要使服务器能将信息写入用户定义的日志目录，请将此目录的许可设置为 755，并将用户定义的服务器用户标识设置为所有者。例如，如果您将服务器的用户标识从缺省值更改为 jdoe，从缺省日志目录更改到 server_root/account，则 server_root/account 目录必须具有许可 755 并从属于 jdoe。

格式

```
UserId {ID_name | number}
```

缺省值

AIX、Linux、Solaris: UserId nobody

HP-UX: UserId www

V2CipherSpecs - 列出 SSL V2 的支持密码规范

此伪指令列出 SSL V2 的可用密码规范。

注: SSL 伪指令在 SuSE Linux 上不支持。

格式

V2CipherSpecs *specification*

可接受的值是下列的任何组合。它们都不可使用两次。

- 1 — RC4 US
- 2 — RC4 Export
- 3 — RC2 US
- 4 — RC2 Export
- 6 — DES 56-bit
- 7 — Triple DES US
- NULL - 使用缺省密码规范

示例

- 对于美国: V2CipherSpecs '137624'
- 对于出口: V2 Cipherspecs '246'

缺省值

无 (缺省情况下禁用 SSL)

V3CipherSpecs - 列出 SSL V3 的支持密码规范

此伪指令列出 SSL V3 的可用密码规范。

注: SSL 伪指令在 SuSE Linux 上不支持。

格式

V3CipherSpecs *specification*

可接受的值包含下列:

- 00 — NULL NULL
- 01 — NULL MD5
- 02 — NULL SHA
- 03 — RC4 MD5 Export
- 04 — RC4 MD5 US
- 05 — RC4 SHA US
- 06 — RC2 MD5 Export

- 09 — DES SHA Export
- 0A — Triple DS SHA US
- 62 — 56-bit DES CBC SHA
- 64 — 56-bit RC4 SHA
- NULL - 使用缺省密码规范。

示例

- 对于美国: V3CipherSpecs '0A09060564620403020100'
- 对于出口: V3Cipherspecs '0906646203020100'

缺省值

无 (缺省情况下禁用 SSL)

WebMasterEMail - 设置电子邮件地址以接收选择服务器报告

使用此伪指令设置在其上接收选择高速缓存代理报告 (如, 在 SSL 证书到期前 30 天发出通知) 的电子邮件地址。在 Linux 和 UNIX 系统上, Sendmail 进程必须正在运行。对于 Windows 系统, 发送邮件进程构建在高速缓存代理中, 所以无需外部邮件服务器; 但是, 必须设置其它两个伪指令: 『WebMasterSocksServer (仅 Windows) - 设置发送邮件例程的 socks 服务器』和 第 242 页的『SMTPServer (仅 Windows) - 设置发送邮件例程的 SMTP 服务器』。

注: 此电子邮件地址也用作匿名 FTP 密码。

格式

WebMasterEMail *webmastermailaddress*

示例

WebMasterEmail webmaster@computer.com

缺省值

WebMasterEmail webmaster

WebMasterSocksServer (仅 Windows) - 设置发送邮件例程的 socks 服务器

使用此伪指令以设置 Windows 的高速缓存代理中内部发送邮件例程使用的 socks 服务器。还必须为此例程设置以下两个伪指令: 『WebMasterEMail - 设置电子邮件地址以接收选择服务器报告』和 第 242 页的『SMTPServer (仅 Windows) - 设置发送邮件例程的 SMTP 服务器』。

格式

WebMasterSocksServer *IP address or hostname of socks server*

示例

WebMasterSocksServer socks.mybox.com

缺省值

无

Welcome – 指定欢迎文件的名称

使用此伪指令指定服务器查找的欢迎文件名称，以响应不包含特定文件名的请求。您可以通过将多个此伪指令的出现放置于配置文件来构建欢迎文件的列表。

对于不包含文件名或目录名的请求，服务器总是查看与 `Welcome` 伪指令上指定的名称相匹配文件的文件根目录。如果查找到匹配，则文件返回到请求者。

对于包含目录名而不是文件名的请求，`AlwaysWelcome` 伪指令控制服务器是否查看要返回的欢迎文件的目录。缺省情况下，`AlwaysWelcome` 设置为 `On` 值。这意味着服务器总是查看与 `Welcome` 伪指令上指定的名称相匹配文件的请求目录。如果查找到匹配，则文件返回到请求者。

如果服务器在目录中的文件和 `Welcome` 伪指令上的文件名之间查找到一个以上匹配，则 `Welcome` 伪指令的顺序确定返回哪个文件。服务器使用最靠近配置文件开头的 `Welcome` 伪指令。

格式

```
Welcome file_name [server_IP_address | host_name]
```

file_name

指定您要定义为欢迎文件的文件名。

[*server_IP_address* | *host_name*]

如果您使用多个 IP 地址或虚拟主机，则使用该参数指定 IP 地址或主机名。服务器仅对于来到此 IP 地址上服务器的请求或此主机使用伪指令。对于 IP 地址，这是服务器的网络连接的地址，而不是请求客户机的地址。

您可以指定 IP 地址（例如，240.146.167.72），或者您可以指定主机名（例如，hostA.bcd.com）。

此参数是可选的。如果没有该参数，服务器使用该伪指令响应所有请求，无论是请求到达的 IP 地址还是 URL 中的主机名。

不能为服务器的 IP 地址指定通配符。

示例

- 下例定义两个欢迎页面，并假设 `AlwaysWelcome` 伪指令设置为 `On` 缺省。对于不包含文件名的请求，服务器尝试从请求上指定的目录中返回欢迎文件（如果请求不指定文件名或目录，则是文件根目录）。服务器首先查找名为 `letsgo.html` 的文件。如果在此目录中没有此名称的文件，则服务器查找名为 `Welcome.html` 的文件。

```
Welcome letsgo.html  
Welcome Welcome.html
```

- 在下例中，服务器基于请求到达的网络连接的 IP 地址来查找不同的欢迎文件。对于到达 0.67.106.79 的请求，服务器查找名为 `CustomerA.html` 的欢迎文件。对于到达 0.83.100.45 的请求，服务器查找名为 `CustomerB.html` 的欢迎文件。如果请求到达不同的 IP 地址，则服务器查找缺省地址。

```
Welcome CustomerA.html 0.67.106.79  
Welcome CustomerB.html 0.83.100.45
```

- 在下例中，服务器基于 URL 中的主机名查找不同的欢迎文件。对于到达 hostA 的请求，服务器查找名为 CustomerA.html 的欢迎文件。对于到达 hostB 的请求，服务器查找名为 CustomerB.html 的欢迎文件。如果请求到达不同的主机，则服务器查找缺省主机名。

```
Welcome CustomerA.html hostA.bcd.com  
Welcome CustomerB.html hostB.bcd.com
```

缺省值

这些缺省值按缺省配置使用的顺序显示：

```
Welcome Welcome.html  
Welcome welcome.html  
Welcome index.html  
Welcome Frntpage.html
```

声明

第一版（2004 年 11 月）

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档所描述内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Corporation
Attn.: G71A/503
P.O. box 12195
3039 Cornwallis Rd.
Research Triangle Park, N.C. 27709-2195
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：

International Business Machines Corporation “按现状”提供本文档，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证或条件。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其它程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation
ATTN: Software Licensing
11 Stanwix Street
Pittsburgh, PA 15222-9183
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含了在日常业务运作中所使用到的数据和报告的示例。为了尽可能能够完整地对其说明，示例可能包含了个人、公司、商标和产品的名称。所有的这些名称都是虚构的，如有和这些名称及地址相似的真实公司企业，纯属巧合。

如果您正在查看此信息软拷贝，图片和彩色图例可能无法显示。

商标

下列术语是 IBM Corporation 在美国和 / 或其他国家或地区的商标:

- AIX®
- IBM
- Netfinity®
- RS/6000®
- SecureWay®
- Tivoli
- ViaVoice®
- WebSphere

Java™ 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

Microsoft®、Windows、Windows NT® 和 Windows 徽标是 Microsoft Corporation 在美国和 / 或其他国家或地区的商标。

Intel™、Intel Inside（徽标）、MMX™ 和 Pentium® 是Intel Corporation 在美国和 / 或其他国家或地区的商标。

UNIX 是 The Open Group 在美国和其他国家或地区的注册商标。

Linux 是 Linus Torvalds 在美国和 / 或其他国家或地区的商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。



中国印刷

G151-0063-00



Spine information:



**WebSphere Application
Server**

高速缓存代理管理指南

版本 6.0

G151-0063-00