

IBM WebSphere eXtreme Scale Versão 7.1.1  
Versão 7 Release 1

*Guia de Administração*  
*21 de Novembro de 2011*



Esta edição aplica-se à versão 7, release 1, modificação 1 do WebSphere eXtreme Scale e a todos os releases e modificações subsequentes até que seja indicado de outra forma em novas edições.

© Copyright IBM Corporation 2009, 2011.

# Índice

**Figuras . . . . . vii**

**Tabelas . . . . . ix**

**Sobre o Guia de Administração . . . . . xi**

## **Capítulo 1. Introdução. . . . . 1**

Tutorial: Introdução ao WebSphere eXtreme Scale . . . 1  
Introduzindo a Lição 1 do Tutorial: Definindo  
Grades de Dados com Arquivos de Configuração . 1  
Lição 2 do Tutorial de Introdução: Criando um  
Aplicativo Cliente . . . . . 3  
Lição 3 do Tutorial Introdução: Executando o  
Aplicativo Cliente de Amostra Introdução . . . 4  
Lição 4 do Tutorial de Introdução: Monitore seu  
Ambiente . . . . . 6

## **Capítulo 2. Planejamento . . . . . 9**

Visão Geral de Planejamento . . . . . 9  
Planejando a Topologia . . . . . 10  
Cache de Memória Local . . . . . 10  
Cache Local Replicado pelo Peer . . . . . 12  
Cache Integrado . . . . . 13  
Cache Distribuído . . . . . 14  
Integração com o Banco de Dados:  
Armazenamento em Cache Write-behind,  
Sequencial e Lateral . . . . . 17  
Planejando Diversas Topologias do Datacenter. . 35  
Interoperabilidade com Outros Produtos WebSphere 47  
Planejando para Instalação . . . . . 48  
Requisitos de Hardware e Software . . . . . 48  
Considerações sobre o Java SE . . . . . 49  
Considerações sobre o Java EE . . . . . 50  
Convenções de Diretório . . . . . 51  
Capacidade do Ambiente de Planejamento . . . 53  
Dimensionamento de Memória e Cálculo de  
Contagem de Partições . . . . . 53  
Dimensionando a CPU por Partição para  
Transações. . . . . 55  
Dimensionando CPUs para Transações Paralelas 56  
Planejamento de Capacidade de Cache Dinâmico 56  
Planejando para Configuração . . . . . 60  
Lista de Verificação Operacional . . . . . 60  
Planejamento para Portas de Rede. . . . . 62  
Visão Geral de Segurança. . . . . 64

## **Capítulo 3. Tutoriais . . . . . 67**

Tutorial: Configurando a Segurança do Java SE . . 67  
Tutorial de Segurança do Java SE - Etapa 1. . . 68  
Tutorial de Segurança do Java SE - Etapa 2. . . 71  
Tutorial de Segurança do Java SE - Etapa 3. . . 77  
Tutorial de Segurança do Java SE - Etapa 4. . . 80  
Tutorial: Integrar a Segurança do WebSphere  
eXtreme Scale com o WebSphere Application Server. 84

Introdução: Integre a Segurança do WebSphere  
eXtreme Scale com o WebSphere Application  
Server usando os Plug-ins do WebSphere  
Application Server Authentication. . . . . 84  
Módulo 1: Preparar o WebSphere Application  
Server . . . . . 85  
Módulo 2: Configurar o WebSphere eXtreme  
Scale para Usar os Plug-ins de Autenticação do  
WebSphere Application Server . . . . . 91  
Módulo 3: Configurar a Segurança de Transporte 97  
Módulo 4: Usar a Autorização do Java  
Authentication and Authorization Service  
(JAAS) no WebSphere Application Server . . . 100  
Módulo 5: Use a Ferramenta **xscmd** para  
Monitorar as Grades de Dados e Mapas . . . 105  
Tutorial: Integrar a Segurança do WebSphere  
eXtreme Scale em um Ambiente Misto com um  
Autenticador Externo. . . . . 106  
Introdução: Segurança em um Ambiente Misto 107  
Módulo 1: Preparar o Ambiente do WebSphere  
Application Server e Independente Combinado . 108  
Módulo 2: Configurar a Autenticação do  
WebSphere eXtreme Scale em um Ambiente  
Misto . . . . . 114  
Módulo 3: Configurar a Segurança de  
Transporte . . . . . 123  
Módulo 4: Usar a Autorização do Java  
Authentication and Authorization Service  
(JAAS) no WebSphere Application Server . . . 126  
Módulo 5: Use o Utilitário **xscmd** para Monitorar  
as Grades de Dados e Mapas . . . . . 130  
Tutorial: Executando Pacotes Configuráveis do  
eXtreme Scale na Estrutura do OSGi. . . . . 132  
Introdução: Iniciando e Configurando o  
Servidor e o Contêiner do eXtreme Scale para  
Executar Plug-ins na Estrutura do OSGi . . . 132  
Módulo 1: Preparando para Instalar e  
Configurar os Pacotes Configuráveis do  
Servidor eXtreme Scale . . . . . 134  
Módulo 2: Instalando e Iniciando Pacotes  
Configuráveis do eXtreme Scale na Estrutura do  
OSGi . . . . . 138  
Módulo 3: Executando o Cliente de Amostra do  
eXtreme Scale . . . . . 143  
Módulo 4: Consultando e Fazendo Upgrade do  
Pacote Configurável de Amostra . . . . . 145

## **Capítulo 4. Instalando o . . . . . 151**

Visão Geral da Instalação . . . . . 151  
Planejando para Instalação . . . . . 152  
Topologias de Instalação. . . . . 152  
Requisitos de Hardware e Software . . . . . 156  
Considerações sobre o Java SE. . . . . 157  
Considerações sobre o Java EE . . . . . 159  
Convenções de Diretório . . . . . 159

Instalando o WebSphere eXtreme Scale com o Assistente de Instalação . . . . .	161
Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client com WebSphere Application Server. . . . .	161
Instalando WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client Independente . . . . .	191
Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client no Modo Silencioso. . . . .	195
Arquivo de Resposta para Instalação Silenciosa . . . . .	196
Instalando o Serviço de Dados REST . . . . .	198
Instalando a Estrutura do Eclipse Equinox OSGi com o Eclipse Gemini para Clientes e Servidores . . . . .	200
Instalando Pacotes Configuráveis do eXtreme Scale . . . . .	202
Verificando a Instalação . . . . .	204
Executando as Primeiras Etapas Após a Instalação . . . . .	205
Instalação de Resolução de Problemas . . . . .	206
Desinstalando o WebSphere eXtreme Scale . . . . .	206

## Capítulo 5. Fazendo Upgrade e Migrando o WebSphere eXtreme Scale 209

Atualizando Servidores eXtreme Scale . . . . .	209
Migrando para o WebSphere eXtreme Scale Versão 7.1.1 . . . . .	212
Usando o Update Installer para instalar os pacotes de manutenção. . . . .	213
Ferramenta <b>xsadmin</b> para a Migração de Ferramenta <b>xscmd</b> . . . . .	213
Propriedades descontinuadas e APIs . . . . .	216

## Capítulo 6. Configurando . . . . . 219

Métodos de Configuração . . . . .	219
Configurando as Grades de Dados . . . . .	220
Configurando Implementações Locais . . . . .	220
Ativando Evictors com a Configuração XML . . . . .	221
Configurando uma Estratégia de Bloqueio. . . . .	222
Configurando Replicação Ponto a Ponto com o JMS . . . . .	224
Configurando Políticas de Implementação . . . . .	231
Configurando Implementações Distribuídas . . . . .	232
Controlando o Posicionamento de Shard com Zonas . . . . .	234
Configurando Servidores de Catálogo e Contêiner . . . . .	247
Melhor Prática: Armazenando em Cluster o Serviço de Catálogo com Domínios de Serviço de Catálogo . . . . .	248
Ajustando a Configuração do Intervalo de Pulsação para Detecção de Failover . . . . .	249
Configurando o WebSphere eXtreme Scale com o WebSphere Application Server . . . . .	252
Configurando o IBM eXtremeMemory e o IBM eXtremeIO . . . . .	273
Configurando Diversas Topologias do Datacenter . . . . .	276
Configurando Portas . . . . .	280
Configurando Portas no Modo Independente . . . . .	280
Configurando Portas em um Ambiente do WebSphere Application Server. . . . .	283
Servidores com Diversas Placas de Rede . . . . .	283

Configurando Transportes . . . . .	284
Configurando Object Request Brokers . . . . .	284
Configurando Clientes . . . . .	288
Configurando Clientes com a Configuração XML . . . . .	289
Ativando o Mecanismo de Invalidação do Cliente . . . . .	291
Configurando Valores de Tempo Limite de Nova Tentativa de Solicitação . . . . .	293
Configurando a Integração de Cache . . . . .	295
Configurando os Gerenciadores de Sessão HTTP . . . . .	296
Configurando o Provedor de Cache Dinâmico para o WebSphere eXtreme Scale . . . . .	321
Plug-in do Cache JPA Nível 2 (L2) . . . . .	325
Configurando Integração do Banco de Dados. . . . .	347
Configurando Utilitários de Carga do JPA. . . . .	347
Configurando Serviços de Dados REST. . . . .	350
Ativando o Serviço de Dados REST . . . . .	352
Configurando Servidores de Aplicativos para o Serviço de Dados REST . . . . .	360
Configurando Navegadores da Web para Acessar os feeds ATOM do Serviço de Dados REST . . . . .	374
Utilizando um Cliente Java com Serviço de Dados REST. . . . .	376
Cliente Visual Studio 2008 WCF com Serviço de Dados REST. . . . .	378
Configurando Servidores para OSGi. . . . .	380
Configurando os Plug-ins do eXtreme Scale com o OSGi Blueprint . . . . .	380
Configurando Servidores com o OSGi Blueprint . . . . .	383
Configurando Servidores com a Administração de Configuração do OSGI . . . . .	384

## Capítulo 7. Administrando . . . . . 387

Iniciando e Parando Servidores Independentes . . . . .	387
Iniciando Servidores Independentes . . . . .	387
Parando Servidores Independentes . . . . .	397
Iniciando e Parando Servidores em um Ambiente do WebSphere Application Server . . . . .	401
Usando a API do Servidor Integrado para Iniciar e Parar Servidores . . . . .	402
API do Servidor Integrado . . . . .	405
Administrando com o Utilitário <b>xscmd</b> . . . . .	407
Iniciando Servidores do eXtreme Scale Usando a Estrutura do Eclipse Equinox OSGi . . . . .	409
Instalando e Iniciando Plug-ins Ativados pelo OSGi . . . . .	411
Administrando Serviços Ativados pelo OSGi Usando o Utilitário <b>xscmd</b> . . . . .	413
Atualizando Serviços OSGi para Plug-ins do eXtreme Scale com <b>xscmd</b> . . . . .	416
Controlando o Posicionamento . . . . .	419
Gerenciando a Disponibilidade do ObjectGrid . . . . .	421
Gerenciando Falhas do Datacenter . . . . .	423
Administrando com Beans Gerenciados (MBeans) . . . . .	425
Acessando Beans Gerenciados (MBeans) Usando a Ferramenta <b>wsadmin</b> . . . . .	426
Acessando Beans Gerenciados (MBeans) Programaticamente . . . . .	426

## Capítulo 8. Monitoramento. . . . . 433

Visão Geral de Estatísticas . . . . .	433	Segurança da Grade de Dados. . . . .	503
Monitorando com o Console da Web . . . . .	435	Transport Layer Security e Secure Sockets Layer	505
Iniciando e Efetuando Login no Console da Web	435	Configurando Tipos de Transportes Seguros . . .	506
Conectando o Console da Web nos Servidores		Configurando os Parâmetros do Secure Sockets	
de Catálogos . . . . .	437	Layer (SSL) para Clientes ou Servidores . . .	506
Visualizando Estatísticas com o Console da Web	439	Segurança do Java Management Extensions (JMX)	507
Monitorando com Relatórios Customizados . . .	445	Integração de Segurança com Provedores Externos	510
Monitorando com Arquivos CSV . . . . .	446	Protegendo o Serviço de Dados REST . . . . .	511
Definições de Estatísticas do Arquivo CSV. . .	447	Integração de Segurança com o WebSphere	
Monitorando com a API de Estatísticas . . . . .	450	Application Server . . . . .	515
Módulos Estatísticos . . . . .	452	Configurando a Segurança do Cliente em um	
Monitorando com o Utilitário <b>xscmd</b> . . . . .	453	Domínio de Serviço de Catálogo . . . . .	518
Monitorando com a PMI do WebSphere		Ativando a Segurança Local . . . . .	519
Application Server . . . . .	455	Iniciando e Parando Servidores Seguros . . . . .	520
Ativando a PMI . . . . .	455	Iniciando Servidores Seguros em um Ambiente	
Recuperando Estatísticas PMI . . . . .	458	Independente . . . . .	520
Módulos PMI . . . . .	459	Iniciando Servidores Seguros no WebSphere	
Acessando Beans Gerenciados (MBeans) Usando		Application Server . . . . .	521
a Ferramenta wsadmin . . . . .	466	Parando Servidores Seguros . . . . .	522
Monitoramento com Beans Gerenciados (MBeans)	467	Configurando Perfis de Segurança para o Utilitário	
Monitorando com Ferramentas do Fornecedor . . .	468	<b>xscmd</b> . . . . .	522
Monitorando com o IBM Tivoli Enterprise			
Monitoring Agent para WebSphere eXtreme			
Scale . . . . .	468		
Monitorando Aplicativos eXtreme Scale com o			
CA Wily Introscope . . . . .	474		
Monitorando o eXtreme Scale com o Hyperic			
HQ. . . . .	477		
Monitorando as Informações do eXtreme Scale no			
DB2 . . . . .	479		
<b>Capítulo 9. Ajuste do desempenho</b>	<b>481</b>	<b>Capítulo 11. Resolução de Problemas</b>	<b>525</b>
Ajustando Sistemas Operacionais e Configurações		Ativando a Criação de Log. . . . .	525
de Rede . . . . .	481	Coletando Rastreo . . . . .	526
Propriedades do ORB . . . . .	482	Opções de Rastreo . . . . .	527
Ajustando as Java virtual machines . . . . .	486	Analizando Dados de Log e de Rastreo . . . . .	529
Ajustando a Configuração do Intervalo de Pulsação		Visão Geral de Análise de Log. . . . .	530
para Detecção de Failover . . . . .	488	Executando Análise de Log. . . . .	531
Ajustando a Coleta de Lixo com o WebSphere Real		Criando Scanners Customizados para Análise	
Time . . . . .	491	do Log . . . . .	532
WebSphere Real Time em um Ambiente		Resolução de Problemas da Análise do Log . . .	534
Independente . . . . .	491	Instalação de Resolução de Problemas . . . . .	534
WebSphere Real Time no WebSphere		Resolvendo Problemas da Integração de Cache . . .	535
Application Server . . . . .	493	Resolução de Problemas do Plug-in do Cache JPA	536
Ajustando o Provedor de Cache Dinâmico. . . . .	495	Resolução de Problemas de Administração . . . . .	537
		Resolução de Problemas de Várias Configurações	
		do Datacenter . . . . .	538
		Resolução de Problemas de Carregadores . . . . .	538
		Resolução de Problemas de Configuração XML . . .	539
		Resolução de Problemas de Segurança . . . . .	542
		IBM Support Assistant for WebSphere eXtreme	
		Scale . . . . .	543
		<b>Avisos . . . . .</b>	<b>545</b>
		<b>Marcas Registradas . . . . .</b>	<b>547</b>
<b>Capítulo 10. Segurança . . . . .</b>	<b>497</b>	<b>Índice Remissivo . . . . .</b>	<b>549</b>
Autenticação de Cliente do Aplicativo . . . . .	497		
Autorização do Aplicativo Cliente . . . . .	499		
Autenticação da Grade de Dados. . . . .	502		



## Figuras

1. Cenário de Cache em Memória Local . . . . .	11	39. Topologia de Intradomínio do JPA . . . . .	327
2. Cache Replicado pelo Peer com Alterações que são Propagadas com JMS . . . . .	12	40. Topologia Integrado do JPA . . . . .	328
3. Cache Replicado pelo Peer com Alterações que são Propagadas com o Gerenciador de Alta Disponibilidade . . . . .	12	41. Topologia Particionada Integrada do JPA . . . . .	329
4. Cache Integrado . . . . .	14	42. Topologia Remota do JPA . . . . .	331
5. Cache Distribuído . . . . .	15	43. Topologia de Amostra de Introdução . . . . .	351
6. Cache Local . . . . .	16	44. Diagrama do Esquema da Amostra Northwind do Microsoft SQL Server . . . . .	352
7. ObjectGrid como um Buffer de Banco de Dados . . . . .	17	45. Diagrama do Esquema de Entidade Cliente e Ordem . . . . .	353
8. ObjectGrid como um Cache Secundário . . . . .	18	46. Diagrama do Esquema de Entidade Categoria e Produto . . . . .	354
9. Cache Secundário . . . . .	19	47. Diagrama do Esquema de Entidade Cliente e Ordem . . . . .	355
10. Cache Sequencial . . . . .	20	48. Processo do Eclipse Equinox para Instalar e Iniciar Pacotes Configuráveis OSGi com Plug-ins do eXtreme Scale . . . . .	380
11. Armazenamento em Cache Read-through . . . . .	21	49. Processo do Eclipse Equinox para Incluir Toda a Configuração e Todos os Metadados em um Pacote Configurável OSGi . . . . .	410
12. Armazenamento em Cache Write-through . . . . .	21	50. Processo do Eclipse Equinox para Especificar a Configuração e os Metadados Fora de um Pacote Configurável OSGi . . . . .	410
13. Armazenamento em Cache Write-behind . . . . .	22	51. Estados de Disponibilidade de uma Instância do ObjectGrid . . . . .	421
14. Armazenamento em Cache Write-behind . . . . .	23	52. CollectPlacementPlan.java . . . . .	428
15. Utilitário de Carga . . . . .	27	53. CollectContainerStatus.java . . . . .	430
16. Plug-in do Utilitário de Carga . . . . .	29	54. CollectPlacementPlan.java . . . . .	431
17. Utilitário de Carga do Cliente . . . . .	30	55. Visão Geral de Estatísticas . . . . .	433
18. Atualização Periódica . . . . .	31	56. Visão Geral do MBean . . . . .	435
19. Topologia do Tutorial . . . . .	87	57. Estrutura do Módulo ObjectGridModule . . . . .	459
20. Topologia do Tutorial . . . . .	110	58. Exemplo da Estrutura do Módulo ObjectGridModule . . . . .	460
21. Fluxo de Autenticação . . . . .	114	59. Estrutura do mapModule . . . . .	461
22. Nó de Desenvolvimento . . . . .	153	60. Exemplo de Estrutura do Módulo mapModule . . . . .	461
23. Topologia Independente com Dois Datacenters . . . . .	154	61. Estrutura do Módulo hashIndexModule . . . . .	463
24. Exemplo de Topologia do WebSphere Application Server . . . . .	155	62. Exemplo da Estrutura do Módulo hashIndexModule . . . . .	463
25. Exemplo de Topologia Mista . . . . .	156	63. Estrutura agentManagerModule . . . . .	464
26. Arquivos do Serviço de Dados REST do WebSphere eXtreme Scale . . . . .	199	64. Exemplo da Estrutura agentManagerModule . . . . .	465
27. Ativar o Evictor TimeToLive com XML . . . . .	221	65. Estrutura queryModule . . . . .	466
28. Conectando um Evictor Usando XML . . . . .	222	66. Exemplo da Estrutura queryModule QueryStats.jpg . . . . .	466
29. Primários e Réplicas em Zonas . . . . .	241	67. Fluxo de Autenticação para Servidores Dentro do Mesmo Domínio de Segurança . . . . .	516
30. Comparação de eXtremeMemory e Tempos de Resposta de Armazenamento Heap . . . . .	274		
31. Link Entre Domínios de Serviço de Catálogo . . . . .	278		
32. Topologia Hub e Spoke . . . . .	279		
33. Exemplo do Uso da Linha de Comandos . . . . .	281		
34. Escolhendo um ORB . . . . .	287		
35. Arquivo objectGrid.xml . . . . .	311		
36. Arquivo objectGridDeployment.xml . . . . .	312		
37. Arquivo objectGridStandAlone.xml . . . . .	314		
38. Arquivo objectGridDeploymentStandAlone.xml . . . . .	315		





---

## Tabelas

1. Abordagens de Arbitragem . . . . .	43	16. Argumentos da Etapa configureClientSecurity	256
2. Recursos que Requerem o Java SE 5 ou Java SE 6 . . . . .	50	17. Argumentos de Comando modifyXSDomain	259
3. Lista de Verificação Operacional . . . . .	60	18. Argumentos da Etapa modifyEndpoints	260
4. Recursos que Requerem o Java SE 5 ou Java SE 6. . . . .	158	19. Argumentos da Etapa addEndpoints	261
5. Arquivos de Tempo de Execução para o WebSphere eXtreme Scale . . . . .	163	20. Argumentos da Etapa removeEndpoints	262
6. Arquivos de Tempo de Execução para o WebSphere eXtreme Scale Client . . . . .	164	21. Argumentos da Etapa configureClientSecurity	263
7. Arquivos de Tempo de Execução para Instalação Completa do WebSphere eXtreme Scale . . . . .	193	22. Status do Terminal do Servidor de Catálogos	268
8. Arquivos de Tempo de Execução para o WebSphere eXtreme Scale Client . . . . .	194	23. Propriedades Customizadas para Gerenciamento de Sessão SIP com ObjectGrid.	306
9. Argumentos para os Comandos Equivalentes <b>xsadmin</b> e <b>xscmd</b> . . . . .	214	24. Incluir Archive no Repositório . . . . .	365
10. Propriedades descontinuadas e APIs . . . . .	216	25. Instalar Novos Aplicativos . . . . .	366
11. Propriedades descontinuadas e APIs . . . . .	217	26. Incluir Archive no Repositório . . . . .	366
12. Propriedades descontinuadas e APIs . . . . .	217	27. Instalar Novos Aplicativos . . . . .	367
13. Intervalos de Pulsações . . . . .	250	28. Archive para Repositório. . . . .	368
14. Argumentos de Comando createXSDomain	255	29. Valores de Instalação . . . . .	369
15. Argumentos da Etapa defineDomainServers	255	30. Intervalos de Pulsações . . . . .	489
		31. Autenticação de credencial nas configurações do cliente e do servidor . . . . .	498
		32. Protocolo de Transporte a Ser Utilizado nas Configurações de Transporte do Cliente e Transporte do Servidor . . . . .	506
		33. Direitos de Acesso de Entidade . . . . .	514



---

## Sobre o *Guia de Administração*

O conjunto da documentação do WebSphere eXtreme Scale inclui três volumes que fornecem as informações necessárias para utilizar, programar e administrar o produto WebSphere eXtreme Scale.

### **Biblioteca do WebSphere eXtreme Scale**

A biblioteca do WebSphere eXtreme Scale contém os seguintes livros:

- O *Visão Geral do Produto* contém uma visualização de alto nível dos conceitos do WebSphere eXtreme Scale, incluindo cenários de caso de uso e tutoriais.
- O *Guia de Instalação* descreve como instalar topologias comuns do WebSphere eXtreme Scale.
- O *Guia de Administração* contém as informações necessárias para os administradores de sistema, incluindo como planejar implementações do aplicativo, planejar capacidade, instalar e configurar o produto, iniciar e parar servidores, monitorar o ambiente e proteger o ambiente.
- O *Guia de Programação* contém informações para desenvolvedores de aplicativos sobre como desenvolver aplicativos para o WebSphere eXtreme Scale utilizando as informações da API incluídas.

Para fazer download dos manuais, vá para a Página da Biblioteca do WebSphere eXtreme Scale.

Também é possível acessar as mesmas informações nesta biblioteca no Centro de Informações do WebSphere eXtreme Scale Versão 7.1.1.

### **Usando Manuais Off-line**

Todos os manuais na biblioteca do WebSphere eXtreme Scale contém links para o centro de informações com a seguinte URL raiz: <http://publib.boulder.ibm.com/infocenter/wxsinfo/v7r1m1>. Esses links levam diretamente para as informações relacionadas. No entanto, se estiver trabalhando off-line e encontrar um desses links, será possível procurar pelo título do link nos outros manuais na biblioteca. A documentação da API, o glossário e a referência de mensagens não estão disponíveis em manuais PDF.

### **Quem Deve Utilizar este Manual**

Este manual é destinado principalmente para administradores de sistema, administradores de segurança e operadores de sistema.

### **Obtendo Atualizações para este Manual**

É possível obter as atualizações para esse manual ao fazer download da versão mais recente da Página da Biblioteca do WebSphere eXtreme Scale.

### **Como Enviar Seus Comentários**

Entre em contato com a equipe de documentação. Você localizou o que precisava? O conteúdo era exato e completo? Envie seus comentários sobre esta documentação por e-mail para [wasdoc@us.ibm.com](mailto:wasdoc@us.ibm.com).



---

## Capítulo 1. Introdução



Depois de instalar o produto, será possível usar a introdução de amostra para testar a instalação e usar o produto pela primeira vez.

---

### Tutorial: Introdução ao WebSphere eXtreme Scale

Depois de instalar o WebSphere eXtreme Scale em um ambiente independente, é possível usar o aplicativo de amostra de introdução como uma introdução simples para sua capacidade como uma grade de dados em memória.

#### Objetivos do aprendizado

- Aprenda sobre o arquivo descritor XML do ObjectGrid e os arquivos do descritor XML da política de implementação que são usados para configurar seu ambiente.
- Inicie os servidores de catálogos e de contêiner usando os arquivos de configuração.
- Aprenda sobre como desenvolver um aplicativo cliente.
- Execute o aplicativo cliente para inserir dados na grade de dados.
- Monitore as grades de dados com o console da web

#### Tempo necessário

60 minutos

### Introduzindo a Lição 1 do Tutorial: Definindo Grades de Dados com Arquivos de Configuração

Para configurar grades de dados simples, use os arquivos `objectgrid.xml` e `deployment.xml` que são fornecidos na amostra Introdução.

A amostra usa os arquivos `objectgrid.xml` e `deployment.xml` que estão no diretório `wxs_install_root/ObjectGrid/gettingstarted/xml`. Estes arquivos são transmitidos para os comandos iniciais para iniciar servidores de contêiner e um servidor de catálogos. O arquivo `objectgrid.xml` é o arquivo descritor XML do ObjectGrid. O arquivo `objectgrid.xml` é o arquivo descritor XML da política de implementação do ObjectGrid. Esses arquivos definem uma topologia distribuída.

#### Arquivo XML descritor do ObjectGrid

Um arquivo XML descritor de ObjectGrid é usado para definir a estrutura do ObjectGrid que será usada pelo aplicativo. Ele inclui uma lista de configurações de mapa de apoio. Esses mapas de apoio armazenam os dados de cache. O exemplo a seguir é um arquivo `objectgrid.xml` de amostra. As primeiras linhas do arquivo incluem o cabeçalho obrigatório de cada arquivo XML do ObjectGrid. O arquivo de exemplo a seguir define Grid ObjectGrid com os mapas de apoio Map1 e Map2.

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
```

```
<objectGrids>
```

```

        <objectGrid name="Grid">
            <backingMap name="Map1" />
            <backingMap name="Map2" />
        </objectGrid>
    </objectGrids>
</objectGridConfig>

```

## Arquivo Descritor XML de Política de Implementação

Um arquivo descritor XML de política de implementação é passado para o servidor de contêiner durante a inicialização. Uma política de implementação deve ser usada com o arquivo XML de ObjectGrid e deve ser compatível com o XML de ObjectGrid que é usado com ele. Para cada elemento `objectgridDeployment` na política de implementação, você deve ter um elemento `ObjectGrid` correspondente no arquivo XML do ObjectGrid. Os elementos de `backingMap` que são definidos dentro do elemento `objectgridDeployment` devem ser consistentes com os `backingMaps` localizados no XML de ObjectGrid. Cada `backingMap` deve ser referido dentro de um e apenas um `mapSet`.

O arquivo XML descritor de política de implementação deve igualar-se com o arquivo XML ObjectGrid correspondente, o arquivo `objectgrid.xml`. No seguinte exemplo, as primeiras linhas do arquivo `deployment.xml` incluem o cabeçalho obrigatório de cada arquivo XML de política de implementação. O arquivo define o elemento `objectgridDeployment` para o ObjectGrid da Grade que está definido no arquivo `objectgrid.xml`. Ambos os `BackingMaps`, `Map1` e `Map2`, que são definidos dentro do ObjectGrid da Grade estão incluídos no `mapSet` que tem os atributos `numberOfPartitions`, `minSyncReplicas` e `maxSyncReplicas` configurados.

```

<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
  ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="Grid">
        <mapSet name="mapSet" numberOfPartitions="13" minSyncReplicas="0"
maxSyncReplicas="1" >
            <map ref="Map1"/>
            <map ref="Map2"/>
        </mapSet>
    </objectgridDeployment>

</deploymentPolicy>

```

O atributo `numberOfPartitions` do elemento `mapSet` especifica o número de partições para o `mapSet`. Ele é um atributo opcional e o padrão é 1. O número deve ser apropriado para a capacidade antecipada da grade de dados.

O atributo `minSyncReplicas` de `mapSet` especifica o número mínimo de réplicas síncronas para cada partição no `mapSet`. Ele é um atributo opcional e o padrão é 0. Primária e réplica não são colocadas até que o domínio possa suportar o número mínimo de réplicas síncronas. Para suporte do valor `minSyncReplicas`, é preciso de mais um contêiner do que o valor de `minSyncReplicas`. Se o número de réplicas síncronas ficar abaixo do valor de `minSyncReplicas`, grave transações que não são mais permitidas àquela partição.

O atributo `maxSyncReplicas` de `mapSet` especifica o número máximo de réplicas síncronas para cada partição no `mapSet`. Ele é um atributo opcional e o padrão é 0. Nenhuma outra réplica síncrona é posicionada para uma partição após um domínio alcançar este número de réplicas síncronas para esta partição específica. A

inclusão de contêineres que podem suportar esse ObjectGrid pode resultar em um aumento no número de réplicas síncronas se seu valor de `maxSyncReplicas` ainda não tiver sido atingido. A amostra define o `maxSyncReplicas` para 1, que significa que o domínio posicionará, no máximo, uma réplica síncrona. Se você iniciar mais de uma instância do servidor de contêineres, haverá somente uma réplica síncrona posicionada em uma das instâncias do servidor de contêineres.

## Ponto de verificação de lições

Nesta lição, você aprendeu:

- Como definir os mapas que armazenam os dados no arquivo descritor XML do ObjectGrid.
- Como usar o arquivo descritor XML de implementação para definir o número de partições e réplicas para a grade de dados.

## Lição 2 do Tutorial de Introdução: Criando um Aplicativo Cliente

Para inserir, excluir, atualizar e recuperar dados de sua grade de dados, você deverá gravar um aplicativo cliente. A introdução de amostra inclui um aplicativo cliente que pode ser usado para saber mais sobre a criação de seu próprio aplicativo cliente.

O arquivo `Client.java` no diretório `wxs_install_root/ObjectGrid/gettingstarted/client/src/` é o programa cliente que demonstra como se conectar a um servidor de catálogos, obter a instância do ObjectGrid e usar a API ObjectMap. A API ObjectMap armazena dados como pares de valores de chave e é ideal para armazenar em cache os objetos que não possuem nenhum relacionamento envolvido.

Se você precisar armazenar em cache objetos que possuem relacionamentos, use a API `EntityManager`.

1. Conexão com o serviço de catálogo por meio da obtenção de uma instância de `ClientClusterContext`

Para se conectar a um servidor de catálogos, use o método `connect` da API `ObjectGridManager`. O método `connect` que é usado requer apenas o terminal do servidor de catálogos no formato de `hostname:port`. É possível indicar diversos terminais do servidor de catálogos separando a lista de valores de `hostname:port` com vírgulas. O fragmento de código a seguir demonstra como se conectar a um servidor de catálogos e obter uma instância de `ClientClusterContext`:

```
ClientClusterContext ccc = ObjectGridManagerFactory.getObjectGridManager().connect("localhost:2809", null, null);
```

Se as conexões com os servidores de catálogos forem bem-sucedidas, o método `connect` retorna uma instância de `ClientClusterContext`. A instância do `ClientClusterContext` é necessária para a obtenção do ObjectGrid a partir da API do `ObjectGridManager`.

2. Obter uma instância do ObjectGrid.

Para obter uma instância do ObjectGrid, use o método `getObjectGrid` da API do `ObjectGridManager`. O método `getObjectGrid` requer a instância de `ClientClusterContext` e o nome da instância da grade de dados. A instância do `ClientClusterContext` é obtida durante a conexão com o servidor de catálogos. O nome da instância de ObjectGrid é `Grid` que é especificado no arquivo `objectgrid.xml`. O fragmento de código a seguir demonstra como obter a grade de dados chamando o método `getObjectGrid` da API `ObjectGridManager`.

```
ObjectGrid grid = ObjectGridManagerFactory.getObjectGridManager().getObjectGrid(ccc, "Grid");
```

### 3. Obter uma instância de Sessão.

É possível obter uma Sessão da instância do ObjectGrid obtida. Uma instância da Sessão é necessária para obter a instância do ObjectMap e executar a demarcação da transação. O fragmento de código a seguir demonstra como obter uma instância da Sessão chamando o método getSession da API ObjectGrid.

```
Session sess = grid.getSession();
```

### 4. Obter uma instância do ObjectGrid.

Após obter uma Sessão, é possível obter uma instância do ObjectMap a partir de uma instância da Sessão chamando o método getMap da API de Sessão. Você deve transmitir o nome do mapa como parâmetro para o método getMap para obter a instância de ObjectMap. O fragmento de código a seguir demonstra como obter o ObjectMap chamando o método getMap da API de Sessão.

```
ObjectMap map1 = sess.getMap("Map1");
```

### 5. Use os métodos ObjectMap.

Depois que uma instância ObjectMap for obtida, será possível usar a API ObjectMap. Lembre-se de que a interface ObjectMap é um mapa transacional e requer demarcação de transação usando os métodos begin e commit da API Session. Se não houver demarcação de transação explícita no aplicativo, as operações do ObjectMap serão executadas com transações de confirmação automática.

O fragmento de código a seguir demonstra como usar a API ObjectMap com uma transação de confirmação automática.

```
map1.insert(key1, value1);
```

O fragmento de código a seguir demonstra como usar a API ObjectMap com demarcação de transação explícita.

```
sess.begin();  
map1.insert(key1, value1);  
sess.commit();
```

## Ponto de verificação de lições

Nesta lição, você aprendeu como criar um aplicativo cliente simples para executar operações da grade de dados.

## Lição 3 do Tutorial Introdução: Executando o Aplicativo Cliente de Amostra Introdução

Use as seguintes etapas para iniciar sua primeira grade de dados e executar um cliente para interagir com a grade de dados.

O script env.sh|bat é chamado pelos outros scripts para configurar as variáveis necessárias do ambiente. Normalmente não é necessário alterar esse script.

- **UNIX** **Linux** ./env.sh
- **Windows** env.bat

Para executar o aplicativo, primeiro você deve iniciar o processo do serviço de catálogo. O serviço de catálogo é o centro de controle da grade de dados. Ele controla os locais dos servidores de contêiner e também o posicionamento dos dados para hospedar servidores de contêiner. Após o serviço de catálogo ser iniciado, é possível iniciar os servidores de contêiner, que armazenam os dados do



aplicativo para a grade de dados. Para armazenar diversas cópias dos dados, diversos servidores de contêiner podem ser iniciados. Quando todos os servidores forem iniciados, será possível executar o aplicativo cliente para inserir, atualizar, remover e obter dados da grade de dados.

1. Abra uma janela de sessão do terminal ou linha de comandos.
2. Utilize o seguinte comando para navegar para o diretório `gettingstarted`:

```
cd wxs_install_root/ObjectGrid/gettingstarted
```

Substitua *wxs\_install\_root* pelo caminho para o diretório-raiz da instalação do eXtreme Scale ou pelo caminho de arquivo raiz de teste do eXtreme Scale extraído *wxs\_install\_root*.

3. Execute o script a seguir para iniciar um processo de serviço de catálogo no host local:

- **UNIX** **Linux** `./runcat.sh`

- **Windows** `runcat.bat`

O processo do serviço de catálogo executa na janela do terminal atual.

Também é possível iniciar o serviço de catálogo com o comando **startOgServer**. Execute **startOgServer** a partir do diretório *wxs\_install\_root*/ObjectGrid/bin:

- **UNIX** **Linux** `startOgServer.sh cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809`

- **Windows** `startOgServer.bat cs0 -catalogServiceEndPoints cs0:localhost:6600:6601 -listenerPort 2809`

4. Abra outra janela de sessão de terminal ou de linha de comandos e execute o seguinte comando para iniciar uma instância do servidor de contêiner:

- **UNIX** **Linux** `./runcontainer.sh server0`

- **Windows** `runcontainer.bat server0`

O servidor de contêiner será executado na janela do terminal atual. É possível repetir esta etapa com um nome de servidor diferente se desejar iniciar mais instâncias do servidor de contêiner para suportar a replicação.

Os servidores de contêiner também podem ser iniciados com o comando **startOgServer**. Execute **startOgServer** a partir do diretório *wxs\_install\_root*/ObjectGrid/bin:

- **UNIX** **Linux** `startOgServer.sh c0 -catalogServiceEndPoints localhost:2809 -objectgridFile gettingstarted\xml\objectgrid.xml -deploymentPolicyFile gettingstarted\xml\deployment.xml`

- **Windows** `startOgServer.bat c0 -catalogServiceEndPoints localhost:2809 -objectgridFile gettingstarted\xml\objectgrid.xml -deploymentPolicyFile gettingstarted\xml\deployment.xml`

5. Abra outra janela de sessão de terminal ou linha de comandos para executar comandos do cliente.

O script `runclient.sh|bat` executa o cliente CRUD simples e inicia a operação especificada. O script `runclient.sh|bat` é executado com os seguintes parâmetros:

- **UNIX** **Linux** `./runclient.sh command value1 value2`

- **Windows** `runclient.bat command value1 value2`

Para *command*, use uma das seguintes opções:

- Especifique *i* para inserir *value2* na grade de dados com a chave *value1*
- Especifique *u* para atualizar o objeto com chave pelo *value1* para o *value2*

- Especifique `d` para excluir o objeto com chave pelo *value1*
- Especifique `g` para recuperar e exibir o objeto com chave pelo *value1*
- a. Inclua dados na grade de dados:
  - `UNIX` `Linux` `./runclient.sh i key1 helloWorld`
  - `Windows` `runclient.bat i key1 helloWorld`
- b. Procurar e exibir o valor:
  - `UNIX` `Linux` `./runclient.sh g key1`
  - `Windows` `runclient.bat g key1`
- c. Atualizar o valor:
  - `UNIX` `Linux` `./runclient.sh u key1 goodbyeWorld`
  - `Windows` `runclient.bat u key1 goodbyeWorld`
- d. Excluir o valor:
  - `UNIX` `Linux` `./runclient.sh d key1`
  - `Windows` `runclient.bat d key1`

### Ponto de verificação de lições

Nesta lição, você aprendeu:

- Como iniciar os servidores de catálogos e servidores de contêiner
- Como executar o aplicativo cliente de amostra

## Lição 4 do Tutorial de Introdução: Monitore seu Ambiente

É possível usar o utilitário `xscmd` e as ferramentas do console da web para monitorar o ambiente da grade de dados.

### Monitorando com o Console da Web


Com o console da Web, é possível registrar no gráfico as estatísticas atuais e de histórico. Este console fornece alguns gráficos pré-configurados para visões gerais de alto nível e tem uma página de relatórios customizada que pode ser usada para construir gráficos a partir das estatísticas disponíveis. É possível usar os recursos gráficos no console de monitoramento do WebSphere eXtreme Scale para visualizar o desempenho geral das grades de dados em seu ambiente.

Instale o console da web como um recurso opcional no qual você executa o assistente de instalação.

1. Inicie o servidor do console. O script `startConsoleServer.bat|sh` para iniciar o servidor do console está no diretório `wxs_install_root/ObjectGrid/bin` de sua instalação.
  2. Efetue logon no console.
    - a. No seu navegador da web, acesse `https://your.console.host:7443`, substituindo `your.console.host` pelo nome do host do servidor na qual o console foi instalado.
    - b. Efetue logon no console.
      - **ID do usuário:** admin
      - **Senha:** admin
- A página de boas-vindas do console é exibida.

3. Edite a configuração do console. Clique em **Definições > Configuração** para revisar a configuração do console. A configuração do console inclui informações como:
  - Cadeia de rastreamento para o cliente do WebSphere eXtreme Scale, como `*=all=disabled`
  - O nome e a senha do Administrador
  - O endereço de e-mail do Administrador
4. Estabeleça e mantenha conexões com servidores de catálogos que você deseja monitorar. Repita as seguintes etapas para incluir cada servidor de catálogos na configuração.
  - a. Clique em **Configurações > Servidores de Catálogos do eXtreme Scale**.
  - b. Inclua um novo servidor de catálogos.



- 1) Clique no ícone Incluir (  ) para registrar um servidor de catálogos existente.
  - 2) Forneça informações, como o nome do host e a porta do listener. Consulte “Planejamento para Portas de Rede” na página 62 para obter mais informações sobre a configuração da porta e os padrões.
  - 3) Clique em **OK**.
  - 4) Verifique se o servidor de catálogos foi incluído na árvore de navegação.
5. Visualize o status da conexão. O campo **Domínio Atual** indica o nome do domínio do serviço de catálogo que está atualmente sendo usado para exibir informações no console da web. O status da conexão é exibido ao lado do nome do domínio de serviço de catálogo.
  6. Visualize as estatísticas para as grades de dados e servidores ou crie um relatório customizado.

### Monitorando com o Utilitário xscmd

1. Abra uma janela de linha de comandos. Na linha de comandos, configure as variáveis de ambiente apropriadas.
  - a. Configure a variável de ambiente `CLIENT_AUTH_LIB`:
    - **Windows** `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`
    - **UNIX** `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`  
`export CLIENT_AUTH_LIB`
2. Acesse o diretório `wxs_home/bin`.  
`cd wxs_home/bin`
3. Execute vários comandos para exibir informações sobre seu ambiente.
  - Mostre todos os servidores de contêiner on-line para a grade de dados Grid e o conjunto de mapas `mapSet`:  
`xscmd -c showPlacement -g Grid -ms mapSet`
  - Exiba as informações de roteamento para a grade de dados.  
`xscmd -c routetable -g Grid`
  - Exiba o número de entradas do mapa na grade de dados.  
`xscmd -c showMapSizes -g Grid -ms mapSet`

### Parando os Servidores

Depois de usar o aplicativo cliente e o monitorar a amostra do ambiente de introdução, os servidores poderão ser interrompidos.

- Se os arquivos de script forem usados para iniciar os servidores, utilize <ctrl+c> para parar o processo de serviço de catálogo e os servidores de contêiner em suas respectivas janelas.
- Se o comando **startOgServer** foi usado para iniciar seus servidores, use o comando **stopOgServer** para parar os servidores.

#### **Pare o servidor de contêiner:**

- **UNIX** **Linux** stopOgServer.sh c0 -catalogServiceEndpoints  
localhost:2809
- **Windows** stopOgServer.bat c0 -catalogServiceEndpoints  
localhost:2809

#### **Pare o servidor de catálogos:**

- **UNIX** **Linux** stopOgServer.sh cs1 -catalogServiceEndpoints  
localhost:2809
- **Windows** stopOgServer.bat cs1 -catalogServiceEndpoints  
localhost:2809

### **Ponto de verificação de lições**

Nesta lição, você aprendeu:

- Como iniciar o console da web e conectá-lo ao servidor de catálogos
- Como monitorar a grade de dados e as estatísticas do servidor
- Como parar os servidores

---

## Capítulo 2. Planejamento



Antes de instalar o WebSphere eXtreme Scale e de implementar seus aplicativos de grade de dados, você deve decidir sobre sua topologia de armazenamento em cache, concluir o planejamento da capacidade, revisar os requisitos de hardware e de software, as configurações de rede e ajustes, e assim por diante. Também é possível usar a lista de verificação operacional para garantir que seu ambiente esteja pronto para ter um aplicativo implementado.

Para uma discussão das boas práticas que é possível usar ao projetar seus aplicativos WebSphere eXtreme Scale, leia o seguinte artigo em developerWorks: Princípios e boas práticas para construir aplicativos WebSphere eXtreme Scale de alta execução e alta resiliência.

---

### Visão Geral de Planejamento

Antes de utilizar o WebSphere eXtreme Scale em um ambiente de produção, considere os seguintes problemas para otimizar sua implementação.

#### Considerações sobre Instalação

É possível instalar o WebSphere eXtreme Scale em um ambiente independente ou integrar a instalação com o WebSphere Application Server. Para assegurar um upgrade normal de seus servidores no futuro, você deve planejar seu ambiente de acordo. Para obter melhor desempenho, os servidores de catálogos devem ser executados em máquinas diferentes dos servidores de contêiner. Se os seus servidores de catálogos e de contêiner tiverem que ser executados na mesma máquina, use instalações separadas do WebSphere eXtreme Scale para os servidores de catálogos e de contêiner. Ao usar as duas instalações, é possível fazer upgrade da instalação que está executando o servidor de catálogos primeiro. Consulte

#### Considerações sobre Topologia de Armazenamento em Cache

Sua arquitetura pode usar armazenamento em cache de dados em memória local ou pode usar o armazenamento em cache de dados de cliente/servidor distribuído. Cada tipo de topologia de cache tem suas vantagens e desvantagens. A topologia de armazenamento em cache implementada depende dos requisitos do ambiente e do aplicativo. Para obter mais informações sobre as diferentes topologias de armazenamento em cache, consulte o “Planejando a Topologia” na página 10.

#### Considerações Sobre Capacidade de Dados

A lista a seguir inclui itens a serem considerados:

- **Número de sistemas e processadores:** Quantas máquinas e processadores físicos são necessários no ambiente?
- **Número de servidores:** Quantos servidores do eXtreme Scale são necessários para hospedar mapas do eXtreme Scale?
- **Número de partições:** A quantidade de dados armazenados nos mapas é um fator para determinar o número de partições necessárias.
- **Número de réplicas:** Quantas réplicas são necessárias para cada principal no domínio?

- **Replicação síncrona ou assíncrona:** Os dados são vitais de modo que a replicação síncrona seja necessária? Ou o desempenho é a maior prioridade, tornando a replicação assíncrona a escolha correta?
- **Tamanhos de heap:** Qual é quantidade de dados a ser armazenada em cada servidor?

Para obter uma discussão detalhada de cada uma dessas considerações, consulte “Capacidade do Ambiente de Planejamento” na página 53..

---

## Planejando a Topologia

Com WebSphere eXtreme Scale, sua arquitetura pode utilizar armazenamento em cache de dados em memória local ou armazenamento em cache de dados de cliente/servidor distribuídos. A arquitetura pode ter relacionamentos variados com seus bancos de dados. Também é possível configurar a topologia para estender diversos datacenters.

O WebSphere eXtreme Scale requer infraestrutura adicional mínima para operar. A infraestrutura consiste em scripts para instalar, iniciar e parar um aplicativo Java Platform, Enterprise Edition em um servidor. Os dados em cache são armazenados nos servidores de contêiner e os clientes se conectam remotamente com o servidor.

### Ambientes em Memória

Quando implementar em um local, um ambiente em memória, o WebSphere eXtreme Scale é executado em uma única Java Virtual Machine e não é replicado. Para configurar um ambiente local, é possível usar um arquivo XML do ObjectGrid ou as APIs do ObjectGrid.

### Ambientes Distribuídos

Quando você implementa em um ambiente distribuído, o WebSphere eXtreme Scale é executado em um conjunto de Java Virtual Machines, aumentando o desempenho, a disponibilidade e a escalabilidade. Com essa configuração, poderá utilizar a replicação de dados e criação de partições. Também é possível incluir servidores adicionais sem restaurar seus servidores eXtreme Scale existentes. Assim como ocorre com um ambiente local, um arquivo XML do ObjectGrid, ou uma configuração programática equivalente, é necessário em um ambiente distribuído. Você também deve fornecer um arquivo XML de política de implementação com detalhes de configuração.

É possível criar implementações simples ou grandes a nível de terabytes, em que milhares de servidores são necessários.

## Cache de Memória Local

Em um caso mais simples, o WebSphere eXtreme Scale pode ser utilizado como um cache de dados em memória local (não distribuído). O caso local pode beneficiar especialmente aplicativos de alta simultaneidade nos quais vários encadeamentos precisam acessar e modificar dados transientes. Os dados mantidos em uma grade de dados local podem ser indexados e recuperados utilizando consultas. As consultas ajudam você a trabalhar com grandes conjuntos de dados de memória. O suporte fornecido com o Java Virtual Machine (JVM), embora esteja pronto para uso, tem uma estrutura de dados limitada.

A topologia de cache em memória local para WebSphere eXtreme Scale é usado para oferecer acesso transacional e consistente aos dados temporários dentro de uma única Java virtual machine.

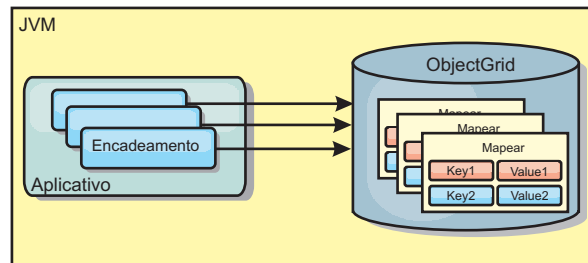


Figura 1. Cenário de Cache em Memória Local

## Vantagens

- Configuração simples: Um ObjectGrid pode ser criado programaticamente ou declarativamente com o arquivo XML do descritor de implementação ObjectGrid ou com outras estruturas como Spring.
- Rápido: Cada BackingMap pode ser ajustado de maneira independente para utilização de memória e simultaneidade ideais.
- Ideal para topologias de uma única Java virtual machine com pequenos conjuntos de dados ou para armazenamento em cache de dados frequentemente acessados.
- Transacional. As atualizações BackingMap podem ser agrupadas em uma única unidade de trabalho e podem ser integradas como um último participante nas transações de duas fases como transações JTA (Java Transaction Architecture).

## Desvantagens

- Não tolerante a falhas.
- Os dados não são replicados. Caches em memória são melhores para dados de referência somente para leitura.
- Não escalável. A quantidade de memória necessária pelo banco de dados pode ultrapassar a capacidade da Java virtual machine.
- Ocorrem problemas na inclusão de Java virtual machines:
  - Os dados não podem ser facilmente particionados
  - Você deve replicar manualmente o estado entre as Java virtual machines ou cada instância do cache poderá ter diferentes versões dos mesmos dados.
  - A invalidação é custosa.
  - Cada cache deve ser aquecido de maneira independente. O aquecimento é o período de carregamento de um conjunto de dados para que o cache seja preenchido com dados válidos.

## Quando Utilizar

A topologia de implementação de cache em memória local deve ser usada somente quando a quantidade de dados a serem armazenados em cache for pequena (puder ser colocada em uma única Java virtual machine) e for relativamente estável. Dados antigos devem ser tolerados com esta abordagem. A utilização de evictors para manter os dados mais frequentemente ou recentemente usados no cache pode ajudar a diminuir o tamanho do cache e a aumentar a relevância dos dados.

## Cache Local Replicado pelo Peer

Você deverá assegurar-se de que o cache esteja sincronizado se vários processos com instâncias de cache independentes existirem. Para assegurar-se de que as instâncias de cache estejam sincronizadas, ative um cache replicado por peer com o Java Message Service (JMS).

WebSphere eXtreme Scale inclui dois plug-ins que propagam automaticamente mudanças de transação entre instâncias do ObjectGrid peer. O plug-in JMSObjectGridEventListener propaga automaticamente as mudanças do eXtreme Scale usando JMS.

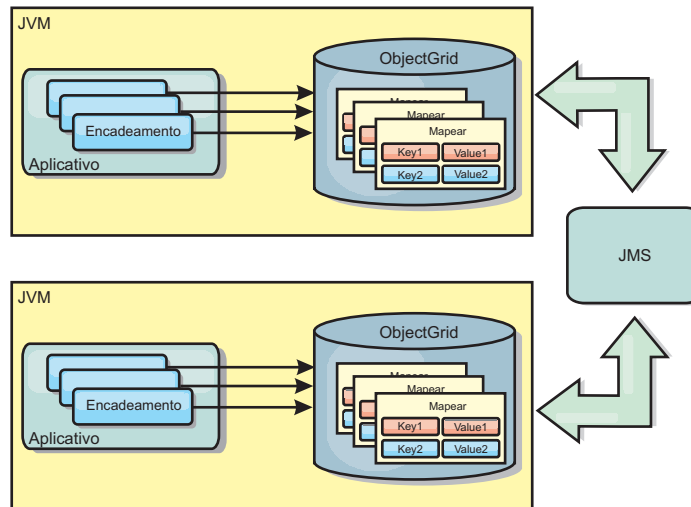


Figura 2. Cache Replicado pelo Peer com Alterações que são Propagadas com JMS

Se você estiver executando um ambiente do WebSphere Application Server, o plug-in TranPropListener também estará disponível. O plug-in TranPropListener usa o gerenciador de alta disponibilidade (HA) para propagar as mudanças em cada instância do cache de peer.

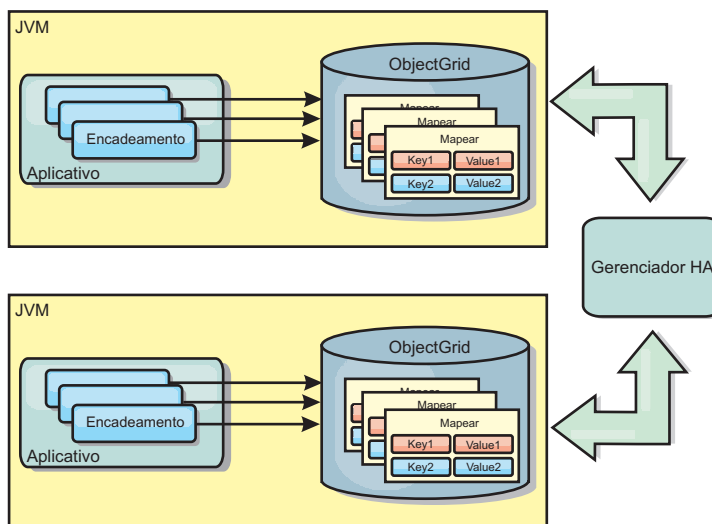


Figura 3. Cache Replicado pelo Peer com Alterações que são Propagadas com o Gerenciador de Alta Disponibilidade



## Vantagens

- Os dados são mais válidos porque os dados são atualizados mais frequentemente.
- Com o plug-in TranPropListener, como no ambiente local, o eXtreme Scale pode ser criado programaticamente ou declarativamente com o arquivo XML descritor de implementação do eXtreme Scale ou com outras estruturas como Spring. A integração com o gerenciador de alta disponibilidade é feita automaticamente.
- Cada BackingMap pode ser independentemente ajustado para melhor utilização da memória e concorrência.
- As atualizações BackingMap podem ser agrupadas em uma única unidade de trabalho e podem ser integradas como um último participante nas transações de duas fases como transações JTA (Java Transaction Architecture).
- Ideal para poucas topologias JVM com um conjunto de dados razoavelmente pequeno ou para armazenamento em cache de dados frequentemente acessados.
- As atualizações em cada eXtreme Scale são replicadas para todas as instâncias do eXtreme Scale do peer. As alterações são consistentes desde que uma assinatura durável seja utilizada.

## Desvantagens

- A configuração e a manutenção para o JMSObjectGridEventListener podem ser complexas. O eXtreme Scale pode ser criado programaticamente ou declarativamente com o arquivo XML descritor de implementação do eXtreme Scale ou com outras estruturas tais como Spring.
- Não escalável: A quantidade de memória necessária para que o banco de dados possa dominar a JVM.
- Funciona inadequadamente ao incluir Java Virtual Machines:
  - Os dados não podem ser facilmente particionados
  - A invalidação é custosa.
  - Cada cache deve ser aquecido de maneira independente

## Quando Utilizar

Use a topologia de implementação apenas quando a quantidade de dados a ser armazenada em cache for pequena, podendo ajustar-se a uma única JVM e se for relativamente estável.

## Cache Integrado

As grades do WebSphere eXtreme Scale podem ser executadas nos processos existentes como servidores eXtreme Scale integrados ou podem ser gerenciadas como processos externos.

As grades integradas são úteis quando você está executando em um servidor de aplicativos, como o WebSphere Application Server. É possível iniciar servidores eXtreme Scale que não são integrados usando scripts da linha de comandos e executar em um processo Java.

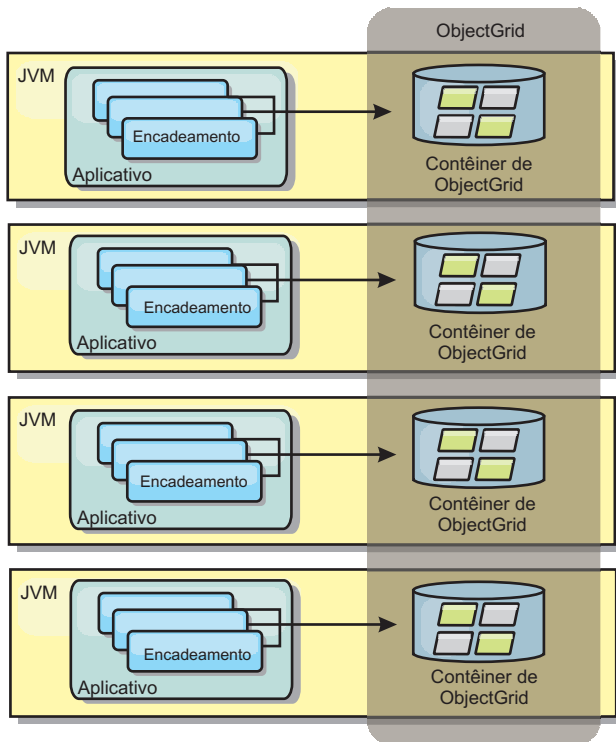


Figura 4. Cache Integrado

#### Vantagens

- Administração simplificada já que há menos processos para gerenciar.
- Implementação do aplicativo simplificada, já que a grade usa o carregador de classe do aplicativo cliente.
- Suporta particionamento e alta disponibilidade.

#### Desvantagens

- Aumento da área de cobertura da memória no processo do cliente já que todos os dados são colocados no processo.
- Aumento da utilização da CPU para atender pedidos de clientes.
- Mais difícil para manipular atualizações de aplicativo, pois os clientes estão usando os mesmos arquivos archive de Java do aplicativo que os servidores.
- Menos flexível. A escala dos clientes e servidores de grade não pode aumentar na mesma proporção. Quando os servidores são externamente definidos, é possível ter mais flexibilidade no gerenciamento do número de processos.

#### Quando Utilizar

Utilize grades integradas quando há grande quantidade de memória livre no processo do cliente para dados da grade e dados de failover potenciais.

Para obter mais informações, consulte o tópico sobre como ativar o mecanismo de invalidação do cliente no *Guia de Administração*.

## Cache Distribuído

O WebSphere eXtreme Scale é mais frequentemente usado como um cache compartilhado, para fornecer acesso transacional a dados para múltiplos

componentes onde, caso contrário, um banco de dados tradicional seria usado. O cache compartilhado elimina a necessidade de configurar um banco de dados.

## Coerência do Cache

O cache é coerente porque todos os clientes veem os mesmos dados no cache. Cada pedaço de dado é armazenado em exatamente um servidor no cache, evitando cópias de registros desperdiçadas que poderiam potencialmente conter diferentes versões dos dados. Um cache coerente também pode conter mais dados, à medida que mais servidores são incluídos na grade de dados, e escalado linearmente à medida que a grade cresce em tamanho. Como os clientes acessam dados a partir desta grade de dados com chamadas de processo remotas, ela também é conhecida como um cache remoto, ou cache distante. Através do particionamento de dados, cada processo contém um subconjunto exclusivo do conjunto de dados total. As grades de dados maiores podem conter mais dados e atender mais solicitações para esses dados. A coerência também elimina a necessidade de enviar dados de invalidação ao redor da grade de dados porque não há dados antigos. O cache coerente retém somente a cópia mais recente de cada pedaço de dados.

Se você estiver executando um ambiente do WebSphere Application Server, o plug-in TranPropListener também estará disponível. O plug-in TranPropListener usa o componente de alta disponibilidade (Gerenciador HA) do WebSphere Application Server para propagar as alterações para cada instância de cache ObjectGrid de peer.

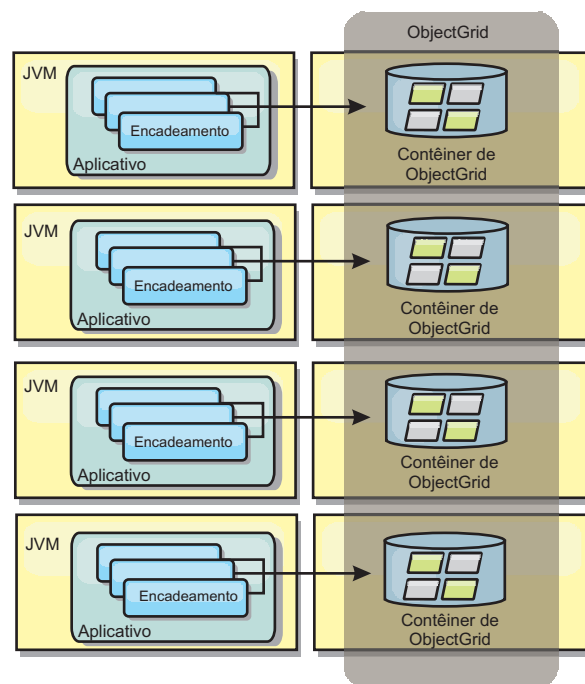


Figura 5. Cache Distribuído

## Cache Local

Opcionalmente, os clientes têm um cache local, sequencial quando o eXtreme Scale é usado em uma topologia distribuída. Este cache opcional é chamado de cache local, um ObjectGrid independente em cada cliente, servindo como um cache para

o cache remoto, do lado do cliente. O cache local é ativado por padrão quando o bloqueio é configurado como otimista ou nenhum e não pode ser utilizado quando é configurado como pessimista.

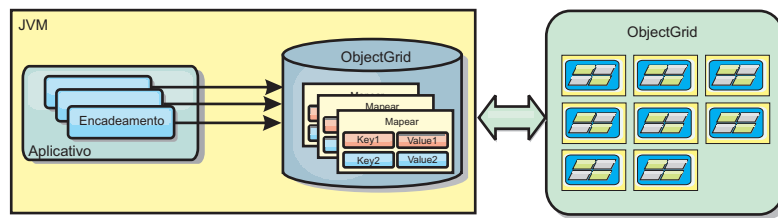


Figura 6. Cache Local

Um cache local é muito rápido porque fornece acesso em memória a um subconjunto do conjunto inteiro de dados em cache que é armazenado remotamente nos servidores do eXtreme Scale. O cache local não é particionado e contém dados de qualquer uma das partições eXtreme Scale remotas. O WebSphere eXtreme Scale pode ter até três camadas de cache, como a seguir.

1. O cache da camada da transação contém todas as alterações para uma única transação. O cache da transação contém uma cópia de trabalho dos dados até que a transação seja confirmada. Quando uma transação do cliente solicita dados de um ObjectMap, a transação é verificada primeiro
2. O cache local na camada do cliente contém um subconjunto de dados da camada do servidor. Quando a camada da transação não possui os dados, eles são buscados em uma camada do cliente, se disponíveis, e inseridos no cache da transação.
3. A grade de dados na camada do servidor contém a maioria dos dados e é compartilhada entre todos os clientes. A camada do servidor pode ser particionada, o que permite que uma grande quantidade de dados seja armazenada em cache. Quando o cache local do cliente não possui os dados, eles são buscados na camada do servidor e inseridos no cache cliente. A camada do servidor também pode ter um plug-in do Utilitário de Carga. Quando a grade não tem os dados necessários, o Utilitário de Carga é chamado e os dados resultantes são inseridos do armazém de dados de backend para a grade.

Para desativar o cache local, configure o atributo `numberOfBuckets` como 0 no arquivo descritor do ObjectGrid de substituição do cliente. Consulte o tópico sobre bloqueio de entrada de mapa para obter detalhes sobre estratégias de bloqueio do eXtreme Scale. O cache local também pode ser configurado para ter uma política de despejo configurada e diferentes plug-ins usando uma configuração do descritor do eXtreme Scale de substituição.

### Vantagem

- Tempo de resposta rápido porque todos os acessos aos dados é local. Procurar pelos dados no cache próximo primeiro economiza acesso à grade de servidores, o que torna até mesmo os dados remotos acessíveis localmente.

### Desvantagens

- Aumenta a duração dos dados antigos porque o cache próximo em cada camada pode estar fora de sincronização com os dados atuais na grade de dados.
- Depende de um evictor para invalidar dados a fim de evitar a falta de memória.

### Quando Utilizar

Utilize quando o tempo de resposta for importante e dados antigos puderem ser tolerados.

## Integração com o Banco de Dados: Armazenamento em Cache Write-behind, Sequencial e Lateral

O WebSphere eXtreme Scale é usado para colocar um banco de dados tradicional na frente e eliminar a atividade de leitura que normalmente é armazenada no banco de dados. Um cache coerente pode ser utilizado com um aplicativo direta ou indiretamente, utilizando um mapeador relacional de objeto. O cache coerente pode transferir o banco de dados ou o backend a partir das leituras. Em um cenário levemente mais complexo, tal como o acesso transacional a um conjunto de dados no qual apenas parte dos dados requer garantias de persistência tradicional, a filtragem pode ser utilizada para transferir até mesmo transações de gravação.

É possível configurar o WebSphere eXtreme Scale para funcionar como um espaço de processamento de banco de dados em memória altamente flexível. Entretanto, o WebSphere eXtreme Scale não é um object relational mapper (ORM). Ele não reconhece de onde vieram os dados na grade de dados. Um aplicativo ou um ORM pode colocar dados em um servidor eXtreme Scale. É responsabilidade da origem dos dados certificar-se de que eles permaneçam consistentes com o banco de dados no qual os dados se originaram. Isto significa que o eXtreme Scale não pode invalidar dados que são extraídos de um banco de dados automaticamente. O aplicativo ou mapeador deve fornecer esta função e gerenciar os dados armazenados no eXtreme Scale.

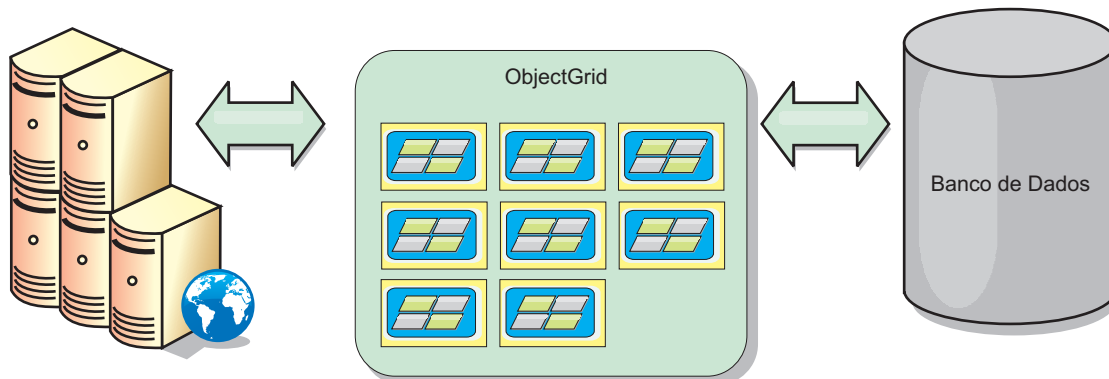


Figura 7. ObjectGrid como um Buffer de Banco de Dados

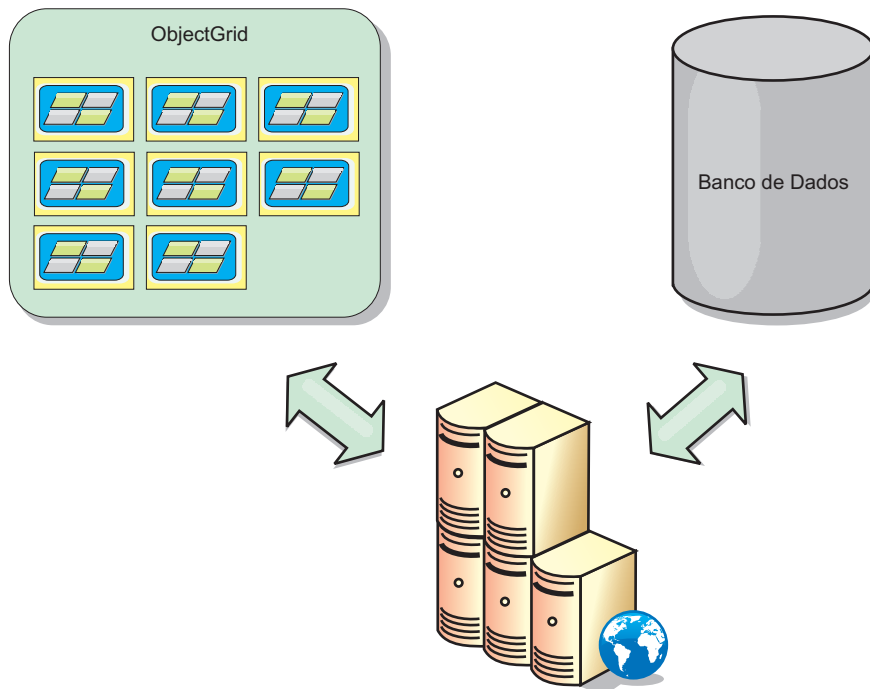


Figura 8. ObjectGrid como um Cache Secundário

### Cache Disperso e Completo

O WebSphere eXtreme Scale pode ser utilizado como um cache disperso ou um cache completo. Um cache disperso mantém apenas um subconjunto do total de dados, enquanto que um cache completo mantém todos os dados. Ele também pode ser preenchido gradualmente, conforme os dados são necessários. Os caches dispersos normalmente são acessados usando chaves (ao invés de índices ou consultas) porque os dados estão disponíveis apenas parcialmente.

### Cache Disperso

Quando uma chave não está presente em um cache disperso, ou os dados não estão disponíveis e uma falta de cache ocorre, a próxima camada é chamada. Os dados são buscados, a partir de um banco de dados, por exemplo, e inseridos na camada de cache da grade de dados. Se estiver usando uma consulta ou um índice, apenas os valores atualmente carregados serão acessados e as solicitações não serão encaminhadas para as outras camadas.

### Cache Completo

Um cache completo contém todos os dados necessários e pode ser acessado usando atributos não-chaves com índices ou consultas. Um cache completo é pré-carregado com dados a partir do banco de dados antes que o aplicativo tente acessar os dados. Um cache completo pode funcionar como uma substituição do banco de dados após os dados serem carregados. Como todos os dados estão disponíveis, as consultas e índices podem ser usados para localizar e agregar dados.

### Cache Secundário

Quando o WebSphere eXtreme Scale é usado como um cache secundário, o backend é usado com a grade de dados.

## Cache Secundário

É possível configurar o produto como um cache secundário para a camada de acesso a dados de um aplicativo. Neste cenário, o WebSphere eXtreme Scale é utilizado para armazenar temporariamente objetos que normalmente poderiam ser recuperados de um banco de dados de backend. Aplicativos verificam se a grade de dados contém os dados. Se os dados estiverem na grade de dados, eles serão retornados para o responsável pela chamada. Se os dados não existirem, eles serão recuperados a partir do banco de dados de backend. Os dados são então inseridos na grade de dados para que a próxima solicitação possa usar a cópia em cache. O diagrama a seguir ilustra como o WebSphere eXtreme Scale pode ser usado como um cache secundário com uma camada de acesso a dados arbitrários, como OpenJPA ou Hibernate.

### Plug-ins do cache secundário para Hibernate e OpenJPA

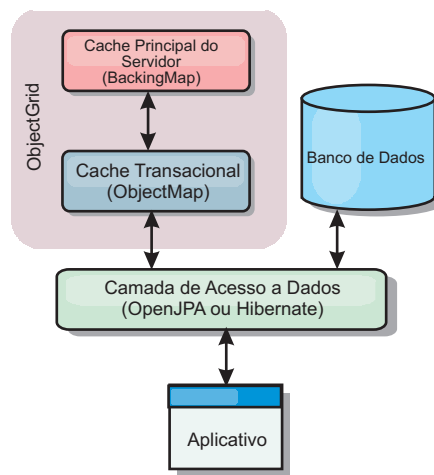


Figura 9. Cache Secundário

Os plug-ins de cache para ambos OpenJPA e Hibernate são incluídos no WebSphere eXtreme Scale, o que permite usar o produto como um cache secundário automático. Usar o WebSphere eXtreme Scale como um provedor de cache aumenta o desempenho ao ler e enfileirar dados e reduz a carga para o banco de dados. Existem vantagens que o WebSphere eXtreme Scale tem sobre as implementações de cache integrado porque o cache é automaticamente replicado entre todos os processos. Quando um cliente armazena em cache um valor, todos os outros clientes podem usar o valor em cache.

## Cache Sequencial

É possível configurar em cache sequencial para um backend de banco de dados ou como um cache secundário para um banco de dados. O armazenamento em cache sequencial utiliza o eXtreme Scale como o meio principal de interação com os dados. Quando o eXtreme Scale é usado como um cache sequencial, o aplicativo interage com o backend usando um plug-in Loader.

## Cache Sequencial

Quando usado como um cache sequencial, o WebSphere eXtreme Scale interage com o backend usando um plug-in Loader. Este cenário pode simplificar o acesso a dados porque os aplicativos podem acessar as APIs do eXtreme Scale diretamente. Vários cenários de armazenamento em cache diferentes são suportados no eXtreme

Scale para garantir que os dados no cache e os dados no backend sejam sincronizados. O diagrama a seguir ilustra como um cache sequencial interage com o aplicativo e o back end.

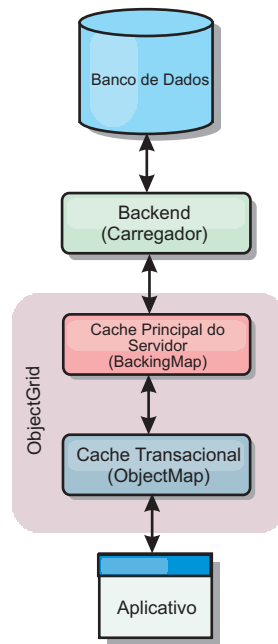


Figura 10. Cache Sequencial

A opção de armazenamento em cache em linha simplifica o acesso aos dados pois ela permite que os aplicativos acessem diretamente as APIs do eXtreme Scale. O WebSphere eXtreme Scale suporta diversos cenários de armazenamento em cache em linha, como os seguintes:

- Read-through
- Write-through
- Write-behind

### Cenário de Armazenamento em Cache Read-through

Um cache read-through é um cache disperso que lentamente carrega entradas de dados por chave à medida que elas são solicitadas. Isto é feito sem exigir que o responsável pela chamada saiba quais entradas estão preenchidas. Se os dados não puderem ser localizados no cache do eXtreme Scale, o eXtreme Scale irá recuperar os dados ausentes do plug-in do utilitário de carga, que carrega os dados do banco de dados backend e insere os dados no cache. Pedidos subsequentes para a mesma chave de dados serão localizados no cache até que ele possa ser removido, invalidado ou despejado.



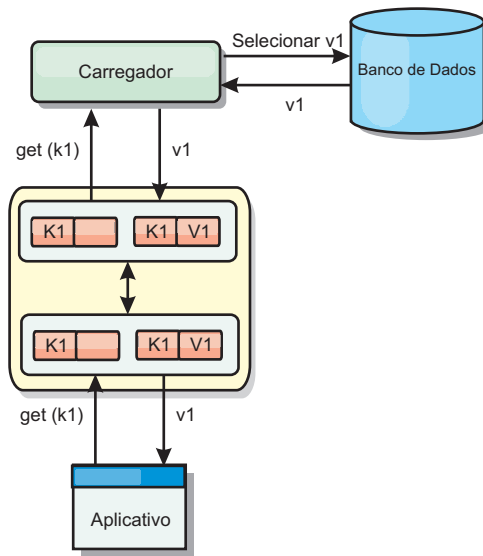


Figura 11. Armazenamento em Cache Read-through

### Cenário de Armazenamento em Cache Write-through

Em um cache write-through, cada gravação no cache é gravada de maneira síncrona no banco de dados utilizando o Utilitário de Carga. Este método fornece consistência com o backend, mas diminui o desempenho de gravação pois a operação do banco de dados é síncrona. Como o cache e o banco de dados são ambos atualizados, as leituras subsequentes para os mesmos dados serão localizadas no cache, evitando a chamada do banco de dados. Um cache write-through sempre é utilizado em conjunto com um cache read-through.

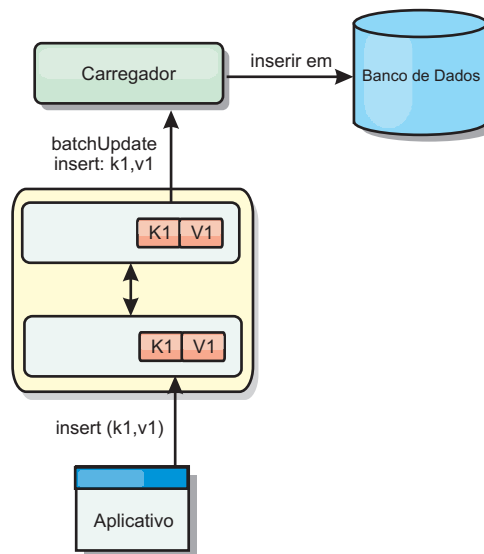


Figura 12. Armazenamento em Cache Write-through

### Cenário de Armazenamento em Cache Write-behind

A sincronização do banco de dados pode ser aprimorada pela gravação de alterações de maneira assíncrona. Isto é conhecido como um cache write-behind ou write-back. Alterações que normalmente poderiam ser gravadas de maneira

síncrona no utilitário de carga são, ao invés disso, armazenadas em buffer no eXtreme Scale e gravadas no banco de dados utilizando um encadeamento secundário. O desempenho de gravação é significativamente aumentado pois a operação do banco de dados é removida da transação do cliente e as gravações do banco de dados podem ser compactadas.

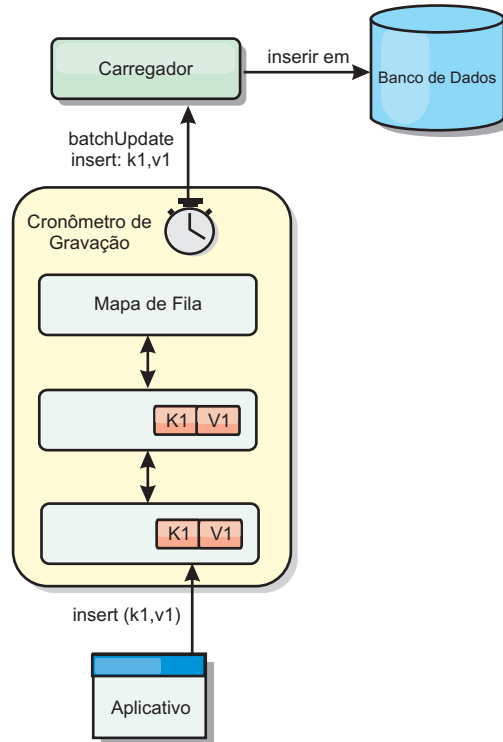


Figura 13. Armazenamento em Cache Write-behind

## Armazenamento em Cache Write-behind

É possível utilizar armazenamento em cache write-behind para reduzir o gasto adicional que ocorre durante a atualização de um banco de dados que você está utilizando como back end.

### Visão Geral do Armazenamento em Cache Write-Behind

O armazenamento em cache write-behind enfileira assincronamente as atualizações no plug-in do Utilitário de Carga. É possível melhorar o desempenho desconectando atualizações, inserções e remoções para um mapa, a sobrecarga de atualização do banco de dados de backend. A atualização assíncrona é executada após um atraso baseado em tempo (por exemplo, cinco minutos) ou um atraso baseado em entradas (1000 entradas).

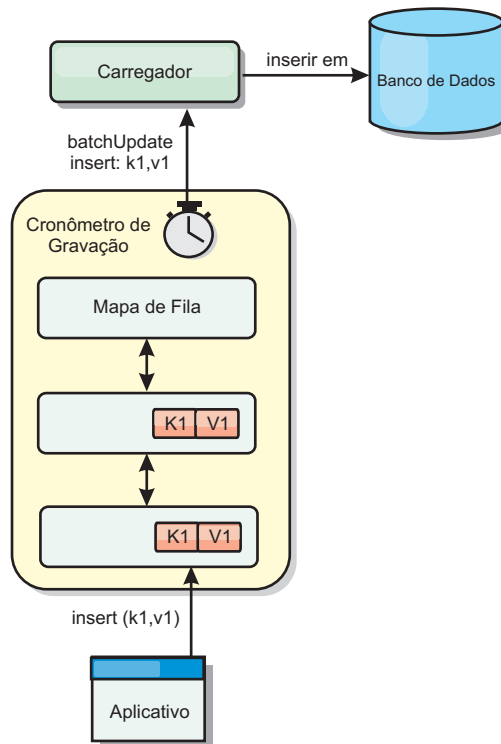


Figura 14. Armazenamento em Cache Write-behind

A configuração write-behind em um BackingMap cria um encadeamento entre o utilitário de carga e o mapa. O utilitário de carga então delega pedidos de dados através do encadeamento de acordo com as definições da configuração no método BackingMap.setWriteBehind. Quando uma transação do eXtreme Scale insere, atualiza ou remove uma entrada de um mapa, um objeto LogElement é criado para cada um destes registros. Estes elementos são enviados para o utilitário de carga write-behind e enfileirados em um ObjectMap especial denominado mapa de fila. Cada mapa de apoio com a configuração write-behind ativada possui seus próprios mapas de fila. Um encadeamento write-behind remove periodicamente os dados enfileirados dos mapas de fila e executa o push deles para o utilitário de carga de backend real.

O utilitário de carga write-behind enviará apenas os tipos insert, update e delete dos objetos LogElement para o utilitário de carga real. Todos os outros tipos de objetos LogElement, por exemplo, o tipo EVICT, são ignorados.

O suporte write-behind é uma extensão do plug-in do Carregador, que você usa para integrar o eXtreme Scale ao banco de dados. Por exemplo, consulte as informações do “Configurando Utilitários de Carga do JPA” na página 347 sobre como configurar um carregador JPA.

## Benefícios

Ativar o suporte write-behind possui os seguintes benefícios:

- **Isolamento de falha de backend:** O armazenamento em cache write-behind fornece uma camada de isolamento das falhas de backend. Quando o banco de dados de backend falha, as atualizações são enfileiradas no mapa de fila. Os

aplicativos podem continuar a conduzir transações para o eXtreme Scale. Quando o backend se recupera, os dados no mapa de fila são enviados para o backend.

- **Carga de backend reduzida:** O utilitário de carga write-behind mescla as atualizações em uma base de chave, portanto, apenas uma atualização mesclada por chave existe no mapa de fila. Esta mesclagem diminui o número de atualizações no backend.
- **Desempenho de transação aprimorado:** Tempos de transação do eXtreme Scale individuais são reduzidos porque a transação não precisa aguardar até que os dados sejam sincronizados com o backend.

## Considerações de Design do Aplicativo

Ativar o suporte write-behind é simples, mas o design de um aplicativo para trabalhar com o suporte write-behind precisa de consideração cuidadosa. Sem o suporte de write-behind, a transação de ObjectGrid engloba a transação de backend. A transação do ObjectGrid inicia antes da transação de backend iniciar e termina após a transação de backend terminar.

Com suporte write-behind ativado, a transação do ObjectGrid é concluída antes que a transação de backend inicie. A transação do ObjectGrid e a transação de backend não estão acopladas.

## Limitadores de Integridade Referencial

Cada mapa de apoio que é configurado com suporte write-behind possui seu próprio encadeamento write-behind para enviar os dados para o backend. Portanto, os dados que são atualizados em diferentes mapas em uma transação do ObjectGrid são atualizados no backend em diferentes transações de backend. Por exemplo, a transação T1 atualiza a chave key1 no mapa Map1 e a chave key2 no mapa Map2. A atualização da key1 para o mapa Map1 é atualizada no backend em uma transação de backend e a key2 atualizada para o mapa Map2 é atualizada no backend em outra transação de backend por encadeamentos write-behind diferentes. Se os dados armazenados no Map1 e Map2 possuírem relações, tais como limitadores de chave estrangeira no backend, as atualizações podem falhar.

Ao projetar os limitadores de integridade referencial em seu banco de dados de backend, certifique-se de que atualizações fora de ordem sejam permitidas.

## Comportamento do Bloqueio de Mapa de Fila

Outra grande diferença de comportamento da transação é o comportamento do bloqueio. O ObjectGrid suporta três diferentes estratégias de bloqueio: PESSIMISTIC, OPTIMISITIC e NONE. Os mapas de fila write-behind utilizam a estratégia de bloqueio pessimista não importando qual estratégia de bloqueio está configurada para seu mapa de apoio. Existem dois diferentes tipos de operações que adquirem um bloqueio no mapa de fila:

- Quando uma transação do ObjectGrid é confirmada ou um flush (flush de mapa ou flush de sessão) acontece, a transação lê a chave no mapa de fila e coloca um bloqueio S na chave.
- Quando uma transação do ObjectGrid é confirmada, a transação tenta atualizar o bloqueio S para o bloqueio X na chave.

Devido a este comportamento do mapa de fila extra, é possível visualizar algumas diferenças de comportamento de bloqueio.

- Se o mapa do usuário for configurado como a estratégia de bloqueio PESSIMISTIC, não há muita diferença no comportamento de bloqueio. Sempre que um flush ou commit é chamado, um bloqueio S é colocado na mesma chave no mapa de fila. Durante o momento do commit, um bloqueio X não é adquirido apenas para a chave no mapa do usuário, ele também é adquirido para a chave no mapa de fila.
- Se o mapa do usuário for configurado com a estratégia de bloqueio OPTIMISTIC ou NONE, a transação do usuário seguirá o padrão de estratégia de bloqueio PESSIMISTIC. Sempre que um flush ou commit é chamado, um bloqueio S é adquirido para a mesma chave no mapa de fila. Durante o momento do commit, um bloqueio X é adquirido para a chave no mapa de fila utilizando a mesma transação.

## Novas Tentativas de Transações do Utilitário de Carga

O ObjectGrid não suporta transações 2-phase ou XA. O encadeamento write-behind remove registros do mapa de fila e atualiza os registros no backend. Se o servidor falhar no meio da transação, algumas atualizações de backend podem ser perdidas.

O utilitário de carga write-behind automaticamente tentará gravar novamente transações falhas e enviará uma LogSequence duvidosa para o backend para evitar a perda de dados. Esta ação requer que o utilitário de carga seja idempotente, o que significa que o Loader.batchUpdate(TxId, LogSequence) é chamado duas vezes com o mesmo valor, ele fornece o mesmo resultado como se tivesse sido aplicado uma vez. As implementações do utilitário de carga devem implementar a interface RetryableLoader para ativar este recurso. Consulte a documentação da API para obter mais detalhes.

## Falha do Utilitário de Carga

O plug-in do utilitário de carga pode falhar quando não consegue se comunicar com o back end do banco de dados. Isto pode acontecer se o servidor de banco de dados ou a conexão de rede estiver inativa. O utilitário de carga write-behind irá enfileirar as atualizações e tentará executar o push das alterações de dados para o utilitário de carga periodicamente. O utilitário de carga deve notificar o tempo de execução do ObjectGrid que há um problema de conectividade do banco de dados lançando uma exceção LoaderNotAvailableException.

Portanto, a implementação do Utilitário de Carga deve poder distinguir uma falha de dados ou uma falha de utilitário de carga físico. A falha de dados deve ser lançada ou relançada como uma LoaderException ou uma OptimisticCollisionException, mas uma falha de utilitário de carga físico deve ser lançada ou relançada como uma LoaderNotAvailableException. O ObjectGrid manipula estas exceções de maneira diferente:

- Se uma LoaderException for capturada pelo utilitário de carga write-behind, o utilitário de carga write-behind a considerará falha devido a alguma falha de dados, tal como uma falha de chave duplicada. O utilitário de carga write-behind irá remover a atualização do lote e tentará atualizar um registro em um momento para isolar a falha de dados. Se uma {{LoaderException}} for capturada durante uma atualização de registro, um registro de atualização falho é criado e registrado no mapa de atualização falho.
- Se uma LoaderNotAvailableException for capturada pelo utilitário de carga write-behind, o utilitário de carga write-behind a considerará falha porque não pode se conectar ao final do banco de dados, por exemplo, porque o backend do

banco de dados estiver inativo, uma conexão com o banco de dados não estiver disponível ou a rede estiver inativa. O utilitário de carga write-behind aguardará por 15 segundo e, em seguida, tentará novamente executar uma atualização de lote no banco de dados.

O erro comum é lançar uma `LoaderException` enquanto uma `LoaderNotAvailableException` deve ser lançada. Todos os registros enfileirados no utilitário de carga write-behind se tornarão atualizações de registro falhas, o que frustra o propósito do isolamento de falha do backend.

## Considerações sobre Desempenho

O suporte ao armazenamento em cache write-behind aumenta o tempo de resposta removendo a atualização do utilitário de carga da transação. Ele também aumenta o rendimento do banco de dados porque as atualizações de banco de dados são combinadas. É importante compreender o gasto adicional introduzido pelo encadeamento write-behind, que executa o pull dos dados da mapa de fila e executa o push para o utilitário de carga.

A contagem máxima de atualização ou o tempo máximo de atualização necessário a ser ajustado com base nos padrões de uso e no ambiente esperados. Se o valor da contagem máxima de atualização ou o tempo máximo de atualização for muito pequeno, o gasto adicional do encadeamento write-behind pode exceder os benefícios. Configurar um valor maior para estes dois parâmetros também pode aumentar o uso da memória para enfileirar os dados e aumentar o tempo de envelhecimento dos registros do banco de dados.

Para obter um melhor desempenho, ajuste os parâmetros write-behind com base nos seguintes fatores:

- Proporção de transações de leitura e gravação
- Mesma frequência de atualização de registro
- Latência de atualização de banco de dados.

## Utilitários de Carga

Com um plug-in Carregador, uma grade de dados pode se comportar como um cache de memória para dados que normalmente são mantidos em um armazenamento persistente no mesmo sistema ou em outro sistema. Geralmente, um banco de dados ou sistema de arquivos é utilizado como o armazenamento persistente. Uma JVM (Java Virtual Machine) também pode ser usada como a origem de dados, permitindo que caches baseados em hub seja construído usando o eXtreme Scale. Um utilitário de carga possui a lógica para leitura e gravação de dados para um armazenamento persistente e a partir dele.

## Visão Geral

Os utilitários de carga são plug-ins de mapa de apoio que são chamados quando são feitas alterações no mapa de apoio ou quando o mapa de apoio não pode atender a um pedido de dados (um erro de cache). O utilitário de carga é chamado quando o cache não pode satisfazer uma solicitação para uma chave, fornecendo capacidade read-through e lazy-population do cache. Um utilitário de carga também permite atualizações no banco de dados quando os valores do cache mudam. Todas as mudanças dentro de uma transação são agrupadas para permitir que o número de interações do banco de dados seja minimizado. Um plug-in `TransactionCallback` é usado em conjunto com o utilitário de carga para acionar a demarcação da transação backend. O uso deste plug-in é importante quando

múltiplos mapas são incluídos em uma única transação ou quando os dados da transação forem enviados para o cache sem consolidação.

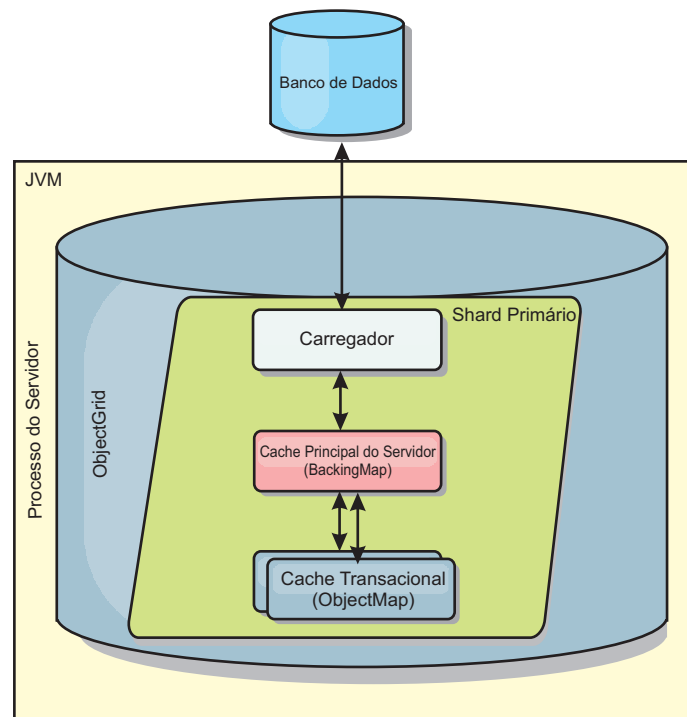


Figura 15. Utilitário de Carga

O utilitário de carga também pode usar atualizações super qualificadas para evitar manter bloqueios do banco de dados. Armazenar um atributo de versão no valor do cache, permite ao utilitário de carga ver a imagem antes e depois do valor quando ele for atualizado no cache. Este valor pode assim ser usado ao atualizar o banco de dados ou backend para verificar se os dados foram atualizados. Um Loader também pode ser configurado para pré-carregar a grade de dados quando for iniciado. Quando particionado, uma instância do utilitário de carga é associada a cada partição. Se o Mapa "Company" tiver dez partições, haverá dez instâncias do utilitário de carga, uma por partição primária. Quando shard primário para o Mapa é ativado, o método preloadMap para o utilitário de carga é chamado síncrona ou assincronamente, o qual permite o carregamento da partição do mapa com dados a partir do backend ocorra automaticamente. Quando chamado sincronamente, todas as transações do cliente são bloqueadas, evitando o acesso inconsistente à grade de dados. Como alternativa, um pré-utilitário de cliente pode ser usado para carregar a grade de dados inteira.

Dois utilitários de carga integrados podem simplificar muito a integração com back ends de banco de dados relacional. Os utilitários de carga JPA utilizam os recursos ORM (Object-Relational Mapping) de ambas as implementações OpenJPA e Hibernate da especificação JPA (Java Persistence API). Consulte Carregadores JPA para obter mais informações.

Se estiver usando carregadores em uma configuração de diversos datacenters, você deverá considerar como dados de revisão e a consistência de cache são mantidos entre as grades de dados. Para obter informações adicionais, consulte "Considerações Sobre o Carregador em uma Topologia Multimestre" na página 40.

## Configuração do Utilitário de Carga

Para incluir um Utilitário de Carga na configuração do BackingMap, é possível utilizar a configuração programática ou a configuração do XML. Um utilitário de carga possui o seguinte relacionamento com um mapa de apoio.

- Um mapa de apoio pode ter apenas um utilitário de carga.
- Um mapa de apoio de cliente (cache local) não pode ter um utilitário de carga.
- Uma definição de utilitário de carga pode ser aplicado a múltiplos mapas de apoio, mas cada mapa de apoio possui sua própria instância do utilitário de carga.

## Pré-carregamento de Dados e Aquecimento

Em vários cenários que incorporam o uso de um carregador, é possível preparar sua grade de dados ao pré-carregá-lo com dados.

Quando usado como um cache completo, a grade de dados deve manter todos os dados e deve ser carregada antes que quaisquer clientes possam se conectar a ele. Quando estiver usando um cache esparso, é possível efetuar um warm-up do cache com dados para que os clientes possam ter acesso imediato aos dados quando eles se conectarem.

Existem duas abordagens para o pré-carregamento de dados na grade de dados: Usando um plug-in do Carregador ou usando um carregador do cliente, conforme descrito nas seguintes seções.

## Plug-in do Utilitário de Carga

O plug-in do carregador é associado a cada mapa e é responsável pela sincronização de um único shard de partição primário com o banco de dados. O método `preloadMap` do plug-in do utilitário de carga é chamado automaticamente quando um shard é ativado. Por exemplo, se você tiver 100 partições, existem 100 instâncias do carregador, cada um carregando os dados para sua partição. Quando executado de modo síncrono, todos os clientes serão bloqueados até que o pré-carregamento seja concluído.



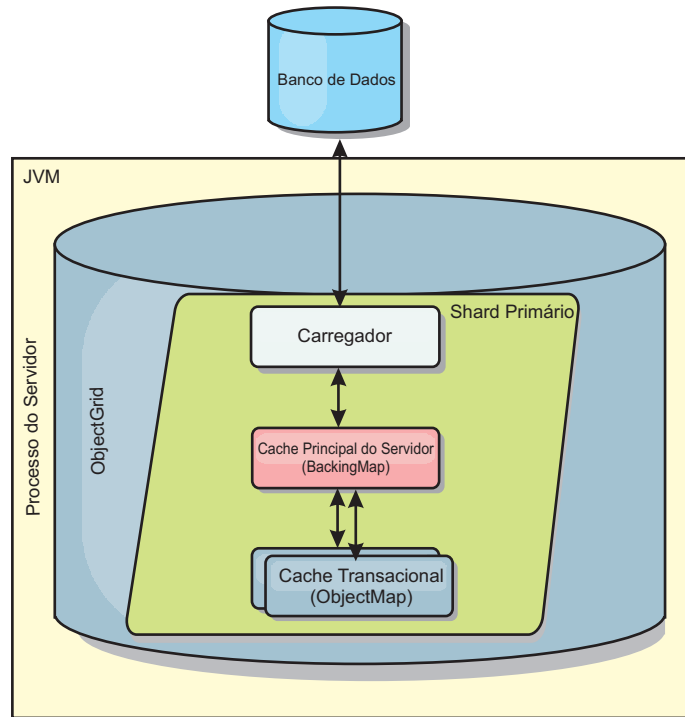


Figura 16. Plug-in do Utilitário de Carga

### Utilitário de Carga do Cliente

Um utilitário de carga do cliente é um padrão para uso de um ou mais clientes para carregar a grade com dados. O uso de múltiplos clientes para carregamento de dados da grade pode ser efetivo quando o esquema de partições não está armazenado no banco de dados. É possível chamar os carregadores de cliente manual ou automaticamente quando a grade de dados é iniciada. Os carregadores do cliente podem usar, opcionalmente, o StateManager para configurar o estado da grade de dados no modo de pré-carregamento, para que os clientes não possam acessar a grade enquanto ela estiver pré-carregando os dados. WebSphere eXtreme Scale inclui um carregador baseado em Java Persistence API (JPA) pode ser usado para carregar automaticamente a grade de dados com os provedores JPA OpenJPA ou Hibernate. Para obter mais informações sobre os provedores de cache, consulte “Plug-in do Cache JPA Nível 2 (L2)” na página 325.

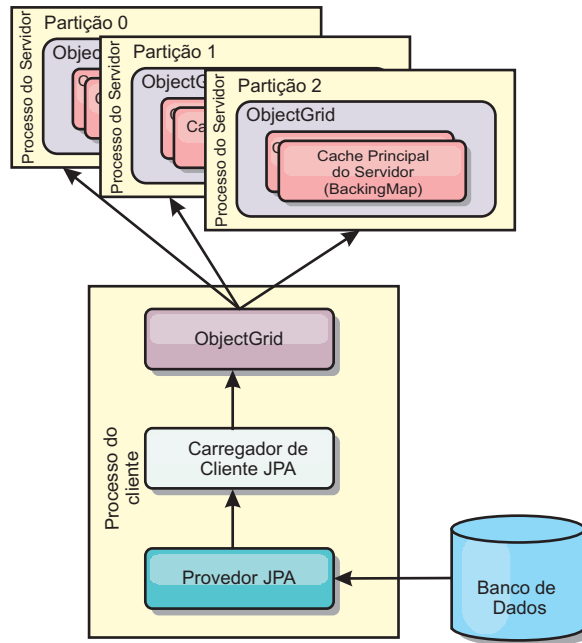


Figura 17. Utilitário de Carga do Cliente

## Técnicas de Sincronização de Banco de Dados

Quando o WebSphere eXtreme Scale é utilizado como um cache, os aplicativos devem ser criados para tolerar dados antigos se o banco de dados puder ser atualizado de maneira independente de uma transação do eXtreme Scale. Para atuar como um espaço de processamento de banco de dados de memória sincronizado, o eXtreme Scale fornece várias maneiras de manter o cache atualizado.

## Técnicas de Sincronização de Banco de Dados

### Atualização periódica

O cache pode ser automaticamente invalidado ou atualizado automaticamente usando o atualizador de banco de dados baseado em tempo JPA (Java Persistence API). O atualizador periodicamente consulta o banco de dados usando um provedor JPA para todas as atualizações ou inserções que ocorreram desde a atualização anterior. Quaisquer alterações identificadas são automaticamente invalidadas ou atualizadas quando utilizadas com um cache disperso. Se utilizadas com um cache completo, as entradas podem ser descobertas e inseridas no cache. As entradas nunca são removidas do cache.

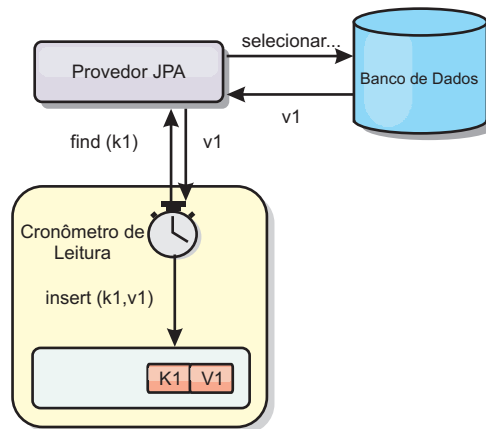


Figura 18. Atualização Periódica

### Despejo

Os caches dispersos podem utilizar políticas de despejo para automaticamente remover dados do cache sem afetar o banco de dados. Há três políticas integradas incluídas no eXtreme Scale: time-to-live, least-recently-used e least-frequently-used. Todas as três políticas podem opcionalmente despejar dados mais agressivamente à medida que a memória torna-se restrita ao ativar a opção de despejo baseada em memória.

### Invalidação Baseada em Eventos

Caches dispersos e completos podem ser invalidados ou atualizados usando um gerador de eventos como JMS (Java Message Service). A invalidação utilizando JMS pode ser manualmente vinculada a qualquer processo que atualiza o backend utilizando um acionador do banco de dados. Um plug-in `ObjectGridEventListener` do JMS é fornecido no eXtreme Scale que pode notificar quando o cache do servidor tiver qualquer alteração. Isto pode diminuir a quantidade de tempo que o cliente pode visualizar dados antigos.

### Invalidação programática

As APIs do eXtreme Scale permitem interação manual do cache local e do servidor usando os métodos de API `Session.beginNoWriteThrough()`, `ObjectMap.invalidate()` e `EntityManager.invalidate()`. Se um processo do cliente ou servidor não precisar mais de uma parte dos dados, os métodos de invalidação podem ser utilizados para remover dados do cache local ou do servidor. O método `beginNoWriteThrough` aplica qualquer operação `ObjectMap` ou `EntityManager` para o cache local sem chamar o utilitário de carga. Se chamada a partir de um cliente, a operação é aplicável apenas para o cache local (o utilitário de carga remoto não é chamado). Se chamada no servidor, a operação é aplicável apenas ao cache principal do servidor sem chamar o utilitário de carga.

### Invalidação de Dados

Para remover os dados de cache de escala, é possível usar um mecanismo de invalidação programático ou baseado em evento.

## Invalidação Baseada em Evento

Caches dispersos e completos podem ser invalidados ou atualizados usando um gerador de eventos como JMS (Java Message Service). A invalidação utilizando JMS pode ser manualmente vinculada a qualquer processo que atualiza o backend utilizando um acionador do banco de dados. É fornecido um plug-in JMS `ObjectGridEventListener` no eXtreme Scale que pode notificar os clientes quando o cache do servidor é alterado. Esse tipo de notificação diminui a quantidade de tempo em que o cliente pode ver dados antigos.

A invalidação baseada em evento normalmente consiste nos três componentes a seguir.

- **Fila de eventos:** Uma fila de eventos armazena os eventos de mudança de dados. Ela pode ser uma fila JMS, um banco de dados, uma fila FIFO em memória ou qualquer tipo de manifesto, contanto que possa gerenciar os eventos de mudança de dados.
- **Publicador de evento:** Um publicador de evento publica os eventos de mudança de dados na fila de eventos. Um publicador de evento geralmente é um aplicativo que você cria ou uma implementação de plug-in do eXtreme Scale. O publicador de evento sabe quando os dados são alterados ou quando ele mesmo altera os dados. Quando uma transação é confirmada, eventos são gerados para os dados alterados e o publicador de eventos publica esses eventos na fila de eventos.
- **Consumidor de evento:** Um consumidor de evento consome eventos de mudança de dados. O consumidor de evento geralmente é um aplicativo para garantir que os dados da grade de destino sejam atualizados com a mudança mais recente das outras grades. Esse consumidor de evento interage com a fila de eventos para obter a mudança de dados mais recente, além de aplicar as mudanças de dados na grade de destino. Os consumidores de evento podem utilizar APIs do eXtreme Scale para invalidar dados antigos ou atualizar a grade com os dados mais recentes.

Por exemplo, `JMSObjectGridEventListener` tem uma opção para um modelo de cliente/servidor, no qual a fila de eventos é um destino JMS designado. Todos os processos do servidor são publicadores de eventos. Quando uma transação é confirmada, o servidor obtém as mudanças de dados e as publica no destino JMS designado. Todos os processos do cliente são consumidores de evento. Eles recebem as mudanças de dados do destino JMS designado e aplicam as mudanças no cache perto do cliente.

Consulte o tópico sobre a ativação do mecanismo de invalidação de cliente no *Guia de Administração* para obter mais informações.

## Invalidação Programática

As APIs do WebSphere eXtreme Scale permitem interação manual do cache local e do servidor usando os métodos de API `Session.beginNoWriteThrough()`, `ObjectMap.invalidate()` e `EntityManager.invalidate()`. Se um processo do cliente ou servidor não precisar mais de uma parte dos dados, os métodos de invalidação podem ser utilizados para remover dados do cache local ou do servidor. O método `beginNoWriteThrough` aplica qualquer operação `ObjectMap` ou `EntityManager` para o cache local sem chamar o utilitário de carga. Se chamada a partir de um cliente, a operação é aplicável apenas para o cache local (o utilitário de carga remoto não é chamado). Se chamada no servidor, a operação é aplicável apenas ao cache principal do servidor sem chamar o utilitário de carga.

É possível utilizar invalidação programática com outras técnicas para determinar quando invalidar os dados. Por exemplo, esse método de invalidação utiliza mecanismos de invalidação baseados em evento para receber eventos de mudança de dados e depois utiliza as APIs para invalidar os dados antigos.

## Indexação

Use o plug-in `MapIndexPlugin` para construir um índice ou vários índices em um `BackingMap` para suportar acesso a dados sem chave.

## Tipos e Configuração de Índice

O recurso de indexação é representado pelo plug-in `MapIndexPlugin` ou `Index`, para abreviar. O Índice é um plug-in `BackingMap`. Um `BackingMap` pode ter múltiplos plug-ins de Índice configurados, enquanto cada um seguir as regras de configuração de Índice.

O recurso de indexação pode ser usado para construir um ou mais índices em um `BackingMap`. Um índice é construído a partir de um atributo ou uma lista de atributos de um objeto no `BackingMap`. Este recurso fornece uma maneira para os aplicativos localizarem determinados objetos mais rapidamente. Com o recurso de indexação, os aplicativos podem localizar objetos com um valor específico ou em um intervalo com os valores de atributos indexados.

Dois tipos de indexação são possíveis: estática e dinâmica. Com a indexação estática, é necessário configurar o plug-in de índice no `BackingMap` antes de inicializar a instância do `ObjectGrid`. É possível fazer esta configuração com a configuração XML ou programática do `BackingMap`. A indexação estática inicia a construção de um índice durante a inicialização do `ObjectGrid`. O índice é sempre sincronizado com o `BackingMap` e está pronto para utilização. Depois de o processo de indexação estático iniciar, a manutenção do índice é parte do processo de gerenciamento de transação do `eXtreme Scale`. Quando as consolidações de transações mudam, estas alterações também atualizam o índice estático, e as alterações de índice são recuperadas se a transação for recuperada.

Com a indexação dinâmica, é possível criar um índice num `BackingMap` antes ou depois da inicialização da instância do `ObjectGrid` que o contém. Os aplicativos possuem controle de ciclo de vida sobre o processo de indexação dinâmica para que você possa remover um índice dinâmico quando ele não for mais necessário. Quando um aplicativo cria um índice dinâmico, o índice pode não estar pronto para utilização imediata devido ao tempo gasto na conclusão do processo de construção do índice. Como a quantidade de tempo depende da quantidade de dados indexados, a interface `DynamicIndexCallback` é fornecido para aplicativos que desejam receber notificações quando ocorrem determinados eventos de indexação. Estes eventos incluem `ready`, `error` e `destroy`. Os aplicativos podem implementar esta interface de retorno de chamada e registrar-se no processo de indexação dinâmica.

Se um `BackingMap` tiver um plug-in de índice configurado, você pode obter o objeto de proxy do índice do aplicativo a partir do `ObjectMap` correspondente. Chamar o método `getIndex` no `ObjectMap` e passar o nome do plug-in de índice retornam o objeto de proxy do índice. Você deve efetuar o cast do objeto de proxy do índice para uma interface de índice adequada do aplicativo, como `MapIndex`, `MapRangeIndex` ou uma interface de índice customizada. Após obter o objeto de proxy do índice, é possível utilizar métodos definidos na interface do índice do aplicativo para localizar objetos armazenados em cache.

As etapas para utilizar a indexação estão resumidas na lista a seguir:

- Incluir plug-ins de indexação, estáticos ou dinâmicos, no BackingMap;
- Obter um objeto de proxy de indexação do aplicativo, emitindo o método `getIndex` do `ObjectMap`.
- Direcione o objeto de proxy de índice a uma interface de índice de aplicativo apropriado, como `MapIndex`, `MapRangeIndex`, ou uma interface de índice customizada.
- Utilizar os métodos definidos na interface `Index` do aplicativo para localizar objetos no cache.

A classe `HashIndex` é a implementação de plug-in de índice integrada que pode suportar ambas as interfaces integradas de índice do aplicativo: `MapIndex` e `MapRangeIndex`. Também é possível criar seus próprios índices. É possível incluir o `HashIndex` como um índice estático ou dinâmico no `BackingMap`, obter o objeto proxy do índice `MapIndex` ou `MapRangeIndex` e usar o objeto proxy do índice para localizar objetos em cache.

### Índice Padrão

Se desejar iterar por meio das chaves em um mapa local, o índice padrão poderá ser usado. Este índice não requer nenhuma configuração, porém ele deve ser usado com relação ao shard, usando um agente ou uma instância do `ObjectGrid` recuperados a partir do método `ShardEvents.shardActivated(shard do ObjectGrid)`.

### Consideração sobre a Qualidade dos Dados

Os resultados dos métodos de consulta de índice somente representar uma captura instantânea de dados em um determinado ponto no tempo. Nenhum bloqueio contra as entradas de dados é obtido depois do retorno dos resultados para o aplicativo. O aplicativo deve estar ciente de que podem ocorrer atualizações de dados em um conjunto de dados retornado. Por exemplo, o aplicativo obtém a chave de um objeto armazenado em cache executando o método `findAll` de `MapIndex`. Este objeto de chave retornado está associado a uma entrada de dados no cache. O aplicativo deve poder executar o método `get` no `ObjectMap` para localizar um objeto fornecendo o objeto chave. Se outra transação remover o objeto de dados do cache imediatamente antes de o método `get` ser chamado, o resultado retornado será nulo.

### Considerações sobre Desempenho de Indexação

Um dos principais objetivos do recurso de indexação é melhorar o desempenho geral do `BackingMap`. Se a indexação não for utilizada corretamente, o desempenho do aplicativo poderá ficar comprometido. Considere os seguintes fatores antes de usar este retorno.

- **A quantidade de transações de gravação concorrentes:** O processamento de índices pode ocorrer todas as vezes que uma transação gravar dados em um `BackingMap`. O desempenho será afetado se muitas transações gravarem dados no mapa simultaneamente quando um aplicativo tentar operações de consulta ao índice.
- **O tamanho do conjunto de resultados que é retornado por uma operação de consulta:** Como o tamanho do conjunto de resultados aumenta, o desempenho da consulta cai. O desempenho tende a degradar quando o tamanho do conjunto de resultados é de 15% (ou mais) do `BackingMap`.

- **A quantidade de índices construídos sobre o mesmo BackingMap:** Cada índice consome os recursos do sistema. Conforme a quantidade dos índices construídos sobre o BackingMap aumenta, o desempenho diminui.

A função de indexação pode aumentar significativamente o desempenho do BackingMap. Os casos ideais ocorrem quando o BackingMap possui a maioria de operações de leitura, o conjunto de resultados da consulta é de uma porcentagem pequena das entradas do BackingMap, e somente poucos índices são construídos sobre o BackingMap.

## Planejando Diversas Topologias do Datacenter

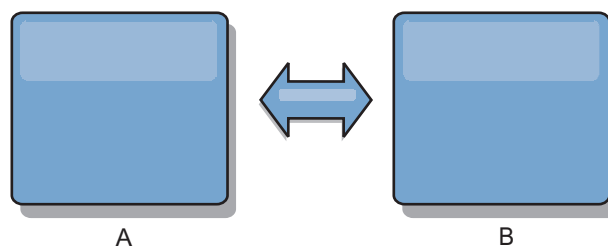
Ao usar a replicação assíncrona multimestre, duas ou mais grades de dados podem se tornar cópias exatas de uns dos outros. Cada grade de dados é hospedada em um domínio do serviço de catálogo independente, com seu próprio serviço de catálogo, servidores de contêiner e um nome exclusivo. Com a replicação assíncrona multimestre, é possível usar links para conectar uma coleção de domínios do serviço de catálogo. Os domínios do serviço de catálogo são então sincronizados usando a replicação sobre os links. É possível construir quase qualquer topologia por meio da definição de links entre os domínios de serviço de catálogo.

### Topologias de Replicação Multimestre

Há várias opções diferentes quando escolher a topologia para sua implementação que incorpora a replicação multimestre.

#### Links Conectando Domínios de Serviço de Catálogos

Uma infraestrutura de grade de dados de replicação é um gráfico conectado dos domínios do serviço de catálogo com links bidirecionais entre eles. Com um link, dois domínios de serviço de catálogo podem comunicar as mudanças de dados. Por exemplo, a topologia mais simples é um par de domínios do serviço de catálogo com um único link entre eles. Os domínios do serviço de catálogo são denominados em ordem alfabética: A, B, C e assim por diante, a partir da esquerda. Um link pode cruzar uma rede de longa distância (WAN), expandindo-se para grandes distâncias. Mesmo se o link for interrompido, os dados ainda poderão ser alterados em qualquer um dos domínios de serviço de catálogo. A topologia reconcilia as mudanças quando o link reconecta os domínios do serviço de catálogo. Os links tentarão automaticamente reconectar se a conexão com a rede for interrompida.

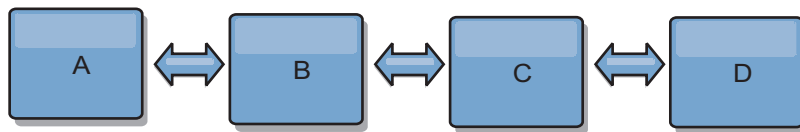


Após configurar os links, o eXtreme Scale primeiro tentará tornar idêntico cada domínio do serviço de catálogo. Em seguida, o eXtreme Scale tenta manter as condições idênticas conforme as mudanças ocorrerem em qualquer domínio de serviço de catálogo. O objetivo é que cada domínio de serviço de catálogo seja um espelho exato de qualquer outro domínio de serviço de catálogo conectado pelos links. Os links de replicação entre os domínios de serviço de catálogo ajudam a

assegurar que quaisquer mudanças feitas em um domínio sejam copiadas para os outros domínios.

### Topologias em Linha

Embora esta seja uma implementação muito simples, uma topologia em linha demonstra algumas qualidades dos links. Primeiro, um domínio de serviço de catálogo não precisa estar conectado diretamente a outro domínio de serviço de catálogo para receber as mudanças. O Domínio B pega as mudanças de Domínio A. O Domínio C recebe mudanças do Domínio A por meio do Domínio B, que se conecta os Domínios A e C. Da mesma forma, o Domínio D recebe as mudanças dos outros domínios por meio do Domínio C. Esta habilidade envia o carregamento das mudanças de distribuição para longe da origem das mudanças.



Observe que, se o Domínio C falhar, as seguintes ações ocorrerão:

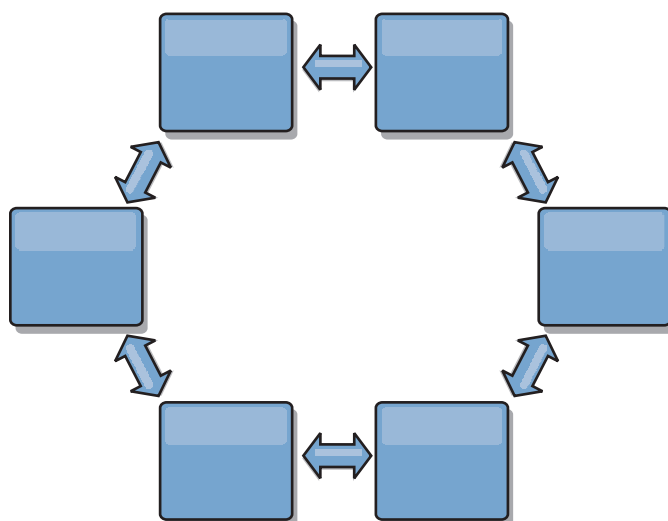
1. O Domínio D pode ficar órfão até o Domínio C ser reiniciado
2. O Domínio C pode se sincronizar com o Domínio B, que é uma cópia do Domínio A
3. O Domínio D pode usar o Domínio C para se sincronizar com as mudanças nos Domínios A e B. Essas mudanças inicialmente ocorreram enquanto o Domínio D estava órfão (enquanto o Domínio C estava inativo).

Por fim, os Domínios A, B, C e D podem se tornar todos idênticos entre si novamente.

### Topologias em Anel

As topologias em anel são um exemplo de uma topologia mais resiliente. Quando um domínio do serviço de catálogo ou um único link falhar, os domínios do serviço de catálogo sobreviventes ainda podem obter as mudanças. Os domínios do serviço de catálogo se deslocam ao redor do anel, longe da falha. Cada domínio de serviço de catálogo tem no máximo dois links para outros domínios de serviço de catálogo, independente do tamanho da topologia em anel. A latência para propagar as mudanças pode ser grande. As mudanças a partir de um domínio de serviço de catálogo particular podem precisar percorrer vários links antes que todos os domínios de serviço de catálogo possuam as mudanças. Uma topologia em linha tem a mesma característica.

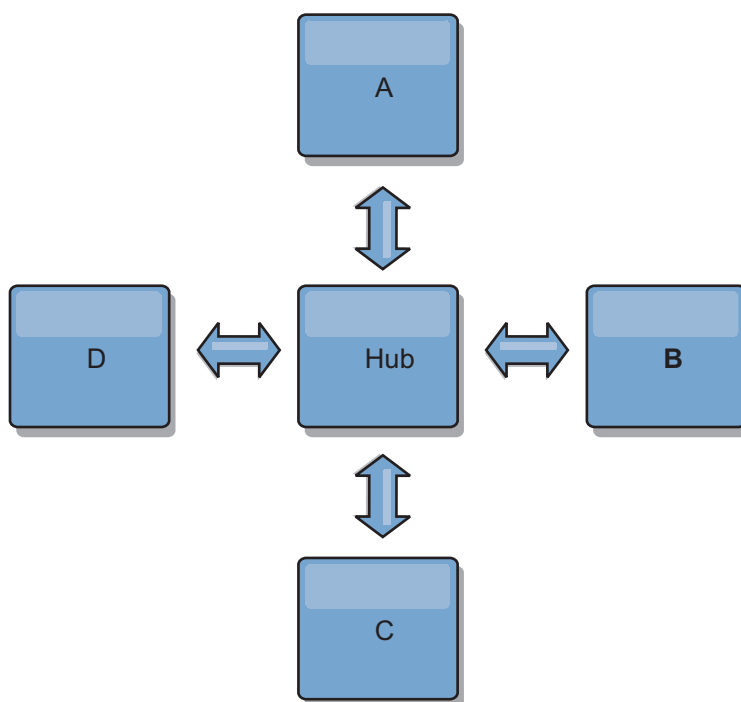




Também é possível implementar uma topologia em anel mais sofisticada, com um domínio de serviço de catálogo raiz no centro do anel. O domínio de serviço de catálogo raiz funciona como o ponto central da reconciliação. Os outros domínios de serviço de catálogo atuam como pontos remotos de reconciliação para que as mudanças ocorram no domínio de serviço de catálogo raiz. O domínio de serviço de catálogo raiz pode arbitrar mudanças entre os domínios de serviço de catálogo. Se uma topologia em anel contiver mais de um anel ao redor de um domínio de serviço de catálogo raiz, o domínio poderá arbitrar apenas as mudanças entre o anel mais interno. No entanto, os resultados da arbitragem se propagam por todos os domínios de serviço de catálogo nos outros anéis.

### **Topologias Hub-and-Spoke**

Com uma topologia hub-and-spoke, as mudanças se deslocam por um domínio de serviço de catálogo hub. Como o hub é apenas o domínio de serviço de catálogo intermediário que é especificado, as topologias hub e spoke possuem baixa latência. O domínio hub é conectado a cada domínio de spoke por meio de um link. O hub distribui as mudanças entre os domínios de serviço de catálogo. O hub atua como um ponto de reconciliação de colisões. Em um ambiente com uma alta taxa de atualização, o hub pode precisar executar em mais hardware do que os spokes devem permanecer sincronizados. O WebSphere eXtreme Scale é projetado para escalar linearmente, o que significa que é possível aumentar o hub, conforme necessário, sem dificuldade. No entanto, se o hub falhar, as mudanças não são distribuídas até ele ser reiniciado. As mudanças nos domínios de serviço de catálogo spoke serão distribuídas depois que o hub for reconectado.



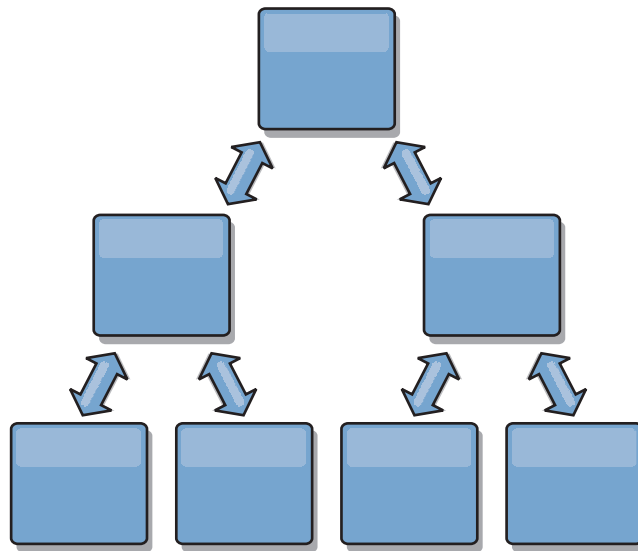
Também é possível usar uma estratégia com clientes totalmente replicados, uma variação de topologia que usa um par de servidores eXtreme Scale sendo executados como um hub. Cada cliente cria uma grade de dados autocontida de contêiner único com um catálogo na JVM do cliente. Um cliente usa a grade de dados para se conectar com o catálogo do hub. Esta conexão faz com que o cliente seja sincronizado com o hub assim que o cliente estabelecer uma conexão com o hub.

Todas as mudanças feitas pelo cliente são locais para o cliente e são replicadas de forma assíncrona para o hub. O hub age como um domínio de arbitragem, distribuindo mudanças para todos os clientes conectados. A topologia de clientes totalmente replicados fornece um cache L2 confiável para um mapeador relacional de objeto, como OpenJPA. As mudanças serão distribuídas rapidamente entre as JVMs de cliente por meio do hub. Se o tamanho do cache puder estar contido no espaço de heap disponível, a topologia será uma arquitetura confiável para este estilo de L2.

Use várias partições para escalar o domínio do hub em várias JVMs, se necessário. Como todos os dados ainda devem se ajustar em uma única JVM de cliente, diversas partições aumentam a capacidade do hub para distribuir e arbitrar as mudanças. No entanto, ter diversas partições não altera a capacidade de um único domínio.

### Topologias em Árvore

Também é possível usar uma árvore direcionada acíclica. Uma árvore acíclica não tem ciclos ou loops e uma configuração direcionada limita a criação de links apenas entre pais e filhos. Esta configuração pode ser útil para topologias que possuem muitos domínios de serviço de catálogo e não é viável ter um hub central conectado a cada spoke possível. Este tipo de topologia também pode ser útil quando os domínios de serviço de catálogo filho tiverem que ser incluídos sem atualizar o domínio de serviço de catálogo raiz.



Uma topologia em árvore ainda pode ter um ponto central de reconciliação no domínio de serviço de catálogo raiz. O segundo nível pode ainda funcionar como um ponto de reconciliação remoto para que as mudanças ocorram no domínio de serviço de catálogo abaixo deles. O domínio do serviço de catálogo raiz pode arbitrar as mudanças entre os domínios de serviço de catálogo apenas no segundo nível. Também é possível usar  $N$  árvores, cada uma delas possuindo  $N$  filhos em cada nível. Cada domínio de serviço de catálogo se conecta com  $n$  links .

### **Clientes Totalmente Replicados**

Essa variação de topologia envolve um par de servidores do eXtreme Scale sendo executados como um hub. Cada cliente cria uma grade de dados autocontida de contêiner único com um catálogo na JVM do cliente. Um cliente usa a grade de dados para se conectar com o catálogo do hub, fazendo com que o cliente seja sincronizado com o hub assim que estabelecer uma conexão com o hub.

Todas as mudanças feitas pelo cliente são locais para o cliente e são replicadas de forma assíncrona para o hub. O hub age como um domínio de arbitragem, distribuindo mudanças para todos os clientes conectados. A topologia de clientes totalmente replicada fornece um bom cache L2 para um mapeador relacional de objeto, como OpenJPA. As mudanças serão distribuídas rapidamente entre as JVMs de cliente por meio do hub. Contudo que o tamanho do cache possa estar contido no espaço de heap disponível dos clientes, esta topologia será uma boa arquitetura para este estilo de L2.

Use várias partições para escalar o domínio do hub em várias JVMs, se necessário. Como todos os dados ainda devem se ajustar em uma única JVM de cliente, usar várias partições aumenta a capacidade do hub para distribuir e arbitrar mudanças, mas não altera a capacidade de um único domínio.

### **Considerações de Configuração para Topologias Multimestre**

Considere os seguintes problemas ao decidir se e como usar as topologias de replicação multimestre.

- **Requisitos do conjunto de mapa**

Os conjuntos de mapa devem ter as seguintes características para replicar as mudanças nos links de domínio do serviço de catálogo:

- O nome do ObjectGrid e o nome do conjunto de mapas dentro de um domínio do serviço de catálogo devem corresponder ao nome do ObjectGrid e ao nome do conjunto de mapas de outros domínios de serviço de catálogo. Por exemplo, o ObjectGrid "og1" e o conjunto de mapas "ms1" devem ser configurados no domínio do serviço de catálogo A e no domínio do serviço de catálogo B para replicar os dados no conjunto de mapas entre os domínios do serviço de catálogo.
- Ser uma grade de dados FIXED\_PARTITION. As grades de dados PER\_CONTAINER não podem ser replicadas.
- Ter o mesmo número de partições em cada domínio do serviço de catálogo. O conjunto de mapa pode ou não ter o mesmo número e tipos de réplicas.
- Ter os mesmos tipos de dados sendo replicados em cada domínio do serviço de catálogo.
- Contém os mesmos mapas e modelos de mapas dinâmicos em cada domínio do serviço de catálogo.
- Não usar o gerenciador de entidades. Um conjunto de mapas contendo um mapa de entidade não é replicado entre os domínios do serviço de catálogo.
- Não usar o suporte de armazenamento em cache write-behind. Um conjunto de mapas contendo um mapa configurado com o suporte write-behind não é replicado entre os domínios de serviço de catálogo.

Quaisquer conjuntos de mapas com as características anteriores começam a serem replicados depois que os domínios do serviço de catálogo na topologia forem iniciados.

- **Carregadores de classe com diversos domínios do serviço de catálogo**

Os domínios do serviço de catálogo devem ter acesso a todas as classes que são usadas como chaves e valores. Todas as dependências devem ser refletidas em todos os caminhos da classe para as Java virtual machine do contêiner da grade para todos os domínios. Se um plug-in CollisionArbiter recuperar o valor para uma entrada de cache, as classes para os valores deverão estar presentes para o domínio que está iniciando o mecanismo de resolução de conflitos.

## **Considerações Sobre o Carregador em uma Topologia Multimestre**

Quando estiver usando os carregadores em uma topologia multimestre, você deve considerar a possibilidade de colisão e desafios de manutenção das informações de revisão. A grade de dados mantém as informações de revisão sobre os itens nela para que colisões possam ser detectadas quando outros shards primários na configuração gravarem entradas na grade de dados. Quando as entradas são incluídas a partir de um carregador, essas informações de revisão não são incluídas e a entrada assume uma nova revisão. Como a revisão da entrada parece ser uma nova inserção, uma colisão false poderá ocorrer se outro shard primário também alterar esse estado ou obtiver as mesmas informações a partir de um carregador.

As mudanças na replicação chamam o método get no carregador com uma lista das chaves que ainda não estão na grade de dados, porém serão alteradas durante a transação da replicação. Quando a replicação ocorre, essas entradas são entradas de colisão. Quando as colisões são definidas e a revisão é aplicada, uma atualização em lote é chamada no carregador para aplicar as mudanças no banco de dados. Todos os mapas que foram alterados na janela de revisão são atualizados na mesma transação.

## **Desafio do Pré-Carregamento**

Considere uma topologia de dois datacenters, com o datacenter A e o datacenter B. Ambos os datacenters possuem bancos de dados independente, mas apenas o datacenter A possui uma grade de dados em execução. Quando você estabelece um link entre os datacenters por uma configuração multimestre, as grades de dados no datacenter A começam a enviar dados para as novas grades de dados do datacenter B, causando uma colisão com cada entrada. Outro maior problema que ocorre é com os dados que estão no banco de dados do datacenter B, mas não no banco de dados no datacenter A. Essas linhas não são preenchidas e definidas, resultando em inconsistências que não são resolvidas.

## **Solução para o Desafio de Pré-Carregamento**

Como os dados que residem apenas no banco de dados não podem ter revisões, a grade de dados sempre deve ser pré-carregada completamente a partir do banco de dados local antes de estabelecer o link multimestre. Em seguida, ambas as grades de dados podem revisar e definir os dados, eventualmente chegando a um estado consistente.

## **Desafio de Cache Disperso**

Com um cache disperso, o primeiro aplicativo tenta localizar os dados na grade de dados. Se os dados não estiverem na grade de dados, eles serão procurados no banco de dados usando o carregador. As entradas são despejadas da grade de dados periodicamente para manter um tamanho de cache pequeno.

Este tipo de cache pode ser problemático em um cenário de configuração multimestre porque as entradas dentro da grade de dados possuem metadados de revisão que ajudam a detectar quando colisões ocorrem e qual lado fez as mudanças. Quando os links entre os datacenters não estiverem funcionando, um datacenter pode atualizar uma entrada e depois, eventualmente, atualizar o banco de dados e invalidar a entrada na grade de dados. Quando o link é recuperado, os datacenters tentam sincronizar as revisões entre si. No entanto, como o banco de dados foi atualizado e a entrada da grade de dados foi invalidada, a mudança é perdida a partir da perspectiva do datacenter que ficou inativo. Como resultado, os dois lados da grade de dados estão fora de sincronização e não estão consistentes.

## **Solução para o Desafio de Cache Disperso**

### **Topologia Hub e Spoke:**

É possível executar o carregador apenas no hub de uma topologia de hub e spoke, mantendo a consistência dos dados, enquanto a escala da grade de dados é ajustada. No entanto, se você estiver considerando essa implementação, observe que os carregadores podem permitir que a grade de dados seja parcialmente carregada, significando que um evictor foi configurado. Se o spokes de sua configuração forem caches dispersos, mas não possuírem nenhum carregador, quaisquer perdas de cache não terão como recuperar dados do banco de dados. Devido a esta restrição, você deve usar uma topologia de cache totalmente preenchida com uma configuração de hub e spoke.

## **Invalidações e Despejo**

Uma invalidação cria inconsistência entre a grade de dados e o banco de dados. Os dados podem ser removidos da grade de dados, seja de modo programático ou

com o despejo. Ao desenvolver seu aplicativo, você deve estar ciente de que a manipulação de revisão não replica mudanças que são invalidadas, resultando em inconsistências entre os shards primários.

Os eventos de invalidação não são mudanças de estado de cache e não resultam em replicação. Todos os evictors configurados são executados independentemente de outros evictors na configuração. Por exemplo, você pode ter um evictor configurado para um limite de memória em um domínio do serviço de catálogo, e um tipo diferente de evictor menos agressivo no outro domínio do serviço de catálogo vinculado. Quando as entradas da grade dedados são removidas devido à política de limite de memória, as entradas no outro domínio do serviço de catálogo não são afetadas.

### **Atualizações do banco de dados e invalidação da grade de dados**

Problemas ocorrem quando o banco de dados é atualizado diretamente no plano de fundo ao chamar a invalidação na grade de dados para as entradas atualizadas em uma configuração multimestre. Esse problema ocorre porque a grade de dados não pode replicar a mudança para os outros shards primários até que algum tipo de acesso ao cache mova a entrada para a grade de dados.

### **Diversos Gravadores para um Único Banco de Dados Lógico**

Quando um banco de dados único é usado com diversos shards primários conectados por meio de um carregador, isso pode resultar em conflitos transacionais. Sua implementação do carregador deve manipular especialmente esses tipos de cenários.

### **Espelhando Dados Usando Replicação Multimestre**

É possível configurar bancos de dados independentes que estão conectados a domínios do serviço de catálogo independentes. Nessa configuração, o carregador pode enviar mudanças de um datacenter para o outro.

### **Considerações de Design para Replicação Multimestre**

Ao implementar a replicação multimestre, você deve considerar aspectos de design, como arbitragem, vinculação e desempenho.

### **Considerações sobre Arbitragem no Design de Topologia**

Poderão ocorrer colisões de mudanças se os mesmos registros puderem ser alterados simultaneamente em dois locais. Configure cada domínio de serviço de catálogo para ter aproximadamente a mesma quantia de processador, memória e recursos de rede. Observe que os domínios de serviço de catálogo que executam a manipulação da colisão de mudanças (arbitragem) usam mais recursos que outros domínios de serviço de catálogo. As colisões são detectadas automaticamente. Elas são manipuladas com um desses dois mecanismos:

- **Árbitro de conflito padrão:** O protocolo padrão deve usar as mudanças a partir do domínio de serviço de catálogo mais baixo denominado de maneira lexical. Por exemplo, se os domínios de serviço de catálogo A e B gerarem um conflito para um registro, a mudança no domínio de serviço de catálogo B será ignorada. O domínio de serviço de catálogo A mantém sua versão e o registro no domínio de serviço de catálogo B é alterado para corresponder ao registro do domínio de serviço de catálogo A. Esse comportamento também se aplica aos aplicativos nos quais os usuários ou as sessões são normalmente ligados ou têm afinidade com uma das grades de dados.

- **Árbitro de colisão customizado:** Os aplicativos podem fornecer um árbitro customizado. Quando um domínio do serviço de catálogo detecta uma colisão, ele inicia o árbitro. Para obter informações sobre como desenvolver um árbitro útil customizado, consulte Desenvolvendo Árbitros Customizados para a Replicação Multimestre.

Para topologias nas quais as colisões são possíveis, considere implementar uma topologia hub-and-spoke ou uma topologia em árvore. Essas duas topologias tendem a evitar colisões constantes, o que pode acontecer nos seguintes cenários:

1. Diversos domínios de serviço de catálogo experimentam uma colisão.
2. Cada domínio de serviço de catálogo manipula a colisão localmente, produzindo revisões.
3. As revisões colidem, resultando em revisões de revisões

Para evitar colisões, escolha um domínio de serviço de catálogo específico, chamado de *domínio de serviço de catálogo de arbitragem* como o árbitro de colisão para um subconjunto de domínios de serviço de catálogo. Por exemplo, uma topologia hub-and-spoke pode usar o hub como o manipulador de colisão. O manipulador de colisão spoke ignora quaisquer colisões que forem detectadas pelos domínios de serviço de catálogo spoke. O domínio de serviço de catálogo do hub cria revisões, evitando revisões de colisão inesperadas. O domínio de serviço de catálogo designado para manipular colisões deve ser vinculado a todos os domínios os quais ele é responsável por manipular as colisões. Em uma topologia em árvore, os domínios pais internos manipulam colisões para seus filhos imediatos. Em contraste, se usar uma topologia em anel, não será possível designar um domínio de serviço de catálogo no anel como o árbitro.

A tabela a seguir resume as abordagens de arbitragem que são mais compatíveis com várias topologias.

*Tabela 1. Abordagens de Arbitragem.* Esta tabela define se a arbitragem de aplicativo é compatível com várias tecnologias.

Topologia	Arbitragem do Aplicativo?	Notes
Uma linha de dois domínios de serviço de catálogo	Sim	Escolha um domínio do serviço de catálogo como o árbitro.
Uma linha de três domínios de serviço de catálogo	Sim	O domínio de serviço de catálogo médio deve ser o árbitro. Considere o domínio de serviço de catálogo médio como sendo o hub em uma topologia hub-and-spoke simples.
Uma linha de mais de três domínios de serviço de catálogo	Não	A arbitragem de aplicativo não é suportada.
Um hub com N spokes	Sim	O hub com links para todos os spokes deve ser o domínio de serviço de catálogo de arbitragem.
Um anel de N domínios de serviço de catálogo	Não	A arbitragem de aplicativo não é suportada.
Uma árvore acíclica, direcionada (árvore n-ary)	Sim	Todos os nós-raiz devem classificar apenas seus descendentes diretos.



## Considerações sobre Links no Design de Topologia

De forma ideal, uma topologia inclui um número mínimo de links enquanto otimiza trade-offs entre latência de mudança, tolerância a falhas e características de desempenho.

### • Latência de mudança

A latência de mudança é determinada pelo número de domínios de serviço de catálogo intermediários onde uma mudança deve passar antes de chegar a um domínio do serviço de catálogo específico.

Uma topologia tem a melhor latência de mudança quando ele elimina os domínios de serviço de catálogo intermediários ao vincular cada domínio de serviço de catálogo a outro domínio. No entanto, um domínio de serviço de catálogo deve executar o trabalho de replicação proporcionalmente ao seu número de links. Para topologias grandes, o número de links absoluto a ser definido pode causar uma carga administrativa.

A velocidade com que uma mudança é copiada para outros domínios do serviço de catálogo depende de fatores adicionais, como:

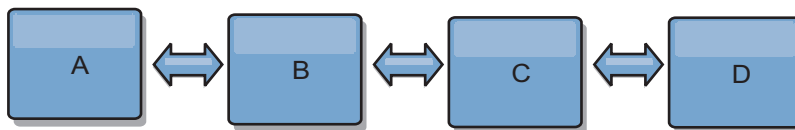
- Processador e largura da banda da rede no domínio de serviço de catálogo de origem
- O número de domínios de serviço de catálogo intermediários e de links entre o domínio de serviço de catálogo de origem e de destino
- Os recursos de processador e de rede disponíveis para os domínios de serviço de catálogo de origem, de destino e intermediários

### • Tolerância a falhas

A tolerância a falhas é determinada pela quantia de caminhos que existem entre dois domínios de serviço de catálogo para a replicação de mudança.

Se houver apenas um link entre um determinado par de domínios de serviço de catálogo, uma falha de link não permitirá a propagação das mudanças. Da mesma forma, as mudanças não serão propagadas entre os domínios de serviço de catálogo se ocorrer uma falha de link em qualquer um dos domínios intermediários. Sua topologia pode ter um único link a partir de um domínio de serviço de catálogo para outro, de modo que o link passe pelos domínios intermediários. Caso positivo, as mudanças não serão propagadas se qualquer um dos domínios de serviço de catálogo intermediários estiver inativo.

Considere a topologia em linha com quatro domínios de serviço de catálogo, A, B, C e D:

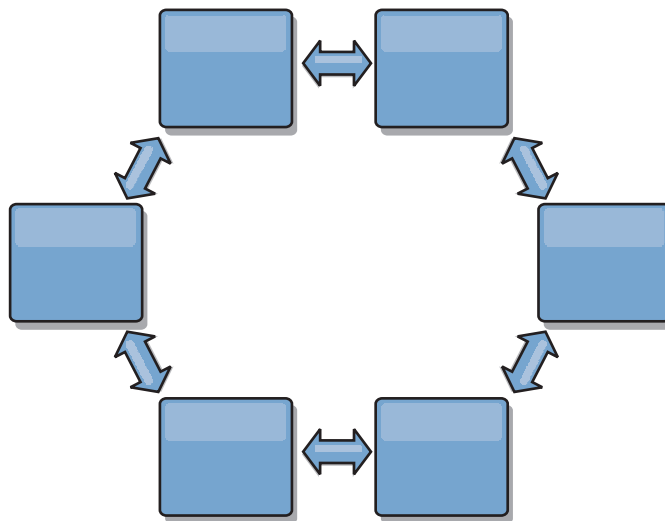


Se qualquer uma dessas condições for mantida, o Domínio D não verá nenhuma mudança a partir do A:

- O Domínio A está ativo e o B está inativo
- Os Domínios A e B estão ativos e C está inativo
- O link entre A e B está inativo
- O link entre B e C está inativo
- O link entre C e D está inativo

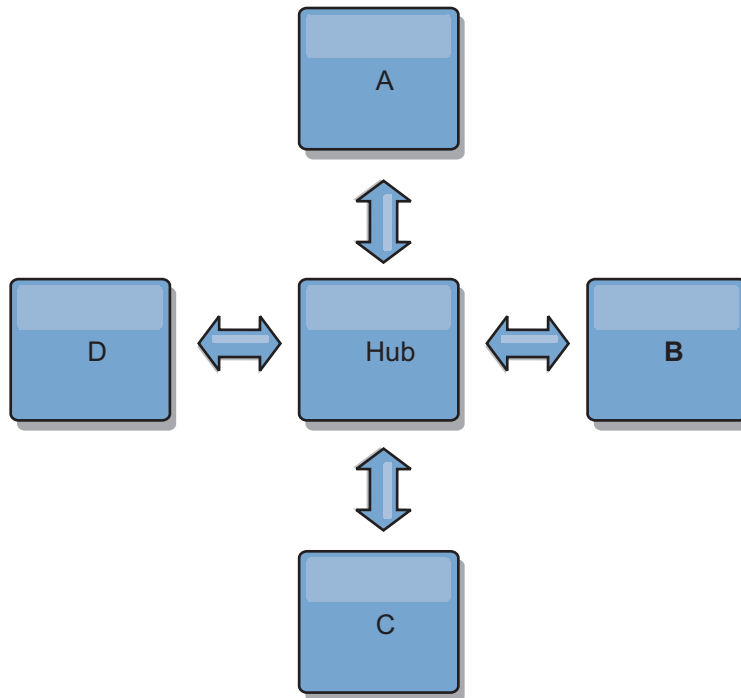
Em contraste, com uma topologia em anel, cada domínio de serviço de catálogo pode receber mudanças a partir de qualquer direção.





Por exemplo, se um determinado serviço de catálogo em sua topologia em anel estiver inativo, os dois domínios adjacentes ainda poderão obter as mudanças diretamente entre si.

Todas as mudanças são propagadas por meio do hub. Assim, em oposição à topologia de linha e em anel, o design hub-and-spoke será suscetível à interrupção se o hub falhar.



Um domínio de serviço de catálogo único é resiliente a uma determinada perda de quantidade de serviço. No entanto, falhas maiores, como interrupções ou perda de links em uma rede maior entre os datacenters físicos pode interromper qualquer um dos domínios de serviço de catálogo.

- **Vínculo e desempenho**

O número de links definido em um domínio de serviço de catálogo afeta o desempenho. Mais links usam mais recursos e o desempenho de replicação pode diminuir como resultado. A capacidade de recuperar mudanças para um domínio A por meio de outros domínios isenta efetivamente o domínio A da replicação de suas transações em todo lugar. O carregamento de distribuição de

mudança em um domínio é limitado pelo número de links que ele usa, e não pela quantidade de domínios presentes na topologia. Esta propriedade de carregamento fornece escalabilidade, de modo que os domínios na topologia possam compartilhar a carga de distribuição de mudança.

Um domínio de serviço de catálogo pode recuperar as mudanças indiretamente por meio de outros domínios de serviço de catálogo. Considere uma topologia em linha com cinco domínios de serviço de catálogo.

A <=> B <=> C <=> D <=> E

- A pega mudanças de B, C, D e E por meio de B
- B pega mudanças de A e C diretamente e mudanças de D e E por meio de C
- C pega mudanças de B e D diretamente e mudanças de A por meio de B e E por meio de D
- D pega mudanças de C e E diretamente e mudanças de A e B por meio de C
- E pega mudanças de D diretamente e mudanças de A, B e C por meio de D

O carregamento de distribuição nos domínios de serviço de catálogo A e E é o mais baixo, porque cada um deles possui um link apenas para um domínio de serviço de catálogo único. Cada um dos domínios B, C e D possui um link para dois domínios. Assim, o carregamento de distribuição nos domínios B, C e D é o dobro do carregamento nos domínios A e E. A carga de trabalho depende do número de links em cada domínio, e não do número geral de domínios na topologia. Assim, a distribuição dos carregamentos descrita permaneceria constante, mesmo se a linha contivesse 1.000 domínios.

## Considerações de Desempenho de Replicação Multimestre

Leve em consideração as seguintes limitações quando usar topologias de replicação multimestre:

- **Alterar ajuste de distribuição**, conforme discutido na seção anterior.
- **Desempenho do link de replicação** O WebSphere eXtreme Scale cria um único soquete TCP/IP entre qualquer par de JVMs. Todo o tráfego entre as JVMs ocorre por meio deste soquete único, incluindo o tráfego de replicação multimestre. Os domínios do serviço de catálogo são hospedados em pelo menos  $n$  JVMs de contêiner, fornecendo pelo menos  $n$  links TCP para os domínios de serviço de catálogo equivalentes. Assim, os domínios de serviço de catálogo com números maiores de contêineres têm níveis maiores de desempenho de replicação. Mais contêineres requerem mais recursos de processador e de rede.
- **Ajuste da janela deslizante TCP e o RFC 1323** O suporte do RFC 1323 em ambas as extremidades de um link geram mais dados para um roundtrip. Este suporte resulta em maior rendimento, expandindo a capacidade da janela em um fator de aproximadamente 16.000.

A chamada desses soquetes TCP usa um mecanismo de janela deslizante para controlar o fluxo de dados em massa. Este mecanismo geralmente limita o soquete para 64 KB para um intervalo de roundtrip. Se o intervalo de roundtrip for de 100 ms, a largura de banda será limitada a 640 KB/segundo sem ajuste adicional. Usar totalmente a largura de banda disponível em um link pode exigir um ajuste específico para um sistema operacional. A maioria dos sistemas operacionais inclui parâmetros de ajuste, inclusive opções RFC 1323, para aprimorar o rendimento por meio dos links de alta latência.

Vários fatores podem afetar o desempenho de replicação:

- A velocidade com que o eXtreme Scale recupera as mudanças.

- A velocidade com que o eXtreme Scale pode atender às solicitações de replicação de recuperação.
- A capacidade da janela de deslizamento.
- Com o ajuste do buffer de rede em ambos os lados de um link, o eXtreme Scale recupera as mudanças por meio do soquete de modo eficiente.
- **Serialização de Objeto** Todos os dados devem ser serializáveis. Se um domínio de serviço de catálogo não estiver usando COPY\_TO\_BYTES, o domínio de serviço de catálogo deverá usar a serialização Java ou ObjectTransformers para otimizar o desempenho da serialização.
- **Compactação** O WebSphere eXtreme Scale compacta todos os dados enviados entre os domínios de serviço de catálogo por padrão. Desativar a compactação não está disponível atualmente.
- **Ajuste de memória** O uso de memória para uma topologia de replicação multimestre é altamente independente do número de domínios de serviço de catálogo na topologia.

A replicação multimaster inclui uma quantidade fixa de processamento por entrada de Mapa para manipular a versão. Cada contêiner também controla uma quantidade fixa de dados em cada domínio de serviço de catálogo na topologia. Uma topologia com dois domínios de serviço de catálogo usa aproximadamente a mesma memória que uma topologia com 50 domínios de serviço de catálogo. O WebSphere eXtreme Scale não usa logs de reprodução ou filas semelhantes em sua implementação. Assim, não há nenhuma estrutura de recuperação pronta caso um link de replicação esteja indisponível por um período prolongado e reinícios posteriores.

---

## Interoperabilidade com Outros Produtos WebSphere

Você pode integrar o WebSphere eXtreme Scale com outros produtos servidores como WebSphere Application Server e WebSphere Application Server Community Edition.

### WebSphere Application Server

É possível integrar o WebSphere Application Server em vários aspectos de sua configuração do WebSphere eXtreme Scale. É possível implementar os aplicativos da grade de dados e usar o WebSphere Application Server para hospedar os servidores de contêiner e de catálogos. Também é possível usar a segurança do WebSphere Application Server no ambiente do WebSphere eXtreme Scale .

### Portal WebSphere

É possível persistir as sessões HTTP a partir do WebSphere Portal em uma grade de dados no WebSphere eXtreme Scale.

### WebSphere Application Server Community Edition

O WebSphere Application Server Community Edition pode compartilhar estado de sessão, mas não de uma maneira eficiente e escalável. O WebSphere eXtreme Scale fornece uma camada de persistência distribuída e de alto desempenho, que pode ser utilizada para replicar o estado, mas não se integra prontamente a qualquer servidor de aplicativos fora do WebSphere Application Server. Estes dois produtos podem ser integrados para fornecer uma solução de gerenciamento de sessões escalável.

## WebSphere Real Time

Com suporte para o WebSphere Real Time, a oferta Java real líder de mercado, o WebSphere eXtreme Scale permite que os aplicativos do Extreme Transaction Processing (XTP) tenham tempos de resposta mais consistentes e previsíveis.

---

## Planejando para Instalação

Antes de instalar o produto, você deve considerar em seu ambiente.

### Requisitos de Hardware e Software

Pesquisar uma visão geral de hardware e de requisitos do sistema operacional. Embora não seja necessário usar um nível específico de hardware ou sistema operacional para o WebSphere eXtreme Scale, as opções de hardware e software formalmente suportadas estão disponíveis na página Requisitos do Sistema do site de suporte do produto. Se existir um conflito entre o centro de informações e a página Requisitos do Sistema, as informações no website terão precedência. As informações de pré-requisito no centro de informações são fornecidas apenas como uma conveniência.

Consulte a Página de Requisitos do Sistema para obter o conjunto de oficial de requisitos de hardware e software.

Não é necessário instalar e implementar o eXtreme Scale em um nível específico do sistema operacional. Cada instalação do Java Platform, Standard Edition (Java SE) e do Java Platform, Enterprise Edition (Java EE) requer níveis ou correções de sistema operacional diferentes.

É possível instalar e implementar o produto nos ambientes do Java EE e do Java SE. Também é possível incluir em pacote configurável o componente do cliente com os aplicativos Java EE diretamente sem integrar com o WebSphere Application Server. O WebSphere eXtreme Scale suporta o Java SE 5 e posterior e o WebSphere Application Server Versão 6.1 e posterior.

### Requisitos de Hardware

O WebSphere eXtreme Scale não requer um nível específico de hardware. Os requisitos de hardware dependem do hardware suportado para a instalação do Java Platform, Standard Edition que é utilizado para executar o WebSphere eXtreme Scale. Se você estiver usando o eXtreme Scale com o WebSphere Application Server ou outra implementação do Java Platform, Enterprise Edition, os requisitos de hardware dessas plataformas são suficientes para o WebSphere eXtreme Scale.

### Requisitos do Sistema Operacional

- **Sem o console da web**

O eXtreme Scale não requer um nível de sistema operacional específico. Cada implementação de Java SE e Java EE necessita de diferentes níveis de sistema operacional ou correções para problemas que são descobertos durante o teste da implementação de Java. Os níveis que as implementações necessitam são suficientes para o eXtreme Scale.

- **Com o console da web**

Os seguintes requisitos se aplicarão a cada sistema operacional se usar o console:

- Linux: JVM de 32 bits ou 64 bits

- Linux PPC: Somente JVM de 32 bits
- Windows: Somente JVM de 32 bits
- AIX: Somente JVM de 32 bits

## Requisitos do Navegador da Web

O console da web suporta os seguintes navegadores da Web:

- Mozilla Firefox, versão 3.5.x e posterior
- Mozilla Firefox, versão 3.6.x e posterior
- Microsoft Internet Explorer, versão 7 ou 8

## Requisitos do WebSphere Application Server

- WebSphere Application Server Versão 6.1.0.39 ou posterior
- WebSphere Application Server Versão 7.0.0.19 ou posterior
- WebSphere Application Server Versão 8.0.0.1 ou posterior

Consulte as Correções recomendadas para o WebSphere Application Server para obter informações adicionais.

## Outros Requisitos do Servidor de Aplicativos

Outras implementações de Java EE podem usar o tempo de execução do eXtreme Scale como uma instância local ou como um cliente para servidores eXtreme Scale. Para implementar o Java SE, você deve usar a Versão 5 ou posterior.

## Considerações sobre o Java SE

O WebSphere eXtreme Scale requer o Java SE 5 ou superior. Em geral, as versões mais recentes do Java SE possuem melhor funcionalidade e desempenho.

### Versões Suportadas

O WebSphere eXtreme Scale pode ser usado com o Java SE 5 ou superior. A versão que deve ser usada deve ser atualmente suportada pelo fornecedor do Java Runtime Environment (JRE).

Um JRE totalmente suportado é instalado como parte das instalações do WebSphere eXtreme Scale e do WebSphere eXtreme Scale Client no diretório *wxs\_install\_root/java* e está disponível para ser usado por ambos clientes e servidores. Se estiver instalando o WebSphere eXtreme Scale no WebSphere Application Server, o JRE incluído na instalação do WebSphere Application Server poderá ser usado.

O WebSphere eXtreme Scale aproveita as funcionalidades do Java Development Kit (JDK) Versão 5 ou posterior conforme é disponibilizado. Geralmente, as versões mais recentes do Java Development Kit (JDK) e do Java SE têm melhor desempenho e funcionalidade.

Consulte Software Suportado para obter mais informações.

## Recursos do WebSphere eXtreme Scale Dependentes de Java

Tabela 2. Recursos que Requerem o Java SE 5 ou Java SE 6.

O WebSphere eXtreme Scale usa a funcionalidade que é introduzida no Java SE 5 ou no Java SE 6 para fornecer os seguintes recursos do produto.

Recurso	Suportado no Java SE 5 e superior	Suportado no Java SE 6 e superior
Anotações EntityManager API (Opcional: Também é possível usar os arquivos XML)	X	X
Java Persistence API (JPA): Carregador JPA, carregador de cliente do JPA e atualizador baseado em tempo do JPA	X	X
Despejo com base em memória (usa MemoryPoolMXBean)	X	X
Agentes de instrumentação: <ul style="list-style-type: none"><li>• wxsizeagent.jar: Aumenta a exatidão das métricas de mapa de bytes usados.</li><li>• ogagent.jar: Aumenta o desempenho de entidades de acesso a campo.</li></ul>	X	X
Console da web para monitoramento		X

## Considerações sobre o Java EE

Conforme você se prepara para integrar o WebSphere eXtreme Scale em um ambiente Java Platform, Enterprise Edition, considere determinados itens, como versões, opções de configuração, requisitos e limitações e implementação e gerenciamento de aplicativo.

### Executando Aplicativos eXtreme Scale em um Ambiente Java EE

Um aplicativo Java EE pode se conectar a um aplicativo eXtreme Scale remoto. Além disso, o ambiente do WebSphere Application Server suporta o início de um servidor eXtreme Scale conforme um aplicativo é iniciado no servidor de aplicativos.

Se você utilizar um arquivo XML para criar uma instância de ObjectGrid, e o arquivo XML estiver no módulo do arquivo enterprise archive (EAR), acesse o arquivo utilizando o método `getClass().getClassLoader().getResource("META-INF/objGrid.xml")` para obter um objeto URL para utilizar para criar uma instância de ObjectGrid. Substitua o nome do arquivo XML que você está utilizando na chamada de método.

É possível utilizar beans de inicialização para um aplicativo para autoinicializar uma instância do ObjectGrid quando o aplicativo for iniciado e para destruir a instância quando o aplicativo for parado. Um bean de inicialização é um bean de

sessão stateless com um local remoto com `com.ibm.websphere.startupservice.AppStartUpHome` e uma interface remota com `com.ibm.websphere.startupservice.AppStartUp`. A interface remota possui dois métodos: o método `start` e o método `stop`. Utilize o método `start` para autoinicializar a instância e o método `stop` para destruir a instância. O aplicativo usa o método `ObjectGridManager.getObjectGrid` para manter uma referência à instância. Consulte o informações sobre como acessar um `ObjectGrid` com o `ObjectGridManager` no *Guia de Programação* para obter mais informações.

## Utilizando Carregadores de Classes

Quando módulos aplicativos que utilizam carregadores de classes diferentes compartilharem uma única instância de `ObjectGrid` em um aplicativo Java EE, verifique se os objetos estão armazenados no eXtreme Scale e se os plug-ins para o produto estão em um carregador comum no aplicativo.

## Gerenciando o Ciclo de Vida de Instâncias do ObjectGrid em um Servlet

Para gerenciar o ciclo de vida de uma instância do `ObjectGrid` em um servlet, é possível usar o método `init` para criar a instância e o método `destroy` para remover a instância. Se a instância estiver armazenada em cache, ela será recuperada e manipulada no código do servlet. Consulte o informações sobre como acessar um `ObjectGrid` com a interface `ObjectGridManager` no *Guia de Programação* para obter mais informações.

## Convenções de Diretório

As seguintes convenções de diretório devem ser usadas em todo o documento para referenciar diretórios especiais, como `wxs_install_root` e `wxs_home`. Acesse esses diretórios durante vários cenários diferentes, inclusive durante a instalação e durante o uso das ferramentas de linha de comandos.

### `wxs_install_root`

O diretório `wxs_install_root` é o diretório-raiz no qual os arquivos do produto WebSphere eXtreme Scale são instalados. O diretório `wxs_install_root` pode ser o diretório no qual o archive de teste é extraído ou o diretório no qual o produto WebSphere eXtreme Scale é instalado.

- Exemplo ao extrair o teste:

**Exemplo:** `/opt/IBM/WebSphere/eXtremeScale`

- Exemplo quando o WebSphere eXtreme Scale é instalado em um diretório independente:

**Exemplo:** `/opt/IBM/eXtremeScale`

- Exemplo quando o WebSphere eXtreme Scale é integrado com o WebSphere Application Server:

**Exemplo:** `/opt/IBM/WebSphere/AppServer`

### `wxs_home`

O diretório `wxs_home` é o diretório raiz das bibliotecas, amostras e componentes do produto WebSphere eXtreme Scale. Esse diretório é o mesmo que o diretório `wxs_install_root` quando o teste é extraído. Para instalações independentes, o diretório `wxs_home` é o subdiretório `ObjectGrid` no diretório `wxs_install_root`. Para instalações que estão integradas ao WebSphere Application Server, esse diretório é o diretório `optionalLibraries/ObjectGrid` dentro do diretório `wxs_install_root`.

- Exemplo ao extrair o teste:



**Exemplo:** /opt/IBM/WebSphere/eXtremeScale

- Exemplo quando o WebSphere eXtreme Scale é instalado em um diretório independente:

**Exemplo:** /opt/IBM/eXtremeScale/ObjectGrid

- Exemplo quando o WebSphere eXtreme Scale é integrado com o WebSphere Application Server:

**Exemplo:** /opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid

#### **was\_root**

O diretório *was\_root* é o diretório-raiz de uma instalação do WebSphere Application Server:

**Exemplo:** /opt/IBM/WebSphere/AppServer

#### **restservice\_home**

O diretório *restservice\_home* é o diretório no qual as bibliotecas e amostras do serviço de dados REST do WebSphere eXtreme Scale estão localizadas. Este diretório é denominado *restservice* e é um subdiretório sob o diretório *wxs\_home*.

- Exemplo para implementações independentes:

**Exemplo:** /opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice

- Exemplo para implementações integradas do WebSphere Application Server:

**Exemplo:** /opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid/restservice

#### **tomcat\_root**

O *tomcat\_root* é o diretório-raiz da instalação do Apache Tomcat.

**Exemplo:** /opt/tomcat5.5

#### **wasce\_root**

O *wasce\_root* é o diretório-raiz da instalação do WebSphere Application Server Community Edition.

**Exemplo:** /opt/IBM/WebSphere/AppServerCE

#### **java\_home**

*java\_home* é o diretório-raiz de uma instalação de Java Runtime Environment (JRE).

**Exemplo:** /opt/IBM/WebSphere/eXtremeScale/java

#### **samples\_home**

O *samples\_home* é o diretório no qual os arquivos de amostra usados para os tutoriais são extraídos.

**Exemplo:** /wxs-samples/

#### **dvd\_root**

O diretório *dvd\_root* é o diretório-raiz do DVD que contém o produto.

**Exemplo:** dvd\_root/docs/

#### **equinox\_root**

O diretório *equinox\_root* é o diretório raiz da instalação da estrutura do Eclipse Equinox OSGi.

**Exemplo:**/opt/equinox

#### **user\_home**

O diretório *user\_home* é o local no qual arquivos de usuário são armazenados, tais como perfis de segurança.



**Windows** c:\Documents and Settings\user\_name

**UNIX** /home/user\_name

---

## Capacidade do Ambiente de Planejamento

Se você tiver um tamanho de conjunto de dados inicial e um tamanho de conjunto de dados projetado, é possível planejar a capacidade necessária para executar o WebSphere eXtreme Scale. Usando esses exercícios de planejamento, é possível implementar o WebSphere eXtreme Scale eficientemente para futuras mudanças e maximizar a elasticidade da grade de dados, o que não haveria com um cenário diferente, como um banco em memória ou outro tipo de banco de dados.

## Dimensionamento de Memória e Cálculo de Contagem de Partições

É possível calcular a quantidade de memória e partições necessárias para sua configuração específica.

**Atenção:** Este tópico se aplica quando o modo de cópia COPY\_TO\_BYTES não estiver sendo usado. Se estiver usando o modo COPY\_TO\_BYTES, o tamanho da memória será muito menor e o procedimento de cálculo será diferente.

O WebSphere eXtreme Scale armazena dados no espaço de endereço das Java Virtual Machines (JVM). Cada JVM fornece espaço no processador para criar, recuperar, atualizar e excluir chamadas para os dados que estão armazenados na JVM. Além disso, cada JVM fornece espaço de memória para entradas de dados e réplicas. Objetos Java variam de tamanho, assim você deve medir para fazer uma estimativa de quanta memória você precisa.

Para dimensionar a memória necessária, carregue os dados do seu aplicativo em uma única JVM. Quando o uso do heap alcança 60%, observe o número de objetos que são utilizados. Este número é a contagem máxima recomendada de objetos para cada uma de suas Java Virtual Machines. Para obter o dimensionamento mais exato, utilize dados realistas e inclua quaisquer índices definidos em seu dimensionamento porque os índices também consomem memória. A melhor forma de dimensionar o uso da memória é executar a saída **verbosegc** da coleta de lixo pois esta saída fornece os números após a coleta de lixo. É possível consultar o uso do heap em qualquer ponto específico por meio de MBeans ou programaticamente, porém essas consultas fornecem apenas uma captura instantânea atual do heap. Essa captura instantânea pode incluir lixo não coletado, portanto, usar esse método não indica precisamente a memória consumida.

## Escalando a Configuração

### Quantidade de shards por partição (numShardsPerPartition value)

Para calcular a quantidade de shards por partição, ou o valor de numShardsPerPartition, inclua 1 para o shard primário mais a quantidade total de shards de réplica que desejar.

`numShardsPerPartition = 1 + total_number_of_replicas`

### Quantidade de Java Virtual Machines (valor minNumJVMs)

Para escalar sua configuração, primeiro, decida o número máximo de objetos necessários a serem armazenados no total. Para determinar a quantidade de Java Virtual Machines que precisa, use a seguinte fórmula:

$$\text{minNumJVMs} = (\text{numShardsPerPartition} * \text{numObjs}) / \text{numObjsPerJVM}$$

Arredonde este valor para cima para o valor de número inteiro mais próximo.

### Quantidade de shards (valor numShards)

No tamanho de crescimento final, use 10 shards para cada JVM. Conforme descrito anteriormente, cada JVM possui um shard principal e (N-1) shards para as réplicas, ou neste caso, nove réplicas. Como você já possui uma quantidade de Java Virtual Machines para armazenar os dados, você pode multiplicar a quantidade de Java Virtual Machines por 10 para determinar a quantidade de shards:

$$\text{numShards} = \text{minNumJVMs} * 10 \text{ shards/JVM}$$

### Número de partições

Se uma partição tiver um shard primário e um shard de réplica, então a partição possui dois shards (principal e de réplica). O número de partições é a contagem de shards dividido por 2, arredondado para o número primo mais próximo. Se a partição possui um primário e duas réplicas, então, o número de partições é a contagem de shards dividida por 3, arredondada para o número primo mais próximo.

$$\text{numPartitions} = \text{numShards} / \text{numShardsPerPartition}$$

## Exemplo de Escala

Neste exemplo, o número de entrada inicia em 250 milhões. A cada ano, o número de entradas aumenta em cerca de 14%. Após sete anos, o número total de entrada será de 500 milhões, portanto, você deve planejar sua capacidade de acordo. Para alta disponibilidade, uma única réplica é necessária. Com uma réplica, o número de entradas duplica, ou seja, passa a ser 1 bilhão de réplicas. Como um teste, 2 milhões de entradas podem ser armazenadas em cada JVM. Utilizando os cálculos neste cenário, a seguinte configuração é necessária:

- 500 Java Virtual Machines para armazenar o número final de entradas.
- 5000 shards, calculados ao multiplicar 500 Java Virtual Machines por 10.
- 2500 partições, ou 2503 como o próximo número primo mais alto, calculados ao obter 5000 shards, divididos por dois para shards primários e de réplica.

### Iniciando a Configuração

Com base nos cálculos anteriores, inicie com 250 Java Virtual Machines e cresça até 500 Java Virtual Machines ao longo de cinco anos. Com essa configuração, é possível gerenciar o crescimento incremental até chegar ao número final de entradas.

Nessa configuração, cerca de 200.000 entradas são armazenadas por partição (500 milhões de entradas dividido por 2503 partições). Configure o parâmetro **numberOfBuckets** no mapa que mantém as entradas para o número primo maior mais próximo, que neste exemplo é 70887, o que mantém a proporção por volta de três.

### Quando o número máximo de Java Virtual Machines é atingido

Quando atingir o número máximo de 500 Java Virtual Machines, sua grade de dados ainda poderá crescer. À medida que o número de Java Virtual Machines aumenta além de 500, a contagem de shards começa a diminuir abaixo de 10 para cada JVM, que está abaixo do número recomendado. Os shards começam a ficar maiores, o que pode causar problemas. Repita o processo de dimensionamento considerando novamente um crescimento futuro e reconfigure a contagem da partição. Esta prática requer um reinício completo da grade de dados, ou uma ocorrência de uma interrupção da sua grade de dados.

## Número de Servidores

**Atenção:** Não utilize paginação em um servidor sob nenhuma circunstância.

Uma única JVM utiliza mais memória do que o tamanho do heap. Por exemplo, 1 GB de heap para uma JVM na verdade utiliza 1.4 GB de memória real. Determine a RAM livre disponível no servidor. Divida a quantidade de RAM pela memória por JVM para obter o número máximo da Java Virtual Machines no servidor.

## Dimensionando a CPU por Partição para Transações

Embora a maior funcionalidade do eXtreme Scale seja sua capacidade de efetuar escala elástica, também é importante considerar o dimensionamento e o ajuste do número ideal de CPUs para efetuar scale up.

Os custos do processador incluem:

- Custos dos serviços das operações create, retrieve, update e delete a partir dos clientes
- Custo de replicação do outro Java Virtual Machines.
- Custo de invalidação
- Custo da política de desocupação
- Custo da coleta de lixo
- Custo da lógica de aplicativo
- Custo de serialização

## Java Virtual Machines por servidor

Utilize dois servidores e inicie a contagem máxima de JVM por servidor. Utilize as contagens de partição calculadas da seção anterior. Em seguida, pré-carregue a Java Virtual Machines com dados suficientes para ajuste apenas nestes dois computadores. Utilize um servidor separado como um cliente. Execute uma simulação de transação realista junto a esta grade de dados de dois servidores.

Para calcular a linha de base, tente saturar o uso do processador. Se não for possível saturar o processador, então, é provável que a rede esteja saturada. Se a rede estiver saturada, inclua mais cartões de rede e execute o round robin da Java Virtual Machines sobre os vários cartões de rede.

Execute os computadores em 60% de uso do processador, meça a taxa de transações de criação, recuperação, atualização e exclusão. Esta medida fornece o rendimento nos dois servidores. Este número duplica com quatro servidores, duplica novamente em 8 servidores e assim por diante. Esta escala assume que a capacidade da rede e a capacidade do cliente também tenham a capacidade de escalar.

Como resultado, o tempo de resposta do eXtreme Scale deve ser estável à medida que o número de servidores é escalado para cima. O rendimento da transação deve escalar linearmente à medida que os computadores são incluídos na grade de dados.

## Dimensionando CPUs para Transações Paralelas

As transações de partição única possuem escala de rendimento linear à medida que a grade de dados cresce. Transações paralelas são diferentes de transações de partição única, pois elas acessam um conjunto de servidores (pode ser todos os servidores).

Se uma transação acessar todos os servidores, o rendimento será limitado ao rendimento do cliente que inicia a transação ou ao servidor mais lento que está sendo acessado. Grades de dados maiores propagam mais os dados e fornecem mais espaço do processador, memória, rede e assim por diante. Entretanto, o cliente deve aguardar que o servidor mais lento responda, e o cliente deve consumir os resultados da transação.

Quando uma transação acessa um subconjunto de servidores,  $M$  entre  $N$  servidores obtêm uma solicitação. O rendimento é então dividido  $N$  por  $M$  vezes mais rápido do que o rendimento do servidor mais lento. Por exemplo, se você tiver 20 servidores e uma transação que acessa 5 servidores, o rendimento será 4 vezes o rendimento do servidor mais lento na grade de dados.

Quando uma transação paralela é concluída, os resultados são enviados para o encadeamento do cliente que iniciou a transação. Este cliente deve então agregar os resultados únicos encadeados. Este tempo de agregação aumenta à medida que os número de servidores acessados pela transação aumentam. Entretanto, este tempo depende do aplicativo porque é possível que cada servidor retorne um resultado menor à medida que a grade de dados aumenta.

Normalmente, as transações paralelas acessam todos os servidores na grade de dados porque as partições são distribuídas de maneira uniforme pela grade. Neste caso, o rendimento é limitado ao primeiro caso.

### Resumo

Com este dimensionamento, são obtidas três métricas como a seguir:

- Número de partições.
- Número de servidores que são necessários para a memória que é necessária.
- Número de servidores que são necessários para o rendimento necessário.

Se você precisar de 10 servidores para requisitos de memória mas estiver obtendo apenas 50% do rendimento necessário devido à saturação do processador, então precisa do dobro de servidores.

Para maior estabilidade, você deve executar seus servidores a um carregamento de processador de 60% e JVMheaps a um carregamento de heap de 60%. Os picos podem então conduzir o uso do processador a 80-90%, mas não execute seus servidores regularmente em níveis superiores a este.

## Planejamento de Capacidade de Cache Dinâmico

A API de Cache Dinâmico está disponível para os aplicativos Java EE que são implementados no WebSphere Application Server. O cache dinâmico pode ser

potencializado para dados de negócios em cache, HTML gerado ou para sincronizar os dados em cache na célula usando o data replication service (DRS).

## Visão Geral

Todas as instâncias de cache dinâmico criadas com o provedor de cache dinâmico do WebSphere eXtreme Scale são altamente disponíveis por padrão. O nível e custo de memória de alta disponibilidade depende da topologia usada.

Ao usar a topologia integrada, o tamanho do cache é limitado à quantidade de memória livre em um único processo do servidor, e cada processo do servidor armazena uma cópia completa do cache. Enquanto um único processo do servidor continua a executar, o cache sobrevive. Os dados do cache somente serão perdidos se todos os servidores que acessam o cache forem desligados.

Para armazenamento em cache que usa topologia particionada integrada, o tamanho do cache é limitado a um agregado do espaço livre disponível em todos os processos do servidor. Por padrão, o provedor de cache dinâmico do eXtreme Scale usa 1 réplica para cada shard primário, assim cada parte dos dados armazenados em cache é armazenada duas vezes.

Use a seguinte fórmula A para determinar a capacidade de um cache particionado integrado.

### Fórmula A

$$F * C / (1 + R) = M$$

Em que:

- F = Memória livre por processo de contêiner
- C = número de contêineres
- R = número de réplicas
- M = Tamanho total do cache

Para que uma grade de dados do WebSphere Application Server Network Deployment tenha 256 MB de espaço disponível em cada processo, com um total de 4 processos do servidor, uma instância de cache por todos esses servidores poderia armazenar até 512 megabytes de dados. Deste modo, o cache pode sobreviver a um dano no servidor sem perder dados. Também, até dois servidores poderiam ser desligados sequencialmente sem perda de dado algum. Assim, para o exemplo anterior, a fórmula é a seguinte:

$$256\text{mb} * 4 \text{ contêineres} / (1 \text{ primário} + 1 \text{ réplica}) = 512\text{mb.}$$

Caches que usam a topologia remota têm características de dimensionamento similares às dos caches que usam topologia particionada integrada, mas eles são limitados pela quantidade de espaço disponível em todos os processos de contêiner do eXtreme Scale.

Em topologias remotas, é possível aumentar o número de réplicas para fornecer um nível mais alto de disponibilidade ao custo de gasto adicional de memória. Na maioria dos aplicativos com cachê dinâmico, isso seria desnecessário, porém o arquivo `dynacache-remote-deployment.xml` pode ser editado para aumentar o número de réplicas.

Use as seguintes fórmulas, B e C, para determinar o efeito da inclusão de mais réplicas na alta disponibilidade do cache.

#### **Fórmula B**

$$N = \text{Mínimo}(T - 1, R)$$

Em que:

- N = o número de processos que podem travar simultaneamente
- T = o número total de contêineres
- R = o número total de réplicas

#### **Fórmula C**

$$\text{Limite}(T / (1+N)) = m$$

Em que:

- T = Número total de contêineres
- N = Número total de réplicas
- m = número mínimo de contêineres necessários para suportar os dados em cache.

Para ajuste de desempenho com o provedor de cache dinâmico, consulte “Ajustando o Provedor de Cache Dinâmico” na página 495.

### **Dimensionamento do Cache**

Antes que um aplicativo que usa o provedor de cache dinâmico do WebSphere eXtreme Scale possa ser implementado, os princípios gerais descritos na seção anterior devem ser combinados com os dados ambientais para os sistemas de produção. O primeiro valor a estabelecer é o número total de processos do contêiner e a quantidade de memória disponível em cada processo para conter os dados do cache. Ao usar a topologia integrada, os contêineres de cache serão colocados dentro dos processos do servidor do WebSphere Application, assim, há um contêiner para cada servidor que estiver compartilhando o cache. Determinar o gasto adicional de memória do aplicativo sem o armazenamento em cache ativado e o WebSphere Application Server é a melhor maneira de descobrir quanto espaço está disponível no processo. Isso pode ser feito por meio de análise detalhada dos dados da coleta de lixo. Ao usar a topologia remota, essas informações podem ser localizadas por meio da verificação da saída detalhada da coleta de lixo de um contêiner independente recentemente iniciado que ainda não foi preenchido com dados do cache. A última coisa a lembrar para descobrir quanto espaço por processo está disponível para dados em cache, é reservar algum espaço de heap para a coleta de lixo. O gasto adicional do contêiner, o WebSphere Application Server ou independente, mais o tamanho reservado para o cache não deve ser maior que 70% do heap total.

Depois que essas informações são coletadas, os valores poderão ser inseridos na fórmula A, descritos anteriormente, para determinar o tamanho máximo para o cache particionado. Depois que o tamanho máximo é conhecido, a próxima etapa é determinar o número total de entradas do cache que pode ser suportado, o que requer determinar o tamanho médio por entrada do cache. A maneira mais simples de fazer isso é incluir 10% ao tamanho do objeto do cliente. Consulte o Guia de

Ajuste para Cache Dinâmico e Serviço de Replicação de Dados para obter mais informações detalhadas sobre o dimensionamento de entradas de cache ao usar o cache dinâmico.

Quando a compactação está ativada, ela afeta o tamanho do objeto do cliente, não o gasto adicional do sistema de armazenamento em cache. Use a fórmula a seguir para determinar o tamanho de um objeto em cache ao usar a compactação:

$$S = O * C + O * 0.10$$

Em que:

- S = Tamanho médio do objeto em cache
- O = Tamanho médio do objeto do cliente não-compactado
- C = Proporção de compactação expressa como uma fração.

Assim, uma proporção de compactação 2 para 1 é  $1/2 = 0.50$ . Para este valor, quanto menor, melhor. Se o objeto que está sendo armazenado é um POJO normal cheio principalmente tipos primitivos, então assuma uma proporção de compactação de 0.60 para 0.70. Se o objeto em cache é um objeto Servlet, JSP ou WebServices, o método ideal para determinar a taxa de compactação é compactar uma amostra representativa com um utilitário de compactação ZIP. Se isso não for possível, então uma proporção de compactação de 0.2 para 0.35 é comum para esses tipos de dados.

Depois, use estas informações para determinar o número total de entradas do cache que pode ser suportado. Use a seguinte fórmula D:

#### **Fórmula D**

$$T = S / A$$

Em que:

- T= Número total de entradas do cache
- S = Tamanho total disponível para dados em cache conforme computados usando a fórmula A
- A = Tamanho médio de cada entrada do cache

Finalmente, você deve configurar o tamanho do cache na instância do cache dinâmico para impor este limite. O provedor de cache dinâmico do WebSphere eXtreme Scale difere do provedor de cache dinâmico padrão neste aspecto. Use a fórmula a seguir para determinar o valor para definir o tamanho do cache na instância do cache dinâmico. Use a seguinte fórmula E:

#### **Fórmula E**

$$Cs = Ts / Np$$

Em que:

- Ts = Tamanho de destino total para o cache
- Cs = Configuração do Tamanho do Cache para definir na instância do cache dinâmico
- Np = Número de partições. O padrão é 47.

Configure o tamanho da instância do cache dinâmico para um valor calculado pela fórmula E em cada servidor que compartilhar a instância do cache.

## Planejando para Configuração

Antes de configurar o hardware ou software, entenda as considerações a seguir.

### Lista de Verificação Operacional

Use a lista de verificação operacional para preparar o ambiente para implementar o WebSphere eXtreme Scale.

Tabela 3. Lista de Verificação Operacional

Item de Lista de Verificação	Para obter informações adicionais
<p>Se você estiver usando o AIX, ajuste as seguintes configurações do sistema operacional:</p> <p><b>TCP_KEEPINTVL</b></p> <p>A configuração TCP_KEEPINTVL faz parte de um protocolo de soquete keep-alive que permite a detecção de uma interrupção de rede. A propriedade especifica o intervalo entre os pacotes que são enviados para validar a conexão. Ao usar o WebSphere eXtreme Scale, configure o valor para 10. Para verificar a configuração atual, execute o seguinte comando:</p> <pre># no -o tcp_keepintvl</pre> <p>Para alterar a configuração atual, execute o seguinte comando:</p> <pre># no -o tcp_keepintvl=10</pre> <p>A configuração TCP_KEEPINTVL está em unidades de meio segundo.</p> <p><b>TCP_KEEPINIT</b></p> <p>A configuração TCP_KEEPINIT faz parte de um protocolo de soquete keep-alive que permite a detecção de uma interrupção de rede. A propriedade especifica o valor de tempo limite inicial para a conexão TCP. Ao usar o WebSphere eXtreme Scale, configure o valor para 40. Para verificar a configuração atual, execute os seguintes comandos:</p> <pre># no -o tcp_keepinit</pre> <p>Para alterar a configuração atual, execute o seguinte comando:</p> <pre># no -o tcp_keepinit=40</pre> <p>A configuração TCP_KEEPINIT está em unidades de meio segundo.</p>	<ul style="list-style-type: none"><li>Para obter informações de ajuste do AIX, consulte Ajustando Sistemas AIX.</li></ul>
<p>Atualize o arquivo <code>orb.properties</code> para modificar o comportamento de transporte da grade. O arquivo <code>orb.properties</code> está no diretório <code>java/jre/lib</code>.</p>	<p>“Propriedades do ORB” na página 482</p>



Tabela 3. Lista de Verificação Operacional (continuação)

Item de Lista de Verificação	Para obter informações adicionais
<p>Use os parâmetros no script <b>start0gServer</b>. Em particular, use os seguintes parâmetros:</p> <ul style="list-style-type: none"> <li>• Defina as configurações de heap com o parâmetro <b>-jvmArgs</b>.</li> <li>• Defina o caminho de classe e as propriedades do aplicativo com o parâmetro <b>-jvmArgs</b>.</li> <li>• Defina os parâmetros <b>-jvmArgs</b> para configurar o monitoramento de agente.</li> </ul> <p><b>Configurações de Porta</b> O WebSphere eXtreme Scale precisa abrir portas para comunicações para alguns transportes. Essas portas são todas definidas dinamicamente. Porém, se um firewall estiver em uso entre os contêineres, será necessário especificar as portas. Use as seguintes informações sobre as portas:</p> <p><b>Porta Listener</b> Você pode usar o argumento <b>-listenerPort</b> para especificar a porta que é usada para comunicação entre processos.</p> <p><b>Porta do grupo principal</b> Você pode usar o argumento <b>-haManagerPort</b> para especificar a porta que é usada para detecção de falha. Este argumento é o mesmo que <b>peerPort</b>. Observe que os grupos principais não precisam se comunicar entre as zonas, portanto, pode não ser necessário configurar essa porta se o firewall estiver aberto para todos os membros de uma única zona.</p> <p><b>Porta de serviço JMX</b> Você pode usar o argumento <b>-JMXServicePort</b> para especificar a porta que o serviço JMX deve usar.</p> <p><b>Porta SSL</b> Transmitir <b>-Dcom.ibm.CSI.SSLPort=1234</b> como um argumento <b>-jvmArgs</b> configura a porta SSL para 1234. A porta SSL é a porta protegida equivalente à porta do listener.</p> <p><b>Porta do cliente</b> Usada apenas no serviço de catálogo. Você pode especificar esse valor com o argumento <b>-catalogServiceEndpoints</b>. O formato do valor desse parâmetro está no formato: <code>serverName:hostName:clientPort:peerPort</code></p>	<p>“Script<b>start0gServer</b>” na página 393</p>
<p>Verifique se as configurações de segurança estão definidas corretamente:</p> <ul style="list-style-type: none"> <li>• Transporte (SSL)</li> <li>• Aplicativo (Autenticação e Autorização)</li> </ul> <p>Para verificar as configurações de segurança, é possível tentar usar um cliente malicioso para conectar-se com a sua configuração. Por exemplo, quando a configuração SSL necessária for definida, um cliente que possuir a configuração TCP_IP ou um cliente com o truststore incorreto não conseguirá se conectar ao servidor. Quando a autenticação é necessária, um cliente com nenhuma credencial, como um ID de usuário e senha, não conseguirá se conectar ao servidor. Quando a autorização é forçada, um cliente com nenhuma autorização de acesso não conseguirá acessar os recursos do servidor.</p>	<p>“Integração de Segurança com Provedores Externos” na página 510</p>
<p>Escolha como você monitorará seu ambiente.</p> <ul style="list-style-type: none"> <li>• Ferramenta: <b>xscmd</b> <ul style="list-style-type: none"> <li>– As portas JMX dos servidores de catálogos precisam estar visíveis para a ferramenta <b>xscmd</b>. As portas do servidor de contêiner também precisam estar acessíveis para alguns comandos que reúnem informações a partir dos contêineres.</li> </ul> </li> <li>• Monitorando o console: Com o console de monitoramento, é possível criar um gráfico das estatísticas atuais e históricas.</li> <li>• Ferramentas de monitoramento do fornecedor: <ul style="list-style-type: none"> <li>– Tivoli Enterprise Monitoring Agent</li> <li>– CA Wily Introscope</li> <li>– Hyperic HQ</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• “Monitorando com o Utilitário <b>xscmd</b>” na página 453</li> <li>• “Segurança do Java Management Extensions (JMX)” na página 507</li> <li>• “Monitorando com o Console da Web” na página 435</li> <li>• “Monitorando com o IBM Tivoli Enterprise Monitoring Agent para WebSphere eXtreme Scale” na página 468</li> <li>• “Monitorando o eXtreme Scale com o Hyperic HQ” na página 477</li> <li>• “Monitorando Aplicativos eXtreme Scale com o CA Wily Introscope” na página 474</li> </ul>

## Planejamento para Portas de Rede

O WebSphere eXtreme Scale é um cache distribuído que requer portas de abertura para se comunicar com a pilha Object Request Broker (ORB) e Protocolo de Controle de Transmissões (TCP) entre as Java virtual machines. Planeje e controle suas portas, principalmente em um ambiente com firewall, e quando estiver usando um serviço de catálogo e contêineres em diversas portas.

**Importante:** Quando estiver especificando os números de portas, evite configurar portas que estejam no intervalo temporário para seu sistema operacional. Se usar uma porta que esteja no intervalo temporário, poderão ocorrer conflitos de porta.

### Domínio do Serviço de Catálogo

Um domínio do serviço de catálogo requer as seguintes portas para ser definido:

#### peerPort

Especifica a porta para o gerenciador de alta disponibilidade (HA) se comunicar entre servidores de catálogo de peer sobre uma pilha TCP. No WebSphere Application Server, esta configuração é herdada pela configuração da porta do gerenciador de alta disponibilidade.

#### clientPort

Especifica a porta para servidores de catálogo acessarem dados do serviço de catálogo. No WebSphere Application Server, essa porta é configurada por meio da configuração do domínio de serviço de catálogo.

#### listenerPort

Especifica o número da porta ao qual o Object Request Broker (ORB) se conecta. Esta configuração define que contêineres e clientes se comunicam com o serviço de catálogo por meio do ORB. No WebSphere Application Server, a listenerPort é herdada pela configuração de porta BOOTSTRAP\_ADDRESS. Esta propriedade se aplica ao servidor de contêiner e ao serviço de catálogo.

**Padrão:**2809

#### JMXConnectorPort

Define a porta Secure Sockets Layer (SSL) à qual o serviço Java Management Extensions (JMX) se conecta.

### Servidores de Contêineres

Os servidores de contêineres do WebSphere eXtreme Scale também precisam de várias portas para operar. Por padrão, o servidor de contêineres do eXtreme Scale gera sua porta do gerenciador HA e porta do listener ORB automaticamente com portas dinâmicas. Para um ambiente que possui um firewall, é recomendado planejar e controlar as portas. Para os servidores de contêiner iniciarem com portas específicas, as seguintes opções poderão ser usadas no comando **startOgServer**.

#### haManagerPort

Sinônimo de porta de peer. Especifica o número da porta usada pelo gerenciador de alta disponibilidade. Se essa propriedade não for configurada, o serviço de catálogo gera uma porta disponível automaticamente. Essa propriedade se aplica ao servidor de contêiner e ao serviço de catálogo. (Necessário somente para ambientes do WebSphere Application Server.)

#### listenerPort

Especifica o número da porta ao qual o Object Request Broker (ORB) se

conecta. Esta configuração define que contêineres e clientes se comunicam com o serviço de catálogo por meio do ORB. No WebSphere Application Server, a listenerPort é herdada pela configuração de porta BOOTSTRAP\_ADDRESS. Esta propriedade se aplica ao servidor de contêiner e ao serviço de catálogo.

**Padrão:**2809

#### **JMXConnectorPort**

Define a porta Secure Sockets Layer (SSL) à qual o serviço Java Management Extensions (JMX) se conecta.

##### **7.1.1+ xioChannel.xioContainerTCPSecure.Port**

Especifica o número de porta SSL do eXtremeIO no servidor. Esta propriedade é usada somente quando a propriedade **transportType** é configurada como SSL-Supported ou SSL-Required.

##### **7.1.1+ xioChannel.xioContainerTCPNonSecure.Port**

Especifica o número de porta listener não segura do eXtremeIO no servidor. Se esse valor não for configurado, uma porta temporária será usada. Esta propriedade é usada somente quando a propriedade **transportType** é configurada como TCP/IP.

Planejar adequadamente o controle da porta é essencial quando centenas de Java virtual machines são iniciadas em um servidor. Se existir um conflito de portas, os servidores de contêiner não são iniciados.

## **Clientes**

Os clientes do WebSphere eXtreme Scale podem receber retornos de chamada de servidores quando estiver usando a API do DataGrid ou vários outros comandos. Use a propriedade **listenerPort** no arquivo de propriedades do cliente para especificar a porta na qual o cliente atende aos retornos de chamada do servidor.

#### **haManagerPort**

Sinônimo de porta de peer. Especifica o número da porta usada pelo gerenciador de alta disponibilidade. Se essa propriedade não for configurada, o serviço de catálogo gera uma porta disponível automaticamente. Essa propriedade se aplica ao servidor de contêiner e ao serviço de catálogo. (Necessário somente para ambientes do WebSphere Application Server.)

#### **jvmArgs (opcional)**

Especifica uma lista de argumentos da Java virtual machine (JVM). Quando a segurança está ativada, você deve usar o argumento a seguir para configurar a porta Secure Socket Layer (SSL): -jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>.

#### **listenerPort**

Especifica o número da porta ao qual o Object Request Broker (ORB) se conecta. Esta configuração define que contêineres e clientes se comunicam com o serviço de catálogo por meio do ORB. No WebSphere Application Server, a listenerPort é herdada pela configuração de porta BOOTSTRAP\_ADDRESS. Esta propriedade se aplica ao servidor de contêiner e ao serviço de catálogo.

**Padrão:**2809

## Portas no WebSphere Application Server

- O valor de **listenerPort** é herdado do valor de **BOOTSTRAP\_ADDRESS** para cada servidor de aplicativos do WebSphere Application Server.
- Os valores **haManagerPort** e **peerPort** herdados do valor **DCS\_UNICAST\_ADDRESS** de cada servidor de aplicativos WebSphere Application Server.

É possível definir um domínio de serviço de catálogo no console administrativo, conforme descrito em “Criando Domínios do Serviço de Catálogo no WebSphere Application Server” na página 252.

Você pode visualizar as portas para um determinado servidor ao clicar em um dos seguintes caminhos no console administrativo:

- WebSphere Application Server Network Deployment Versão 6.1: **Servidores > Servidores de Aplicativos > server\_name > Portas > end\_point\_name.**
- WebSphere Application Server Network Deployment Versão 7.0: **Servidores > Tipos de Servidor > WebSphere Application Servers > server\_name > Portas > port\_name**

## Visão Geral de Segurança

WebSphere eXtreme Scale pode proteger o acesso a dados, incluindo permissão para integração com provedores de segurança externos.

**Nota:** Em um armazenamento de dados fora do cache existente, como um banco de dados, provavelmente é necessário ter recursos de segurança integrados que podem não ser necessários para configuração ou ativação de modo ativo. Entretanto, após ter armazenado seus dados em cache com o eXtreme Scale, você deve considerar a importante situação resultante de que seus recursos de segurança de backend não estão mais em vigor. É possível configurar a segurança do eXtreme Scale no níveis necessários para que a nova arquitetura armazenada em cache para os seus dados também fique segura.

A seguir é apresentado um breve resumo sobre os recursos de segurança do eXtreme Scale. Para obter mais informações detalhadas sobre como configurar a segurança, consulte o *Guia de Administração* e o *Guia de Programação*.

## Fundamentos sobre Segurança Distribuída

A segurança distribuída do eXtreme Scale é baseada em três conceitos fundamentais:

### *Autenticação confiável*

A habilidade de determinar a identidade do solicitante. O WebSphere eXtreme Scale suporta autenticação cliente-para-servidor e servidor-para-servidor.

### *Autorização*

A habilidade de dar permissões para conceder direitos de acesso ao solicitante. O WebSphere eXtreme Scale suporta diferentes autorizações para várias operações.

### *Transporte Seguro*

A transmissão segura dos dados sobre uma rede. O WebSphere eXtreme Scale suporta os protocolos TLS/SSL (Transport Layer Security/Secure Sockets Layer).

## **Autenticação**

O WebSphere eXtreme Scale suporta uma estrutura de cliente e servidor distribuída. Uma infraestrutura de segurança de cliente e servidor está estabelecida para proteger o acesso aos servidores eXtreme Scale. Por exemplo, quando a autenticação é necessária pelo servidor do eXtreme Scale, um cliente do eXtreme Scale deve fornecer credenciais para se autenticar no servidor. Essas credenciais podem ser um par de nome de usuário e senha, um certificado cliente, um ticket Kerberos ou dados que são apresentados em um formato acordado entre o cliente e o servidor.

## **Autorização**

As autorizações do WebSphere eXtreme Scale são baseadas em assuntos e permissões. É possível utilizar o Java Authentication and Authorization Services (JAAS) para autorizar o acesso ou é possível conectar uma abordagem customizada, tal como Tivoli Access Manager (TAM), para tratar as autorizações. As seguintes autorizações podem ser fornecidas a um cliente ou grupo:

### **Autorização de mapa**

Execute operações insert, read, update, evict ou delete nos Mapas.

### **Autorização do ObjectGrid**

Execute consultas em objetos ou entidades e consultas em fluxos nos objetos do ObjectGrid.

### **Autorização do agente do DataGrid**

Permita que os agentes do DataGrid sejam implementados em um ObjectGrid.

### **Autorização do mapa do lado do servidor**

Replique um mapa de servidor para o lado do cliente ou crie um índice dinâmico para o mapa do servidor.

### **Autorização de administração**

Execute tarefas de administração.

## **Segurança do Transporte**

Para garantir a segurança da comunicação entre cliente e o servidor, o WebSphere eXtreme Scale suporta TLS/SSL. Estes protocolos fornecem segurança da camada de transporte com autenticidade, integridade e confidencialidade para uma conexão segura entre um cliente e um servidor do eXtreme Scale.

## **Segurança da Grade**

Em um ambiente seguro, um servidor deve poder verificar a autenticidade de outro servidor. O WebSphere eXtreme Scale utiliza um mecanismo de cadeia de chave secreta compartilhado para este propósito. Este mecanismo de chave secreta é semelhante a uma senha compartilhada. Todos os servidores eXtreme Scale aceitam uma cadeia secreta compartilhada. Quando um servidor se junta à grade de dados, o servidor é desafiado a apresentar a sequência secreta. Se a cadeia secreta do servidor que está se juntando corresponder a uma cadeia no servidor principal, então o servidor que está se juntando pode ser unido à grade. Caso contrário, o pedido de junção será rejeitado.

Não é seguro enviar um segredo em texto não-criptografado. A infraestrutura de segurança do eXtreme Scale fornece um plug-in SecureTokenManager para

possibilitar que o servidor faça a segurança deste segredo antes de enviá-lo. É possível escolher como implementar a operação segura. O WebSphere eXtreme Scale fornece uma implementação, na qual a operação segura é implementada para criptografar e assinar o segredo.

## **Segurança Java Management Extensions (JMX) em uma Topologia de Implementação Dinâmica**

A segurança JMX MBean é suportada em todas as versões do eXtreme Scale. Clientes dos MBeans do servidor de catálogos e MBeans do servidor de contêineres podem ser autenticados e o acesso às operações do MBean podem ser impostos.

## **Segurança Local do eXtreme Scale**

A segurança local do eXtreme Scale é diferente do modelo distribuído do eXtreme Scale porque o aplicativo instancia diretamente e utiliza uma instância do ObjectGrid. Seu aplicativo e as instâncias do eXtreme Scale estão na mesma Java virtual machine (JVM). Como não há nenhum conceito de cliente/servidor neste modelo, a autenticação não é suportada. Seu aplicativo deve gerenciar sua própria autenticação e, então, passar o objeto Subject autenticado para o eXtreme Scale. Porém, o mecanismo de autorização usado para o modelo de programação do eXtreme Scale local é o mesmo que o usado para o modelo cliente/servidor.

## **Configuração e Programação**

Para obter mais informações sobre como configurar e programar a segurança, consulte o “Integração de Segurança com Provedores Externos” na página 510 e o API de Segurança.

---

## Capítulo 3. Tutoriais



Os tutoriais podem ser usados para ajudar a entender os cenários de uso do produto, incluindo o gerenciador, consultas e segurança da entidade.

---

### Tutorial: Configurando a Segurança do Java SE

Com o seguinte tutorial, é possível criar um ambiente eXtreme Scale distribuído em um ambiente Java Platform, Standard Edition.

#### Antes de Iniciar

Certifique-se de estar familiarizado com os conceitos básicos de uma configuração distribuída do eXtreme Scale.

#### Sobre Esta Tarefa

Neste tutorial, o servidor de catálogo, servidor de contêiner e o cliente são todos executados em um ambiente do Java SE. Cada etapa no tutorial é baseada na anterior. Siga cada etapa para proteger um eXtreme Scale distribuído e a desenvolver um aplicativo simples Java SE para acessar o eXtreme Scale protegido.

Iniciar o tutorial

#### Procedimento

1. “Tutorial de Segurança do Java SE - Etapa 1” na página 68
  - Inicie um servidor de catálogos não-seguro
  - Inicie um servidor de contêineres não-seguro
  - Inicie um cliente para acessar os dados
  - Use o utilitário `xscmd` para mostrar o tamanho do mapa
  - Pare o servidor
2. “Tutorial de Segurança do Java SE - Etapa 2” na página 71
  - Uso do CredentialGenerator
  - Uso do Autenticador
  - Inicie um servidor de catálogos seguro
  - Inicie um servidor de contêineres seguro
  - Inicie o cliente para acessar o ObjectGrid seguro
  - Use o utilitário `xscmd` para mostrar o tamanho do mapa
  - Pare o servidor protegido
3. “Tutorial de Segurança do Java SE - Etapa 3” na página 77
  - Uso da política de autorização do JAAS
4. “Tutorial de Segurança do Java SE - Etapa 4” na página 80
  - Crie um key store e trust store
  - Configure propriedades SSL para o servidor
  - Configure propriedades SSL para o cliente
  - Use o utilitário `xscmd` para mostrar o tamanho do mapa

- Pare o servidor protegido

## Tutorial de Segurança do Java SE - Etapa 1

Este tópico descreve uma *amostra não-segura simples*. Recursos de segurança adicionais são incluídos de maneira incremental nas etapas do tutorial para aumentar a quantidade de segurança integrada que está disponível.

### Antes de Iniciar

**Nota:** Todos os arquivos necessários para essa etapa do tutorial são fornecidos na seguinte seção.

### Procedimento

#### Executando a amostra

Inicie o serviço de catálogo usando os seguintes scripts. Para obter mais informações sobre como iniciar o serviço de catálogo, consulte “Iniciando um Serviço de Catálogo Independente” na página 387.

1. Navegue até o diretório bin: `cd objectgridRoot/bin`
2. Inicie um servidor de catálogos denominado `catalogServer`:
  - **UNIX Linux** `startOgServer.sh catalogServer`
  - **Windows** `startOgServer.bat catalogServer`
3. Navegue até o diretório bin `cd objectgridRoot/bin`
4. Em seguida, ative um servidor de contêiner denominado `c0` com o script a seguir:

- **UNIX Linux**  

```
startOgServer.sh c0 -objectGridFile ../xml/SimpleApp.xml -deploymentPolicyFile ../xml/SimpleDP.xml  
-catalogServiceEndPoints localhost:2809
```
- **Windows**  

```
startOgServer.bat c0 -objectGridFile ../xml/SimpleApp.xml - deploymentPolicyFile ../xml/SimpleDP.xml  
-catalogServiceEndPoints localhost:2809
```

### Exemplo

Para obter informações adicionais sobre o início dos servidores de contêiner, consulte “Iniciando Servidores de Contêiner” na página 390.

Após o servidor de catálogos e o servidor de contêineres terem sido iniciados, ative o cliente conforme a seguir.

1. Navegue até o diretório bin mais uma vez.
2. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.websphere.objectgrid.security.sample.guide.SimpleApp`

O arquivo `secsample.jar` contém a classe `SimpleApp`.

A saída deste programa é:

```
The customer name for ID 0001 is fName lName
```

Também possível recorrer ao utilitário `xscmd` para mostrar os tamanhos de mapa da grade "accounting".

- Navegue até o diretório `objectgridRoot/bin`
- Use o utilitário `xscmd` para mostrar os tamanhos de mapa:



- `UNIX` `Linux` `xscmd.sh -c showMapSizes -g accounting -ms mapSet1`
- `Windows` `xscmd.bat -c showMapSizes -g accounting -ms mapSet1`

## Parando servidores

### *Servidor de Contêiner*

Utilize o seguinte comando para parar o servidor de contêiner c0.

```
UNIX Linux stopOgServer.sh c0 -catalogServiceEndPoints localhost:2809
```

```
Windows stopOgServer.bat c0 -catalogServiceEndPoints localhost:2809
```

A seguinte mensagem será exibida.

```
CWOBJ2512I: ObjectGrid server c0 stopped.
```

### *Servidor de catálogos*

É possível parar um servidor de catálogos usando o seguinte comando.

```
UNIX Linux stopOgServer.sh catalogServer -catalogServiceEndPoints localhost:2809
```

```
Windows stopOgServer.bat catalogServer -catalogServiceEndPoints localhost:2809
```

Se você encerrar o servidor de catálogos, a seguinte mensagem será exibida.

```
CWOBJ2512I: ObjectGrid server catalogServer stopped.
```

## Arquivos necessários

O arquivo a seguir é a classe Java para SimpleApp.

```
SimpleApp.java
// This sample program is provided AS IS and may be used, executed, copied and modified
// without royalty payment by customer
// (a) for its own instruction and study,
// (b) in order to develop applications designed to run with an IBM WebSphere product,
// either for customer's own internal use or for redistribution by customer, as part of such an
// application, in customer's own products.
// Licensed Materials - Property of IBM
// 5724-J34 (C) COPYRIGHT International Business Machines Corp. 2007-2009
package com.ibm.websphere.objectgrid.security.sample.guide;

import com.ibm.websphere.objectgrid.ClientClusterContext;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManager;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.ObjectMap;
import com.ibm.websphere.objectgrid.Session;

public class SimpleApp {

    public static void main(String[] args) throws Exception {

        SimpleApp app = new SimpleApp();
        app.run(args);
    }

    /**
```

```

    * read and write the map
    * @throws Exception
    */
protected void run(String[] args) throws Exception {
    ObjectGrid og = getObjectGrid(args);

    Session session = og.getSession();

    ObjectMap customerMap = session.getMap("customer");

    String customer = (String) customerMap.get("0001");

    if (customer == null) {
        customerMap.insert("0001", "fName lName");
    } else {
        customerMap.update("0001", "fName lName");
    }
    customer = (String) customerMap.get("0001");

    System.out.println("The customer name for ID 0001 is " + customer);
}

/**
 * Get the ObjectGrid
 * @return an ObjectGrid instance
 * @throws Exception
 */
protected ObjectGrid getObjectGrid(String[] args) throws Exception {
    ObjectGridManager ogManager = ObjectGridManagerFactory.getObjectGridManager();

    // Create an ObjectGrid
    ClientClusterContext ccContext = ogManager.connect("localhost:2809", null, null);
    ObjectGrid og = ogManager.getObjectGrid(ccContext, "accounting");

    return og;
}
}

```

O método getObjectGrid nesta classe obtém o um ObjectGrid e o método run lê um registro a partir do mapa do cliente e atualiza o valor.

Para executar essa amostra em um ambiente distribuído, um arquivo descritor XML do ObjectGrid SimpleApp.xml e um arquivo de implementação XML SimpleDP.xml são criados. Os arquivos são apresentados no exemplo a seguir:

#### SimpleApp.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="accounting">
            <backingMap name="customer" readOnly="false" copyKey="true"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>

```

O arquivo XML a seguir configura o ambiente de implementação.

#### SimpleDP.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

    <objectgridDeployment objectgridName="accounting">
        <mapSet name="mapSet1" numberOfPartitions="1" minSyncReplicas="0" maxSyncReplicas="2"
maxAsyncReplicas="1">
            <map ref="customer"/>
        </mapSet>
    </objectgridDeployment>
</deploymentPolicy>

```

Essa é uma configuração simples do ObjectGrid com uma instância do ObjectGrid chamada "accounting" e um mapa chamado "customer" (dentro do mapSet "mapSet1"). O arquivo SimpleDP.xml apresenta um conjunto de mapa que é configurado com 1 partição e com o mínimo de 0 réplicas necessárias.

Próxima etapa do tutorial

## Tutorial de Segurança do Java SE - Etapa 2

Baseado na etapa anterior, o seguinte tópico mostra como implementar a autenticação do cliente em um ambiente do eXtreme Scale distribuído.

### Antes de Iniciar

Certifique-se de ter concluído o "Tutorial de Segurança do Java SE - Etapa 1" na página 68.

### Sobre Esta Tarefa

Com a autenticação de cliente ativada, um cliente é autenticado antes de conectar-se ao servidor eXtreme Scale. Esta seção demonstra como a autenticação de cliente pode ser feita em um ambiente de servidor eXtreme Scale, incluindo código de amostra e scripts para demonstrar.

Como qualquer outro mecanismo de autenticação, a autenticação mínima consiste nas seguintes etapas:

1. O administrador altera as configurações feitas para tornar a autenticação um requisito.
2. O cliente oferece uma credencial para o servidor.
3. O servidor autentica a credencial para o registro.

### Procedimento

#### 1. Credencial de cliente

Uma credencial de cliente é representada por uma interface `com.ibm.websphere.objectgrid.security.plugins.Credential`. Uma credencial de cliente pode ser um par de nome de usuário e senha, um registro do Kerberos, um certificado cliente ou dados em qualquer formato concordado entre o cliente e o servidor. Consulte a documentação da API de Credencial para obter mais detalhes.

Esta interface define explicitamente os métodos `equals(Object)` e `hashCode()`. Estes métodos são importantes porque os objetos `Subject` autenticados são armazenados em cache utilizando o objeto `Credential` como a chave no lado do servidor.

O eXtreme Scale também fornece um plug-in para gerar uma credencial. Este plug-in é representado pela interface `com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator` e é utilizado para gerar uma credencial de cliente. Isso é útil quando a credencial pode expirar. Neste caso, o método `getCredential()` é chamado para renovar uma credencial. Consulte a Documentação da API do `CredentialGenerator` para obter mais detalhes.

É possível implementar estas duas interfaces para o tempo de execução do cliente do eXtreme Scale para obter credenciais de cliente.

Essa amostra usa as seguintes duas implementações de plug-in de amostra fornecidas pelo eXtreme Scale.

```
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredential
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
```

Para obter mais informações sobre esses plug-ins, consulte o Programação de Autenticação de Cliente.

2. **Autenticação do servidor** Após o cliente do eXtreme Scale recuperar o objeto Credential utilizando o objeto CredentialGenerator, este objeto Credential do cliente é enviado junto o pedido do cliente para o servidor eXtreme Scale. O servidor eXtreme Scale autentica o objeto Credential antes de processar o pedido. Se o objeto Credential for autenticado com êxito, um objeto Subject será retornado para representar este cliente.

Este objeto Subject é então armazenado em cache e expira após seu tempo de vida alcançar o valor de tempo limite da sessão. O valor de tempo limite da sessão de login pode ser configurado utilizando a propriedade loginSessionExpirationTime no arquivo XML do cluster. Por exemplo, configurar loginSessionExpirationTime="300" fará com que o objeto Subject expire em 300 segundos. Esse objeto Subject é, então, utilizado para autorizar o pedido, que é mostrado posteriormente.

Um servidor eXtreme Scale utiliza o plug-in do Autenticador para autenticar o objeto Credential. Consulte a Documentação da API do Autenticador para obter mais detalhes.

Este exemplo utiliza uma implementação integrada do eXtreme Scale: KeyStoreLoginAuthenticator, que é propósitos de teste e amostra (um armazenamento de chaves é um registro do usuário simples e não deve ser utilizado para produção). Para obter mais informações, consulte o tópico sobre o plug-in do autenticador em Programação de Autenticação de Cliente.

Este KeyStoreLoginAuthenticator utiliza um KeyStoreLoginModule para autenticar o usuário com o armazenamento de chaves utilizando o módulo de login do JAAS "KeyStoreLogin". O armazenamento de chaves é configurado como uma opção para a classe KeyStoreLoginModule. O exemplo a seguir ilustra o alias keyStoreLogin definido no arquivo de configuração do JAAS og\_jaas.config:

```
KeyStoreLogin{
com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule required
  keyStoreFile="../security/sampleKS.jks" debug = true;
};
```

Os seguintes comandos criam um armazenamento de chaves sampleKS.jks no diretório %OBJECTGRID\_HOME%/security com a senha como sampleKS1. Além disso, três certificados de usuário, representando o usuário administrador, o usuário gerente e o usuário caixa são criados com suas próprias senhas.

- a. Navegue para o diretório-raiz eXtreme Scale.  
cd objectgridRoot
- b. Crie um diretório chamado "security".  
mkdir security
- c. Navegue até o diretório de segurança recém criado.  
cd security
- d. Use o keytool (no diretório javaHOME/bin) para criar um usuário "administrador" com a senha "administrator1" no armazenamento de chaves sampleKS.jks.  
keytool -genkey -v -keystore ./sampleKS.jks -storepass sampleKS1  
-alias administrator -keypass administrator1  
-dname CN=administrator,O=acme,OU=OGSample -validity 10000
- e. Use o keytool (no diretório javaHOME/bin) para criar um usuário "manager" com a senha "manager1" no armazenamento de chaves sampleKS.jks.

```
keytool -genkey -v -keystore ./sampleKS.jks -storepass sampleKS1
-alias manager -keypass manager1
-dname CN=manager,O=acme,OU=OGSample -validity 10000
```

- f. Use o keytool (no diretório javaHOME/bin) para criar um usuário "cashier" com a senha "cashier1" no armazenamento de chaves sampleKS.jks.

```
keytool -genkey -v -keystore ./sampleKS.jks -storepass sampleKS1
-alias cashier -keypass cashier1 -dname CN=cashier,O=acme,OU=OGSample
-validity 10000
```

A configuração de segurança do cliente é definida no arquivo de propriedades do cliente. Utilize o seguinte comando para criar uma cópia no diretório %OBJECTGRID\_HOME%/security:

- a. Mude para o diretório security.
 

```
cd objectgridRoot/security
```
- b. Copie o arquivo sampleClient.properties para o arquivo client.properties.
 

```
cp ../properties/sampleClient.properties client.properties
```

As seguintes propriedades são realçadas no arquivo client.properties no diretório de segurança.

- a. **securityEnabled:** Configurar securityEnabled como true (valor padrão) ativa a segurança do cliente, que inclui autenticação.
- b. **credentialAuthentication:** Configure credentialAuthentication autenticação de credencial.
- c. **transportType:** Configure transportType como TCP/IP, o que significa que nenhum SSL será utilizado.
- d. **singleSignOnEnabled:** Configure-o como false (valor padrão). A conexão única não está disponível.

### 3. Configuração de segurança do servidor

A configuração de segurança do servidor é especificada no arquivo XML descritor de segurança e o arquivo de propriedades de segurança do servidor. O arquivo XML descritor de segurança descreve as propriedades de segurança comuns para todos os servidores (incluindo servidores de catálogo e servidores de contêiner). Um exemplo de propriedade é a configuração do autenticador que representa o registro do usuário e o mecanismo de autenticação.

Aqui está o arquivo security.xml a ser utilizado nesta amostra:

```
<?xml version="1.0" encoding="UTF-8"?>
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config/security">

  <security securityEnabled="true" loginSessionExpirationTime="300" >

    <authenticator className ="com.ibm.websphere.objectgrid.security
      .plugins.builtins.KeyStoreLoginAuthenticator">
      </authenticator>
    </security>

  </securityConfig>
```

- a. **securityEnabled:** Configure para true para ativar a segurança do servidor, incluindo autenticação.
- b. **loginSessionExpirationTime:** Configure o valor como 300 (valor-padrão).
- c. **authenticator:** Inclua a classe do autenticador KeyStoreLoginAuthenticator no arquivo XML do cluster, conforme a seguir:

```
<authenticator className ="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
  </authenticator>
```

- d. **credentialAuthentication:** Configure o atributo credentialAuthentication para Necessário para que o servidor exija autenticação.

Para obter uma explicação mais detalhada no arquivo `security.xml`, consulte Arquivo XML Descritor de Segurança.

Copie o arquivo de propriedades do servidor no diretório de segurança. Neste momento, não é necessário modificar nada neste arquivo.

- a. Navegue até o diretório `security`.  
`cd objectgridRoot/security`
- b. Copie o arquivo `sampleServer.properties` de amostra do `objectGrid` do diretório de propriedades para o novo arquivo `server.properties`.  
`cp ../properties/containerServer.properties server.properties`

Faça as seguintes alterações no arquivo `server.properties`:

- a. **securityEnabled:** Configure o atributo **securityEnabled** como `true`.
- b. **transportType:** Configure o atributo **transportType** como `TCP/IP`, o que significa que nenhum SSL será utilizado.
- c. **secureTokenManagerType:** Configure o atributo **secureTokenManagerType** como `none` para não configurar o gerenciador de tokens seguros.

#### 4. Cliente seguro Conecte o aplicativo cliente ao servidor de maneira segura conforme demonstrado no exemplo a seguir:

```
package com.ibm.websphere.objectgrid.security.sample.guide;

import com.ibm.websphere.objectgrid.ClientClusterContext;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManager;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
import com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration;
import com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory;
import com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator;
import com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator;

public class SecureSimpleApp extends SimpleApp {

    public static void main(String[] args) throws Exception {

        SecureSimpleApp app = new SecureSimpleApp();
        app.run(args);
    }

    /**
     * Get the ObjectGrid
     * @return an ObjectGrid instance
     * @throws Exception
     */
    protected ObjectGrid getObjectGrid(String[] args) throws Exception {
        ObjectGridManager ogManager = ObjectGridManagerFactory.getObjectGridManager();
        ogManager.setTraceFileName("logs/client.log");
        ogManager.setTraceSpecification("ObjectGrid*=all=enabled:ORBRas=all=enabled");

        // Creates a ClientSecurityConfiguration object using the specified file
        ClientSecurityConfiguration clientSC = ClientSecurityConfigurationFactory
            .getClientSecurityConfiguration(args[0]);

        // Creates a CredentialGenerator using the passed-in user and password.
        CredentialGenerator credGen = new UserPasswordCredentialGenerator(args[1], args[2]);
        clientSC.setCredentialGenerator(credGen);

        // Create an ObjectGrid by connecting to the catalog server
        ClientClusterContext ccContext = ogManager.connect("localhost:2809", clientSC, null);
        ObjectGrid og = ogManager.getObjectGrid(ccContext, "accounting");

        return og;
    }
}
```

Existem três coisas diferentes do aplicativo não-seguro:

- a. Objeto `ClientSecurityConfiguration` criado ao transmitir o arquivo `client.properties` configurado.
- b. Criou um `UserPasswordCredentialGenerator` utilizando o ID do usuário e a senha passados.

- c. Conectado com o servidor de catálogo para obter um ObjectGrid a partir ClientClusterContext ao transmitir um objeto ClientSecurityConfiguration.

## 5. Execute o aplicativo

Para iniciar o aplicativo, inicie o servidor de catálogos. Emita as opções da linha de comandos `-clusterFile` e `-serverProps` para transmitir nas propriedades de segurança:

- a. Navegue até o diretório `bin`:

```
cd objectgridRoot/bin
```

- b. Ative o servidor de catálogos:

- **UNIX** **Linux**

```
startOgServer.sh catalogServer -clusterSecurityFile ../security/security.xml
-serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config=../security/og_jaas.config"
```
- **Windows**

```
startOgServer.bat catalogServer -clusterSecurityFile ../security/security.xml
-serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config=../security/og_jaas.config"
```

Em seguida, ative um servidor de contêiner seguro, utilizando o seguinte script:

- a. Navegue até o diretório `bin` novamente:

```
cd objectgridRoot/bin
```

- b. Ative o servidor de contêineres seguro:

- **Linux** **UNIX**

```
startOgServer.sh c0 -objectgridFile ../xml/SimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml
-catalogServiceEndpoints localhost:2809
-serverProps ../security/server.properties
-jvmArgs -Djava.security.auth.login.config=../security/og_jaas.config"
```
- **Windows**

```
startOgServer.bat c0 -objectgridFile ../xml/SimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml
-catalogServiceEndpoints localhost:2809
-serverProps ../security/server.properties
-jvmArgs -Djava.security.auth.login.config=../security/og_jaas.config"
```

O arquivo de propriedades do servidor é passado executando `-serverProps`.

Depois que o servidor for iniciado, inicie o cliente usando o seguinte comando:

- a. `cd objectgridRoot/bin`

- b.

```
java -classpath ../lib/objectgrid.jar;../applib/secsample.jar
com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp
../security/client.properties manager manager1
```

**Linux** Utilize dois pontos (`:`) para o separador de caminho de classe em vez de usar ponto e vírgula (`;`), como no exemplo anterior.

O arquivo `secsample.jar` contém a classe `SimpleApp`.

O `SecureSimpleApp` utiliza três parâmetros que são fornecidos na lista a seguir:

- a. O arquivo `../security/client.properties` está no arquivo de propriedades de segurança do cliente.
- b. `manager` é o ID do usuário.
- c. `manager1` é a senha.

Após executar a classe, o resultado é a seguinte saída:

O nome do cliente para o ID `0001` é `fName 1Name`.



Também possível recorrer ao utilitário **xscmd** para mostrar os tamanhos de mapa da grade "accounting".

- Navegue até o diretório `objectgridRoot/bin`
- Use o comando **xscmd** com a opção de comando `-c showMapSizes` da seguinte forma.
  - **UNIX** **Linux** `xscmd.sh -c showMapSizes -g accounting -m mapSet1 -username manager -password manager1`
  - **Windows** `xscmd.bat -c showMapSizes -g accounting -m mapSet1 -username manager -password manager1`

Agora é possível usar o comando **stopOgServer** para parar o processo do servidor de contêiner ou do serviço de catálogo. Porém, é necessário fornecer um arquivo de configuração de segurança. O arquivo de propriedades do cliente de amostra define as seguintes duas propriedades para gerar uma credencial `userID/password` (`manager/manager1`).

```
credentialGeneratorClass=com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
credentialGeneratorProps=manager manager1
```

Pare o contêiner `c0` com o seguinte comando.

- **UNIX** **Linux** `stopOgServer.sh c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`
- **Windows** `stopOgServer.bat c0 -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`

Se a opção **-clientSecurityFile** não for fornecida, uma exceção será exibida com a seguinte mensagem.

```
>> SERVER (id=39132c79, host=9.10.86.47) TRACE START:
```

```
>> org.omg.CORBA.NO_PERMISSION: O servidor requer uma autenticação de credencial mas não há nenhum contexto de segurança a partir do cliente. Isso geralmente acontece quando o cliente não transmite uma credencial para o servidor.
```

```
vmcid: 0x0
```

```
código secundário: 0
```

```
completed: No
```

Também é possível encerrar o servidor de catálogos usando o seguinte comando. Porém, se você deseja continuar tentando a próxima etapa do tutorial, poderá deixar que o servidor de catálogo permaneça em execução.

- **UNIX** **Linux** `stopOgServer.sh catalogServer -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`
- **Windows** `stopOgServer.bat catalogServer -catalogServiceEndPoints localhost:2809 -clientSecurityFile ..\security\client.properties`

Se você encerrar o servidor de catálogos, a seguinte saída será exibida.

```
CW0BJ2512I: ObjectGrid server catalogServer stopped
```

Agora, você tornou seu sistema parcialmente seguro com sucesso, ativando a autenticação. Você configurou o servidor para conexão no registro do usuário, configurou o cliente para fornecer credenciais do cliente e alterou o arquivo de propriedades do cliente e o arquivo XML do cluster para ativar autenticação.

Se você fornecer uma senha inválida, verá uma exceção que informa que o nome de usuário ou a senha não está correto.

Para obter mais detalhes sobre a autenticação do cliente, consulte "Autenticação de Cliente do Aplicativo" na página 497.



Próxima etapa do tutorial

## Tutorial de Segurança do Java SE - Etapa 3

Após autenticar um cliente, na etapa anterior, é possível fornecer privilégios de segurança por meio dos mecanismos de autorização do eXtreme Scale.

### Antes de Iniciar

Certifique-se de ter concluído o “Tutorial de Segurança do Java SE - Etapa 2” na página 71 antes de continuar com esta tarefa.

### Sobre Esta Tarefa

A etapa anterior deste tutorial demonstrou como ativar a autenticação em uma grade do eXtreme Scale. Como resultado, nenhum cliente não autenticado pode se conectar a seu servidor e submeter pedidos para seu sistema. Entretanto, todo cliente autenticado tem a mesma permissão ou privilégios para o servidor, como de leitura, gravação ou exclusão de dados armazenados nos mapas do ObjectGrid. Os clientes também podem emitir qualquer tipo de consulta. Esta seção demonstra como utilizar a autorização do eXtreme Scale para conceder vários privilégios de usuário autenticado.

Semelhante a vários outros sistemas, o eXtreme Scale adota um mecanismo de autorização baseado em permissão. WebSphere eXtreme Scale tem categorias de permissão diferentes que são representadas por diferentes classes de permissão. Este tópico descreve o MapPermission. Para a categoria completa de permissões, consulte o Programação de Autorização de Cliente.

No WebSphere eXtreme Scale, a classe `com.ibm.websphere.objectgrid.security.MapPermission` representa permissões para os recursos do eXtreme Scale, especificamente os métodos das interfaces `ObjectMap` ou `JavaMap`. O WebSphere eXtreme Scale define as seguintes cadeias de permissões para acesso aos métodos de `ObjectMap` e `JavaMap`:

- `read`: Concede permissão para ler os dados do mapa.
- `write`: Concede permissão para atualizar os dados no mapa.
- `insert`: Concede permissão para inserir os dados no mapa.
- `remove`: Concede permissão para remover os dados do mapa.
- `invalidate`: Concede permissão para invalidar os dados do mapa.
- `all`: Concede todas as permissões para ler, gravar, inserir, remover e invalidar.

A autorização ocorre quando um cliente chama um método de `ObjectMap` ou `JavaMap`. O tempo de execução do eXtreme Scale verifica permissões de mapa diferentes para métodos diferentes. Se as permissões requeridas não forem concedidas ao cliente, isso resultará em um `AccessControlException`.

Este tutorial demonstra como utilizar a autorização Java Authentication and Authorization Service (JAAS) para conceder acessos do mapa de autorização para diferentes usuários.

### Procedimento

1. **Ative a autorização do eXtreme Scale.** Para ativar a autorização no ObjectGrid, você precisa configurar o atributo `securityEnabled` como `true` para esse ObjectGrid específico no arquivo XML. A ativação da segurança no ObjectGrid

significa que você está ativando a autorização. Utilize os seguintes comandos para criar um novo arquivo XML do ObjectGrid com a segurança ativada.

a. Navegue até o diretório xml.

```
cd objectgridRoot/xml
```

b. Copie o arquivo SimpleApp.xml no arquivo SecureSimpleApp.xml.

```
cp SimpleApp.xml SecureSimpleApp.xml
```

c. Abra o arquivo SecureSimpleApp.xml e inclua securityEnabled="true" no nível do ObjectGrid como mostra o seguinte XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="accounting" securityEnabled="true">
      <backingMap name="customer" readOnly="false" copyKey="true"/>
    </objectGrid>
  </objectGrids>
</objectGridConfig>
```

2. **Defina a política de autorização.** Na seção de autenticação pré-cliente, você criou três usuários no keystore: cashier, manager e administrator. Neste exemplo, o usuário "cashier" só tem permissões de leitura a todos os mapas, e o usuário "manager" tem todas as permissões. A autorização JAAS é usada neste exemplo. A autorização JAAS utiliza o arquivo de políticas de autorização para conceder permissões aos principais. O seguinte arquivo é definido no diretório de segurança:

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
  principal javax.security.auth.x500.X500Principal "CN=cashier,O=acme,OU=OGSample" {
  permission com.ibm.websphere.objectgrid.security.MapPermission "accounting.*", "read ";
};

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
  principal javax.security.auth.x500.X500Principal "CN=manager,O=acme,OU=OGSample" {
  permission com.ibm.websphere.objectgrid.security.MapPermission "accounting.*", "all";
};
```

Nota:

- A codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction" é uma URL reservada especialmente para ObjectGrid. Todas as permissões do ObjectGrid concedidas a principals devem utilizar esse código base especial.
- A primeira instrução de concessão concede permissão de mapa "read" ao principal "CN=cashier,O=acme,OU=OGSample", de modo que cashier tenha apenas permissão de leitura ao mapa para todos os mapas na contabilidade do ObjectGrid.
- A segunda instrução de concessão concede permissão de mapa "all" ao principal "CN=manager,O=acme,OU=OGSample", de modo que manager tenha todas as permissões para os mapas na contabilidade do ObjectGrid.

Agora você pode ativar um servidor com uma política de autorização. O arquivo da política de autorização JAAS pode ser configurado utilizando a propriedade -D padrão: -Djava.security.auth.policy=../security/ogAuth.policy

3. **Execute o aplicativo.**

Depois de criar os arquivos acima, será possível executar o aplicativo.

Utilize os seguintes comandos para iniciar o servidor de catálogos. Para obter mais informações sobre como iniciar o serviço de catálogo, consulte o "Iniciando um Serviço de Catálogo Independente" na página 387.

a. Navegue até o diretório bin: cd objectgridRoot/bin

b. Inicie o servidor de catálogos.

- **UNIX** **Linux** `startOgServer.sh catalogServer -clusterSecurityFile ../security/security.xml -serverProps ../security/server.properties -jvmArgs -Djava.security.auth.login.config=../security/og_jaas.config"`
- **Windows** `startOgServer.bat catalogServer -clusterSecurityFile ../security/security.xml -serverProps ../security/server.properties -jvmArgs -Djava.security.auth.login.config=../security/og_jaas.config"`

Os arquivos `security.xml` e `server.properties` foram criados na etapa anterior deste tutorial.

- c. É possível iniciar um servidor de contêiner seguro utilizando o seguinte script. Execute o seguinte script a partir do diretório `bin`:

- **UNIX** **Linux** `# startOgServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml -deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809 -serverProps ../security/server.properties -jvmArgs -Djava.security.auth.login.config=../security/og_jaas.config -Djava.security.auth.policy=../security/og_auth.policy"`
- **Windows** `startOgServer.bat c0 -objectGridFile ../xml/SecureSimpleApp.xml -deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndPoints localhost:2809 -serverProps ../security/server.properties -jvmArgs -Djava.security.auth.login.config=../security/og_jaas.config -Djava.security.auth.policy=../security/og_auth.policy"`

Observe as seguintes diferenças do comando para iniciar o servidor de contêiner anterior:

- Utilize o arquivo `SecureSimpleApp.xml` em vez de o arquivo `SimpleApp.xml`.
- Inclua outro argumento `-Djava.security.auth.policy` para configurar o arquivo de política de autorização JAAS para o processo do servidor de contêiner.

Utilize o mesmo comando da etapa anterior do tutorial:

- a. Navegue até o diretório `bin`.
- b. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp ../security/client.properties manager manager1`

Como o usuário "manager" possui todas as permissões para mapas no ObjectGrid de contabilidade, o aplicativo é executado apropriadamente.

Agora, em vez de utilizar o usuário "manager", utilize o usuário "cashier" para ativar o aplicativo cliente.

- c. Navegue até o diretório `bin`.
- d. `java -classpath ../lib/objectgrid.jar;../applib/secsample.jar com.ibm.ws.objectgrid.security.sample.guide.SecureSimpleApp ../security/client.properties cashier cashier1`

Resulta na seguinte exceção:

```
Exception in thread "P=387313:0=0:CT" com.ibm.websphere.objectgrid.TransactionException:
rolling back transaction, see caused by exception
at com.ibm.ws.objectgrid.SessionImpl.rollbackPMapChanges(SessionImpl.java:1422)
at com.ibm.ws.objectgrid.SessionImpl.commit(SessionImpl.java:1149)
at com.ibm.ws.objectgrid.SessionImpl.mapPostInvoke(SessionImpl.java:2260)
at com.ibm.ws.objectgrid.ObjectMapImpl.update(ObjectMapImpl.java:1062)
at com.ibm.ws.objectgrid.security.sample.guide.SimpleApp.run(SimpleApp.java:42)
at com.ibm.ws.objectgrid.security.sample.guide.SecureSimpleApp.main(SecureSimpleApp.java:27)
Caused by: com.ibm.websphere.objectgrid.ClientServerTransactionCallbackException:
```

```

Client Services - received exception from remote server:
com.ibm.websphere.objectgrid.TransactionException: transaction rolled back, see caused by Throwable
  at com.ibm.ws.objectgrid.client.RemoteTransactionCallbackImpl.processReadWriteResponse(
    RemoteTransactionCallbackImpl.java:1399)
  at com.ibm.ws.objectgrid.client.RemoteTransactionCallbackImpl.processReadWriteRequestAndResponse(
    RemoteTransactionCallbackImpl.java:2333)
  at com.ibm.ws.objectgrid.client.RemoteTransactionCallbackImpl.commit(RemoteTransactionCallbackImpl.java:557)
  at com.ibm.ws.objectgrid.SessionImpl.commit(SessionImpl.java:1079)
  ... 4 more
Caused by: com.ibm.websphere.objectgrid.TransactionException: transaction rolled back, see caused by Throwable
  at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processLogSequence(ServerCoreEventProcessor.java:1133)
  at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processReadWriteTransactionRequest
    (ServerCoreEventProcessor.java:910)
  at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processClientServerRequest(ServerCoreEventProcessor.java:1285)

  at com.ibm.ws.objectgrid.ShardImpl.processMessage(ShardImpl.java:515)
  at com.ibm.ws.objectgrid.partition.IDLShardPOA._invoke(IDLShardPOA.java:154)
  at com.ibm.CORBA.poa.POAServerDelegate.dispatchToServant(POAServerDelegate.java:396)
  at com.ibm.CORBA.poa.POAServerDelegate.internalDispatch(POAServerDelegate.java:331)
  at com.ibm.CORBA.poa.POAServerDelegate.dispatch(POAServerDelegate.java:253)
  at com.ibm.rmi.iiop.ORB.process(ORB.java:503)
  at com.ibm.CORBA.iiop.ORB.process(ORB.java:1553)
  at com.ibm.rmi.iiop.Connection.respondTo(Connection.java:2680)
  at com.ibm.rmi.iiop.Connection.doWork(Connection.java:2554)
  at com.ibm.rmi.iiop.WorkUnitImpl.doWork(WorkUnitImpl.java:62)
  at com.ibm.rmi.iiop.WorkerThread.run(ThreadPoolImpl.java:202)
  at java.lang.Thread.run(Thread.java:803)
Caused by: java.security.AccessControlException: Access denied (
  com.ibm.websphere.objectgrid.security.MapPermission accounting.customer write)
  at java.security.AccessControlContext.checkPermission(AccessControlContext.java:155)
  at com.ibm.ws.objectgrid.security.MapPermissionCheckAction.run(MapPermissionCheckAction.java:141)
  at java.security.AccessController.doPrivileged(AccessController.java:275)
  at javax.security.auth.Subject.doAsPrivileged(Subject.java:727)
  at com.ibm.ws.objectgrid.security.MapAuthorizer$1.run(MapAuthorizer.java:76)
  at java.security.AccessController.doPrivileged(AccessController.java:242)
  at com.ibm.ws.objectgrid.security.MapAuthorizer.check(MapAuthorizer.java:66)
  at com.ibm.ws.objectgrid.security.SecuredObjectMapImpl.checkMapAuthorization(SecuredObjectMapImpl.java:429)
  at com.ibm.ws.objectgrid.security.SecuredObjectMapImpl.update(SecuredObjectMapImpl.java:490)
  at com.ibm.ws.objectgrid.SessionImpl.processLogSequence(SessionImpl.java:1913)
  at com.ibm.ws.objectgrid.SessionImpl.processLogSequence(SessionImpl.java:1805)
  at com.ibm.ws.objectgrid.ServerCoreEventProcessor.processLogSequence(ServerCoreEventProcessor.java:1011)
  ... 14 more

```

Essa exceção ocorre porque o usuário "cashier" não tem permissão de gravação, portanto, ele não pode atualizar o cliente do mapa.

Agora, o seu sistema suporta autorização. É possível definir políticas de autorização para conceder diferentes permissões a diferentes usuários. Para obter mais informações sobre autorização, consulte "Autorização do Aplicativo Cliente" na página 499.

## O que Fazer Depois

Conclua a próxima etapa do tutorial. Consulte "Tutorial de Segurança do Java SE - Etapa 4".

## Tutorial de Segurança do Java SE - Etapa 4

A seguinte etapa explica como uma camada de segurança pode ser ativada para comunicação entre os terminais do ambiente.

### Antes de Iniciar

Certifique-se de ter concluído do "Tutorial de Segurança do Java SE - Etapa 3" na página 77 antes de continuar com esta tarefa.

### Sobre Esta Tarefa

A topologia do eXtreme Scale suporta Transport Layer Security/Secure Sockets Layer (TLS/SSL) para comunicação segura entre terminais do ObjectGrid (cliente,

servidores de contêineres e servidores de catálogos). Esta etapa do tutorial é baseada nas etapas anteriores para ativar a segurança do transporte.

## Procedimento

### 1. Criar chaves TLS/SSL e key stores

Para ativar a segurança do transporte, é necessário criar um key store e um trust store. Este exercício cria apenas um par de chave e trust-store. Estes armazéns são utilizados para clientes do ObjectGrid, servidores de contêineres e servidores de catálogos, e são criados com o JDK keytool.

- *Criar uma chave privada no key store*

```
keytool -genkey -alias ogsample -keystore key.jks -storetype JKS
-keyalg rsa -dname "CN=ogsample, OU=Your Organizational Unit, O=Your
Organization, L=Your City, S=Your State, C=Your Country" -storepass
ogpass -keypass ogpass -validity 3650
```

Utilizando este comando, um key store key.jks é criado com uma chave "ogsample" armazenada nele. Esta key store key.jks será utilizada como o key store SSL.

- *Exportar o certificado público*

```
keytool -export -alias ogsample -keystore key.jks -file temp.key
-storepass ogpass
```

Utilizando este comando, o certificado público da chave "ogsample" é extraído e armazenado no arquivo temp.key.

- *Importar o certificado público do cliente para o trust store*

```
keytool -import -noprompt -alias ogsamplepublic -keystore trust.jks
-file temp.key -storepass ogpass
```

Utilizando este comando, o certificado público foi incluído no key store trust.jks. Este trust.jks é utilizado como o trust store SSL.

### 2. Configurando arquivos de propriedades do ObjectGrid

Neste etapa, é necessário configurar os arquivos de propriedades do ObjectGrid para ativar a segurança do transporte.

Primeiro, copie os arquivos key.jks e trust.jks no diretório objectgridRoot/security.

Configure as seguintes propriedades nos arquivos client.properties e server.properties.

```
transportType=SSL-Required
```

```
alias=ogsample
contextProvider=IBMJSE2
protocol=SSL
keyStoreType=JKS
keyStore=./security/key.jks
keyStorePassword=ogpass
trustStoreType=JKS
trustStore=./security/trust.jks
trustStorePassword=ogpass
```

**transportType:** O valor de transportType é configurado como "SSL-Required", o que significa que o transporte requer SSL. Assim, todos os terminais do ObjectGrid (clientes, servidores de catálogos e servidores de contêineres) devem ter a configuração SSL definida e toda a comunicação de transporte será criptografada.

As outras propriedades são utilizadas para definir as configurações SSL. Consulte o "Transport Layer Security e Secure Sockets Layer" na página 505

para obter uma explicação detalhada. Certifique-se de seguir as seguintes instruções neste tópico para atualizar o arquivo `orb.properties`. Certifique-se de seguir essa página para atualizar o arquivo `orb.properties`. No arquivo `server.properties`, é necessário incluir uma propriedade adicional `clientAuthentication` e configurá-la para `false`. No lado do servidor, não é necessário confiar o cliente.

```
clientAuthentication=false
```

### 3. Execute o aplicativo

Os comandos são os mesmos que os comandos no tópico “Tutorial de Segurança do Java SE - Etapa 3” na página 77.

Utilize os seguintes comandos para iniciar um servidor de catálogos.

a. Navegue até o diretório `bin`: `cd objectgridRoot/bin`

b. Inicie o servidor de catálogos:

- **Linux**      **UNIX**  

```
startOgServer.sh catalogServer -clusterSecurityFile ../security/security.xml  
-serverProps ../security/server.properties -JMXServicePort 11001  
-jvmArgs -Djava.security.auth.login.config=../security/og_jaas.config"
```

- **Windows**  

```
startOgServer.bat catalogServer -clusterSecurityFile ../security/security.xml  
-serverProps ../security/server.properties -JMXServicePort 11001 -jvmArgs  
-Djava.security.auth.login.config=../security/og_jaas.config"
```

Os arquivos `security.xml` e `server.properties` foram criados na página “Tutorial de Segurança do Java SE - Etapa 2” na página 71.

Use a opção **-JMXServicePort** para especificar explicitamente a porta JMX para o servidor. Essa opção é necessária para usar o comando **xscmd**.

Execute um servidor de contêiner ObjectGrid:

c. Navegue até o diretório `bin` novamente: `cd objectgridRoot/bin`

d.

- **Linux**      **UNIX**  

```
startOgServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml  
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndpoints  
localhost:2809 -serverProps ../security/server.properties  
-JMXServicePort 11002 -jvmArgs  
-Djava.security.auth.login.config=../security/og_jaas.config"  
-Djava.security.auth.policy=../security/og_auth.policy"
```

- **Windows**  

```
startOgServer.bat c0 -objectGridFile ../xml/SecureSimpleApp.xml  
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndpoints localhost:2809  
-serverProps ../security/server.properties -JMXServicePort 11002  
-jvmArgs -Djava.security.auth.login.config=../security/og_jaas.config"  
-Djava.security.auth.policy=../security/og_auth.policy"
```

Observe as seguintes diferenças do comando para iniciar o servidor de contêiner anterior:

- Use a propriedade `SecureSimpleApp.xml` em vez dos arquivos `SimpleApp.xml`.
- Incluir outro `-Djava.security.auth.policy` para configurar o arquivo de políticas de autorização ao processo do servidor de contêiner.

Execute o seguinte comando para autenticação de cliente:

a. `cd objectgridRoot/bin`

b.

```
javaHome/java -classpath ../lib/objectgrid.jar;../applib/secsample.jar  
com.ibm.websphere.objectgrid.security.sample.guide.SecureSimpleApp  
../security/client.properties manager manager1
```

Como o usuário "manager" tem permissão para todos os mapas no ObjectGrid de contabilidade, o aplicativo é executado com êxito.



Também é possível usar o utilitário **xscmd** para mostrar os tamanhos de mapa da grade "accounting".

- Navegue até o diretório `objectgridRoot/bin`
- Use o comando **xscmd** para mostrar os tamanhos de mapa:

```
– UNIX Linux
xscmd.sh -c showMapSizes -g accounting -m mapSet1 -jp 11001 -ssl
-ts ..\security\trust.jks -tsp ogpass -tst jks
-user manager -pwd manager1

– Windows
xscmd.bat -c showMapSizes -g accounting -m mapSet1 -jp 11001 -ssl
-ts ..\security\trust.jks -tsp ogpass -tst jks
-user manager -pwd manager1
```

Observe que especificamos a porta JMX do serviço de catálogo usando `-p 11001` aqui.

A seguinte saída será exibida.

```
This administrative utility is provided as a sample only and is not to
be considered a fully supported component of the WebSphere eXtreme Scale product.
Connecting to Catalog service at localhost:1099
***** Displaying Results for Grid - accounting, MapSet - mapSet1 *****
*** Listing Maps for c0 ***
Map Name: customer Partition #: 0 Map Size: 1 Shard Type: Primary
Server Total: 1
Total Domain Count: 1
```

### Executando o aplicativo com um key store incorreto

Se o seu trust store não contiver o certificado público da chave privada no key store, será obtida uma exceção reclamando que a chave pode não ser confiável. Para mostrar isso, crie outro key store, `key2.jks`.

```
keytool -genkey -alias ogsample -keystore key2.jks -storetype JKS
-keyalg rsa -dname "CN=ogsample, OU=Your Organizational Unit, O=Your
Organization, L=Your City, S=Your State, C=Your Country" -storepass
ogpass -keypass ogpass -validity 3650
```

Em seguida, modifique o arquivo `server.properties` para que o `keyStore` aponte para esse novo armazenamento de chaves `key2.jks`:

```
keyStore=../security/key2.jks
```

Execute o seguintes comando para iniciar o servidor de catálogos:

- a. Navegue até o bin: `cd objectgridRoot/bin`
- b. Inicie o servidor de catálogos:

```
Linux UNIX
startOgServer.sh c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndpoints localhost:2809
-serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config=../security/og_jaas.config"
-Djava.security.auth.policy=../security/og_auth.policy"

Windows
startOgServer.bat c0 -objectGridFile ../xml/SecureSimpleApp.xml
-deploymentPolicyFile ../xml/SimpleDP.xml -catalogServiceEndpoints localhost:2809
-serverProps ../security/server.properties -jvmArgs
-Djava.security.auth.login.config=../security/og_jaas.config"
-Djava.security.auth.policy=../security/og_auth.policy"
```

A seguinte exceção será exibida:

```
Caused by: com.ibm.websphere.objectgrid.ObjectGridRPCException:
com.ibm.websphere.objectgrid.ObjectGridRuntimeException:
SSL connection fails and plain socket cannot be used.
```

Por fim, altere o arquivo `server.properties` de volta para usar o arquivo `key.jks`.

---

## Tutorial: Integrar a Segurança do WebSphere eXtreme Scale com o WebSphere Application Server

Este tutorial demonstra como proteger uma implementação do servidor WebSphere eXtreme Scale em um ambiente do WebSphere Application Server.

### Objetivos do aprendizado

Os objetivos do aprendizado para este tutorial são:

- Configurar o WebSphere eXtreme Scale para usar os plug-ins de autenticação do WebSphere Application Server
- Configurar a segurança de transporte do WebSphere eXtreme Scale para usar a configuração do WebSphere Application Server CSIv2
- Usar a autorização do Java Authentication and Authorization Service (JAAS) no WebSphere Application Server
- Usar um módulo de login customizado para a autorização JAAS baseada em grupo
- Usar o utilitário WebSphere eXtreme Scale `xscmd` no ambiente do WebSphere Application Server

### Tempo necessário

Este tutorial leva cerca de 4 horas do início ao fim.

## Introdução: Integre a Segurança do WebSphere eXtreme Scale com o WebSphere Application Server usando os Plug-ins do WebSphere Application Server Authentication

Neste tutorial, integre a segurança do WebSphere eXtreme Scale com o WebSphere Application Server. Primeiro, configure a autenticação com um aplicativo da web simples que usa as credenciais do usuário autenticado a partir do encadeamento atual para se conectar ao ObjectGrid. Em seguida, investigue a criptografia dos dados que são transferidos entre o cliente e o servidor com a segurança da camada de transporte. Para conceder aos usuários diversos níveis de permissões, o Java Authentication and Authorization Service (JAAS) pode ser configurado. Depois de concluir a configuração, será possível usar o utilitário `xscmd` para monitorar suas grades de dados e mapas.

Este tutorial assume que todas os os clientes, servidores de contêineres e servidores de catálogos do WebSphere eXtreme Scale são implementados no ambiente do WebSphere Application Server.

### Objetivos do aprendizado

Os objetivos do aprendizado para este tutorial são:

- Configurar o WebSphere eXtreme Scale para usar os plug-ins de autenticação do WebSphere Application Server
- Configurar a segurança de transporte do WebSphere eXtreme Scale para usar a configuração do WebSphere Application Server CSIv2
- Usar a autorização do Java Authentication and Authorization Service (JAAS) no WebSphere Application Server
- Usar um módulo de login customizado para a autorização JAAS baseada em grupo



- Usar o utilitário WebSphere eXtreme Scale **xscmd** no ambiente do WebSphere Application Server

### **Tempo necessário**

Este tutorial leva cerca de 4 horas do início ao fim.

### **Nível de qualificação**

Intermediário.

### **Público**

Desenvolvedores e administradores interessados na integração de segurança entre o WebSphere eXtreme Scale e o WebSphere Application Server.

### **Requisitos do Sistema e Topologia**

- O WebSphere Application Server Versão 6.1 ou Versão 7.0.0.11 ou posterior
- Atualize o tempo de execução Java para aplicar a seguinte correção: IZ79819: IBMJDK FALHA AO LER A INSTRUÇÃO PRINCIPAL COM ESPAÇO EM BRANCO A PARTIR DO ARQUIVO DE SEGURANÇA

Este tutorial usa quatro servidores de aplicativos WebSphere Application Server e um gerenciador de implementação para demonstrar a amostra.

### **Pré-requisitos**

Um entendimento básico dos seguintes itens é útil antes de iniciar este tutorial:

- Modelo de programação do WebSphere eXtreme Scale
- Conceitos básicos a segurança do WebSphere eXtreme Scale
- Conceitos básicos a segurança do WebSphere Application Server

Para obter informações complementares sobre o WebSphere eXtreme Scale e sobre a integração de segurança do WebSphere Application Server, consulte “Integração de Segurança com o WebSphere Application Server” na página 515.

## **Módulo 1: Preparar o WebSphere Application Server**

Antes de iniciar o tutorial para integração com o WebSphere eXtreme Scale, você deve criar uma configuração de segurança básica no WebSphere Application Server.

### **Objetivos do aprendizado**

Com as lições neste módulo, você aprende como:

- Configurar a segurança do WebSphere Application Server para usar um repositório federado baseado em arquivo interno como um registro de conta do usuário.
- Criar grupos de usuários e usuários.
- Criar clusters para os servidores de aplicativos e do WebSphere eXtreme Scale.

### **Tempo necessário**

Este módulo leva cerca de 60 minutos.

## Lição 1.1: Entendendo a Topologia e Obtendo os Arquivos do Tutorial

Para preparar seu ambiente para o tutorial, você deve configurar a segurança do WebSphere Application Server. Configure a segurança de administração e do aplicativo usando os repositórios federados baseados em arquivo interno como um registro de conta do usuário.

Essa lição o orienta pela topologia de amostra e pelos aplicativos que são usados no tutorial. Para começar a executar o tutorial, você deve fazer download dos aplicativos e colocar os arquivos de configuração nos locais corretos para seu ambiente. É possível fazer download do aplicativo de amostra a partir do Wiki do WebSphere eXtreme Scale.

**Topologia de Amostra do WebSphere Application Server:** Este tutorial guia você na criação de quatro servidores de aplicativos do WebSphere Application Server para demonstrar como usar os aplicativos de amostra com a segurança ativada. Esses servidores de aplicativos são agrupados em dois clusters, cada um com dois servidores:

- **Cluster appCluster:** Hospeda o aplicativo corporativo da amostra EmployeeManagement. Esse cluster possui dois servidores de aplicativos: s1 e s2.
- **Cluster xsCluster:** Hospeda os servidores de contêiner do eXtreme Scale. Esse cluster possui dois servidores de aplicativos: xs1 e xs2.

Nessa topologia de implementação, os servidores de aplicativos s1 e s2 são os servidores de cliente que acessam dados que estão sendo armazenados na grade de dados. Os servidores xs1 e xs2 são os servidores de contêiner que hospedam a grade de dados.

O servidor de catálogos é implementado no processo do gerenciador de implementação por padrão. Este tutorial usa o comportamento padrão. Hospedar o servidor de catálogos no gerenciador de implementação não é uma prática recomendada em um ambiente de produção. Em um ambiente de produção, você deve criar um domínio de serviço de catálogo para definir onde os servidores de catálogos iniciam. Consulte “Criando Domínios do Serviço de Catálogo no WebSphere Application Server” na página 252 para obter mais informações.

**Configuração alternativa:** É possível hospedar todos os servidores de aplicativos em um único cluster, como no cluster appCluster. Com essa configuração, todos os servidores no cluster são ambos servidores de clientes e servidores de contêiner. Este tutorial usa dois clusters para distinguir entre os servidores de aplicativos que hospedam os servidores de clientes e de contêiner.

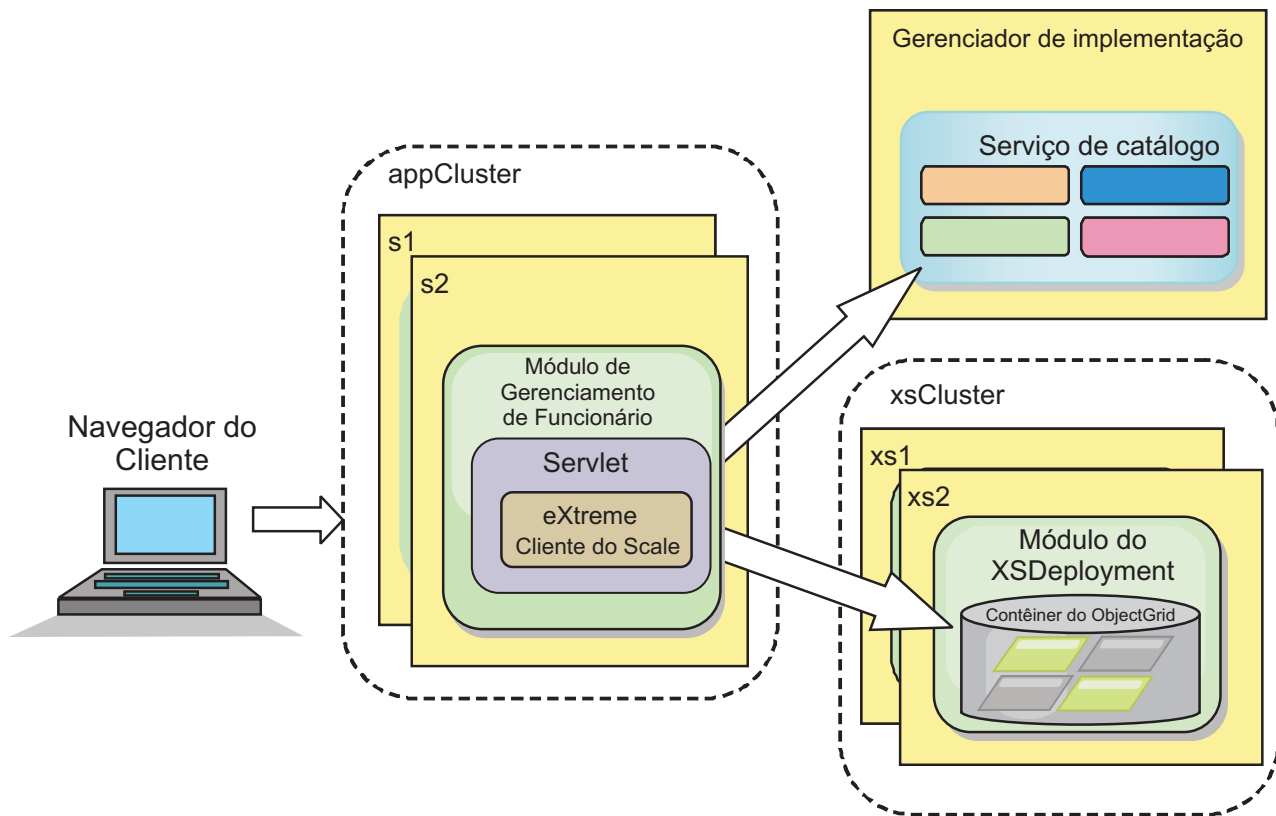


Figura 19. Topologia do Tutorial

**Aplicativos:** Neste tutorial, você está usando dois aplicativos e um arquivo de biblioteca compartilhado:

- **EmployeeManagement.ear:** O aplicativo EmployeeManagement.ear é um aplicativo corporativo simplificado do Java 2 Platform, Enterprise Edition (J2EE). Ele contém um módulo da web para gerenciar os perfis dos funcionários. O módulo da web contém o arquivo management.jsp para exibir, inserir, atualizar e excluir perfis de funcionários que são armazenados nos servidores de contêiner.
- **XSDeployment.ear:** Este aplicativo contém um módulo de aplicativo corporativo sem nenhum artefato do aplicativo. Os objetos de cache são compactados no arquivo EmployeeData.jar. O arquivo EmployeeData.jar é implementado como uma biblioteca compartilhada para o arquivo XSDeployment.ear, de modo que o arquivo XSDeployment.ear possa acessar as classes. O propósito deste aplicativo é compactar os arquivos de configuração do eXtreme Scale. Quando este aplicativo corporativo é iniciado, os arquivos de configuração do eXtreme Scale são automaticamente detectados pelo tempo de execução do eXtreme Scale, portanto, os servidores de contêiner são criados. Esses arquivos de configuração incluem os arquivos objectGrid.xml e objectGridDeployment.xml.
- **EmployeeData.jar:** Este arquivo jar contém uma classe: a classe com.ibm.websphere.sample.xs.data.EmployeeData. Esta classe representa dados do funcionário que são armazenados na grade. Este arquivo Java Archive (JAR) é implementado com os arquivos EmployeeManagement.ear e XSDeployment.ear como uma biblioteca compartilhada.

#### Obter os Arquivos do Tutorial:

1. Faça download dos arquivos WASSecurity.zip e security.zip. É possível fazer download do aplicativo de amostra a partir do Wiki do WebSphere eXtreme Scale.

2. Extraia o arquivo WASecurity.zip em um diretório para visualizar o binário e os artefatos de origem, por exemplo, o diretório /wxs\_samples/. Esse diretório é referido como *samples\_home* para o restante do tutorial. Para obter uma descrição do conteúdo do arquivo WASecurity.zip e de como carregar a origem na área de trabalho do Eclipse, consulte o arquivo README.txt no pacote.
3. Extraia o arquivo security.zip no diretório *samples\_home*. O arquivo security.zip contém os seguintes arquivos de configuração de segurança que são usados neste tutorial:
  - catServer2.props
  - server2.props
  - client2.props
  - securityWAS2.xml
  - xsAuth2.props

### Sobre os Arquivos de Configuração:

Os arquivos objectGrid.xml e objectGridDeployment.xml criam as grades de dados e mapas que armazenam os dados do aplicativo.

Esses arquivos de configuração devem ser nomeados como objectGrid.xml e objectGridDeployment.xml. Quando o servidor de aplicativos é iniciado, o eXtreme Scale detecta estes arquivos no diretório META-INF dos módulos EJB e da web. Se estes arquivos forem localizados, supõe-se que a Java virtual machine (JVM) atua como um servidor de contêiner para as grades de dados definidas nos arquivos de configuração.

### Arquivo objectGrid.xml

O arquivo objectGrid.xml define um ObjectGrid denominado Grid. A grade de dados Grid tem um mapa, o mapa Map1, que armazena o perfil do funcionário para o aplicativo.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">

  <objectGrids>
    <objectGrid name="Grid" txTimeout="15">
      <backingMap name="Map1" />
    </objectGrid>
  </objectGrids>

</objectGridConfig>
```

### Arquivo objectGridDeployment.xml

O arquivo objectGridDeployment.xml especifica como implementar a grade de dados Grid. Quando a grade é implementada, ela tem cinco partições e uma réplica síncrona.

```
<?xml version="1.0" encoding="UTF-8"?>

<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

  <objectgridDeployment objectgridName="Grid">
    <mapSet name="mapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="1" >
      <map ref="Map1"/>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

```
</mapSet>
</objectgridDeployment>

</deploymentPolicy>
```

### Ponto de verificação de lições:

Nesta lição, você aprendeu sobre a topologia do tutorial e incluiu os arquivos de configuração e os aplicativos de amostra no seu ambiente.

Se desejar saber mais sobre como iniciar automaticamente os servidores de contêiner, consulte “Configurando os Aplicativos do WebSphere Application Server para Iniciar Automaticamente os Servidores de Contêiner” na página 270.

## Lição 1.2: Configurar o Ambiente do WebSphere Application Server

Para preparar seu ambiente para o tutorial, você deve configurar a segurança do WebSphere Application Server. Ative a segurança de administração e do aplicativo usando repositórios federados baseados em arquivo internos como um registro de conta do usuário. Em seguida, é possível criar clusters de servidores para hospedar o aplicativo cliente e os servidores de contêiner.

As seguintes etapas foram gravadas usando o WebSphere Application Server Versão 7.0. No entanto, também podem ser aplicados os conceitos que se aplicam às versões anteriores do WebSphere Application Server.

### Configurar a Segurança do WebSphere Application Server:

1. Configure a segurança do WebSphere Application Server.
  - a. No console administrativo do WebSphere Application Server, clique em **Segurança > Segurança Global**.
  - b. Selecione **Repositórios Federados** como o **Repositório de conta do usuário**. Clique em **Configurar como atual**.
  - c. Clique em **Configurar** para acessar o painel **Repositórios Federados**.
  - d. Digite o **Nome do usuário administrativo principal**, por exemplo, admin. Clique em **Aplicar**.
  - e. Quando solicitado, digite a senha do usuário administrativo e clique em **OK**. Salve as alterações.
  - f. Na página **Segurança Global**, verifique se a configuração de **Repositórios Federados** está definida para o registro de conta do usuário atual.
  - g. Selecione os seguintes itens: **Ativar Segurança Administrativa**, **Ativar Segurança do Aplicativo** e **Usar segurança Java 2 para restringir o acesso do aplicativo aos recursos locais**. Clique em **Aplicar** e salve suas alterações.
  - h. Reinicie o gerenciador de implementação e quaisquer servidores de aplicativo em execução.

A Segurança Administrativa do WebSphere Application Server é ativada usando os repositórios federados baseados em arquivo internos como o registro de conta do usuário.

2. Crie dois grupos: adminGroup e operatorGroup.
  - a. Clique em **Usuários e Grupos > Gerenciar grupos > Criar...**
  - b. Digite adminGroup como o nome do grupo. Insira Grupo de Administração como a descrição. Clique em **Criar**.
  - c. Clique em **Criar semelhante**. Digite operatorGroup como o nome do grupo. Insira Grupo de Operador como a descrição. Clique em **Criar**.

- d. Clique em **Fechar**.
3. Crie usuários admin1 e operator1.
  - a. Clique em **Usuários e Grupos > Gerenciar Usuários > Criar...**
  - b. Crie um usuário chamado admin1 com o nome Joe e o sobrenome Doe com a senha do admin1. Clique em **Criar**.
  - c. Crie um segundo usuário. Clique em **Criar semelhante** para criar um usuário chamado operator1 com o nome Jane e sobrenome Doe com a senha do operator1. Clique em **Criar**. Clique em **Fechar**.
4. Inclua usuários nos grupos de usuários. Inclua o usuário admin1 no adminGroup e o usuário operador1 no operatorGroup.
  - a. Clique em **Usuários e Grupos > Gerenciar Usuários**
  - b. Procure por usuários para incluir nos grupos. Clique em **Procurar**, e configure a procura de valor como um asterisco (\*) para exibir todos os usuários.
  - c. No resultado da procura, selecione o usuário admin1 e clique na guia **Grupos**. Clique em **Incluir** para incluir o grupo.
  - d. Procure pelos grupos para localizar os grupos disponíveis. Clique em adminGroup e depois em **Incluir**.
  - e. Repita essas etapas para incluir o usuário operador1 no grupo de usuários operatorGroup.
5. Salve suas mudanças, efetue logout do console administrativo e reinicie o gerenciador de implementação e o agente do nó para ativar as configurações de segurança.

Você ativou a segurança e criou usuários e grupos de usuários que possuem acesso administrativo e de operador à sua configuração do WebSphere Application Server .

#### **Criar Clusters de Servidores:**

Crie dois clusters de servidores em sua configuração do WebSphere Application Server: O cluster appCluster para hospedar o aplicativo de amostra do tutorial e o cluster xsCluster para hospedar a grade de dados.

1. No console administrativo do WebSphere Application Server, abra o painel de clusters. Clique em **Servidores > Clusters > Clusters de Servidores de Aplicativos do WebSphere > Novo**.
2. Digite appCluster como o nome do cluster, deixe a opção **Preferir local** selecionada e clique em **Avançar**.
3. Crie servidores no cluster. Crie um servidor denominado s1, mantendo as opções padrão. Inclua um membro de cluster adicional denominado s2.
4. Conclua as etapas restantes no assistente para criar o cluster. Salve as mudanças.
5. Repita essas etapas para criar o cluster xsCluster. Este cluster possui dois servidores denominados xs1 e xs2.

#### **Ponto de verificação de lições:**

Você ativou a segurança global para a célula do WebSphere Application Server, criou usuários e grupos de usuários e criou clusters para hospedar o aplicativo e a grade e dados.

## Módulo 2: Configurar o WebSphere eXtreme Scale para Usar os Plug-ins de Autenticação do WebSphere Application Server

Após criar a configuração do WebSphere Application Server, será possível integrar a autenticação do WebSphere eXtreme Scale com o WebSphere Application Server.

Quando um cliente do WebSphere eXtreme Scale se conecta a um servidor de contêiner que requer autenticação, o cliente deve fornecer um gerador de credencial representado pela interface `com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator`. Um gerador de credenciais é um `factory` para criar uma credencial do cliente. Uma credencial de cliente pode ser: um par nome de usuário e senha, um ticket do Kerberos, um certificado de cliente ou dados de identificação de cliente em qualquer formato concordado entre o cliente e o servidor. Consulte a Documentação da API de Credencial para obter mais detalhes. Nessa amostra, o cliente do WebSphere eXtreme Scale é o aplicativo da web `EmployeeManagement` que é implementado no cluster `appCluster`. A credencial do cliente é um token de segurança do WebSphere que representa a identidade do usuário da web.

### Objetivos do aprendizado

Com as lições neste módulo, você aprende como:

- Configurar a segurança do servidor de cliente.
- Configurar a segurança do servidor de catálogos.
- Configurar a segurança do servidor de contêiner.
- Instale e execute o aplicativo de amostra.

### Tempo necessário

Este módulo leva cerca de 60 minutos.

### Lição 2.1: Configurar a Segurança do Servidor do Cliente

O arquivo de propriedades do cliente indica a classe de implementação `CredentialGenerator` a ser usada.

Configure o arquivo de propriedades do cliente com a propriedade da JVM **`-Dobjectgrid.client.props`**. O nome do arquivo especificado para esta propriedade é um caminho de arquivo absoluto, como `samples_home/tmp/security/client2.props`. Consulte Arquivo de Propriedades do Cliente para obter mais informações sobre o arquivo de propriedades do cliente.

#### Conteúdo do Arquivo de Propriedades do Cliente:

Este exemplo usa os tokens de segurança do WebSphere Application Server como a credencial do cliente. O arquivo `client2.props` está no diretório `samples_home/security`. O arquivo `client2.props` inclui as seguintes configurações.

##### **securityEnabled**

Quando configurado para `true`, indica que o cliente deve enviar informações de segurança disponíveis para o servidor.

##### **credentialAuthentication**

Quando configurado para `Supported`, indica que o cliente suporta a autenticação de credencial.

##### **credentialGeneratorClass**

Indica a classe `com.ibm.websphere.objectgrid.security.plugins.builtins`.



WSTokenCredentialGenerator para que o cliente recupere os tokens de segurança a partir do encadeamento. Consulte o “Integração de Segurança com o WebSphere Application Server” na página 515 para obter mais informações sobre como os tokens de segurança são recuperados.

### **Configurando o Arquivo de Propriedades do Cliente Usando as Propriedades da Java virtual machine (JVM):**

No console administrativo, conclua as seguintes etapas para ambos servidores s1 e s2 no cluster appCluster. Se uma topologia diferente estiver sendo usada, conclua as seguintes etapas para todos os servidores de aplicativos nos quais o aplicativo EmployeeManagement é implementado.

1. **Servidores > Servidores de aplicativo WebSphere > server\_name > Java and Process Management > Definição de processo > Java Virtual Machine.**
2. Crie a seguinte propriedade JVM genérica para configurar o local do arquivo de propriedades do cliente:  
`-Dobjectgrid.client.props=samples_home/security/client2.props`
3. Clique em **OK** e salve as mudanças.

### **Ponto de verificação de lições:**

Você editou o arquivo de propriedades do cliente e configurou os servidores no cluster appCluster para usar o arquivo de propriedades do cliente. Este arquivo de propriedades indica a classe de implementação CredentialGenerator a ser usada.

### **Lição 2.2: Configurar a Segurança do Servidor de Catálogos**

Um servidor de catálogos contém dois níveis de informações de segurança: As propriedades de segurança que são comuns a todos os servidores do WebSphere eXtreme Scale, incluindo o serviço de catálogo e os servidores de contêiner, e as propriedades de segurança que são específicas ao servidor de catálogos.

As propriedades de segurança que são comuns aos servidores de catálogos e aos servidores de contêiner são configuradas no arquivo descritor XML de segurança. Um exemplo das propriedades comuns é a configuração do autenticador, que representa o registro do usuário e o mecanismo de autenticação. Consulte o Arquivo XML Descritor de Segurança para obter mais informações sobre as propriedades de segurança.

Para configurar o arquivo descritor XML de segurança, crie uma propriedade `-Dobjectgrid.cluster.security.xml.url` no argumento Java virtual machine (JVM). O nome do arquivo especificado para esta propriedade deve estar em um formato de URL, como `file:///samples_home/security/securityWAS2.xml`.

### **Arquivo securityWAS2.xml:**

Neste tutorial, o arquivo `securityWAS2.xml` está no diretório `samples_home/security`. O conteúdo do arquivo `securityWAS2.xml` com os comentários removidos é o seguinte:

```
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config/security">
  <security securityEnabled="true">
    <authenticator
      className="com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenAuthenticator">
    </authenticator>
  </security>
</securityConfig>
```



As seguintes propriedades são definidas no arquivo `securityWAS2.xml`:

#### **securityEnabled**

A propriedade `securityEnabled` é configurada para `true`, indicando que a segurança global do WebSphere eXtreme Scale do servidor de catálogos foi ativada.

#### **autenticador**

O autenticador é configurado como a classe `com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenAuthenticator`. Com essa implementação integrada do plug-in Autenticador, o servidor WebSphere eXtreme Scale pode converter os tokens de segurança em um objeto `Subject`. Consulte o “Integração de Segurança com o WebSphere Application Server” na página 515 para obter mais informações sobre como os tokens de segurança são convertidos.

#### **Arquivo `catServer2.props`:**

O arquivo de propriedades do servidor armazena as propriedades específicas do servidor, dentre elas as propriedades de segurança específicas do servidor. Consulte o Arquivo de Propriedades do Servidor para obter informações adicionais. O arquivo de propriedades do servidor pode ser configurado com a propriedade `-Dobjectgrid.server.props` no argumento da JVM. O valor do nome do arquivo para essa propriedade é especificado como um caminho absoluto, como `samples_home/security/catServer2.props`. Para este tutorial, um arquivo `catServer2.props` está incluído no diretório `samples_home/security`. O conteúdo do arquivo `catServer2.props` com os comentários removidos é o seguinte:

#### **securityEnabled**

A propriedade `securityEnabled` é configurada para `true` para indicar que este servidor de catálogos é um servidor seguro.

#### **credentialAuthentication**

A propriedade `credentialAuthentication` é configurada para `Necessária`, portanto, qualquer cliente que esteja conectado ao servidor precisa fornecer uma credencial.

#### **secureTokenManagerType**

O `secureTokenManagerType` está configurado para `none` para indicar que o segredo da autenticação não é criptografado quando junta os servidores existentes.

#### **authenticationSecret**

A propriedade `authenticationSecret` está configurada para `ObjectGridDefaultSecret`. Esta sequência secreta é usada para juntar o cluster de servidores eXtreme Scale. Quando um servidor junta a grade de dados, ele é desafiado a apresentar a sequência secreta. Se a sequência secreta do servidor de junção corresponder à sequência no servidor de catálogos, o servidor de junção será aceito. Se a cadeia não corresponder, o pedido de junção é rejeitado.

#### **transportType**

A propriedade `transportType` é configurada inicialmente para `TCP/IP`. Posteriormente no tutorial, a segurança de transporte é ativada.

#### **Configurando o arquivo de propriedades de servidor com propriedades JVM:**

Configure o arquivo de propriedades do servidor no servidor do gerenciador de implementação. Se uma topologia diferente da topologia deste tutorial estiver

sendo usada, configure o arquivo de propriedades do servidor para todos os servidores de aplicativos que estiverem sendo usados para hospedar servidores de contêiner.

1. Abra a configuração da Java virtual machine para o servidor. No console administrativo, clique em **Administração do Sistema > Gerenciador de Implementação > Gerenciamento Java e Processos > Definição de Processo > Java Virtual Machine**.
2. Inclua os seguintes argumentos JVM genéricos:  

```
-Dobjectgrid.cluster.security.xml.url=file:///samples_home/security/securityWAS2.xml  
-Dobjectgrid.server.props=samples_home/security/catServer2.props
```
3. Clique em **OK** e salve as mudanças.

#### **Ponto de verificação de lições:**

Você configurou a segurança do servidor de catálogos ao associar os arquivos `securityWAS2.xml` e `catServer2.props` com o gerenciador de implementação, que hospeda o processo do servidor de catálogos na configuração do WebSphere Application Server.

### **Lição 2.3: Configurar a Segurança do Servidor do Contêiner**

Quando um servidor de contêiner se conecta ao serviço de catálogo, o servidor de contêiner obtém todas as configurações de segurança que estiverem definidas no arquivo XML ObjectGrid Security, como a configuração do autenticador, o valor do tempo limite de sessão de login e outras informações de configuração. Um servidor de contêiner também possui suas próprias propriedades de segurança específicas do servidor no arquivo de propriedade do servidor.

Configure o arquivo de propriedades do servidor com a propriedade `-Dobjectgrid.server.props` da Java virtual machine (JVM). O nome do arquivo especificado para esta propriedade é um caminho de arquivo absoluto, como `samples_home/security/server2.props`.

Neste tutorial, os servidores de contêiner são hospedados nos servidores `xs1` e `xs2` no cluster `xsCluster`.

#### **Arquivo `server2.props`:**

O arquivo `server2.props` está no diretório `samples_home/security` sob o diretório `WASSecurity`. As propriedades definidas no arquivo `server2.props` são:

##### **`securityEnabled`**

A propriedade `securityEnabled` é configurada para `true` para indicar que este servidor de contêiner é um servidor seguro.

##### **`credentialAuthentication`**

A propriedade `credentialAuthentication` é configurada para `Necessária`, portanto, qualquer cliente que esteja conectado ao servidor precisa fornecer uma credencial.

##### **`secureTokenManagerType`**

O `secureTokenManagerType` está configurado para `none` para indicar que o segredo da autenticação não é criptografado quando junta os servidores existentes.

##### **`authenticationSecret`**

A propriedade `authenticationSecret` está configurada para `ObjectGridDefaultSecret`. Esta sequência secreta é usada para juntar o cluster de servidores eXtreme Scale. Quando um servidor junta a grade de

dados, ele é desafiado a apresentar a sequência secreta. Se a sequência secreta do servidor de junção corresponder à sequência no servidor de catálogos, o servidor de junção será aceito. Se a cadeia não corresponder, o pedido de junção é rejeitado.

### Configurando o arquivo de propriedades de servidor com propriedades JVM:

Configure o arquivo de propriedades do servidor nos servidores xs1 e xs2. Se a topologia não estiver sendo usada para este tutorial, configure o arquivo de propriedades do servidor para todos os servidores de aplicativos que estiverem sendo usados para hospedar servidores de contêiner.

1. Abra a página da Java virtual machine para o servidor. **Servidores > Servidores de Aplicativos > *server\_name* > Gerenciamento Java e Processos > Definição de Processo > Java Virtual Machine.**
2. Inclua os argumentos da JVM genéricos:  
`-Dobjectgrid.server.props=samples_home/security/server2.props`
3. Clique em **OK** e salve as mudanças.

### Ponto de verificação de lições:

Agora, a autenticação do servidor WebSphere eXtreme Scale está protegida. Ao configurar esta segurança, todos os aplicativos que tentam se conectar com os servidores WebSphere eXtreme Scale são necessárias para fornecer uma credencial. Neste tutorial, o WSTokenAuthenticator é o autenticador. Como resultado, o cliente deverá fornecer um token de segurança do WebSphere Application Server.

## Lição 2.4: Instalar e Executar a Amostra

Depois que a autenticação for configurada, o aplicativo de amostra poderá ser instalado e executado.

### Criando uma Biblioteca Compartilhada para o Arquivo EmployeeData.jar:

1. No console administrativo do WebSphere Application Server, abra a página **Bibliotecas Compartilhadas**. Clique em **Ambiente > Bibliotecas Compartilhadas**.
2. Escolha o escopo **célula**.
3. Crie a biblioteca compartilhada. Clique em **Novo**. Insira `EmployeeManagementLIB` como o **Nome**. Insira o caminho para o arquivo `EmployeeData.jar` no caminho de classe, por exemplo, `samples_home/WASSecurity/EmployeeData.jar`.
4. Dê um clique em **Aplicar**.

### Instalando a Amostra:

1. Instale o arquivo `EmployeeManagement.ear`.
  - a. Para iniciar a instalação, clique em **Aplicativos > Novo Aplicativo > Novo Aplicativo Corporativo**. Escolha o caminho detalhado para instalar o aplicativo.
  - b. Na etapa **Mapear módulos para servidores**, especifique o cluster `appCluster` para instalar o módulo `EmployeeManagementWeb`.
  - c. Na etapa **Mapear bibliotecas compartilhadas**, selecione o módulo `EmployeeManagementWeb`.
  - d. Clique em **Referenciar bibliotecas compartilhadas**. Selecionar a biblioteca `EmployeeManagementLIB`.
  - e. Mapeie a função `webUser` para **Tudo Autenticado na Região do Aplicativo**.
  - f. Clique em **OK**.

Os clientes são executados nos servidores s1 e s2 neste cluster.

2. Instale o arquivo de amostra `XSDeployment.ear`.
  - a. Para iniciar a instalação, clique em **Aplicativos > Novo Aplicativo > Novo Aplicativo Corporativo**. Escolha o caminho detalhado para instalar o aplicativo.
  - b. Na etapa **Mapear módulos para servidores**, especifique o cluster `xsCluster` para instalar o módulo da web `XSDeploymentWeb`.
  - c. Na etapa **Mapear bibliotecas compartilhadas**, selecione o módulo `XSDeploymentWeb`.
  - d. Clique em **Referenciar bibliotecas compartilhadas**. Selecionar a biblioteca `EmployeeManagementLIB`.
  - e. Clique em **OK**.

Os servidores `xs1` e `xs2` neste cluster hospedam os servidores de contêiner.

3. Reinicie o gerenciador de implementação. Quando o gerenciador de implementação é iniciado, o servidor de catálogos também inicia. Se observar o arquivo `SystemOut.log` do gerenciador de implementação, será possível ver a seguinte mensagem indicando que o arquivo de propriedades do servidor `eXtreme Scale` foi carregado.

CW0BJ0913I: Os arquivos de propriedade de servidor foram carregados:  
/wxs\_samples/security/catServer2.props.

4. Reinicie o cluster `xsCluster`. Quando o `xsCluster` é iniciado, o aplicativo `XSDeployment` é iniciado e um servidor de contêiner é iniciado nos servidores `xs1` e `xs2` respectivamente. Se você olhar o arquivo `SystemOut.log` dos servidores `xs1` e `xs2`, a seguinte mensagem que indica que o arquivo de propriedades do servidor foi carregado é exibida:

CW0BJ0913I: Os arquivos de propriedade de servidor foram carregados:  
/wxs\_samples/security/server2.props.

5. Reinicie o cluster `appClusters`. Quando o cluster `appClusters` é iniciado, o aplicativo `EmployeeManagement` também é iniciado. Se observar o arquivo `SystemOut.log` dos servidores `s1` e `s2`, será possível ver a seguinte mensagem indicando que o arquivo de propriedades do cliente foi carregado.

CW0BJ0924I: 0 arquivo de propriedades do cliente {0} foi carregado.

As mensagens de aviso sobre as propriedades `authenticationRetryCount`, `transportType`, e `clientCertificateAuthentication` podem ser ignoradas. Os valores padrão devem ser usados porque os valores não foram especificados no arquivo de propriedades. Se estiver usando o `WebSphere eXtreme Scale Versão 7.0`, uma mensagem `CW0BJ9000I` somente em inglês será exibida para indicar que o arquivo de propriedades do cliente foi carregado. Se a mensagem esperada não for exibida, verifique se a propriedade `-Dobjectgrid.server.props` ou `-Dobjectgrid.client.props` foi configurada no argumento da JVM. Se as propriedades estiverem realmente configuradas, certifique-se de que o traço (-) seja um caractere UTF.

#### Executando o Aplicativo de Amostra:

1. Execute o arquivo `management.jsp`. Em um navegador da web, acesse `http://<your_servername>:<port>/EmployeeManagementWeb/management.jsp`. Por exemplo, é possível usar a seguinte URL: `http://localhost:9080/EmployeeManagementWeb/management.jsp`.
2. Forneça autenticação para o aplicativo. Insira as credenciais do usuário que foram mapeadas para a função `webUser`. Por padrão, essa função do usuário é

mapeada para todos os usuários autenticados. Digite admin1 como seu ID de usuário e admin1 como a senha. Uma página para exibir, incluir, atualizar e excluir funcionários é exibida.

3. Exibir Funcionários Clique em **Exibir um Funcionário**. Insira emp1@acme.com como o endereço de email e clique em **Enviar**. Uma mensagem é exibida informando que o usuário não pode ser localizado.
4. Incluir um funcionário. Clique em **Incluir um Funcionário**. Insira emp1@acme.com como o endereço de email, insira Joe como o nome e insira Doe como o sobrenome. Clique em **Enviar**. Uma mensagem é exibida informando que um funcionário com o endereço emp1@acme.com foi incluído.
5. Exibir o novo funcionário. Clique em **Exibir um Funcionário**. Insira emp1@acme.com como o endereço de email com campos vazios para o nome e sobrenome e clique em **Enviar**. Uma mensagem é exibida informando que o funcionário foi localizado e que os nomes corretos são exibidos nos campos nome e sobrenome.
6. Excluir o funcionário. Clique em **Excluir um funcionário**. Insira emp1@acme.com e clique em **Enviar**. Uma mensagem é exibida informando que o funcionário foi excluído.

#### Ponto de verificação de lições:

Você instalou e executou o aplicativo de amostra. Como este tutorial usa a integração do WebSphere Application Server, não será possível ver o cenário quando um cliente falhar ao se autenticar no servidor eXtreme Scale. Se o usuário se autenticar no WebSphere Application Server com êxito, o eXtreme Scale também é autenticado com êxito.

## Módulo 3: Configurar a Segurança de Transporte

Configure a segurança de transporte para proteger a transferência de dados entre os clientes e servidores na configuração.

No módulo anterior no tutorial, você ativou a autenticação do WebSphere eXtreme Scale. Com a autenticação, qualquer aplicativo que tentar se conectar com o servidor WebSphere eXtreme Scale precisará fornecer uma credencial. Portanto, nenhum cliente não autenticado pode se conectar com o servidor WebSphere eXtreme Scale. Os clientes devem ser um aplicativo autenticado que esteja em execução em uma célula do WebSphere Application Server.

Com a configuração até esse módulo, a transferência de dados entre os clientes no cluster appCluster e os servidores no cluster xsCluster não é criptografada. Essa configuração poderá ser aceitável apenas se seus clusters do WebSphere Application Server estiverem instalados em servidores por trás de um firewall. Entretanto, em alguns cenários, um tráfego não criptografado não é aceito por algumas razões, embora a topologia esteja protegida por um firewall. Por exemplo, uma política de controle pode aplicar o tráfego criptografado. O WebSphere eXtreme Scale suporta a Segurança da Camada de Transporte/Secure Sockets Layer (TLS/SSL) para uma comunicação segura entre os terminais do ObjectGrid, que incluem servidores de cliente, servidores de contêineres e servidores de catálogos.

Nesta implementação de amostra, os clientes e servidores de contêiner do eXtreme Scale estão todos executando no ambiente do WebSphere Application Server. As propriedades do cliente ou do servidor não são necessárias para definir as configurações SSL porque a segurança de transporte do eXtreme Scale é gerenciada pelas configurações de transporte do Common Secure Interoperability Protocol Versão 2 (CSIV2) do WebSphere Application Server. Os servidores do WebSphere

eXtreme Scale usam a mesma instância do Object Request Broker (ORB) como os servidores de aplicativos nos quais eles são executados. Especifique todas as configurações de SSL para servidores de cliente e de contêiner na configuração do WebSphere Application Server usando estas configurações de transporte de CSIv2. O servidor de catálogos possui seus próprios caminhos de transporte proprietários que não usam Internet Inter-ORB Protocol (IIOP) ou Remote Method Invocation (RMI). Devido a estes caminhos de transporte proprietários, o servidor de catálogos não pode ser gerenciado pelas configurações de transporte WebSphere Application Server CSIv2. Portanto, você deve configurar as propriedades SSL no arquivo de propriedades do servidor para o servidor de catálogos.

## Objetivos do aprendizado

Depois de concluir as lições neste módulo, você saberá como:

- Configurar o transporte de entrada e de saída CSIv2
- Incluir propriedades SSL no arquivo de propriedades do servidor de catálogos
- Verificar o arquivo de propriedades do ORB.
- Executar a amostra.

## Tempo necessário

Este módulo leva cerca de 60 minutos.

## Pré-requisitos

Esta etapa do tutorial é construída sobre os módulos anteriores. Conclua os módulos anteriores neste tutorial antes de configurar a segurança do transporte.

### Lição 3.1: Configurar o Transporte de Entrada e de Saída CSIv2

Para configurar o protocolo Segurança da Camada de Transporte/Secure Sockets Layer (TLS/SSL) para o transporte do servidor, configure o transporte de entrada e o transporte de saída do Common Secure Interoperability Protocol Versão 2 (CSIv2) para SSL Necessário para todos os servidores do WebSphere Application Server que hospedam clientes, servidores de catálogo e servidores de contêineres.

Na topologia de exemplo do tutorial, você deve configurar essas propriedades para os servidores de aplicativos s1, s2, xs1 e xs2. As etapas a seguir configuram os transportes de entrada e saída para todos os servidores na configuração.

Configure os transportes de entrada e saída no console administrativo. Certifique-se de que a segurança administrativa esteja ativada.

- **WebSphere Application Server Versão 6.1:** Clique em **Segurança > Administração Segura > Aplicativo.** > **Segurança RMI/IIOP** e altere o tipo de transporte para **SSL Necessário**.
- **WebSphere Application Server Versão 7.0:** Clique em **Segurança > Segurança Global > Segurança RMI/IIOP > Comunicações de entrada CSIv2**. Altere o tipo de transporte na Camada de Transporte CSIv2 como **SSL Necessário**. Repita esta etapa para configurar as comunicações de saída CSIv2.

As configurações de segurança do terminal gerenciadas centralmente podem ser usadas ou os repositórios SSL podem ser configurados. Consulte Configurações do Transporte de Entrada Common Secure Interoperability Versão 2 para obter mais informações.



## Lição 3.2: Incluir Propriedades SSL no Arquivo de Propriedades do Servidor de Catálogos

O servidor de catálogos tem seus próprios caminhos de transporte proprietários que não podem ser gerenciados pelas configurações de transporte do WebSphere Application Server Common Secure Interoperability Protocol Version 2 (CSIV2). Portanto, você deve configurar as propriedades do Secure Sockets Layer (SSL) no arquivo de propriedades do servidor para o servidor de catálogos.

Para configurar a segurança do servidor de catálogos, etapas adicionais são necessárias porque o servidor de catálogos tem seus próprios caminhos de transporte proprietários. Esses caminhos de transporte não podem ser gerenciados pelas configurações de transporte do Application Server CSIV2.

1. Edite as propriedades SSL no arquivo `catServer2.props`. Para configurar a segurança do servidor de catálogos, remova o comentário das seguintes propriedades SSL no arquivo de propriedades do servidor de catálogos. Para este tutorial, as propriedades do servidor de catálogos estão no arquivo `catServer2.props`. Atualize as propriedades `keyStore` e `trustStore` para referenciar o local apropriado em seu ambiente.

```
#alias=default
#contextProvider=IBMJSE2
#protocol=SSL
#keyStoreType=PKCS12
#keyStore=/<WAS_HOME>/IBM/WebSphere/AppServer/profiles/<DMGR_NAME>/config/
cells/<CELL_NAME>/nodes/<NODE_NAME>/key.p12
#keyStorePassword=WebAS
#trustStoreType=PKCS12
#trustStore=/<WAS_HOME>/IBM/WebSphere/AppServer/profiles/<DMGR_NAME>/config/
cells/<CELL_NAME>/nodes/<NODE_NAME>/trust.p12
#trustStorePassword=WebAS
#clientAuthentication=false
```

O arquivo `catServer2.props` usa o keystore e o armazenamento confiável no nível do nó WebSphere Application Server padrão. Se estiver implementando um ambiente de implementação mais complexo, você deverá escolher o keystore e o armazenamento confiável corretos. Em alguns casos, você deve criar um keystore e um armazenamento confiável e importar as chaves dos keystores de outros servidores. Observe que a sequência `WebAS` é a senha padrão do keystore e do armazenamento confiável do WebSphere Application Server. Consulte *Configuração de Certificado Autoassinado Padrão* para obter mais detalhes.

2. No arquivo `catServer2.props`, atualize o valor da propriedade `transportType`. Para as etapas anteriores do tutorial, o valor era definido para TCP/IP. Altere o valor para SSL Necessário.
3. Reinicie o gerenciador de implementação para ativar as mudanças das configurações de segurança do servidor de catálogos.

### Ponto de verificação de lições:

Você configurou as propriedades SSL para o servidor de catálogos.

## Lição 3.3: Executar a Amostra

Reinicie todos os servidores e execute o aplicativo de amostra novamente. Você deve poder executar as etapas sem nenhum problema.

Consulte “*Lição 2.4: Instalar e Executar a Amostra*” na página 95 para obter mais informações sobre como executar e instalar o aplicativo de amostra.

### Ponto de verificação de lições:

Você executou o aplicativo de amostra com a segurança de transporte ativada.

## Módulo 4: Usar a Autorização do Java Authentication and Authorization Service (JAAS) no WebSphere Application Server

Agora que você configurou a autenticação de clientes, é possível configurar ainda mais a autenticação para conceder aos usuários diferentes permissões. Por exemplo, um usuário operador pode apenas visualizar dados, enquanto que um usuário administrador pode executar todas as operações.

Após autenticar um cliente, como no módulo anterior neste tutorial, é possível fornecer privilégios de segurança por meio dos mecanismos de autorização do eXtreme Scale. O módulo anterior deste tutorial demonstrou como ativar a autenticação para uma grade de dados usando a integração com o WebSphere Application Server. Como resultado, nenhum cliente não autenticado pode se conectar aos servidores eXtreme Scale ou submeter solicitações para seu sistema. Entretanto, todo cliente autenticado tem a mesma permissão ou privilégios para o servidor, como de leitura, gravação ou exclusão de dados armazenados nos mapas do ObjectGrid. Os clientes também podem emitir qualquer tipo de consulta.

Esta parte do tutorial demonstra como usar a autorização do eXtreme Scale para fornecer diferentes privilégios para usuários autenticados. O WebSphere eXtreme Scale usa um mecanismo de autorização baseado em permissão. É possível designar categorias de permissão diferentes que são representadas por diferentes classes de permissão. Esse módulo apresenta a classe `MapPermission`. Para obter uma lista de todas as permissões possíveis, consulte Programação de Autorização de Cliente.

No WebSphere eXtreme Scale, a classe `com.ibm.websphere.objectgrid.security.MapPermission` representa permissões para os recursos do eXtreme Scale, especificamente os métodos das interfaces `ObjectMap` ou `JavaMap`. O WebSphere eXtreme Scale define as seguintes cadeias de permissões para acesso aos métodos de `ObjectMap` e `JavaMap`:

- **read**: Concede permissão para ler os dados do mapa.
- **write**: Concede permissão para atualizar os dados no mapa.
- **insert**: Concede permissão para inserir os dados no mapa.
- **remove**: Concede permissão para remover os dados do mapa.
- **invalidate**: Concede permissão para invalidar os dados a partir do mapa.
- **all**: Concede todas as permissões para ler, gravar, inserir, remover e invalidar.

A autorização ocorre quando um cliente do eXtreme Scale usa uma API de acesso a dados, como o `ObjectMap`, o `JavaMap` ou as APIs `EntityManager`. O tempo de execução do eXtreme Scale verifica as permissões do mapa correspondente quando o método é chamado. Se as permissões necessárias não forem concedidas ao cliente, isso resultará em uma exceção `AccessControlException`. Este tutorial demonstra como usar a autorização do Java Authentication and Authorization Service (JAAS) para conceder acesso de mapa de autorização para diferentes usuários.

### Objetivos do aprendizado

Depois de concluir as lições neste módulo, você saberá como:

- Ativar a autorização para WebSphere eXtreme Scale.



- Ativar a autorização baseada no usuário
- Configurar a autorização baseada em grupo.

## Tempo necessário

Este módulo leva cerca de 60 minutos.

## Pré-requisitos

Você deve concluir os módulos anteriores neste tutorial antes de configurar autenticação.

### Lição 4.1: Ativar a Autorização do WebSphere eXtreme Scale

Para ativar a autorização no WebSphere eXtreme Scale, você deve ativar a segurança em um ObjectGrid específico.

Para ativar a autorização no ObjectGrid, você deve configurar o atributo **securityEnabled** para true desse ObjectGrid específico no arquivo XML. Para este tutorial, é possível usar o arquivo XSDeployment\_sec.ear no diretório *samples\_home/WASSecurity*, que já possui a segurança configurada no arquivo *objectGrid.xml*, ou é possível editar o arquivo *objectGrid.xml* existente para ativar a segurança. Esta lição demonstra como editar o arquivo para ativar a segurança.

1. Extraia os arquivos no arquivo XSDeployment.ear e, em seguida, descompacte o arquivo XSDeploymentWeb.war.
2. Abra o arquivo *objectGrid.xml* e configure o atributo `securityEnabled` para true no nível do ObjectGrid. A seguir há um exemplo desse atributo:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
```

```
  <objectGrids>
    <objectGrid name="Grid" securityEnabled="true">
      <backingMap name="Map1" />
    </objectGrid>
  </objectGrids>
```

```
</objectGridConfig>
```

Se diversos ObjectGrids estiverem definidos, esse atributo deverá ser configurado em cada grade de dados.

3. Recompacte os arquivos XSDeploymentWeb.war e XSDeployment.ear para incluir suas mudanças. Nomeie o arquivo XSDeployment\_sec.ear de modo que o pacote original não seja sobrescrito.
4. Desinstale o aplicativo XSDeployment existente e instale o arquivo XSDeployment\_sec.ear. Consulte “Lição 2.4: Instalar e Executar a Amostra” na página 95 para obter mais informações sobre a implementação de aplicativos.

### Ponto de verificação de lições:

Você ativou a segurança no ObjectGrid, que também ativa a autorização na grade de dados.

## Lição 4.2: Ativar Autorização Baseada no Usuário

No módulo de autenticação deste tutorial, dois usuários foram criados: `operator1` e `admin1`. É possível designar permissões diferentes para esses usuários com a autorização Java Authentication and Authorization Service (JAAS).

### Definindo a Política de Autorização do Java Authentication and Authorization Service (JAAS) Usando os Principals do Usuário:

É possível designar permissões para os usuários criados anteriormente. Designe permissões somente leitura ao usuário `operator1` para todos os mapas. Designe ao usuário `admin1` todas as permissões. Use o arquivo de política de autorização JAAS para conceder permissões para os principals.

Edite o arquivo de autorização do JAAS. O arquivo `xsAuth2.policy` está no diretório `samples_home/security`:

```
grant codebase http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction
Principal com.ibm.ws.security.common.auth.WSPPrincipalImpl "defaultWIMFileBasedRealm/operator1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction
Principal com.ibm.ws.security.common.auth.WSPPrincipalImpl "defaultWIMFileBasedRealm/admin1" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};
```

Neste arquivo, o código base `http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction` é uma URL reservada especialmente para o ObjectGrid. Todas as permissões do ObjectGrid concedidas aos principals devem usar esse código base especial. As seguintes permissões são designadas nesse arquivo:

- A primeira instrução de concessão concede permissão `read` para o principal `operator1`. O usuário `operator1` possui permissão somente leitura para o mapa `Map1` da instância `Grid` do ObjectGrid.
- A segunda instrução de concessão concede a permissão de mapa `'all'` para o principal do `admin1`. O usuário `admin1` possui permissão `all` para o mapa `Map1` na instância `Grid` do ObjectGrid.
- O nome do principal é `defaultWIMFileBasedRealm/operator1`, mas não `operator1`. O WebSphere Application Server inclui automaticamente o nome da região no nome do principal quando repositórios federados são usados como o registro de conta do usuário. Ajuste esse valor, se necessário.

### Configurando o Arquivo de Política de Autorização JAAS Usando as Propriedades da JVM:

Use as seguintes etapas para configurar as propriedades JVM para os servidores `xs1` e `xs2`, que estão no cluster `xsCluster`. Se estiver usando uma topologia diferente da topologia de amostra usada neste tutorial, configure o arquivo em todos os servidores de contêiner.

1. No console administrativo, clique em **Servidores > Servidores de Aplicativos > server\_name > Gerenciamento Java e Processos > Definição de Processo > Java virtual machine**
2. Inclua os seguintes argumentos JVM genéricos:  
`-Djava.security.auth.policy=samples_home/security/xsAuth2.policy`
3. Clique em **OK** e salve as mudanças.

### Executando os Aplicativos de Amostra para Testar a Autorização:

É possível usar o mesmo aplicativo de amostra para testar as configurações de autorização. O usuário administrador possui todas as permissões no mapa Map1, inclusive permissões para exibir e incluir funcionários. O usuário operador pode apenas visualizar os funcionários porque esse usuário recebeu permissão somente leitura.

1. Reinicie todos os servidores de aplicativos que estão executando servidores de contêiner.
2. Abra o aplicativo EmployeeManagementWeb. Em um navegador da web, abra `http://<host>:<port>/EmployeeManagementWeb/management.jsp`.
3. Efetue login no aplicativo como administrador. Use o nome de usuário admin1 e a senha admin1.
4. Tente exibir um funcionário. Clique em **Exibir um Funcionário** e procure pelo endereço de e-mail authemp1@acme.com. Uma mensagem exibe que o usuário não pode ser localizado.
5. Incluir um funcionário. Clique em **Incluir um Funcionário**. Inclua o email authemp1@acme.com, o nome Joe e o sobrenome Doe e, em seguida, clique em **Enviar**. Uma mensagem exibe que o funcionário foi incluído.
6. Efetue login como o usuário operador. Abra uma segunda janela do navegador da web e depois abra a seguinte URL: `http://<host>:<port>/EmployeeManagementWeb/management.jsp`. Use o nome de usuário operator1 e a senha operator1.
7. Tente exibir um funcionário. Clique em **Exibir um Funcionário** e procure pelo endereço de e-mail authemp1@acme.com. O funcionário é exibido.
8. Incluir um funcionário. Clique em **Incluir um Funcionário**. Inclua o email authemp2@acme.com, o nome Joe e o sobrenome Doe e, em seguida, clique em **Enviar**. A mensagem a seguir é exibida:

Ocorre uma exceção ao Incluir o funcionário. Consulte abaixo para obter mensagens de exceção detalhadas.

A seguinte exceção está na cadeia de exceções:

```
java.security.AccessControlException: Acesso negado
(com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
```

Esta mensagem é exibida porque o usuário operator1 não tem permissão para inserir dados no mapa Map1.

Se você estiver executando com uma versão do WebSphere Application Server anterior à Versão 7.0.0.11, um erro `java.lang.StackOverflowError` será exibido no servidor de contêiner. Este erro é causado por um problema com o IBM Developer Kit. O problema é corrigido no IBM Developer Kit fornecido com o WebSphere Application Server Versão 7.0.0.11 e posterior.

#### Ponto de verificação de lições:

Nesta lição, você configurou a autorização ao designar permissões para usuários específicos.

### Lição 4.3: Configurar Autorização Baseada em Grupo

Na lição anterior, você designou autorização baseada em usuário individual com os principais do usuário na política de autorização do Java Authentication and Authorization Service. No entanto, quando você tiver centenas ou milhares de usuários, use a autorização baseado em grupo, que autoriza o acesso baseado em grupos em vez de em usuários individuais.

Infelizmente, o objeto Subject que é autenticado no WebSphere Application Server contém apenas um principal de usuário. Este objeto não contém um principal de grupo. É possível incluir um módulo de login customizado para preencher o principal do grupo no objeto Subject.

Para este tutorial, o módulo de login customizado é denominado `com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule`. O módulo está no arquivo `groupLM.jar`. Coloque o arquivo JAR no diretório `WAS-INSTALL/lib/ext`.

O `WASAddGroupLoginModule` recupera a credencial do grupo público a partir do sujeito WebSphere Application Server e cria um principal de Grupo, `com.ibm.websphere.samples.objectgrid.security.WSGroupPrincipal`, para representar o grupo. Esse principal de grupo pode então ser usado para autorização do grupo. Os grupos são definidos no arquivo `xsAuthGroup2.policy` :

```
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal com.ibm.websphere.sample.xs.security.WSGroupPrincipal
    "defaultWIMFileBasedRealm/cn=operatorGroup,o=defaultWIMFileBasedRealm" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal com.ibm.websphere.sample.xs.security.WSGroupPrincipal
    "defaultWIMFileBasedRealm/cn=adminGroup,o=defaultWIMFileBasedRealm" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};
```

O nome do grupo é o `WSGroupPrincipal`, que representa o grupo.

### Incluindo o Módulo de Login Customizado:

O módulo de login customizado deve ser incluído em cada uma das seguintes entradas do módulo de login do sistema: Se estiver usando Lightweight Third Party Authentication (LTPA), inclua a entrada nos módulos de login `RMI_INBOUND`. O LTPA é o mecanismo de autenticação padrão para o WebSphere Application Server Versão 7.0. Para uma configuração do WebSphere Application Server Network Deployment, é necessário definir apenas as entradas de configuração do mecanismo de autenticação de LTPA.

Use as seguintes etapas para configurar o módulo de login `com.ibm.websphere.samples.objectgrid.security.lm.WASAddGroupLoginModule` fornecido:

1. No console administrativo, clique em **Segurança > Segurança Global > Java Authentication and Authorization Service > Logins do Sistema > login\_module\_name > Módulos de Login do JAAS > Novo**.
2. Insira o nome da classe como `com.ibm.websphere.sample.xs.security.lm.WASAddGroupLoginModule`.
3. Opcional: Inclua uma propriedade debug e configure o valor como `true`.
4. Clique em **Aplicar** para incluir o novo módulo na lista de módulos de login.

### Configurando o Arquivo de Propriedades JAAS Authorization Usando as Propriedades da JVM:

No console administrativo, execute as seguintes etapas para os servidores `xs1` e `xs2` no `xsCluster`. Se uma topologia de implementação diferente for usada, execute as etapas a seguir para os servidores de aplicativos que hospedam os servidores de contêiner.

1. No console administrativo, clique em **Servidores > Servidores de Aplicativos > *server\_name* > Gerenciamento Java e Processos > Definição de Processo > Java virtual machine**
2. Insira os seguintes argumentos Genéricos de JVM ou substitua a entrada-Djava.security.auth.policy com o seguinte texto:  
-Djava.security.auth.policy=*samples\_home*/security/xsAuthGroup2.policy
3. Clique em **OK** e salve as mudanças.

#### Testando a Autorização do Grupo com o Aplicativo de Amostra:

É possível testar se a autorização de grupo foi configurada pelo módulo de login com o aplicativo de amostra.

1. Reinicie os servidores de contêiner. Para este tutorial, os servidores de contêiner são os servidores de xs1 e xs2.
2. Efetue login no aplicativo de amostra. Em um navegador da web, abra `http://<host>:<port>/EmployeeManagementWeb/management.jsp` e efetue login com o nome de usuário `admin1` e senha `admin1`.
3. Exibir um funcionário. Clique em **Exibir um Funcionário** e procure pelo endereço de email `authemp2@acme.com`. Uma mensagem exibe que o usuário não pode ser localizado.
4. Incluir um funcionário. Clique em **Incluir um Funcionário**. Inclua o email `authemp2@acme.com`, o nome `Joe` e o sobrenome `Doe` e, em seguida, clique em **Enviar**. Uma mensagem exibe que o funcionário foi incluído.
5. Efetue login como o usuário operador. Abra uma segunda janela do navegador da web e depois a seguinte URL: `http://<host>:<port>/EmployeeManagementWeb/management.jsp`. Use o nome de usuário `operator1` e a senha `operator1`.
6. Tente exibir um funcionário. Clique em **Exibir um Funcionário** e procure pelo endereço de email `authemp2@acme.com`. O funcionário é exibido.
7. Incluir um funcionário. Clique em **Incluir um Funcionário**. Inclua o email `authemp3@acme.com`, o nome `Joe` e o sobrenome `Doe` e, em seguida, clique em **Enviar**. A mensagem a seguir é exibida:

Ocorre uma exceção ao Incluir o funcionário. Consulte abaixo para obter mensagens de exceção detalhadas.

A seguinte exceção está na cadeia de exceções:

```
java.security.AccessControlException: Acesso negado
(com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
```

Esta mensagem é exibida porque o usuário `operator` não tem permissão para inserir dados no mapa `Map1`.

#### Ponto de verificação de lições:

Você configurou grupos para simplificar a designação de permissão para os usuários de seu aplicativo.

## Módulo 5: Use a Ferramenta `xscmd` para Monitorar as Grades de Dados e Mapas

É possível usar a ferramenta `xscmd` para mostrar as grades de dados e os tamanhos de mapa primários da grade de dados `Grid`. A ferramenta `xscmd` usa o `MBean` para consultar todos os artefatos grade de dados, tais como `shards` primários, `shards` de réplica, servidores de contêiner, tamanhos de mapa, e assim por diante.

Neste tutorial, os servidores de contêiner e de catálogos estão em execução nos servidores de aplicativos WebSphere Application Server. O tempo de execução do WebSphere eXtreme Scale registra os beans gerenciados (MBean) com o servidor MBean que é criado pelo tempo de execução do WebSphere Application Server. A segurança usada pela ferramenta **xscmd** é fornecida pela segurança MBean do WebSphere Application Server. Portanto, uma configuração de segurança específica do WebSphere eXtreme Scale não é necessária.

1. Usando uma ferramenta de linha de comandos, abra o diretório `DMGR_PROFILE/bin`.
2. Executar a ferramenta **xscmd**.

Use o comando `-c listObjectGridPlacement -sf P` para listar o posicionamento dos shards primários.

```
xscmd.sh -g Grid -ms mapSet -c showPlacement -sf P
```

#### Windows

```
xscmd.bat -g Grid -ms mapSet -c showPlacement -sf P
```

Antes de poder visualizar a saída, será solicitado efetuar login com seu ID e senha do WebSphere Application Server.

### Ponto de verificação de lições

Você usou a ferramenta **xscmd** no WebSphere Application Server.

---

## Tutorial: Integrar a Segurança do WebSphere eXtreme Scale em um Ambiente Misto com um Autenticador Externo

Este tutorial demonstra como proteger os servidores do WebSphere eXtreme Scale que são parcialmente implementados em um ambiente do WebSphere Application Server.

Na implementação deste tutorial, os servidores de contêiner são implementados no WebSphere Application Server. O servidor de catálogos é implementado como um servidor independente e é iniciado em um ambiente do Java Standard Edition (Java SE).

Como o servidor de catálogos não é implementado no WebSphere Application Server, não é possível usar os plug-ins do WebSphere Application Server Authentication. Para obter mais informações sobre o processo de configuração dos plug-ins do WebSphere Application Server Authentication, consulte “Tutorial: Integrar a Segurança do WebSphere eXtreme Scale com o WebSphere Application Server” na página 84. Neste tutorial, um autenticador diferente é necessário para a autenticação do servidor de catálogos. Configure um autenticador do keystore para autenticar os clientes.

### Objetivos do aprendizado

Os objetivos do aprendizado para este tutorial são:

- Configurar o WebSphere eXtreme Scale para usar o plug-in `KeyStoreLoginAuthenticator`.
- Configurar a segurança de transporte do WebSphere eXtreme Scale para usar a configuração do WebSphere Application Server CSIv2 e o arquivo de propriedades do WebSphere eXtreme Scale.
- Usar a autorização do Java Authentication and Authorization Service (JAAS) no WebSphere Application Server

- Usar o utilitário `xscmd` para monitorar as grades de dados e mapas criados no tutorial.

### **Tempo necessário**

Este tutorial leva cerca de 4 horas do início ao fim.

## **Introdução: Segurança em um Ambiente Misto**

Neste tutorial, integre a segurança do WebSphere eXtreme Scale em um ambiente misto. Os servidores de contêiner são executados no WebSphere Application Server e o serviço de catálogos é executado no modo independente. Como o servidor de catálogos está no modo independente, você deve configurar um autenticador externo.

**Importante:** Se ambos servidor de contêiner e o servidor de catálogos estiverem em execução no WebSphere Application Server, será possível usar os plug-ins do WebSphere Application Server Authentication ou um autenticador externo. Para obter mais informações sobre como usar os plug-ins do WebSphere Application Server Authentication, consulte “Tutorial: Integrar a Segurança do WebSphere eXtreme Scale com o WebSphere Application Server” na página 84.

### **Objetivos do aprendizado**

Os objetivos do aprendizado para este tutorial são:

- Configurar o WebSphere eXtreme Scale para usar o plug-in KeyStoreLoginAuthenticator.
- Configurar a segurança de transporte do WebSphere eXtreme Scale para usar a configuração do WebSphere Application Server CSIv2 e o arquivo de propriedades do WebSphere eXtreme Scale.
- Usar a autorização do Java Authentication and Authorization Service (JAAS) no WebSphere Application Server
- Usar o utilitário `xscmd` para monitorar as grades de dados e mapas criados no tutorial.

### **Tempo necessário**

Este tutorial leva cerca de 4 horas do início ao fim.

### **Nível de qualificação**

Intermediário.

### **Público**

Desenvolvedores e administradores que estão interessados na integração de segurança entre o WebSphere eXtreme Scale e o WebSphere Application Server e que configuram autenticadores externos.

### **Requisitos do Sistema**

- O WebSphere Application Server Versão 6.1 ou Versão 7.0.0.11 ou posterior com as seguintes correções aplicadas: Correção Temporária PM20613 e Correção Temporária PM15818.



- O servidor de catálogos deve estar em execução em uma instalação independente e não em uma instalação integrada ao WebSphere Application Server.
- Atualize o tempo de execução Java para aplicar a seguinte correção: IZ79819: IBMJDK FALHA AO LER A INSTRUÇÃO PRINCIPAL COM ESPAÇO EM BRANCO A PARTIR DO ARQUIVO DE SEGURANÇA
- O nó independente que executa o serviço de catálogo deve usar o IBM Software Development Kit Versão 1.6 J9. Esse SDK é incluído na instalação do WebSphere Application Server. O nó do servidor de catálogos deve ser uma instalação independente porque não é possível executar o comando **startOgServer** dentro de uma instalação do WebSphere eXtreme Scale no WebSphere Application Server.

Este tutorial usa quatro servidores de aplicativos WebSphere Application Server e um gerenciador de implementação para demonstrar a amostra.

## Pré-requisitos

Um entendimento básico dos seguintes itens é útil antes de iniciar este tutorial:

- Modelo de programação do WebSphere eXtreme Scale
- Conceitos básicos a segurança do WebSphere eXtreme Scale
- Conceitos básicos a segurança do WebSphere Application Server

Para obter informações complementares sobre o WebSphere eXtreme Scale e sobre a integração de segurança do WebSphere Application Server, consulte “Integração de Segurança com o WebSphere Application Server” na página 515.

## Módulo 1: Preparar o Ambiente do WebSphere Application Server e Independente Combinado

Antes de iniciar o tutorial, você deve criar uma topologia básica que inclui servidores de contêiner que são executados no WebSphere Application Server. Neste tutorial, os servidores de catálogos são executados no modo independente.

### Objetivos do aprendizado

Com as lições neste módulo, você aprende como:

- Entender a topologia combinada e os arquivos que são necessários para o tutorial
- Configurar o WebSphere Application Server para executar os servidores de contêiner.

### Tempo necessário

Este módulo leva cerca de 60 minutos.

### Lição 1.1: Entendendo a Topologia e Obtendo os Arquivos do Tutorial

Para preparar seu ambiente para o tutorial, você deve configurar os servidores de catálogos e de contêiner para a topologia.

Essa lição o orienta pela topologia de amostra e pelos aplicativos que são usados no tutorial. Para começar a executar o tutorial, você deve fazer download dos



aplicativos e colocar os arquivos de configuração nos locais corretos para seu ambiente. É possível fazer download do aplicativo de amostra a partir do Wiki do WebSphere eXtreme Scale.

**Topologia:** Neste tutorial, os seguintes clusters são criados na célula WebSphere Application Server:

- **Cluster appCluster:** Hospeda o aplicativo corporativo da amostra EmployeeManagement. Esse cluster possui dois servidores de aplicativos: s1 e s2.
- **Cluster xsCluster:** Hospeda os servidores de contêiner do eXtreme Scale. Esse cluster possui dois servidores de aplicativos: xs1 e xs2.

Nessa topologia de implementação, os servidores de aplicativos s1 e s2 são os servidores de cliente que acessam dados que estão sendo armazenados na grade de dados. Os servidores xs1 e xs2 são os servidores de contêiner que hospedam a grade de dados.

**Configuração alternativa:** É possível hospedar todos os servidores de aplicativos em um único cluster, como no cluster appCluster. Com essa configuração, todos os servidores no cluster são ambos servidores de clientes e servidores de contêiner. Este tutorial usa dois clusters para distinguir entre os servidores de aplicativos que hospedam os servidores de clientes e de contêiner.

Neste tutorial, um domínio de serviço de catálogo é configurado, que consiste de um servidor remoto que não está na célula do WebSphere Application Server. Esta configuração não é a padrão, o que resulta nos servidores de catálogos serem executados no gerenciador de implementação e outros processos na célula do WebSphere Application Server . Consulte o “Criando Domínios do Serviço de Catálogo no WebSphere Application Server” na página 252 para obter mais informações sobre como criar um domínio de serviço de catálogo composto por servidores remotos.

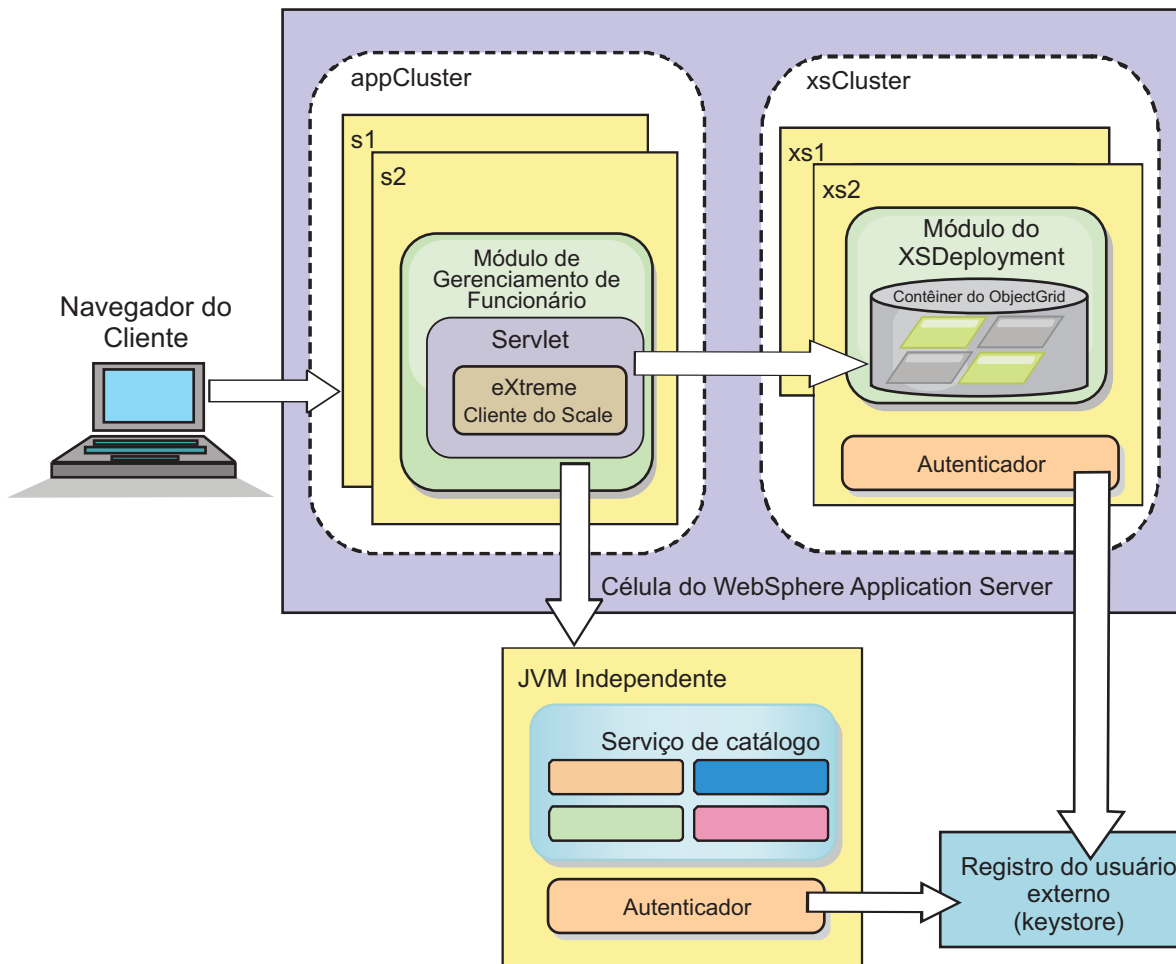


Figura 20. Topologia do Tutorial

**Aplicativos:** Neste tutorial, você está usando dois aplicativos e um arquivo de biblioteca compartilhado:

- **EmployeeManagement.ear:** O aplicativo EmployeeManagement.ear é um aplicativo corporativo simplificado do Java 2 Platform, Enterprise Edition (J2EE). Ele contém um módulo da web para gerenciar os perfis dos funcionários. O módulo da web contém o arquivo management.jsp para exibir, inserir, atualizar e excluir perfis de funcionários que são armazenados nos servidores de contêiner.
- **XSDeployment.ear:** Este aplicativo contém um módulo de aplicativo corporativo sem nenhum artefato do aplicativo. Os objetos de cache são compactados no arquivo EmployeeData.jar. O arquivo EmployeeData.jar é implementado como uma biblioteca compartilhada para o arquivo XSDeployment.ear, de modo que o arquivo XSDeployment.ear possa acessar as classes. O propósito deste aplicativo é compactar o eXtreme Scale no arquivo de configuração e no arquivo de propriedades. Quando este aplicativo corporativo é iniciado, os arquivos de configuração do eXtreme Scale são automaticamente detectados pelo tempo de execução do eXtreme Scale, portanto, os servidores de contêiner são criados. Esses arquivos de configuração incluem os arquivos objectGrid.xml e objectGridDeployment.xml.
- **EmployeeData.jar:** Este arquivo jar contém uma classe: a classe com.ibm.websphere.sample.xs.data.EmployeeData. Esta classe representa dados

do funcionário que são armazenados na grade. Este arquivo Java Archive (JAR) é implementado com os arquivos `EmployeeManagement.ear` e `XSDeployment.ear` como uma biblioteca compartilhada.

### Obter os Arquivos do Tutorial:

1. Faça download dos arquivos `WASSecurity.zip` e `security_extauth.zip` a partir do Wiki do WebSphere eXtreme Scale.
2. Extraia o arquivo `WASSecurity.zip` em um diretório para visualizar os artefatos binários e de origem, por exemplo, em um diretório `wxs_samples/`. Esse diretório é referido como *samples\_home* para o restante do tutorial. Consulte o arquivo `README.txt` no pacote para obter uma descrição do conteúdo e como carregar a origem na área de trabalho do Eclipse. Os seguintes arquivos de configuração do ObjectGrid estão no diretório META-INF:
  - `objectGrid.xml`
  - `objectGridDeployment.xml`
3. Crie um diretório para armazenar os arquivos de propriedade que são usados para proteger este ambiente. Por exemplo, você pode criar o diretório `/opt/wxs/security`.
4. Extraia o arquivo `security_extauth.zip` para o *samples\_home*. O arquivo `security_extauth.zip` contém os seguintes arquivos de configuração de segurança que são usados neste tutorial. Os arquivos de configuração são:
  - `catServer3.props`
  - `server3.props`
  - `client3.props`
  - `security3.xml`
  - `xsAuth3.props`
  - `xsjaas3.config`
  - `sampleKS3.jks`

### Sobre os Arquivos de Configuração:

Os arquivos `objectGrid.xml` e `objectGridDeployment.xml` criam as grades de dados e mapas que armazenam os dados do aplicativo.

Esses arquivos de configuração devem ser nomeados como `objectGrid.xml` e `objectGridDeployment.xml`. Quando o servidor de aplicativos é iniciado, o eXtreme Scale detecta estes arquivos no diretório META-INF dos módulos EJB e da web. Se estes arquivos forem localizados, supõe-se que a Java virtual machine (JVM) atua como um servidor de contêiner para as grades de dados definidas nos arquivos de configuração.

### Arquivo `objectGrid.xml`

O arquivo `objectGrid.xml` define um ObjectGrid denominado Grid. A grade de dados Grid tem um mapa, o mapa Map1, que armazena o perfil do funcionário para o aplicativo.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="Grid" txTimeout="15">
      <backingMap name="Map1" />
    </objectGrid>
  </objectGrids>
</objectGridConfig>
```

```

    </objectGrid>
  </objectGrids>
</objectGridConfig>

```

### Arquivo objectGridDeployment.xml

O arquivo objectGridDeployment.xml especifica como implementar a grade de dados Grid. Quando a grade é implementada, ela tem cinco partições e uma réplica síncrona.

```

<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="Grid">
    <mapSet name="mapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="1" >
      <map ref="Map1"/>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>

```

### Ponto de verificação de lições:

Nesta lição, você aprendeu sobre a topologia do tutorial e incluiu os arquivos de configuração e os aplicativos de amostra no seu ambiente.

## Lição 1.2: Configurar o Ambiente do WebSphere Application Server

Para preparar seu ambiente para o tutorial, você deve configurar a segurança do WebSphere Application Server. Ative a segurança de administração e do aplicativo usando repositórios federados baseados em arquivo internos como um registro de conta do usuário. Em seguida, é possível criar clusters de servidores para hospedar o aplicativo cliente e os servidores de contêiner. Também é necessário criar e iniciar os servidores de catálogos.

As seguintes etapas foram gravadas usando o WebSphere Application Server Versão 7.0. No entanto, também podem ser aplicados os conceitos que se aplicam às versões anteriores do WebSphere Application Server.

### Configurar a Segurança do WebSphere Application Server:

Crie e aumente perfis para o gerenciador de implementação e os nós com o WebSphere eXtreme Scale. Consulte “Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client com WebSphere Application Server” na página 161 para obter mais informações.

Configure a segurança do WebSphere Application Server.

1. No console administrativo do WebSphere Application Server, clique em **Segurança > Segurança Global**.
2. Selecione **Repositórios Federados** como o **Repositório de conta do usuário**. Clique em **Configurar como atual**.
3. Clique em **Configurar** para acessar o painel **Repositórios Federados**.
4. Digite o **Nome do usuário administrativo principal**, por exemplo, admin. Clique em **Aplicar**.
5. Quando solicitado, digite a senha do usuário administrativo e clique em **OK**. Salve as alterações.

6. Na página **Segurança Global**, verifique se a configuração de **Repositórios Federados** está definida para o registro de conta do usuário atual.
7. Selecione os seguintes itens: **Ativar Segurança Administrativa**, **Ativar Segurança do Aplicativo** e **Usar segurança Java 2 para restringir o acesso do aplicativo aos recursos locais**. Clique em **Aplicar** e salve suas alterações.
8. Reinicie o gerenciador de implementação e quaisquer servidores de aplicativo em execução.

A Segurança Administrativa do WebSphere Application Server é ativada usando os repositórios federados baseados em arquivo internos como o registro de conta do usuário.

#### **Criar Clusters de Servidores:**

Crie dois clusters de servidores em sua configuração do WebSphere Application Server: O cluster appCluster para hospedar o aplicativo de amostra do tutorial e o cluster xsCluster para hospedar a grade de dados.

1. No console administrativo do WebSphere Application Server, abra o painel de clusters. Clique em **Servidores > Clusters > Clusters de Servidores de Aplicativos do WebSphere > Novo**.
2. Digite appCluster como o nome do cluster, deixe a opção **Preferir local** selecionada e clique em **Avançar**.
3. Crie servidores no cluster. Crie um servidor denominado s1, mantendo as opções padrão. Inclua um membro de cluster adicional denominado s2.
4. Conclua as etapas restantes no assistente para criar o cluster. Salve as mudanças.
5. Repita essas etapas para criar o cluster xsCluster. Este cluster possui dois servidores denominados xs1 e xs2.

#### **Criar um Domínio de Serviço de Catálogo:**

Depois de configurar o cluster de servidores e a segurança, você deve definir onde os servidores de catálogos serão iniciados.

#### **Defina um domínio de serviço de catálogo no WebSphere eXtreme Scale**

1. No console administrativo do WebSphere Application Server, clique em **Administração do sistema > WebSphere eXtreme Scale > Domínios do serviço de catálogo**.
2. Crie o domínio do serviço de catálogo. Clique em **Novo**. Crie o domínio de serviço de catálogo com o nome catalogService1 e ative o domínio de serviço de catálogo como o padrão.
3. Inclua servidores remotos no domínio de serviço de catálogo. Selecione **Servidor Remoto**. Forneça o nome do host no qual o servidor de catálogos está em execução. Utilize o valor da porta do listener de 16809 para este exemplo.
4. Clique em **OK** e salve as mudanças.

#### **Ponto de verificação de lições:**

Você ativou a segurança no WebSphere Application Server e criou a topologia do servidor para o WebSphere eXtreme Scale.

## Módulo 2: Configurar a Autenticação do WebSphere eXtreme Scale em um Ambiente Misto

Ao configurar autenticação, é possível determinar de maneira confiável a identidade do solicitante. O WebSphere eXtreme Scale suporta autenticação cliente-para-servidor e servidor-para-servidor.

### Fluxo de Autenticação

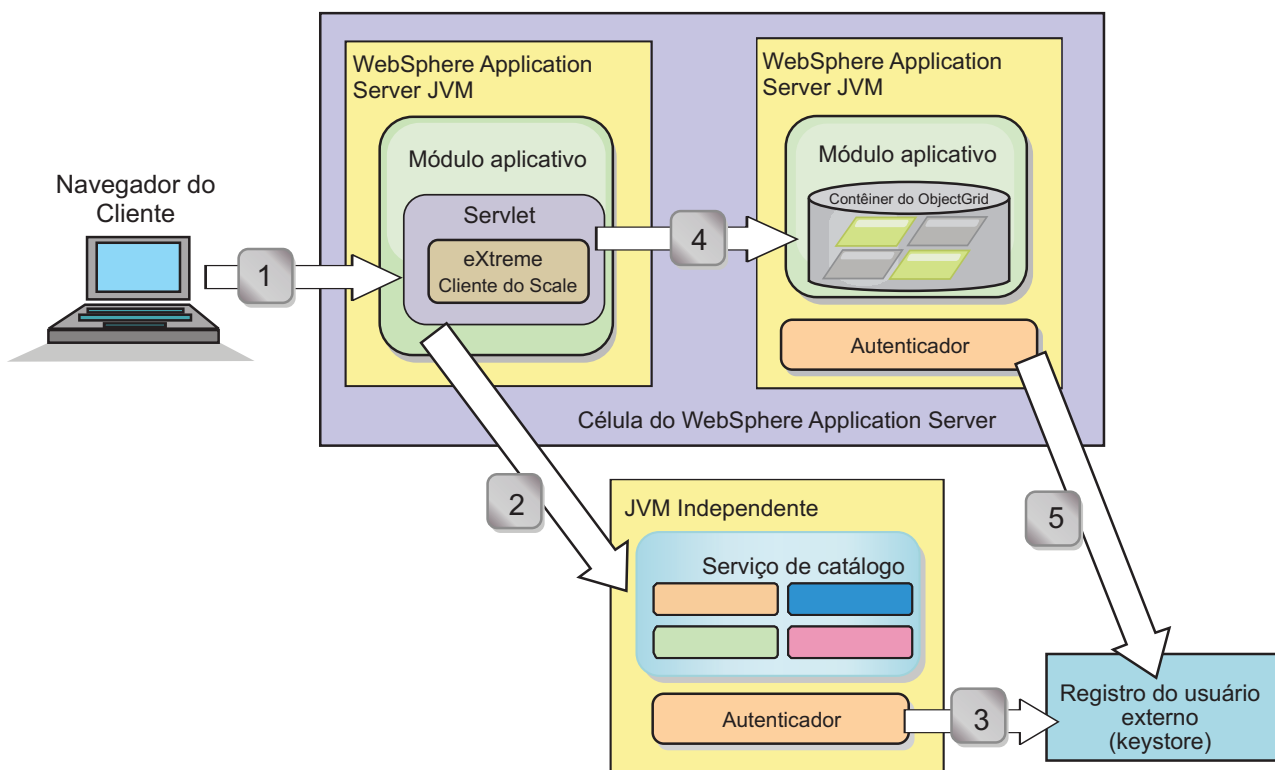


Figura 21. Fluxo de Autenticação

O diagrama anterior mostra dois servidores de aplicativos. O primeiro servidor de aplicativos hospeda o aplicativo da web, que também é um cliente do WebSphere eXtreme Scale. O segundo servidor de aplicativos hospeda um servidor de contêiner. O servidor de catálogos está em execução em uma Java virtual machine (JVM) independente em vez de WebSphere Application Server.

As setas marcadas com números do diagrama indicam o fluxo da autenticação:

1. Um usuário de aplicativo corporativo acessa o navegador da web e efetua login no primeiro servidor de aplicativos com um nome de usuário e senha. O primeiro servidor de aplicativos envia o nome de usuário e senha do cliente para a infraestrutura de segurança para se autenticar no registro do usuário. Esse registro do usuário é um keystore. Como resultado, as informações de segurança são armazenadas no encadeamento do WebSphere Application Server.
2. O arquivo JavaServer Pages (JSP) age como um cliente do WebSphere eXtreme Scale para recuperar as informações de segurança a partir do arquivo de propriedades do cliente. O aplicativo JSP que está agindo como o cliente do WebSphere eXtreme Scale envia a credencial de segurança do cliente do WebSphere eXtreme Scale junto com a solicitação para o servidor de catálogos. Enviar a credencial de segurança com a solicitação é considerado um modelo

*runAs*. Em um modelo *runAs*, o cliente do navegador da web é executado como um cliente do WebSphere eXtreme Scale para acessar os dados armazenados no servidor de contêiner. O cliente usa uma credencial de cliente da Java virtual machine (JSP) para se conectar aos servidores WebSphere eXtreme Scale. Usar o modelo *runAs* é o mesmo que conectar-se a um banco de dados com um ID do usuário e senha no nível da origem de dados.

3. O servidor de catálogos recebe a credencial do cliente do WebSphere eXtreme Scale, que inclui os tokens de segurança do WebSphere Application Server. Em seguida, o servidor de catálogos chama o plug-in do autenticador para autenticar a credencial do cliente. O autenticador se conecta ao registro do usuário externo e envia a credencial do cliente para o registro do usuário para autenticação.
4. O cliente envia o ID do usuário e a senha para o servidor de contêiner que é hospedado no servidor de aplicativos.
5. O serviço de contêiner, hospedado no servidor de aplicativos, recebe a credencial do cliente do WebSphere eXtreme Scale, que é o par de ID do usuário e senha. Em seguida, o servidor de contêiner chama o plug-in do autenticador para autenticar a credencial do cliente. O autenticador se conecta ao registro do usuário de keystore e envia a credencial do cliente para o registro do usuário para autenticação

## Objetivos do aprendizado

Com as lições neste módulo, você aprende como:

- Configure a segurança do cliente do WebSphere eXtreme Scale.
- Configure a segurança do servidor de catálogos do WebSphere eXtreme Scale.
- Configure a segurança do servidor de contêiner do WebSphere eXtreme Scale.
- Instale e execute o aplicativo de amostra.

## Tempo necessário

Este módulo leva cerca de 60 minutos.

### Lição 2.1: Configurar a Segurança do Cliente do WebSphere eXtreme Scale

Configure as propriedades do cliente com um arquivo de propriedades. O arquivo de propriedades do cliente indica a classe de implementação `CredentialGenerator` a ser usada.

#### Conteúdo do Arquivo de Propriedades do Cliente:

O tutorial usa tokens de segurança do WebSphere Application Server para obter a credencial do cliente. O diretório `samples_home/security_extauth` contém o arquivo `client3.props`.

O arquivo `client3.props` inclui as seguintes configurações:

#### `securityEnabled`

Ativa a segurança do cliente do WebSphere eXtreme Scale. O valor é configurado para `true` para indicar que o cliente deve enviar informações de segurança disponíveis para o servidor.



### **credentialAuthentication**

Especifica o suporte de autenticação da credencial do cliente. O valor é configurado para Suportado para indicar que o cliente suporta autenticação de credencial.

### **credentialGeneratorClass**

Especifica o nome da classe que implementa a interface `com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator`. O valor é configurado para a classe `com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator` para que o cliente recupere as informações de segurança a partir da classe `UserPasswordCredentialGenerator`.

### **credentialGeneratorProps**

Especifica o nome do usuário e a senha: `manager manager1`. O nome do usuário é `manager` e a senha é `manager1`. Também é possível usar o comando `FilePasswordEncoder.bat | sh` para codificar essa propriedade usando um algoritmo exclusivo ou (xor).

## **Configurando o Arquivo de Propriedades do Cliente Usando as Propriedades da Java virtual machine (JVM):**

No console administrativo, conclua as seguintes etapas para ambos servidores `s1` e `s2` no cluster `appCluster`. Se uma topologia diferente estiver sendo usada, conclua as seguintes etapas para todos os servidores de aplicativos nos quais o aplicativo `EmployeeManagement` é implementado.

1. **Servidores > Servidores de aplicativo WebSphere > *server\_name* > Java and Process Management > Definição de processo > Java Virtual Machine.**
2. Crie a seguinte propriedade JVM genérica para configurar o local do arquivo de propriedades do cliente:  
`-Dobjectgrid.client.props=samples_home/security_extauth/client3.props`
3. Clique em **OK** e salve as mudanças.

### **Ponto de verificação de lições:**

Você editou o arquivo de propriedades do cliente e configurou os servidores no cluster `appCluster` para usar o arquivo de propriedades do cliente. Este arquivo de propriedades indica a classe de implementação `CredentialGenerator` a ser usada.

## **Lição 2.2: Configurar a Segurança do Servidor de Catálogos**

Um servidor de catálogos contém dois níveis de informações de segurança: O primeiro nível contém as propriedades de segurança que são comuns a todos os servidores do WebSphere eXtreme Scale, incluindo o serviço de catálogo e os servidores de contêiner. O segundo nível contém as propriedades de segurança que são específicas para o servidor de catálogos.

As propriedades de segurança que são comuns aos servidores de catálogos e aos servidores de contêiner são configuradas no arquivo descritor XML de segurança. Um exemplo das propriedades comuns é a configuração do autenticador, que representa o registro do usuário e o mecanismo de autenticação. Consulte o Arquivo XML Descritor de Segurança para obter mais informações sobre as propriedades de segurança.



Para configurar o arquivo descritor XML de segurança em um ambiente Java SE, use uma opção **-clusterSecurityFile** quando executar o comando **startOgServer**. Especifique um valor em um formato de arquivo, como *samples\_home/security\_extauth/security3.xml*.

#### **security3.xml file:**

Neste tutorial, o arquivo *securityWAS2.xml* está no diretório *samples\_home/security\_extauth*. O conteúdo do arquivo *security3.xml* com os comentários removidos é o seguinte:

```
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config/security">

  <security securityEnabled="true">
    <authenticator
      className="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
    </authenticator>
  </security>
</securityConfig>
```

As seguintes propriedades são definidas no arquivo *security3.xml*:

#### **securityEnabled**

A propriedade `securityEnabled` é configurada para `true`, indicando que a segurança global do WebSphere eXtreme Scale do servidor de catálogos foi ativada.

#### **autenticador**

O autenticador é configurado como a classe `com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator`. Com essa implementação integrada do plug-in Autenticador, o ID do usuário e senha são passados para verificar se ele está configurado no arquivo `keystore`. A classe `KeyStoreLoginAuthenticator` usa um alias de módulo de login `KeyStoreLogin`, portanto, uma configuração de login do Java Authentication and Authorization Service (JAAS) é necessária.

#### **Arquivo catServer3.props:**

O arquivo de propriedades do servidor armazena as propriedades específicas do servidor, dentre elas as propriedades de segurança específicas do servidor. Consulte o Arquivo de Propriedades do Servidor para obter informações adicionais. É possível usar a opção **-serverProps** para especificar a propriedade do servidor de catálogos quando executar o comando **startOgServer**. Para este tutorial, um arquivo *catServer3.props* é incluído no diretório. O conteúdo do arquivo *catServer3.props* com os comentários removidos é o seguinte:

```
securityEnabled=true
credentialAuthentication=Required
transportType=TCP/IP
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

#### **securityEnabled**

A propriedade `securityEnabled` é configurada para `true` para indicar que este servidor de catálogos é um servidor seguro.

#### **credentialAuthentication**

A propriedade `credentialAuthentication` é configurada para `Necessária`, portanto, qualquer cliente que esteja conectado ao servidor precisa fornecer uma credencial. No arquivo de propriedades do cliente, o valor

credentialAuthentication é configurado para Suportado, portanto, o servidor recebe as credenciais que são enviados pelo cliente.

#### **secureTokenManagerType**

O secureTokenManagerType está configurado para none para indicar que o segredo da autenticação não é criptografado quando junta os servidores existentes.

#### **authenticationSecret**

A propriedade authenticationSecret está configurada para ObjectGridDefaultSecret. Esta sequência secreta é usada para juntar o cluster de servidores eXtreme Scale. Quando um servidor junta a grade de dados, ele é desafiado a apresentar a sequência secreta. Se a sequência secreta do servidor de junção corresponder à sequência no servidor de catálogos, o servidor de junção será aceito. Se a cadeia não corresponder, o pedido de junção é rejeitado.

#### **transportType**

A propriedade transportType é configurada inicialmente para TCP/IP. Posteriormente no tutorial, a segurança de transporte é ativada.

#### **Arquivo xsjaas3.config:**

Como a implementação KeyStoreLoginAuthenticator usa um módulo de login, você deve configurar o modelo de login com um arquivo de configuração de login de autenticação JAAS. O conteúdo do arquivo xsjaas3.config é:

```
KeyStoreLogin{
com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule required
    keyStoreFile="samples_home/security_extauth/sampleKS3.jks" debug = true;
};
```

Se um local para o *samples\_home* diferente de */wxs\_samples/* for usado, será necessário atualizar o local do keyStoreFile. Esta configuração de login indica que o módulo

com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule é usado como o módulo de login. O arquivo keystore é configurado para o arquivo sampleKS3.jks.

O arquivo keystore sampleKS3.jks de amostra armazena dois IDs do usuário e senhas: manager/manager1 e cashier/cashier1.

Os seguintes comandos **keytool** podem ser usados para criar esse keystore:

- keytool -genkey -v -keystore ./sampleKS3.jks -storepass sampleKS1 -alias manager -keypass manager1 -dname CN=manager,O=acme,OU=OGSample -validity 10000
- keytool -genkey -v -keystore ./sampleKS3.jks -storepass sampleKS1 -alias operator -keypass operator1 -dname CN=operator,O=acme,OU=OGSample -validity 10000

#### **Iniciar o Servidor de Catálogos com a Segurança Ativada:**

Para iniciar o servidor de catálogos, emita o comando **startOgServer** com os parâmetros **-clusterFile** e **-serverProps** a serem passados nas propriedades de segurança.

Use uma instalação independente do WebSphere eXtreme Scale para executar o servidor de catálogos. Ao usar a imagem de instalação independente, o IBM SDK deve ser usado. É possível usar o SDK que está incluído com o WebSphere

Application Server ao definir a variável `JAVA_HOME` para apontar para o IBM SDK. Por exemplo, set `JAVA_HOME=was_root/IBM/WebSphere/AppServer/java/`

1. Acesse o diretório `bin`.

```
cd wxs_home/bin
```

2. Executar o comando **startOgServer**.

Linux

UNIX

```
./startOgServer.sh cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndpoints
cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile samples_home/security_extauth/security3.xml
-serverProps samples_home/security_extauth/catServer3.props -jvmArgs
-Djava.security.auth.login.config="samples_home/security_extauth/xsjaas3.config"
```

Windows

```
startOgServer.bat cs1 -listenerPort 16809 -JMXServicePort 16099 -catalogServiceEndpoints
cs1:[HOST_NAME]:16601:16602 -clusterSecurityFile samples_home/security_extauth/security3.xml
-serverProps samples_home/security_extauth/catServer3.props -jvmArgs
-Djava.security.auth.login.config="samples_home/security_extauth/xsjaas3.config"
```

Depois de executar o comando **startOgServer**, um servidor seguro é iniciado com a porta listener 16809, porta do cliente 16601, porta de peer 16602 e a porta JMX 16099. Se existir um conflito de portas, altere o número da porta para um número de porta não usado.

### Parar um Servidor de Catálogos com a Segurança Ativada:

É possível usar o comando **stopOgServer** para parar o servidor de catálogos.

1. Acesse o diretório `bin`.

```
cd wxs_home/bin
```

2. Execute o comando **stopOgServer**.

Linux

UNIX

```
stopOgServer.sh cs1 -catalogServiceEndpoints localhost:16809 -clientSecurityFile
samples_home/security_extauth/client3.props
```

Windows

```
stopOgServer.bat cs1 -catalogServiceEndpoints localhost:16809 -clientSecurityFile
samples_home/security_extauth/client3.props
```

### Ponto de verificação de lições:

Você configurou a segurança do servidor de catálogos ao associar os arquivos `security3.xml`, `catServer3.props` e `xsjaas3.config` ao serviço de catálogo.

### Lição 2.3: Configurar a Segurança do Servidor do Contêiner

Quando um servidor de contêiner se conecta ao serviço de catálogos, o servidor de contêiner obtém todas as configurações de segurança que estiverem definidas no arquivo XML ObjectGrid Security. O arquivo XML ObjectGrid Security define a configuração do autenticador, o valor de tempo limite da sessão de login e outras informações de configuração. Um servidor de contêiner também possui suas próprias propriedades de segurança específicas do servidor no arquivo de propriedade do servidor.

Configure o arquivo de propriedades do servidor com a propriedade `-Dobjectgrid.server.props` da Java virtual machine (JVM). O nome do arquivo especificado para esta propriedade é um caminho de arquivo absoluto, como `samples_home/security_extauth/server3.props`.

Neste tutorial, os servidores de contêiner são hospedados nos servidores `xs1` e `xs2` no cluster `xsCluster`.

### Arquivo server3.props:

O arquivo server3.props está no diretório *samples\_home/security\_extauth/*. O conteúdo do arquivo server3.props é o seguinte:

```
securityEnabled=true
credentialAuthentication=Required
secureTokenManagerType=none
authenticationSecret=ObjectGridDefaultSecret
```

#### securityEnabled

A propriedade securityEnabled é configurada para true para indicar que este servidor de contêiner é um servidor seguro.

#### credentialAuthentication

A propriedade credentialAuthentication é configurada para Necessária, portanto, qualquer cliente que esteja conectado ao servidor precisa fornecer uma credencial. No arquivo de propriedades do cliente, a propriedade credentialAuthentication é configurada para Suportado, portanto, o servidor recebe a credencial que é enviada pelo cliente.

#### secureTokenManagerType

O secureTokenManagerType está configurado para none para indicar que o segredo da autenticação não é criptografado quando junta os servidores existentes.

#### authenticationSecret

A propriedade authenticationSecret está configurada para ObjectGridDefaultSecret. Esta sequência secreta é usada para juntar o cluster de servidores eXtreme Scale. Quando um servidor junta a grade de dados, ele é desafiado a apresentar a sequência secreta. Se a sequência secreta do servidor de junção corresponder à sequência no servidor de catálogos, o servidor de junção será aceito. Se a cadeia não corresponder, o pedido de junção é rejeitado.

### Configurando o arquivo de propriedades de servidor com propriedades JVM:

Configure o arquivo de propriedades do servidor nos servidores xs1 e xs2. Se a topologia não estiver sendo usada para este tutorial, configure o arquivo de propriedades do servidor para todos os servidores de aplicativos que estiverem sendo usados para hospedar servidores de contêiner.

1. Abra a página da Java virtual machine para o servidor. **Servidores > Servidores de aplicativo WebSphere > server\_name > Java and Process Management > Definição de processo > Java Virtual Machine.**
2. Inclua os argumentos JVM genéricos:  
-Dobjectgrid.server.props=*samples\_home/security\_extauth/server3.props*
3. Clique em **OK** e salve as mudanças.

### Incluindo o Módulo de Login Customizado:

O servidor de contêiner usa a mesma implementação KeyStoreAuthenticator que o servidor de catálogos. A implementação KeyStoreAuthenticator usa um alias de módulo de login **KeyStoreLogin**, portanto, você deve incluir um módulo de login customizado nas entradas do modelo de login do aplicativo.

1. No console administrativo do WebSphere Application Server, clique em **Segurança > Segurança Global > Java Authentication and Authorization Service.**
2. Clique em **Logins do Aplicativo.**

3. Clique em **Novo** e inclua um alias `KeyStoreLogin`. Dê um clique em **Aplicar**.
4. Em **Módulos de login JAAS**, clique em **Novo**.
5. Insira `com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginModule` como o nome de classe do módulo e escolha **SUFFICIENT** como a estratégia de autenticação. Dê um clique em **Aplicar**.
6. Inclua uma propriedade customizada `keyStoreFile` com o valor `samples_home/security_extauth/sampleKS.jks`.
7. Opcional: Inclua a propriedade customizada `debug` com o valor `true`.
8. Salve a configuração.

#### Ponto de verificação de lições:

Agora, a autenticação do servidor WebSphere eXtreme Scale está protegida. Ao configurar esta segurança, todos os aplicativos que tentam se conectar com os servidores WebSphere eXtreme Scale são necessárias para fornecer uma credencial. Nesse tutorial, o `KeyStoreLoginAuthenticator` é o autenticador. Como resultado, o cliente é necessário para fornecer um nome de usuário e senha.

### Lição 2.4: Instalar e Executar a Amostra

Depois que a autenticação for configurada, o aplicativo de amostra poderá ser instalado e executado.

#### Criando uma Biblioteca Compartilhada para o Arquivo `EmployeeData.jar`:

1. No console administrativo do WebSphere Application Server, abra a página **Bibliotecas Compartilhadas**. Clique em **Ambiente > Bibliotecas Compartilhadas**.
2. Escolha o escopo **célula**.
3. Crie a biblioteca compartilhada. Clique em **Novo**. Insira `EmployeeManagementLIB` como o **Nome**. Insira o caminho para o arquivo `EmployeeData.jar` no caminho de classe, por exemplo, `samples_home/WASSecurity/EmployeeData.jar`.
4. Dê um clique em **Aplicar**.

#### Instalando a Amostra:

1. Instale o arquivo `EmployeeManagement_extauth.ear` no diretório `samples_home/security_extauth`.

**Importante:** O arquivo `EmployeeManagement_extauth.ear` é diferente do arquivo `samples_home/WASSecurity/EmployeeManagement.ear`. O modo com que a sessão do ObjectGrid é recuperada foi atualizado para utilizar a credencial que é armazenada em cache no arquivo de propriedades do cliente no aplicativo `EmployeeManagement_extauth.ear`. Consulte os comentários na classe `com.ibm.websphere.sample.xs.DataAccessor` no projeto `samples_home/WASSecurity/EmployeeManagementWeb` para ver o código que foi atualizado para esta mudança.

- a. Para iniciar a instalação, clique em **Aplicativos > Novo Aplicativo > Novo Aplicativo Corporativo**. Escolha o caminho detalhado para instalar o aplicativo.
- b. Na etapa **Mapear módulos para servidores**, especifique o cluster `appCluster` para instalar o módulo `EmployeeManagementWeb`.
- c. Na etapa **Mapear bibliotecas compartilhadas**, selecione o módulo `EmployeeManagementWeb`.
- d. Clique em **Referenciar bibliotecas compartilhadas**. Selecionar a biblioteca `EmployeeManagementLIB`.

- e. Mapeie a função webUser para **Tudo Autenticado na Região do Aplicativo**.
- f. Clique em **OK**.

Os clientes são executados nos servidores s1 e s2 neste cluster.

2. Instale o arquivo de amostra XSDeployment.ear no diretório *samples\_home/WASSecurity*.
  - a. Para iniciar a instalação, clique em **Aplicativos > Novo Aplicativo > Novo Aplicativo Corporativo**. Escolha o caminho detalhado para instalar o aplicativo.
  - b. Na etapa **Mapear módulos para servidores**, especifique o cluster xsCluster para instalar o módulo da web XSDeploymentWeb.
  - c. Na etapa **Mapear bibliotecas compartilhadas**, selecione o módulo XSDeploymentWeb.
  - d. Clique em **Referenciar bibliotecas compartilhadas**. Selecione a biblioteca EmployeeManagementLIB.
  - e. Clique em **OK**.

Os servidores xs1 e xs2 neste cluster hospedam os servidores de contêiner.

3. Verifique se o servidor de catálogos foi iniciado. Para obter mais informações sobre como iniciar um servidor de catálogos para este tutorial, consulte "Iniciar o Servidor de Catálogos com a Segurança Ativada" na página 118.
4. Reinicie o cluster xsCluster. Quando o xsCluster é iniciado, o aplicativo XSDeployment é iniciado e um servidor de contêiner é iniciado nos servidores xs1 e xs2 respectivamente. Se você olhar o arquivo SystemOut.log dos servidores xs1 e xs2, a seguinte mensagem que indica que o arquivo de propriedades do servidor foi carregado é exibida:  
CW0BJ0913I: Os arquivos de propriedade de servidor foram carregados:  
*samples\_home/security\_extauth/server3.props*.
5. Reinicie o cluster appClusters. Quando o cluster appClusters é iniciado, o aplicativo EmployeeManagement também é iniciado. Se observar o arquivo SystemOut.log dos servidores s1 e s2, será possível ver a seguinte mensagem indicando que o arquivo de propriedades do cliente foi carregado.  
CW0BJ0924I: O arquivo de propriedades do cliente {0} foi carregado.

Se estiver usando o WebSphere eXtreme Scale Versão 7.0, uma mensagem CW0BJ9000I somente em inglês será exibida para indicar que o arquivo de propriedades do cliente foi carregado. Se a mensagem esperada não for exibida, verifique se a propriedade -Dobjectgrid.server.props ou -Dobjectgrid.client.props foi configurada no argumento da JVM. Se as propriedades estiverem realmente configuradas, certifique-se de que o traço (-) seja um caractere UTF.

#### Executando o Aplicativo de Amostra:

1. Execute o arquivo management.jsp. Em um navegador da web, acesse `http://<your_servername>:<port>/EmployeeManagementWeb/management.jsp`. Por exemplo, é possível usar a seguinte URL: `http://localhost:9080/EmployeeManagementWeb/management.jsp`.
2. Forneça autenticação para o aplicativo. Insira as credenciais do usuário que foram mapeadas para a função webUser. Por padrão, essa função do usuário é mapeada para todos os usuários autenticados. Digite qualquer nome de usuário e senha válidos, como o nome e a senha do usuário administrativo. Uma página para exibir, incluir, atualizar e excluir funcionários é exibida.



3. Exibir Funcionários Clique em **Exibir um Funcionário**. Insira emp1@acme.com como o endereço de email e clique em **Enviar**. Uma mensagem é exibida informando que o usuário não pode ser localizado.
4. Incluir um funcionário. Clique em **Incluir um Funcionário**. Insira emp1@acme.com como o endereço de email, insira Joe como o nome e insira Doe como o sobrenome. Clique em **Enviar**. Uma mensagem é exibida informando que um funcionário com o endereço emp1@acme.com foi incluído.
5. Exibir o novo funcionário. Clique em **Exibir um Funcionário**. Insira emp1@acme.com como o endereço de email com campos vazios para o nome e sobrenome e clique em **Enviar**. Uma mensagem é exibida informando que o funcionário foi localizado e que os nomes corretos são exibidos nos campos nome e sobrenome.
6. Excluir o funcionário. Clique em **Excluir um funcionário**. Insira emp1@acme.com e clique em **Enviar**. Uma mensagem é exibida informando que o funcionário foi excluído.

Como o tipo de transporte do servidor de catálogos é configurado para TCP/IP, verifique se a configuração do transporte de saída s1 e s2 do servidor não está definida para SSL Necessário. Caso contrário, uma exceção ocorrerá. Se você observar o arquivo de saída do sistema do servidor de catálogos, logs/cs1/SystemOut.log, a seguinte saída de depuração indicará a autenticação do armazenamento de chaves:

```
SystemOut    0 [KeyStoreLoginModule] initialize: Successfully loaded key store
SystemOut    0 [KeyStoreLoginModule] login: entry
SystemOut    0 [KeyStoreLoginModule] login: user entered user name: manager
SystemOut    0   Print out the certificates:
...
```

#### **Ponto de verificação de lições:**

Você instalou e executou o aplicativo de amostra.

## **Módulo 3: Configurar a Segurança de Transporte**

Configure a segurança de transporte para proteger a transferência de dados entre os clientes e servidores na configuração.

No módulo anterior no tutorial, você ativou a autenticação do WebSphere eXtreme Scale. Com a autenticação, qualquer aplicativo que tentar se conectar com o servidor WebSphere eXtreme Scale precisará fornecer uma credencial. Portanto, nenhum cliente não autenticado pode se conectar com o servidor WebSphere eXtreme Scale. Os clientes devem ser um aplicativo autenticado que esteja em execução em uma célula do WebSphere Application Server.

Com a configuração até esse módulo, a transferência de dados entre os clientes no cluster appCluster e os servidores no cluster xsCluster não é criptografada. Essa configuração poderá ser aceitável apenas se seus clusters do WebSphere Application Server estiverem instalados em servidores por trás de um firewall. Entretanto, em alguns cenários, um tráfego não criptografado não é aceito por algumas razões, embora a topologia esteja protegida por um firewall. Por exemplo, uma política de controle pode aplicar o tráfego criptografado. O WebSphere eXtreme Scale suporta a Segurança da Camada de Transporte/Secure Sockets Layer (TLS/SSL) para uma comunicação segura entre os terminais do ObjectGrid, que incluem servidores de cliente, servidores de contêineres e servidores de catálogos.

Nesta implementação de amostra, os clientes e servidores de contêiner do eXtreme Scale estão todos executando no ambiente do WebSphere Application Server. As propriedades do cliente ou do servidor não são necessárias para definir as configurações SSL porque a segurança de transporte do eXtreme Scale é gerenciada pelas configurações de transporte do Common Secure Interoperability Protocol Versão 2 (CSIV2) do WebSphere Application Server. Os servidores do WebSphere eXtreme Scale usam a mesma instância do Object Request Broker (ORB) como os servidores de aplicativos nos quais eles são executados. Especifique todas as configurações de SSL para servidores de cliente e de contêiner na configuração do WebSphere Application Server usando estas configurações de transporte de CSIV2. Você deve configurar as propriedades SSL no arquivo de propriedades do servidor para o servidor de catálogos.

## Objetivos do aprendizado

Depois de concluir as lições neste módulo, você saberá como:

- Configurar o transporte de entrada e de saída CSIV2
- Incluir propriedades SSL no arquivo de propriedades do servidor de catálogos
- Verificar o arquivo de propriedades do ORB.
- Executar a amostra.

## Tempo necessário

Este módulo leva cerca de 60 minutos.

## Pré-requisitos

Esta etapa do tutorial é construída sobre os módulos anteriores. Conclua os módulos anteriores neste tutorial antes de configurar a segurança do transporte.

### Lição 3.1: Configurar o Transporte de Entrada e de Saída CSIV2

Para configurar o protocolo Segurança da Camada de Transporte/Secure Sockets Layer (TLS/SSL) para o transporte do servidor, configure o transporte de entrada e o transporte de saída do Common Secure Interoperability Protocol Versão 2 (CSIV2) para SSL Necessário para todos os servidores do WebSphere Application Server que hospedam clientes, servidores de catálogo e servidores de contêineres.

Na topologia de exemplo do tutorial, você deve configurar essas propriedades para os servidores de aplicativos s1, s2, xs1 e xs2. As etapas a seguir configuram os transportes de entrada e saída para todos os servidores na configuração.

Configure os transportes de entrada e saída no console administrativo. Certifique-se de que a segurança administrativa esteja ativada.

- **WebSphere Application Server Versão 6.1:** Clique em **Segurança > Administração Segura > Aplicativo.** > **Segurança RMI/IIOP** e altere o tipo de transporte para **SSL Necessário**.
- **WebSphere Application Server Versão 7.0:** Clique em **Segurança > Segurança Global > Segurança RMI/IIOP > Comunicações de entrada CSIV2**. Altere o tipo de transporte na Camada de Transporte CSIV2 como **SSL Necessário**. Repita esta etapa para configurar as comunicações de saída CSIV2.

As configurações de segurança do terminal gerenciadas centralmente podem ser usadas ou os repositórios SSL podem ser configurados. Consulte Configurações do Transporte de Entrada Common Secure Interoperability Versão 2 para obter mais informações.



## Lição 3.2: Incluir Propriedades SSL no Arquivo de Propriedades do Servidor de Catálogos

O servidor de catálogos está executando fora do WebSphere Application Server, portanto, você deve configurar as propriedades SSL no arquivo de propriedades do servidor.

A outra razão para configurar as propriedades SSL no arquivo de propriedades do servidor é porque o servidor de catálogos tem seus próprios caminhos de transporte proprietários que não podem ser gerenciados pelas configurações de transporte do WebSphere Application Server Common Secure Interoperability Protocol Versão 2 (CSIV2). Portanto, você deve configurar as propriedades do Secure Sockets Layer (SSL) no arquivo de propriedades do servidor para o servidor de catálogos.

### Propriedades SSL no Arquivo `catServer3.props`:

```
alias=default
contextProvider=IBMJSE2
protocol=SSL
keyStoreType=PKCS12
keyStore=/was_root/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/key.p12
keyStorePassword=WebAS
trustStoreType=PKCS12
trustStore=/was_root/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/trust.p12
trustStorePassword=WebAS
clientAuthentication=false
```

O arquivo `catServer3.props` usa o keystore e o armazenamento confiável de nível do nó do WebSphere Application Server padrão. Se estiver implementando um ambiente de implementação mais complexo, você deverá escolher o keystore e o armazenamento confiável corretos. Em alguns casos, você deve criar um keystore e um armazenamento confiável e importar as chaves dos keystores de outros servidores. Observe que a sequência `WebAS` é a senha padrão do keystore e do armazenamento confiável do WebSphere Application Server. Consulte [Configuração de Certificado Autoassinado Padrão](#) para obter mais detalhes.

Essas entradas já estão incluídas no arquivo `samples_home/security_extauth/catServer3.props` como comentários. É possível remover o comentário das entradas e fazer as atualizações apropriadas de sua instalação nas variáveis `was_root`, `<deployment_manager_name>`, `<cell_name>` e `<node_name>`.

Depois de configurar as propriedades SSL, altere o valor da propriedade `transportType` de `TCP/IP` para `SSL Necessário`.

### Propriedades SSL no arquivo `client3.props`:

As propriedades SSL também devem ser configuradas no arquivo `client3.props` porque este arquivo é usado quando você para o servidor de catálogos que está sendo executado fora do WebSphere Application Server.

Essas propriedades não têm efeito sobre os servidores do cliente que estão em execução no WebSphere Application Server porque eles usam as configurações de transporte WebSphere Application Server Common Security Interoperability Protocol Versão 2 (CSIV2). Entretanto, quando você para o servidor de catálogos, um arquivo de propriedades do cliente deve ser fornecido no comando **stop0gServer**. Configure as seguintes propriedades no arquivo

<SAMPLES\_HOME>/security\_extauth/client3.props) para corresponderem aos valores especificados acima no arquivo catServer3.props:

```
#contextProvider=IBMJSSE2
#protocol=SSL
#keyStoreType=PKCS12
#keyStore=/was_root/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/key.p12
#keyStorePassword=WebAS
#trustStoreType=PKCS12
#trustStore=/was_root/IBM/WebSphere/AppServer/profiles/
<deployment_manager_name>/config/cells/<cell_name>/nodes/
<node_name>/trust.p12
#trustStorePassword=WebAS
```

Assim como no arquivo catServer3.props, é possível usar os comentários que já estão fornecido no arquivo *samples\_home/security\_extauth/client3.props* com as atualizações apropriadas para as variáveis *was\_root*, *<deployment\_manager\_name>*, *<cell\_name>* e *<node\_name>* para corresponderem ao seu ambiente.

#### Ponto de verificação de lições:

Você configurou as propriedades SSL para o servidor de catálogos.

#### Lição 3.3: Executar a Amostra

Reinicie todos os servidores e execute o aplicativo de amostra novamente. Você deve poder executar as etapas sem nenhum problema.

Consulte “Lição 2.4: Instalar e Executar a Amostra” na página 121 para obter mais informações sobre como executar e instalar o aplicativo de amostra.

## Módulo 4: Usar a Autorização do Java Authentication and Authorization Service (JAAS) no WebSphere Application Server

Agora que você configurou a autenticação de clientes, é possível configurar ainda mais a autorização para conceder aos usuários diferentes permissões. Por exemplo, um usuário "operador" pode apenas visualizar dados, enquanto que um usuário "gerente" pode executar todas as operações.

Após autenticar um cliente, como no módulo anterior neste tutorial, é possível fornecer privilégios de segurança por meio dos mecanismos de autorização do eXtreme Scale. O módulo anterior deste tutorial demonstrou como ativar a autenticação para uma grade de dados usando a integração com o WebSphere Application Server. Como resultado, nenhum cliente não autenticado pode se conectar aos servidores eXtreme Scale ou submeter solicitações para seu sistema. Entretanto, todo cliente autenticado tem a mesma permissão ou privilégios para o servidor, como de leitura, gravação ou exclusão de dados armazenados nos mapas do ObjectGrid. Os clientes também podem emitir qualquer tipo de consulta.

Esta parte do tutorial demonstra como usar a autorização do eXtreme Scale para fornecer diferentes privilégios para usuários autenticados. O WebSphere eXtreme Scale usa um mecanismo de autorização baseado em permissão. É possível designar categorias de permissão diferentes que são representadas por diferentes classes de permissão. Esse módulo apresenta a classe MapPermission. Para obter uma lista de todas as permissões possíveis, consulte Programação de Autorização de Cliente.

No WebSphere eXtreme Scale, a classe `com.ibm.websphere.objectgrid.security.MapPermission` representa permissões para os recursos do eXtreme Scale, especificamente os métodos das interfaces `ObjectMap` ou `JavaMap`. O WebSphere eXtreme Scale define as seguintes cadeias de permissões para acesso aos métodos de `ObjectMap` e `JavaMap`:

- **read**: Concede permissão para ler os dados do mapa.
- **write**: Concede permissão para atualizar os dados no mapa.
- **insert**: Concede permissão para inserir os dados no mapa.
- **remove**: Concede permissão para remover os dados do mapa.
- **invalidate**: Concede permissão para invalidar os dados a partir do mapa.
- **all**: Concede todas as permissões para ler, gravar, inserir, remover e invalidar.

A autorização ocorre quando um cliente do eXtreme Scale usa uma API de acesso a dados, como o `ObjectMap`, o `JavaMap` ou as APIs `EntityManager`. O tempo de execução do eXtreme Scale verifica as permissões do mapa correspondente quando o método é chamado. Se as permissões necessárias não forem concedidas ao cliente, isso resultará em uma exceção `AccessControlException`. Este tutorial demonstra como usar a autorização do Java Authentication and Authorization Service (JAAS) para conceder acesso de mapa de autorização para diferentes usuários.

## Objetivos do aprendizado

Depois de concluir as lições neste módulo, você saberá como:

- Ativar a autorização para WebSphere eXtreme Scale.
- Ativar a autorização baseada no usuário

## Tempo necessário

Este módulo leva cerca de 60 minutos.

### Lição 4.1: Ativar a Autorização do WebSphere eXtreme Scale

Para ativar a autorização no WebSphere eXtreme Scale, você deve ativar a segurança em um `ObjectGrid` específico.

Para ativar a autorização no `ObjectGrid`, você deve configurar o atributo **`securityEnabled`** para `true` desse `ObjectGrid` específico no arquivo XML. Para este tutorial, é possível usar o arquivo `XSDeployment_sec.ear` a partir do diretório `samples_home/WASSecurity`, que já possui a segurança configurada no arquivo `objectGrid.xml`, ou é possível editar o arquivo `objectGrid.xml` existente para ativar a segurança. Esta lição demonstra como editar o arquivo para ativar a segurança.

1. Opcional: Extraia os arquivos no arquivo `XSDeployment.ear` e, em seguida, descompacte o arquivo `XSDeploymentWeb.war`.
2. Opcional: Abra o arquivo `objectGrid.xml` e configure o atributo **`securityEnabled`** para `true` no nível do `ObjectGrid`. A seguir há um exemplo desse atributo:

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
```

```
  <objectGrids>
    <objectGrid name="Grid" txTimeout="15" securityEnabled="true">
      <backingMap name="Map1" />
```

```

        </objectGrid>
    </objectGrids>

</objectGridConfig>

```

Se diversos ObjectGrids estiverem definidos, esse atributo deverá ser configurado em cada grade.

3. Opcional: Recompacte os arquivos XSDeploymentWeb.war e XSDeployment.ear para incluir suas mudanças.
4. Necessário: Desinstale o arquivo XSDeployment.ear e, em seguida, instale o arquivo XSDeployment.ear. É possível usar o arquivo modificado nas etapas anteriores ou instalar o arquivo XSDeployment\_sec.ear que é fornecido no diretório *samples\_home/WASSecurity*. Consulte o “Lição 2.4: Instalar e Executar a Amostra” na página 121 para obter informações adicionais sobre a instalação do aplicativo.
5. Reinicie todos os servidores de aplicativos para ativar a autorização do WebSphere eXtreme Scale.

#### Ponto de verificação de lições:

Você ativou a segurança no ObjectGrid, que também ativa a autorização na grade de dados.

### Lição 4.2: Ativar Autorização Baseada no Usuário

No módulo de autenticação deste tutorial, dois usuários foram criados: operator e manager. É possível designar permissões diferentes para esses usuários com a autorização Java Authentication and Authorization Service (JAAS).

#### Definindo a Política de Autorização do Java Authentication and Authorization Service (JAAS) Usando os Principals do Usuário:

É possível designar permissões para os usuários criados anteriormente. Designe permissões somente leitura ao usuário operator para todos os mapas. Designe ao usuário manager todas as permissões. Use o arquivo de política de autorização JAAS para conceder permissões para os principals.

Edite o arquivo de autorização do JAAS. O arquivo *xAuth3.policy* está no diretório *samples\_home/security\_extauth*.

```

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal javax.security.auth.x500.X500Principal
"CN=operator,O=acme,OU=OGSample" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "read";
};

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
principal javax.security.auth.x500.X500Principal
"CN=manager,O=acme,OU=OGSample" {
    permission com.ibm.websphere.objectgrid.security.MapPermission "Grid.Map1", "all";
};

```

Neste arquivo, o código base `http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction` é uma URL reservada especialmente para o ObjectGrid. Todas as permissões do ObjectGrid concedidas aos principals devem usar esse código base especial. As seguintes permissões são designadas nesse arquivo:

- A primeira instrução de concessão concede permissão `read` para o principal `"CN=operator,O=acme,OU=OGSample"`. O usuário `"CN=operator,O=acme,OU=OGSample"` possui permissão somente leitura para o mapa `Map1` da instância `Grid` do ObjectGrid.

- A segunda instrução de concessão concede a permissão de mapa 'all' para o principal do "CN=manager,O=acme,OU=OGSample". O usuário "CN=manager,O=acme,OU=OGSample" possui permissão all para o mapa Map1 na instância Grid do ObjectGrid.

### Configurando o Arquivo de Política de Autorização JAAS Usando as Propriedades da JVM:

Use as seguintes etapas para configurar as propriedades JVM para os servidores xs1 e xs2, que estão no cluster xsCluster. Se estiver usando uma topologia diferente da topologia de amostra usada neste tutorial, configure o arquivo em todos os servidores de contêiner.

1. No console administrativo, clique em **Servidores > Servidores de Aplicativos > *server\_name* > Gerenciamento Java e Processos > Definição de Processo > Java virtual machine**
2. Inclua os seguintes argumentos JVM genéricos:  
-Djava.security.auth.policy=*samples\_home*/security\_extauth/xsAuth3.policy
3. Clique em **OK** e salve as mudanças.

### Executando os Aplicativos de Amostra para Testar a Autorização:

É possível usar o mesmo aplicativo de amostra para testar as configurações de autorização. O usuário gerenciador possui todas as permissões no mapa Map1, inclusive permissões para exibir e incluir funcionários. O usuário operador pode apenas visualizar os funcionários porque esse usuário receber permissão somente leitura.

1. Reinicie todos os servidores de aplicativos que estão executando servidores de contêiner. Para este tutorial, reinicie os servidores xs1 e xs2.
2. Abra o aplicativo EmployeeManagementWeb. Em um navegador da web, abra `http://<host>:<port>/EmployeeManagementWeb/management.jsp`.
3. Efetue login no aplicativo usando qualquer nome de usuário e senha válidos.
4. Tente exibir um funcionário. Clique em **Exibir um Funcionário** e procure pelo endereço de e-mail authemp1@acme.com. Uma mensagem exibe que o usuário não pode ser localizado.
5. Incluir um funcionário. Clique em **Incluir um Funcionário**. Inclua o email authemp1@acme.com, o nome Joe e o sobrenome Doe e, em seguida, clique em **Enviar**. Uma mensagem exibe que o funcionário foi incluído.
6. Edite o arquivo `samples_home/security_extauth/client3.props`. Altere o valor de propriedade credentialGeneratorProps a partir do gerenciador manager1 para operador operator1. Depois de editar o arquivo, o servlet usará o nome do usuário "operador" e a senha "operator1" para se autenticar nos servidoresWebSphere eXtreme Scale.
7. Reinicie o cluster appCluster para selecionar as mudanças no arquivo `samples_home/security_extauth/client3.props`.
8. Tente exibir um funcionário. Clique em **Exibir um Funcionário** e procure pelo endereço de e-mail authemp1@acme.com. O funcionário é exibido.
9. Incluir um funcionário. Clique em **Incluir um Funcionário**. Inclua o email authemp2@acme.com, o nome Joe e o sobrenome Doe e, em seguida, clique em **Enviar**. A mensagem a seguir é exibida:

Ocorre uma exceção ao Incluir o funcionário. Consulte abaixo para obter mensagens de exceção detalhadas.

O texto da exceção detalhado é o seguinte:

```
java.security.AccessControlException: Acesso negado
(com.ibm.websphere.objectgrid.security.MapPermission Grid.Map1 insert)
```

Esta mensagem é exibida porque o usuário `operator` não tem permissão para inserir dados no mapa `Map1`.

Se você estiver executando com uma versão do WebSphere Application Server anterior à Versão 7.0.0.11, um erro `java.lang.StackOverflowError` será exibido no servidor de contêiner. Este erro é causado por um problema com o IBM Developer Kit. O problema é corrigido no IBM Developer Kit fornecido com o WebSphere Application Server Versão 7.0.0.11 e posterior.

#### Ponto de verificação de lições:

Nesta lição, você configurou a autorização ao designar permissões para usuários específicos.

## Módulo 5: Use o Utilitário `xscmd` para Monitorar as Grades de Dados e Mapas

O utilitário `xscmd` pode ser usado para mostrar as grades de dados e tamanhos de mapa principais da grade de dados `Grade`. A ferramenta `xscmd` usa o MBean para consultar todos os artefatos grade de dados, tais como os shards primários, os shards de réplica, servidores de contêiner, tamanhos de mapa, e outros dados.

Para este tutorial, o servidor de catálogos está em execução como um servidor Java SE independente. Os servidores de contêiner estão em execução nos servidores de aplicativos do WebSphere Application Server.

Para o servidor de catálogos, um servidor MBean é criado na Java virtual machine (JVM) padrão. Quando a ferramenta `xscmd` é usada no servidor de catálogos, a segurança do WebSphere eXtreme Scale é usada.

Para os servidores de contêiner, o tempo de execução do WebSphere eXtreme Scale registra o beans gerenciados (MBean) com o servidor MBean que é criado pelo tempo de execução do WebSphere Application Server. A segurança usada pela ferramenta `xscmd` é fornecida pela segurança MBean do WebSphere Application Server.

1. Usando uma ferramenta de linha de comandos, abra o diretório `DMGR_PROFILE/bin`.
2. Executar a ferramenta `xscmd`. Use os parâmetros `-c showPlacement -st P` como nos exemplos a seguir:

Linux      UNIX

```
xscmd.sh -c listObjectGridPlacement -cep localhost:16099 -g Grid -ms mapSet -sf P
-user manager -pwd manager1
```

Windows

```
xscmd.bat -c listObjectGridPlacement -cep localhost:16099 -g Grid -m mapSet -sf P
-user manager -pwd manager1
```

O nome do usuário e a senha são passados para o servidor de catálogos para autenticação.

3. Visualize os resultados do comando.

```
*** Showing all primaries for grid - Grid & mapset - mapSet
Partition Container Host Server
0 myCell102\myNode04\xs2_C-1 myhost.mycompany.com myCell102\myNode04\xs2
```



```

1 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
2 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
3 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2
4 myCell02\myNode04\xs2_C-1 myhost.mycompany.com myCell02\myNode04\xs2

```

4. Executar a ferramenta **xscmd**. Use o parâmetro **-c showMapSizes** como nos exemplos a seguir:

Linux

UNIX

```
xscmd.sh -c showMapSizes -cep localhost:16099 -g Grid -ms mapSet -user manager -pwd manager1
```

Windows

```
xscmd.bat -c showMapSizes -cep localhost:16099 -g Grid -ms mapSet -user manager -pwd manager1
```

O nome do usuário e a senha são passados para o servidor de catálogos para autenticação. Depois de executar o comando, será solicitado fornecer o ID de usuário e senha do WebSphere Application Server para autenticação no WebSphere Application Server. Você deve fornecer estas informações de login porque a opção **-c showMapSizes** obtém o tamanho de mapa de cada servidor de contêiner, o qual requer a segurança do WebSphere Application Server.

5. Opcional: É possível alterar o arquivo `PROFILE/properties/sas.client.props` para executar o comando sem o ID do usuário e a senha necessários. Altere a propriedade `com.ibm.CORBA.loginSource` a partir do prompt para propriedades e, em seguida, forneça o ID de usuário e a senha. Um exemplo das propriedades no arquivo `PROFILE/properties/sas.client.props` é o seguinte:

```

com.ibm.CORBA.loginSource=properties
# RMI/IIOP user identity
com.ibm.CORBA.loginUserId=Admin
com.ibm.CORBA.loginPassword=xxxxxx

```

6. Opcional: Se estiver usando o comando **xscmd** em uma instalação independente do WebSphere eXtreme Scale, você deverá incluir as opções a seguir:

- Se estiver usando a segurança do WebSphere eXtreme Scale:

```
-user
-pwd
```

- Se estiver usando a segurança do WebSphere eXtreme Scale com geração de credencial customizada:

```
-user
-pwd
-cgc
-cgp
```

- Se SSL estiver ativado:

```
-tt
-cxpv
-prot
-ks
-ksp
-kst
-ts
-tsp
-tst
```

Se a segurança do WebSphere eXtreme Scale e o SSL estiverem ativados, ambos os conjuntos de parâmetros serão necessários.

### Ponto de verificação de lições

Você usou a ferramenta **xscmd** para monitorar as grades de dados e os mapas em sua configuração.



---

## Tutorial: Executando Pacotes Configuráveis do eXtreme Scale na Estrutura do OSGi

A amostra OSGi é construída sobre as amostras do serializador do Google Protocol Buffers. Quando esse conjunto de lições for concluído, os plug-ins de amostra do serializador terão sido executados na estrutura OSGi.

### Objetivos do aprendizado

Esta amostra demonstra os pacotes configuráveis OSGi. O plug-in serializador é incidental e não é necessário. A amostra de OSGi está disponível na galeria de amostras do WebSphere eXtreme Scale. Você deve fazer o download da amostra e extraí-la no diretório *wxs\_home/samples*. O diretório raiz para a amostra OSGi é *wxs\_home/samples/OSGiProto*.

A amostra do serializador do Google Protocol Buffers está localizada no diretório *wxs\_home/samples/SerializerProto*.

A amostra do serializador Binary JSON (BSON) está localizada no diretório *wxs\_home/samples/SerializerBSON*.

Os exemplos de comando neste tutorial assumem que você está executando no sistema operacional UNIX. Você deve ajustar o exemplo de comando para ser executado em um sistema operacional Windows.

Depois de concluir as lições neste tutorial, você entenderá os conceitos de amostra do OSGi e saberá como concluir os seguintes objetivos:

- Instalar o pacote configurável do servidor WebSphere eXtreme Scale no contêiner OSGi para iniciar o servidor eXtreme Scale.
- Configurar seu ambiente de desenvolvimento do eXtreme Scale para executar o cliente de amostra.
- Usar o comando `xscmd` para consultar a classificação do serviço do pacote configurável da amostra, fazer upgrade dele para uma nova classificação de serviço e verificar a nova classificação de serviço.

### Tempo necessário

Esse módulo demora aproximadamente 60 minutos para ser concluído.

### Pré-requisitos

Além de fazer download e extrair as amostras do serializador, este tutorial também possui os pré-requisitos a seguir:

- Instalar e extrair o produto eXtreme Scale
- Configurar o Eclipse Equinox Environment

## Introdução: Iniciando e Configurando o Servidor e o Contêiner do eXtreme Scale para Executar Plug-ins na Estrutura do OSGi

Neste tutorial, inicie um servidor eXtreme Scale na estrutura do OSGi, inicie um contêiner do eXtreme Scale e ligue os plug-ins de amostra com o ambiente de tempo de execução do eXtreme Scale.

## Objetivos do aprendizado

Depois de concluir as lições neste tutorial, você entenderá os conceitos de amostra do OSGi e saberá como concluir os seguintes objetivos:

- Instalar o pacote configurável do servidor WebSphere eXtreme Scale no contêiner OSGi para iniciar o servidor eXtreme Scale.
- Configurar seu ambiente de desenvolvimento do eXtreme Scale para executar o cliente de amostra.
- Usar o comando xscmd para consultar a classificação do serviço do pacote configurável da amostra, fazer upgrade dele para uma nova classificação de serviço e verificar a nova classificação de serviço.

## Tempo necessário

Esse tutorial demora aproximadamente 60 minutos para ser concluído. Se você explorar outros conceitos relacionados a este tutorial, poderá demorar mais tempo para ele ser concluído.

## Nível de qualificação

Intermediário.

## Público

Desenvolvedores e administradores que desejam construir, instalar e executar os pacotes configuráveis do eXtreme Scale na estrutura do OSGi.

## Requisitos do Sistema

- Luminis OSGi Configuration Admin Command Line Client, versão 0.2.5
- Apache Felix File Install, versão 3.0.2
- Quando usar o Eclipse Gemini como o provedor de contêiner blueprint, os seguintes itens são necessários:
  - Eclipse Gemini Blueprint, versão 1.0.0
  - Spring Framework, versão 3.0.5
  - SpringSource AOP Alliance API, versão 1.0.0
  - SpringSource Apache Commons Logging, versão 1.1.1
- Quando usar o Aries Apache como o provedor do Blueprint Container, você deve ter os seguintes requisitos:
  - Apache Aries, captura instantânea mais recente
  - Biblioteca ASM
  - Criação de Log PAX

## Pré-requisitos

Para concluir este tutorial, você deve fazer o download da amostra e extraí-la no diretório `wxs_home/samples`. O diretório raiz para a amostra OSGi é `wxs_home/samples/OSGiProto`.

## Resultados Esperados:

Ao concluir este tutorial, você terá instalado os pacotes configuráveis de amostra e executado um cliente do eXtreme Scale para inserir dados na grade. Também

espera-se que esses pacotes configuráveis de amostra sejam consultados e atualizados usando os recursos dinâmicos fornecidos pelo contêiner OSGi.

## Módulo 1: Preparando para Instalar e Configurar os Pacotes Configuráveis do Servidor eXtreme Scale

Conclua este módulo para explorar os pacotes configuráveis de amostra OSGi e examine os arquivos de configuração que você usa para configurar o servidor eXtreme Scale.

### Objetivos do aprendizado

Depois de concluir as lições neste módulo, você entenderá os conceitos e saberá como concluir os seguintes objetivos:

- Localizar e explorar os pacotes configuráveis que estão incluídos na amostra OSGi.
- Examine os arquivos de configuração que são usados para configurar a grade e o servidor do eXtreme Scale.

### Lição 1.1: Entendendo os Pacotes Configuráveis OSGi de Amostra

Conclua esta lição para localizar e explorar os pacotes configuráveis que são fornecidos na amostra do OSGi.

#### Pacotes Configuráveis OSGi de Amostra:

Diferente dos pacotes configuráveis que estão configurados no arquivo `config.ini`, que é mostrado no tópico sobre como configurar o ambiente do Eclipse Equinox, os seguintes pacotes configuráveis adicionais são usados na amostra OSGi:

##### **objectgrid.jar**

O pacote configurável do tempo de execução do servidor WebSphere eXtreme Scale. Este pacote configurável está localizado no diretório `wxs_home/lib` directory.

##### **com.google.protobuf\_2.4.0a.jar**

O pacote configurável Google Protocol Buffers, versão 2.4.0a. Este pacote configurável está localizado no diretório `wxs_sample_osgi_root/lib`.

##### **ProtoBufSamplePlugins-1.0.0.jar**

Versão 1.0.0 do pacote configurável de plug-in do usuário com as implementações de plug-in `ObjectGridEventListener` e `MapSerializerPlugin`. Este pacote configurável está localizado no diretório `wxs_sample_osgi_root/lib`. Os serviços são configurados com a classificação de serviço 1.

Esta versão usa o XML Blueprint padrão para configurar os serviços de plug-in do eXtreme Scale. A classe de serviço é uma classe implementada pelo usuário na interface do WebSphere eXtreme Scale, `com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory`. A classe implementada pelo usuário cria um bean para cada solicitação e funciona de modo semelhante a um bean de protótipo com escopo definido.

##### **ProtoBufSamplePlugins-2.0.0.jar**

Versão 2.0.0 do pacote configurável de plug-in do usuário com as implementações de plug-in `ObjectGridEventListener` e `MapSerializerPlugin`

de amostra. Este pacote configurável está localizado no diretório *wxs\_sample\_osgi\_root/lib*. Os serviços são configurados com a classificação de serviço 2.

Esta versão usa o XML Blueprint padrão para configurar os serviços de plug-in do eXtreme Scale. A classe de serviço está usando uma classe integrada do WebSphere eXtreme Scale, `com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl`, que usa o serviço do `BlueprintContainer`. Usando a configuração XML blueprint padrão, os beans podem ser configurados como um escopo de protótipo ou escopo singleton. O bean não é configurado como escopo de shard.

#### **ProtoBufSamplePlugins-Gemini-3.0.0.jar**

Versão 3.0.0 do pacote configurável de plug-in do usuário com as implementações de plug-in `ObjectGridEventListener` e `MapSerializerPlugin`. Este pacote configurável está localizado no diretório *wxs\_sample\_osgi\_root/lib*. Os serviços são configurados com o serviço de classificação 3.

Esta versão usa o XML blueprint específico do Eclipse Gemini para configurar os serviços de plug-in do eXtreme Scale. A classe de serviço está usando uma classe integrada do WebSphere eXtreme Scale, `com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl`, que usa o serviço do `BlueprintContainer`. A maneira de configurar um bean do escopo de shard é usar uma abordagem específica do Gemini. Esta versão configura o bean `myShardListener` como um bean de escopo de shard ao fornecer `{http://www.ibm.com/schema/objectgrid}shard` como o valor do escopo e configurar um atributo fictício para que o escopo customizado seja reconhecido pelo Gemini. Isso ocorre devido ao seguinte problema do Eclipse: [https://bugs.eclipse.org/bugs/show\\_bug.cgi?id=348776](https://bugs.eclipse.org/bugs/show_bug.cgi?id=348776)

#### **ProtoBufSamplePlugins-Aries-4.0.0.jar**

Versão 4.0.0 do pacote configurável de plug-in do usuário com as implementações de plug-in `ObjectGridEventListener` e `MapSerializerPlugin` de amostra. Este pacote configurável está localizado no diretório *wxs\_sample\_osgi\_root/lib*. Os serviços são configurados com a classificação de serviço 4.

Esta versão usa o XML blueprint padrão para configurar os serviços de plug-in do eXtreme Scale. A classe de serviço está usando uma classe integrada do WebSphere eXtreme Scale, `com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactoryImpl`, que usa o serviço do `BlueprintContainer`. Ao usar a configuração XML blueprint padrão, os beans podem ser configurados usando um escopo customizado. Esta versão configura o `myShardListenerbean` como bean de shard com escopo definido ao fornecer `{http://www.ibm.com/schema/objectgrid}shard` como o valor do escopo.

#### **ProtoBufSamplePlugins-Activator-5.0.0.jar**

Versão 5.0.0 do pacote configurável de plug-in do usuário com as implementações de plug-in `ObjectGridEventListener` e `MapSerializerPlugin` de amostra. Este pacote configurável está localizado no diretório *wxs\_sample\_osgi\_root/lib*. Os serviços são configurados com o serviço de classificação 5.

Esta versão não é usada em todo o contêiner blueprint. Nesta versão, os serviços são registrados usando o registro de serviço OSGi. A classe de serviço é uma classe implementada pelo usuário para a interface do WebSphere eXtreme Scale,

com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory. A classe implementada pelo usuário cria um bean para cada solicitação. Ela funciona de modo semelhante a um bean de protótipo com escopo definido.

#### Ponto de verificação de lições:

Ao explorar os pacotes configuráveis que são fornecidos com a amostra OSGi, é possível entender melhor como desenvolver suas próprias implementações que serão executadas no contêiner OSGi.

Você aprendeu:

- Sobre os pacotes configuráveis que estão incluídos com a amostra OSGi
- O local desses pacotes configuráveis
- A classificação de serviço com a qual cada pacote configurável foi configurado

### Lição 1.2: Entender os Arquivos de Configuração do OSGi

A amostra OSGi inclui três arquivos de configuração. Esses arquivos podem ser usados para iniciar e configurar a grade e o servidor do WebSphere eXtreme Scale.

#### Arquivos de Configuração OSGi:

Nesta lição, os seguintes arquivos de configuração serão explorados:

- `collocated.server.properties`
- `protoBufObjectGrid.xml`
- `protoBufDeployment.xml`

#### `collocated.server.properties`

Uma configuração do servidor é necessária para iniciar um servidor. Quando o pacote configurável do servidor eXtreme Scale é iniciado, ele não inicia um servidor. Ele aguarda o PID de configuração, `com.ibm.websphere.xs.server`, ser criado com um arquivo de propriedades do servidor. Esse arquivo de propriedades do servidor especifica o nome do servidor, o número da porta e outras propriedades do servidor.

Na maioria dos casos, uma configuração é criada para configurar o arquivo de propriedades do servidor. Raramente você pode querer iniciar um servidor apenas com cada propriedade configurada para um valor padrão. Nesse caso, é possível criar uma configuração denominada `com.ibm.websphere.xs.server` com o valor configurado para padrão.

Para obter mais detalhes sobre o arquivo de propriedades de servidor, consulte o tópico Arquivo de Propriedades do Servidor.

A amostra OSGi inclui o arquivo de propriedades do servidor de amostra, `wxs_sample_osgi_root/server/properties/collocated.server.properties`. Esse arquivo de propriedades de amostra inicia um serviço de catálogo e um servidor de contêiner únicos no processo da estrutura do OSGi. Os clientes do eXtreme Scale se conectam à porta 2809 e os clientes do JMX se conectam à porta 1099. O conteúdo do arquivo de propriedades do servidor de amostra é:

```
serverName=collocatedServer
isCatalog=true
catalogClusterEndpoints=collocatedServer:localhost:6601:6602
```

```
traceSpec=ObjectGridOSGi=all=enabled
traceFile=logs/trace.log
listenerPort=2809
JMXServicePort=1099
```

## protoBufObjectGrid.xml

O arquivo XML do descritor do ObjectGrid protoBufObjectGrid.xml de amostra contém o conteúdo a seguir, com comentários removidos.

```
<objectGridConfig>
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">

  <objectGrids>
    <objectGrid name="Grid" txTimeout="15">

      <bean id="ObjectGridEventListener"
        osgiService="myShardListener"/>

      <backingMap name="Map" readOnly="false"
        lockStrategy="PESSIMISTIC" lockTimeout="5"
        copyMode="COPY_TO_BYTES"
        pluginCollectionRef="serializer"/>

    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="serializer">
      <bean id="MapSerializerPlugin"
        osgiService="myProtoBufSerializer"/>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Há dois plug-ins configurados neste arquivo XML do descritor do ObjectGrid:

### ObjectGridEventListener

O plug-in de nível de shard. Para cada instância do ObjectGrid, há uma instância de ObjectGridEventListener. Ela é configurada para usar o myShardListener de serviço do OSGi. Isso significa que quando a grade é criada, o plug-in ObjectGridEventListener usa o serviço OSGi myShardListener com a classificação de serviço mais alta disponível.

### MapSerializerPlugin

O plug-in de nível de mapa. Para o mapa de apoio denominado Map, há um plug-in MapSerializerPlugin configurado. Ele é configurado para usar o serviço OSGi myProtoBufSerializer. Isso significa que quando o mapa é criado, o plug-in MapSerializerPlugin usa o serviço myProtoBufSerializer com a maior classificação de serviço obtida disponível.

## protoBufDeployment.xml

O arquivo XML do descritor de implementação descreve a política de implementação para a grade denominada Grid, que usa cinco partições. Consulte o seguinte código de exemplo desse arquivo XML:

```
<deploymentPolicy>
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

  <objectgridDeployment objectgridName="Grid">
    <mapSet name="MapSet" numberOfPartitions="5">
      <map ref="Map"/>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

## blueprint.xml

Como alternativa para o uso do arquivo `collocated.server.properties` em conjunto com PID de configuração, com `ibm.websphere.xs.server`, o XML do ObjectGrid e os arquivos XML de implementação podem ser incluídos em um pacote configurável OSGi, junto com um arquivo XML blueprint conforme mostrado no exemplo a seguir :

```
<blueprint>
  xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
  xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
  default-activation="lazy">

  <objectgrid:server id="server" isCatalog="true"
    name="server"
    tracespec="ObjectGridOSGi=all=enabled"
    tracefile="C:/Temp/logs/trace.log"
    workingDirectory="C:/Temp/working"
    jmxport="1099">
    <objectgrid:catalog host="localhost" port="2809"/>
  </objectgrid:server>

  <objectgrid:container id="container"
    objectgridxml="/META-INF/objectgrid.xml"
    deploymentxml="/META-INF/deployment.xml"
    server="server"/>
</blueprint>
```

### Ponto de verificação de lições:

Nesta lição, você aprendeu sobre os arquivos de configuração que são usados na amostra OSGi. Agora, quando iniciar e configurar a grade e o servidor eXtreme Scale, você entenderá quais arquivos estão sendo usados nestes processos e como esses arquivos interagem com seus plug-ins na estrutura do OSGi.

## Módulo 2: Instalando e Iniciando Pacotes Configuráveis do eXtreme Scale na Estrutura do OSGi

Utilize os módulos nessas lições para instalar o pacote configurável do servidor eXtreme Scale no contêiner OSGi e para iniciar o servidor WebSphere eXtreme Scale.

Iniciar o servidor na estrutura do OSGi não significa que seus pacotes configuráveis OSGi estão prontos para execução. Você deve configurar as propriedades do servidor e os contêineres para que os pacotes configuráveis OSGi que forem instalados sejam reconhecidos e executados corretamente.

### Objetivos do aprendizado

Depois de concluir as lições neste módulo, você entenderá os conceitos e saberá como concluir as seguintes tarefas:

- Instalar os pacotes configuráveis do eXtreme Scale usando o console do Equinox OSGi.
- Configurar o servidor eXtreme Scale.
- Configure o contêiner do eXtreme Scale.
- Iniciar pacotes configuráveis de amostra do eXtreme Scale.

### Pré-requisitos

Para concluir este módulo, as tarefas a seguir são necessárias antes de iniciar:

- Instalar e extrair o produto eXtreme Scale



- Configurar o Eclipse Equinox Environment

Você também deve preparar para acessar os seguintes arquivos para concluir as lições neste módulo:

- Pacote Configurável objectgrid.jar. Instale esse pacote configurável do eXtreme Scale.
- Arquivo collocated.server.properties. Inclua as propriedades do servidor nesse arquivo de configuração.
- Espera-se que os seguintes pacotes configuráveis sejam instalados e iniciados:
- Pacote configurável protobuf-java-2.4.0a-bundle.jar
- Pacote configurável ProtoBufSamplePlugins-1.0.0.jar
- Pacote configurável ProtoBufSamplePlugins-2.0.0.jar

## Lição 2.1: Iniciar o Console e Instalar o Pacote Configurável do Servidor eXtreme Scale

Nesta lição, use o console do Equinox OSGi para iniciar e instalar um WebSphere eXtreme Scale.

1. Execute o seguinte comando para iniciar o console Equinox OSGi:

```
cd equinox_root

java -jar
plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar
-console
```

2. Após iniciar o console OSGi, emita o comando ss no console e os seguintes pacotes configuráveis serão iniciados:

### Saída do Eclipse Gemini:

```
osgi> ss
Framework is launched.
id State Bundle
0 ACTIVE org.eclipse.osgi_3.6.1.R36x_v20100806
1 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
2 ACTIVE org.eclipse.osgi.util_3.2.100.v20100503
3 ACTIVE org.eclipse.equinox.cm_1.0.200.v20100520
4 ACTIVE com.springsource.org.apache.commons.logging_1.1.1
5 ACTIVE com.springsource.org.aopalliance_1.0.0
6 ACTIVE org.springframework.aop_3.0.5.RELEASE
7 ACTIVE org.springframework.asm_3.0.5.RELEASE
8 ACTIVE org.springframework.beans_3.0.5.RELEASE
9 ACTIVE org.springframework.context_3.0.5.RELEASE
10 ACTIVE org.springframework.core_3.0.5.RELEASE
11 ACTIVE org.springframework.expression_3.0.5.RELEASE
12 ACTIVE org.apache.felix.fileinstall_3.0.2
13 ACTIVE net.luminis.cmc_0.2.5
14 ACTIVE org.eclipse.gemini.blueprint.core_1.0.0.RELEASE
15 ACTIVE org.eclipse.gemini.blueprint.extender_1.0.0.RELEASE
16 ACTIVE org.eclipse.gemini.blueprint.io_1.0.0.RELEASE
```

### Saída do Apache Aries:

```
osgi> ss
Framework is launched.
id State Bundle
0 ACTIVE org.eclipse.osgi_3.6.1.R36x_v20100806
1 ACTIVE org.eclipse.osgi.services_3.2.100.v20100503
2 ACTIVE org.eclipse.osgi.util_3.2.100.v20100503
3 ACTIVE org.eclipse.equinox.cm_1.0.200.v20100520
4 ACTIVE org.ops4j.pax.logging.pax-logging-api_1.6.3
5 ACTIVE org.ops4j.pax.logging.pax-logging-service_1.6.3
6 ACTIVE org.objectweb.asm.all_3.3.0
7 ACTIVE org.apache.aries.blueprint_0.3.2.SNAPSHOT
```

```
8 ACTIVE org.apache.aries.util_0.4.0.SNAPSHOT
9 ACTIVE org.apache.aries.proxy_0.4.0.SNAPSHOT
10 ACTIVE org.apache.felix.fileinstall_3.0.2
11 ACTIVE net.luminis.cmc_0.2.5
```

3. Instale o pacote configurável `objectgrid.jar`. Para iniciar um servidor na Java virtual machine (JVM), é necessário instalar um pacote configurável do servidor eXtreme Scale. Este pacote configurável do servidor eXtreme Scale pode iniciar um servidor e criar contêineres. Use o comando a seguir para instalar o arquivo `objectgrid.jar`:

```
osgi> install file:///wxs_home/lib/objectgrid.jar
```

Consulte o seguinte exemplo:

```
osgi> install
file:///opt/wxs/ObjectGrid/lib/objectgrid.jar
```

O Equinox exibe o ID do pacote configurável, por exemplo:

```
Bundle id is 19
```

**Lembre-se:** Seu ID do pacote configurável pode ser diferente. O caminho do arquivo deve ser uma URL absoluta para o caminho do pacote configurável. Caminhos relativos não são suportados.

#### Ponto de verificação de lições:

Nesta lição, você usou o console do Equinox OSGi para instalar o pacote configurável `objectgrid.jar`, o qual você usará para iniciar um servidor e criar um contêiner posteriormente neste tutorial.

### Lição 2.2: Customizar e Configurar o Servidor eXtreme Scale

Use essa lição para customizar e incluir as propriedades do servidor para o servidor WebSphere eXtreme Scale.

1. Edite o arquivo `wxs_sample_osgi_root/server/properties/collocated.server.properties`.
  - a. Altere a propriedade do `workingDirectory` para `equinox_root`.
  - b. Altere a propriedade do `traceFile` para `equinox_roo/logs/trace.log`.
2. Salve o arquivo.
3. Insira as seguintes linhas de código no console OSGI para criar a configuração do servidor a partir do arquivo:

```
osgi> cm create com.ibm.websphere.xs.server
```

```
osgi> cm put com.ibm.websphere.xs.server
objectgrid.server.props
wxs_sample_osgi_root/server/properties/collocated.server.properties
```

4. Para visualizar a configuração, execute o seguinte comando:

```
osgi> cm get com.ibm.websphere.xs.server
Configuration for service (pid) "com.ibm.websphere.xs.server"
(bundle location = null)
key value
-----
objectgrid.server.props objectgrid.server.props
```

#### Ponto de verificação de lições:

Nesta lição, você editou o arquivo `wxs_sample_osgi_root/server/properties/collocated.server.properties` para especificar as configurações do servidor, como o diretório de trabalho e o local para os arquivos de log de rastreamento.

### Lição 2.3: Configurar o Contêiner do eXtreme Scale

Conclua esta lição para configurar um contêiner, que inclui o arquivo descritor XML e o arquivo XML de implementação do ObjectGrid do WebSphere eXtreme Scale. Esses arquivos incluem a configuração para a grade e sua topologia.

Para criar um contêiner, primeiro crie um serviço de configuração usando o factory de serviço gerenciado pelo número de identificação do processo (PID), `com.ibm.websphere.xs.container`. A configuração do serviço é um factory de serviço gerenciado para que seja possível criar vários PIDs de serviço a partir do PID do factory. Em seguida, para iniciar o serviço do contêiner, configure os PIDs `objectgridFile` e `deploymentPolicyFile` para cada PID de serviço.

Conclua as seguintes etapas para customizar e incluir as propriedades do servidor para a estrutura do OSGi:

1. No console OSGI, insira o seguinte comando para criar o contêiner a partir do arquivo:

```
osgi> cm createf com.ibm.websphere.xs.container
PID: com.ibm.websphere.xs.container-1291179621421-0
```

2. Insira o seguinte comando para ligar o PID recém-criado aos arquivos XML do ObjectGrid.

**Lembre-se:** O número de PID será diferente do número que está incluído neste exemplo.

```
osgi> cm put com.ibm.websphere.xs.container-1291179621421-0
objectgridFile wxs_sample_osgi_root/server/META-INF/protoBufObjectgrid.xml
```

```
osgi> cm put com.ibm.websphere.xs.container-1291179621421-0
deploymentPolicyFile wxs_sample_osgi_root/server/META-INF/protoBufDeployment.xml
```

3. Use o seguinte comando para exibir a configuração:

```
osgi> cm get com.ibm.websphere.xs.container-1291760127968-0
Configuration for service (pid) "com.ibm.websphere.xs.container-1291760127968-0"
(bundle location = null)
```

```
key value
-----
deploymentPolicyFile /opt/wxs/ObjectGrid/samples/OSGiProto/server/META-INF/protoBufDeployment.xml
objectgridFile       /opt/wxs/ObjectGrid/samples/OSGiProto/server/META-INF/protoBufObjectgrid.xml
service.factoryPid   com.ibm.websphere.xs.container
service.pid          com.ibm.websphere.xs.container-1291760127968-0
```

#### Ponto de verificação de lições:

Nesta lição, você criou um serviço de configuração, que foi usado para criar um contêiner do eXtreme Scale. Como os arquivos XML do ObjectGrid contêm a configuração para a grade e sua topologia, foi necessário vincular o contêiner criado para esses arquivos XML do ObjectGrid. Com essa configuração, o contêiner do eXtreme Scale pode reconhecer os pacotes configuráveis do OSGi que serão executados posteriormente neste tutorial.

### Lição 2.4: Instalar o Google Protocol Buffers e os Pacotes Configuráveis do Plug-in de Amostra

Conclua este tutorial para instalar o pacote configurável `protobuf-java-2.4.0a-bundle.jar` e o pacote configurável de plug-in `ProtoBufSamplePlugins-1.0.0.jar` usando o console do Equinox OSGi.

Conclua as seguintes etapas para instalar o pacote configurável Google Protocol Buffers.

No console OSGi, insira o seguinte comando para instalar o pacote configurável:  
osgi> install file:///wxs\_sample\_osgi\_root/common/lib/com.google.protobuf\_2.4.0a.jar

A saída a seguir é exibida:

Bundle ID is 21

### Visão Geral dos Pacotes Configuráveis do Plug-in de Amostra:

A amostra do OSGi inclui cinco pacotes configuráveis que incluem os plug-ins do eXtreme Scale, dentre eles um plug-in ObjectGridEventListener e MapSerializerPlugin customizado. O plug-in MapSerializerPlugin usa a amostra do Google Protocol Buffers e as mensagens fornecidas pela amostra do MapSerializerPlugin.

Os seguintes pacotes configuráveis estão localizados no diretório *wxs\_sample\_osgi\_root/lib*: ProtoBufSamplePlugins-1.0.0.jar e o ProtoBufSamplePlugins-2.0.0.jar.

O arquivo blueprint.xml possui o seguinte conteúdo com comentários removidos:

```
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">
<bean id="myShardListener" class="com.ibm.websphere.samples.xs.proto.osgi.MyShardListenerFactory"/>
<service ref="myShardListener" interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory" ranking="1">
</service>

<bean id="myProtoBufSerializer" class="com.ibm.websphere.samples.xs.proto.osgi.ProtoMapSerializerFactory">
<property name="keyType" value="com.ibm.websphere.samples.xs.serializer.app.proto.DataObjects1$0OrderKey" />
<property name="valueType" value="com.ibm.websphere.samples.xs.serializer.app.proto.DataObjects1$0Order" />
</bean>

<service ref="myProtoBufSerializer" interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory"
ranking="1">
</service>
</blueprint>
```

O arquivo XML Blueprint exporta dois serviços, myShardListener e myProtoBufSerializer. Esses dois serviços são referenciados no arquivo protoBufObjectgrid.xml.

### Instalar o Pacote Configurável do Plug-in de Amostra:

Conclua as seguintes etapas para instalar o pacote configurável ProtoBufSamplePlugins-1.0.0.jar.

Execute o seguinte comando no console do Equinox OSGi para instalar o pacote configurável do plug-in ProtoBufSamplePlugins-1.0.0.jar:

```
osgi> install file:///wxs_sample_osgi_root/common/lib/ProtoBufSamplePlugins-1.0.0.jar
```

A saída a seguir é exibida:

Bundle ID is 22

### Ponto de verificação de lições:

Nesta lição, você instalou o pacote configurável protobuf-java-2.4.0a-bundle.jar e o pacote configurável de plug-in ProtoBufSamplePlugins-1.0.0.jar.

## Lição 2.5: Iniciar os Pacotes Configuráveis do OSGi

O servidor WebSphere eXtreme Scale é incluído em um pacote configurável do OSGi. Conclua esta lição para instalar o pacote configurável do servidor eXtreme Scale e também outros pacotes configuráveis OSGi que forem instalados.

1. Inicie o pacote configurável do plug-in de amostra. Execute o seguinte comando no console do Equinox OSGi para iniciar o pacote configurável. Neste exemplo, o ID do pacote configurável do plug-in de amostra é 22.  
`osgi> start 22`
2. Inicie o pacote configurável do Google Protocol Buffers. Execute o seguinte comando no console do Equinox OSGi para iniciar o pacote configurável. Neste exemplo, o ID do pacote configurável do plug-in Google Protocol Buffers é 21.  
`osgi> start 21`
3. Inicie o pacote configurável do servidor. Execute o seguinte comando no console OSGi para iniciar o servidor. Neste exemplo, o ID do pacote configurável do servidor eXtreme Scale é 19.  
`osgi> start 19`

Depois de iniciar o servidor, o listener de event MyShardListener é iniciado e pronto para inserir ou atualizar registros. A seguinte saída pode ser visualizada no console OSGi para confirmar que o pacote configurável do plug-in foi iniciado com êxito:

```
SystemOut 0 MyShardListener@1253853884(version=1.0.0) order  
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Order$Builder  
@1abalaba(22) inserted
```

### Ponto de verificação de lições:

Nesta lição, você iniciou dois pacotes configuráveis de plug-in e o pacote configurável do servidor no contêiner do eXtreme Scale configurado para a estrutura do OSGi.

## Módulo 3: Executando o Cliente de Amostra do eXtreme Scale

O servidor WebSphere eXtreme Scale agora está em execução em um ambiente do OSGi. Conclua as etapas neste módulo para executar um cliente do WebSphere eXtreme Scale que insere dados na grade.

### Objetivos do aprendizado

Depois de concluir as lições neste módulo, você saberá como concluir as seguintes tarefas:

- Executar um aplicativo cliente que se conecta à grade e insere e recupera dados a partir dele.
- Inicie uma ordem usando um aplicativo cliente não OSGi.

### Pré-requisitos

Conclua Módulo 2: Instalando e Iniciando Pacotes Configuráveis do eXtreme Scale na Estrutura OSGi.

## Lição 3.1: Configurar o Eclipse para Executar o Cliente e Construir as Amostras

Conclua esta lição para importar o projeto Eclipse que será usado para executar o cliente e construir os plug-ins de amostra.

A amostra inclui um programa cliente Java SE que se conecta à grade e insere e recupera dados a partir dele. Ele também inclui projetos que podem ser usados para construir e reimplementar os pacotes configuráveis do OSGi.

O projeto fornecido foi testado com o Eclipse 3.x e posterior e requer somente a perspectiva de projetos de desenvolvimento Java padrão. Conclua as etapas a seguir para configurar o seu ambiente de desenvolvimento do WebSphere eXtreme Scale.

1. Abra o Eclipse para uma área de trabalho nova ou existente.
2. No menu Arquivo, selecione **Importar**.
3. Expanda a pasta Geral. Selecione **Pacotes Existentes na Área de Trabalho** e clique em **Avançar**.
4. No campo **Selecionar Diretório-Raiz**, digite ou navegue até o diretório *wxs\_sample\_osgi\_root*. Clique em **Concluir**. Vários novos projetos são exibidos na sua área de trabalho. Você deve corrigir diversos erros de construção ao definir a biblioteca de usuário do eXtreme Scale. Conclua as próximas etapas para definir a biblioteca de usuário.
5. No menu Janela, selecione **Preferências**.
6. Expanda a ramificação **Java > Caminho de Construção** e selecione **Bibliotecas de Usuário**.
7. Clique em **Novo**.
8. Digite **eXtremeScale** no campo **Nome da Biblioteca de Usuário** e clique em **OK**.
9. Selecione a nova biblioteca de usuário e clique em **Incluir JARs**.
  - a. Procure por e selecione o arquivo `objectgrid.jar` a partir do diretório *wxs\_install\_root/lib*. Clique em **OK**.
  - b. Para incluir a documentação da API para as APIs do ObjectGrid, selecione o local da documentação da API para o arquivo `objectgrid.jar` que você incluiu na etapa anterior. Clique em **Editar**.
  - c. Na caixa do caminho do local para a documentação da API, selecione o arquivo `Javadoc.zip` que está incluído no seguinte diretório:  
*wxs\_install\_root/docs/javadoc.zip*.

#### Ponto de verificação de lições:

Nesta lição, você importou o projeto Eclipse de amostra, definiu a biblioteca de usuário do eXtreme Scale e incluiu uma documentação da API de suporte no projeto de amostra. Agora o aplicativo cliente de amostra está pronto para ser iniciado.

### Lição 3.2: Iniciar um Cliente e Inserir Dados na Grade

Conclua esta lição para iniciar um cliente não OSGi e executar um aplicativo cliente.

O aplicativo de cliente Java é `com.ibm.websphere.samples.xs.proto.client.Client`.

Este cliente usa uma substituição do cliente, o arquivo descritor XML do ObjectGrid para substituir a configuração OSGi, para que o cliente possa ser executado em um ambiente não OSGi. Consulte o conteúdo a seguir do arquivo com comentários e cabeçalhos removidos. Algumas linhas de código são exibidas em diversas linhas para propósitos de formatação.

```
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
```

```

<objectGrids>
  <objectGrid name="Grid" txTimeout="15">
    <bean id="ObjectGridEventListener" className="" osgiService="" />
    <backingMap name="Map" readOnly="false"
      lockStrategy="PESSIMISTIC" lockTimeout="5"
      copyMode="COPY_TO_BYTES" pluginCollectionRef="serializer"/>
  </objectGrid>
</objectGrids>

<backingMapPluginCollections>
  <backingMapPluginCollection id="serializer">

<bean id="MapSerializer"
className="com.ibm.websphere.samples.xs.serializer.proto.ProtoMapSerializer"
osgiService="">
  <property name="keyType" type="java.lang.String"
value="com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$OrderKey" />
  <property name="valueType" type="java.lang.String"
value="com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$Order" />
</bean>
</backingMapPluginCollection>
</backingMapPluginCollections>
</objectGridConfig>

```

Conclua as seguintes etapas para iniciar o aplicativo cliente.

1. Use o exemplo de código a seguir para modificar os atributos da classe Cliente para refletir seu ambiente.

```

private String catHost = "localhost";
private int catListenerPort = 2809;
private String clientOGXML = "wxs_sample_osgi_root/client/META-INF/
clientProtoBufObjectgrid.xml";
private String gridName = "Grid";
private String mapName = "Map";

```

2. Execute o aplicativo cliente.

Quando você executa o aplicativo, a seguinte mensagem é exibida. A mensagem indica que uma solicitação foi inserida:

```

order
com.ibm.websphere.samples.xs.serializer.proto.DataObjects1$Or
der$Builder@5d165d16(5000000) inserted

```

#### Ponto de verificação de lições:

Nesta lição, você iniciou o aplicativo `com.ibm.websphere.samples.xs.proto.client.Client`, que produziu uma solicitação.

## Módulo 4: Consultando e Fazendo Upgrade do Pacote Configurável de Amostra

Conclua as lições neste módulo para usar o comando `xscmd` para consultar a classificação de serviço do pacote configurável da amostra, fazer upgrade dela para uma nova classificação de serviço e verificar o novo serviço de classificação.

Um projeto eclipse é fornecido como uma maneira conveniente de executar os aplicativos de amostra.

### Objetivos do aprendizado

Depois de concluir as lições deste módulo, você saberá como concluir as tarefas:

- Consultar a classificação de serviço atual para um serviço.



- Consultar a classificação atual de todos os serviços.
- Consultar todas as classificações disponíveis para um serviço.
- Consultar todas as classificações de serviço disponíveis.
- Use a ferramenta xscmd para verificar se as classificações específicas do serviço estão disponíveis.
- Atualizar as classificações de serviço dos serviços OSGi de amostra.

## Pré-requisitos

Executar o Módulo 3: Executando o Cliente de Amostra do eXtreme Scale.

### Lição 4.1: Consultar Classificações de Serviço

Conclua esta lição para consultar as classificações de serviço atuais, bem como as classificações de serviço que estão disponíveis para upgrade.

- Consultar a classificação de serviço atual para um serviço. Insira o seguinte comando para consultar a classificação de serviço atual usada para serviço, myShardListener, que é usado pelo ObjectGrid denominado Grid e pelo conjunto de mapas denominado MapSet.

1. Alterne para o diretório a seguir:

```
cd wxs_home/bin
```

2. Insira o seguinte comando para consultar a classificação de serviço atual para o serviço myShardListener.

```
./xscmd.sh -c osgiCurrent -g Grid -ms MapSet -sn myShardListener
```

A saída a seguir é exibida:

```
OSGi Service Name: myShardListener
ObjectGrid Name MapSet Name Server Name      Current Ranking
-----
Grid           MapSet      collocatedServer  1
```

```
CWXS10040I: The command osgiCurrent has completed successfully.
```

- Consultar a classificação atual de todos os serviços. Insira o seguinte comando para consultar as classificações de serviço atuais para todos os serviços usados pelo ObjectGrid denominado Grid e pelo conjunto de mapas denominado MapSet:

1. Alterne para o diretório a seguir:

```
cd wxs_home/bin
```

2. Insira o seguinte comando para consultar a classificação de serviço atual para todos os serviços.

```
./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
```

A saída a seguir é exibida:

```
OSGi Service Name  Current Ranking ObjectGrid Name MapSet Name Server Name
-----
myProtoBufSerializer  1           Grid           MapSet      collocatedServer
myShardListener      1           Grid           MapSet      collocatedServer
```

```
CWXS10040I: The command osgiCurrent has completed successfully.
```

- Consultar todas as classificações disponíveis para um serviço. Insira o seguinte comando para consultar todas as classificações de serviço disponíveis para o serviço denominado myShardListener.

1. Alterne para o diretório a seguir:

```
cd wxs_home/bin
```

2. Insira o seguinte comando para consultar todas as classificações disponíveis para um serviço.

```
./xscmd.sh -c osgiAll -sn myShardListener
```

A saída a seguir é exibida:

```
Server: collocatedServer
  OSGi Service Name Available Rankings
  -----
  myShardListener 1
```

Summary - All servers have the same service rankings.

CWXS10040I: The command osgiAll has completed successfully.

A saída é agrupada pelo servidor. Neste exemplo, apenas o seguinte servidor existe: collocatedServer.

- Consultar todas as classificações de serviço disponíveis. Insira o comando a seguir para consultar todas as classificações de serviço disponíveis para todos os serviços.

1. Alterne para o diretório a seguir:

```
cd wxs_home/bin
```

2. Insira o seguinte comando para consultar todas as classificações de serviço disponíveis.

```
./xscmd.sh -c osgiAll
```

A saída a seguir é exibida:

```
Server: collocatedServer
  OSGi Service Name Available Rankings
  -----
  myProtoBufSerializer 1
  myShardListener 1
```

Summary - All servers have the same service rankings.

- Instale e inicie a Versão 2 do pacote configurável do plug-in. No console OSGi servidor, instale um novo pacote configurável que contém uma nova versão da classe Order e o plug-in MapSerializerPlugin. Consulte Lição 2.4: Instalar o Google Buffers Protocol e os pacotes configuráveis do plug-in de amostra para obter detalhes sobre como instalar o pacote configurável ProtoBufSamplePlugins-2.0.0.jar.

1. Após a instalação, inicie o novo pacote configurável. Os serviços para seu novo pacote configurável estão disponíveis, mas eles ainda não são usados pelo servidor eXtreme Scale. Você deve executar uma solicitação de atualização de serviço para usar um serviço com uma versão específica.

- Agora, quando você consultar todas as classificações de serviço disponíveis novamente, o serviço de classificação 2 é incluído na saída.

1. Alterne para o diretório a seguir:

```
cd wxs_home/bin
```

2. Insira o seguinte comando para consultar todas as classificações de serviço disponíveis.

```
./xscmd.sh -c osgiAll
```

A saída a seguir é exibida:

```
Server: collocatedServer
  OSGi Service Name Available Rankings
  -----
```

```
myProtoBufSerializer 1, 2
myShardListener      1, 2
```

Summary - All servers have the same service rankings.

#### Ponto de verificação de lições:

Neste tutorial, você consultou atualmente todas as classificações de serviço, e classificações de serviço especificadas, disponíveis. Você também exibiu a classificação de serviço para um novo pacote configurável que você instalou e iniciou.

### Lição 4.2: Determinar Se Classificações de um Serviço Específico Estão Disponíveis

Conclua esta lição para determinar se as classificações de um serviço específico estão disponíveis para os nomes de serviço que forem especificados.

1. Insira o seguinte comando para determinar se o serviço `myShardListener`, com a classificação de serviço 2, e o serviço `myProtoBufSerializer`, com a classificação de serviço 2, estão disponíveis. A lista de classificação do serviço é passada usando a opção `-sr`.
  - a. Alterne para o diretório a seguir:

```
cd wxs_home/bin
```
  - b. Insira o seguinte comando para determinar se os serviços estão disponíveis:

```
./xscmd.sh -c osgiCheck -g Grid -ms MapSet -sr
"myShardListener;2,myProtoBufSerializer;2"
```

A saída a seguir é exibida:

```
CWXSII0040I: The command osgiCheck has completed successfully.
```

2. Insira o seguinte comando para determinar se o serviço `myShardListener`, com a classificação de serviço 2, e o serviço `myProtoBufSerializer`, com a classificação de serviço 3 estão disponíveis.
  - a. Alterne para o diretório a seguir:

```
cd wxs_home/bin
```
  - b. Insira o seguinte comando para determinar se os serviços estão disponíveis:

```
./xsadmin.sh -c osgiCheck -g Grid -ms MapSet -sr
"myShardListener;2,myProtoBufSerializer;3"
```

A saída a seguir é exibida:

```
Server  OSGi Service Unavailable Rankings
-----
collocatedServer myProtoBufSerializer 3
```

#### Ponto de verificação de lições:

Nesta lição, você especificou os serviços `myShardListener` e `myProtoBufSerializer`, junto com classificações de serviço específicas para determinar se essas classificações estavam disponíveis.

### Lição 4.3: Atualizar as Classificações do Serviço

Conclua esta lição para atualizar as classificações do serviço atual que você consultou.

1. Insira o comando a seguir atualiza as classificações dos serviços denominados `myShardListener` e `myProtoBufSerializer` o serviço de classificação 2. A lista de classificação do serviço é passada usando a opção `-sr`.
  - a. Alterne para o diretório a seguir:

```
cd wxs_home/bin
```

- b. Insira o seguinte comando para atualizar as classificações de serviço:

```
./xscmd.sh -c osgiUpdate -g Grid -ms MapSet  
-sr "myShardListener;2,myProtoBufSerializer;2"
```

A saída a seguir é exibida:

```
Update succeeded for the following service rankings:
```

```
Service Ranking
```

```
-----
```

```
myProtoBufSerializer 2
```

```
myShardListener 2
```

```
CWXSIO040I: The command osgiUpdate has completed successfully.
```

A seguinte saída é exibida no console do OSGi:

```
SystemOut 0 MyShardListener@326505334(version=2.0.0) order  
com.ibm.websphere.samples.xs.serializer.proto.DataObjects2$Order$Builder@  
22342234(34) updated
```

Observe que o serviço MyShardListener está agora na versão 2.0.0, que possui um serviço de classificação 2.

2. Se você executar o comando **xscmd** para consultar a classificação de serviço atual que está sendo usada para todos os serviços usados pelo ObjectGrid denominado Grid e o conjunto de mapa denominado MapSet.

- a. Alterne para o diretório a seguir:

```
cd wxs_home/bin
```

- b. Insira o seguinte comando para consultar as classificações de serviço para todos os serviços usados por Grid e MapSet:

```
./xscmd.sh -c osgiCurrent -g Grid -ms MapSet
```

A saída a seguir é exibida:

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
```

```
-----
```

```
myProtoBufSerializer 2 Grid MapSet collocatedServer
```

```
myShardListener 2 Grid MapSet collocatedServer
```

```
CWXSIO040I: The command osgiCurrent has completed successfully.
```

### Ponto de verificação de lições:

Nesta lição você atualizou as classificações de serviço para os serviços myShardListener e myProtoBufSerializer.



---

## Capítulo 4. Instalando o



O WebSphere eXtreme Scale é uma grade de dados de memória que pode ser usada para particionar, replicar e gerenciar dados e lógica de negócios dinamicamente por meio de vários servidores. Após determinar os propósitos e os requisitos da sua implementação, instale o eXtreme Scale em seu sistema.

### Antes de Iniciar

- Antes de iniciar a instalação, você deve ter um entendimento das arquiteturas de armazenamento em cache, do cache, e da integração, serialização, escalabilidade e disponibilidade do banco de dados do WebSphere eXtreme Scale. Consulte Visão Geral do Produto para obter informações adicionais.
- Planeje sua implementação do WebSphere eXtreme Scale. Para obter mais informações sobre as diferentes topologias de armazenamento em cache, informações de dimensionamento, etc., consulte Capítulo 2, “Planejamento”, na página 9.
- Verifique se seu ambiente atende aos pré-requisitos para instalar o eXtreme Scale. Consulte “Requisitos de Hardware e Software” na página 48 para obter mais informações.
- Para obter mais informações sobre ambientes e outros requisitos, consulte “Planejando para Instalação” na página 48.
- Se estiver instalando um upgrade em uma versão anterior do WebSphere eXtreme Scale, siga as etapas descritas no “Atualizando Servidores eXtreme Scale” na página 209.

---

## Visão Geral da Instalação

É possível usar a instalação completa ou do cliente para instalar o WebSphere eXtreme Scale em um ambiente independente ou do WebSphere Application Server.

### Tipos de Instalação

O instalador completo e o instalador do cliente separado que podem ser transferidos por download a partir do site de suporte fornecem uma variedade de opções de instalação. Quando usar o instalador completo, ambos os servidores de catálogos e servidores de contêiner podem ser executados. Nos servidores que estiverem executando aplicativos clientes e que acessam a grade de dados, é possível usar uma instalação apenas de cliente. Use a instalação do servidor ou a instalação do servidor e do cliente nos nós que executam servidores de catálogos ou servidores de contêiner.

- **Instalação Completa:**
  - Quando estiver instalando no WebSphere Application Server, é possível optar por instalar apenas o cliente ou ambos servidor e o cliente.
  - Quando estiver instalando em um ambiente independente, tanto o cliente quanto o servidor podem ser instalados. Se desejar instalar apenas o cliente, use a instalação do WebSphere eXtreme Scale Client.
- **Instalação do Cliente:**

É possível usar a instalação apenas de cliente nos nós que estão executando os aplicativos clientes. Para instalar apenas o cliente, é possível fazer download apenas do instalador do cliente para a plataforma apropriada a partir da seção de downloads no Site de Suporte.

## Opções do Ambiente

É possível instalar o WebSphere eXtreme Scale em um ambiente independente ou do WebSphere Application Server.

- **Ambiente WebSphere Application Server:**

Ao instalar o WebSphere eXtreme Scale nos nós em seu ambiente do WebSphere Application Server, é possível iniciar automaticamente os servidores de catálogos e os servidores de contêiner na mesma célula do seu gerenciador de implementação e em outros servidores de aplicativos.

- **Ambiente independente:**

Em uma instalação independente, o WebSphere eXtreme Scale é instalado em um ambiente que não tem o WebSphere Application Server. Com um ambiente independente, configure e inicie manualmente os processos do servidor de catálogos e do servidor de contêiner.

---

## Planejando para Instalação

Antes de instalar o produto, você deve considerar em seu ambiente.

### Topologias de Instalação

Com o WebSphere eXtreme Scale, é possível criar muitas topologias de instalação que incluem servidores independentes, WebSphere Application Server, ou ambos. Os exemplos a seguir são algumas topologias possíveis que podem ser criadas.

### Nó de Desenvolvimento

O cenário de instalação mais simples é criar um nó de desenvolvimento. Neste cenário, você instala a instalação de cliente e servidor do WebSphere eXtreme Scale uma vez no nó no qual deseja desenvolver seu aplicativo.



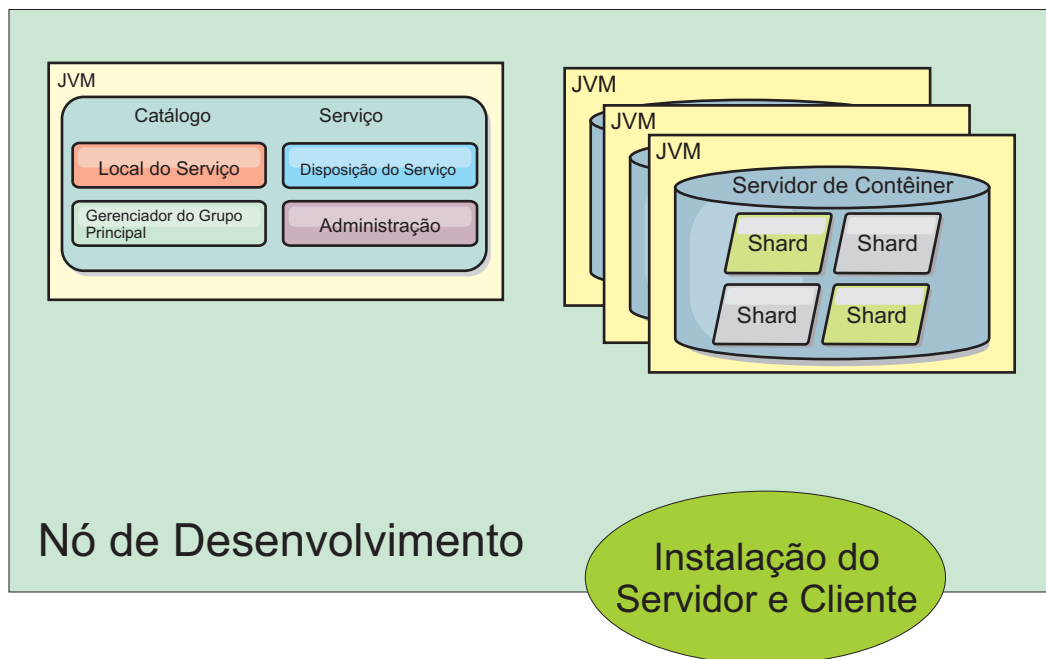


Figura 22. Nó de Desenvolvimento

Depois de concluir a instalação em seu modo de desenvolvimento, é possível configurar seu ambiente de desenvolvimento e começar a gravar seus aplicativos.

### Topologia Independente

Uma topologia independente consiste em servidores que não estão em execução no WebSphere Application Server. É possível criar muitas topologias independentes diferentes, mas a topologia a seguir é incluída como um exemplo. Nesta topologia, dois datacenters estão presentes. Em cada datacenter, instalações completas (cliente e servidor) e instalações apenas do cliente do WebSphere eXtreme Scale são instaladas nos servidores físicos. As instalações apenas do cliente estão nos nós que estão executando os aplicativos da web que estão usando a grade de dados. Estes nós não executam quaisquer servidores de catálogos ou de contêiner, portanto, a instalação de servidor não é necessária. Um link multimaster conecta os dois domínios de serviço de catálogo na configuração. O link multimaster ativa a replicação entre os shards nos servidores de contêiner nos diferentes datacenters.

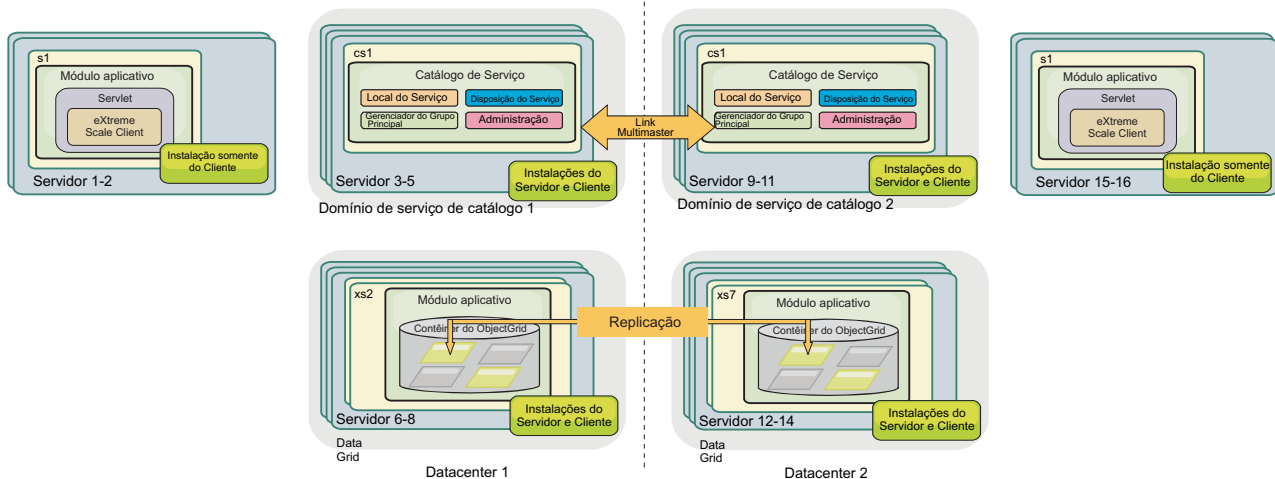


Figura 23. Topologia Independente com Dois Datacenters

Vantagens de usar uma topologia independente:

- Opções de integração flexíveis que podem ser integradas com estruturas de fornecedor e bibliotecas.
- Área de cobertura menor do que uma topologia do WebSphere Application Server.
- Menos requisitos de licenciamento de que uma topologia do WebSphere Application Server.
- Opções de Java Runtime Environment (JRE) expandidas.

## Topologia do WebSphere Application Server

Também é possível criar uma instalação que é executada inteiramente em um célula do WebSphere Application Server. Os clientes, servidores de catálogos e servidores de contêiner possuem, cada um, um cluster associado. Os nós que executam o aplicativo têm a instalação apenas do cliente. Os outros nós possuem a instalação de cliente e servidor.

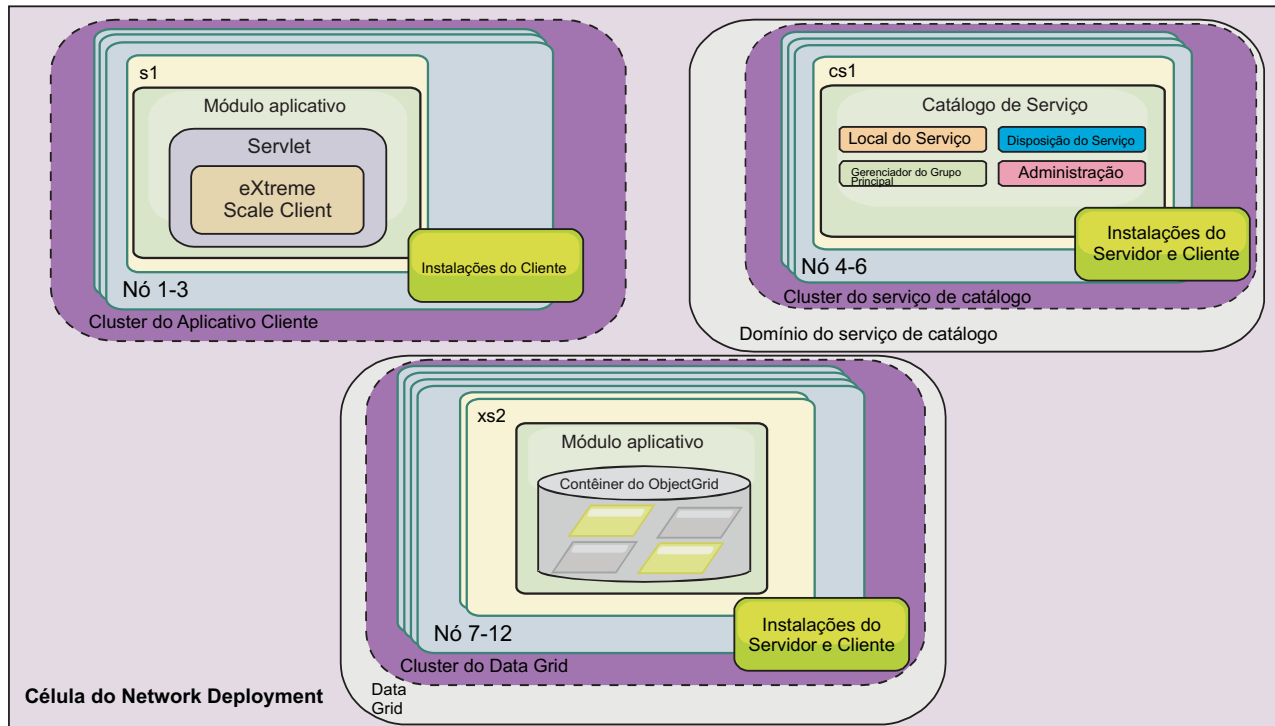


Figura 24. Exemplo de Topologia do WebSphere Application Server

Vantagens de usar uma topologia do WebSphere Application Server.

- Administração e configuração centralizadas e consistentes.
- Integração de segurança.
- Integração do aplicativo Java EE.
- Integração de Performance Monitoring Infrastructure (PMI).
- Integração com os componentes do WebSphere Application Server a seguir: cache OpenJPA L2, cache dinâmico e persistência de sessão HTTP.

### Topologia Mista

É possível criar uma topologia mista que contém o WebSphere Application Server e servidores independentes. No exemplo a seguir, os aplicativos clientes estão em execução na célula do WebSphere Application Server, enquanto os servidores de catálogos e servidores de contêiner estão em execução no modo independente.

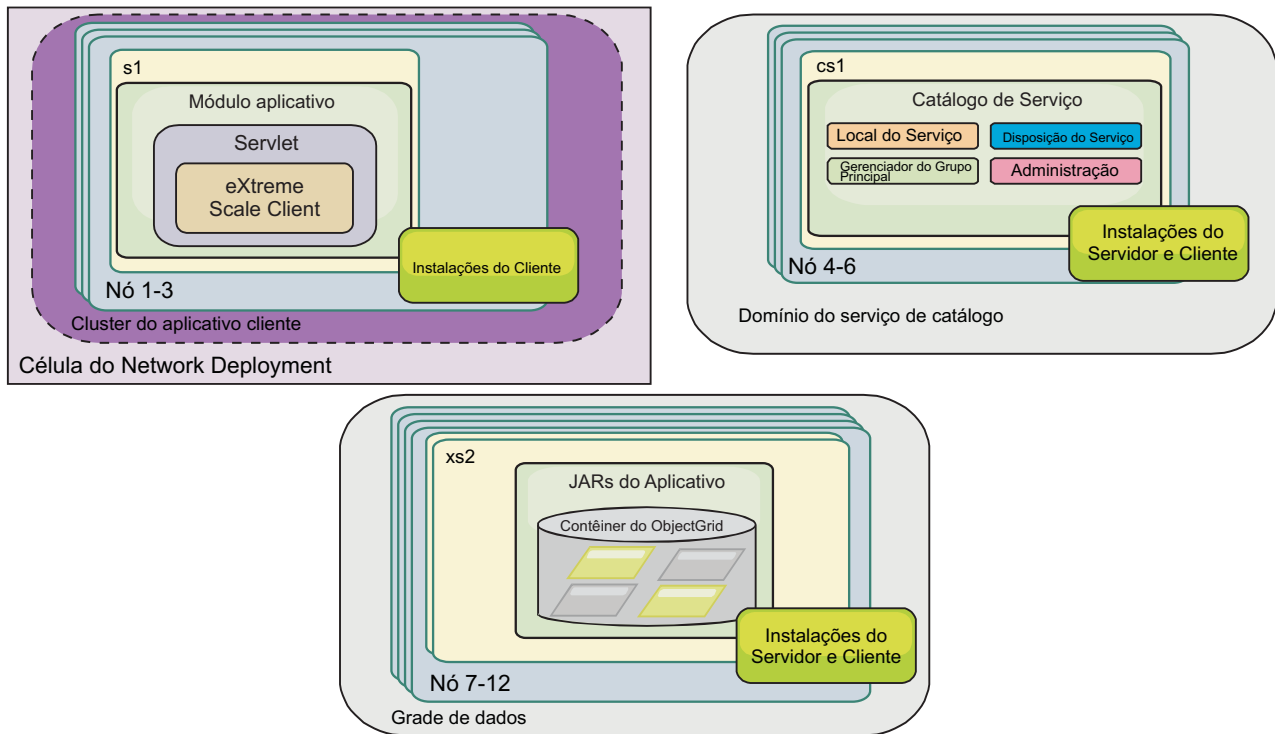


Figura 25. Exemplo de Topologia Mista

## Requisitos de Hardware e Software

Pesquisar uma visão geral de hardware e de requisitos do sistema operacional. Embora não seja necessário usar um nível específico de hardware ou sistema operacional para o WebSphere eXtreme Scale, as opções de hardware e software formalmente suportadas estão disponíveis na página Requisitos do Sistema do site de suporte do produto. Se existir um conflito entre o centro de informações e a página Requisitos do Sistema, as informações no website terão precedência. As informações de pré-requisito no centro de informações são fornecidas apenas como uma conveniência.

Consulte a Página de Requisitos do Sistema para obter o conjunto de oficial de requisitos de hardware e software.

Não é necessário instalar e implementar o eXtreme Scale em um nível específico do sistema operacional. Cada instalação do Java Platform, Standard Edition (Java SE) e do Java Platform, Enterprise Edition (Java EE) requer níveis ou correções de sistema operacional diferentes.

É possível instalar e implementar o produto nos ambientes do Java EE e do Java SE. Também é possível incluir em pacote configurável o componente do cliente com os aplicativos Java EE diretamente sem integrar com o WebSphere Application Server. O WebSphere eXtreme Scale suporta o Java SE 5 e posterior e o WebSphere Application Server Versão 6.1 e posterior.

## Requisitos de Hardware

O WebSphere eXtreme Scale não requer um nível específico de hardware. Os requisitos de hardware dependem do hardware suportado para a instalação do Java Platform, Standard Edition que é utilizado para executar o WebSphere eXtreme Scale. Se você estiver usando o eXtreme Scale com o WebSphere Application Server ou outra implementação do Java Platform, Enterprise Edition, os requisitos de hardware dessas plataformas são suficientes para o WebSphere eXtreme Scale.

## Requisitos do Sistema Operacional

- **Sem o console da web**

O eXtreme Scale não requer um nível de sistema operacional específico. Cada implementação de Java SE e Java EE necessita de diferentes níveis de sistema operacional ou correções para problemas que são descobertos durante o teste da implementação de Java. Os níveis que as implementações necessitam são suficientes para o eXtreme Scale.

- **Com o console da web**

Os seguintes requisitos se aplicarão a cada sistema operacional se usar o console:

- Linux: JVM de 32 bits ou 64 bits
- Linux PPC: Somente JVM de 32 bits
- Windows: Somente JVM de 32 bits
- AIX: Somente JVM de 32 bits

## Requisitos do Navegador da Web

O console da web suporta os seguintes navegadores da Web:

- Mozilla Firefox, versão 3.5.x e posterior
- Mozilla Firefox, versão 3.6.x e posterior
- Microsoft Internet Explorer, versão 7 ou 8

## Requisitos do WebSphere Application Server

- WebSphere Application Server Versão 6.1.0.39 ou posterior
- WebSphere Application Server Versão 7.0.0.19 ou posterior
- WebSphere Application Server Versão 8.0.0.1 ou posterior

Consulte as Correções recomendadas para o WebSphere Application Server para obter informações adicionais.

## Outros Requisitos do Servidor de Aplicativos

Outras implementações de Java EE podem usar o tempo de execução do eXtreme Scale como uma instância local ou como um cliente para servidores eXtreme Scale. Para implementar o Java SE, você deve usar a Versão 5 ou posterior.

## Considerações sobre o Java SE

O WebSphere eXtreme Scale requer o Java SE 5 ou superior. Em geral, as versões mais recentes do Java SE possuem melhor funcionalidade e desempenho.

## Versões Suportadas

O WebSphere eXtreme Scale pode ser usado com o Java SE 5 ou superior. A versão que deve ser usada deve ser atualmente suportada pelo fornecedor do Java Runtime Environment (JRE).

Um JRE totalmente suportado é instalado como parte das instalações do WebSphere eXtreme Scale e do WebSphere eXtreme Scale Client no diretório *wxs\_install\_root/java* e está disponível para ser usado por ambos clientes e servidores. Se estiver instalando o WebSphere eXtreme Scale no WebSphere Application Server, o JRE incluído na instalação do WebSphere Application Server poderá ser usado.

O WebSphere eXtreme Scale aproveita as funcionalidades do Java Development Kit (JDK) Versão 5 ou posterior conforme é disponibilizado. Geralmente, as versões mais recentes do Java Development Kit (JDK) e do Java SE têm melhor desempenho e funcionalidade.

Consulte Software Suportado para obter mais informações.

## Recursos do WebSphere eXtreme Scale Dependentes de Java

Tabela 4. Recursos que Requerem o Java SE 5 ou Java SE 6.

O WebSphere eXtreme Scale usa a funcionalidade que é introduzida no Java SE 5 ou no Java SE 6 para fornecer os seguintes recursos do produto.

Recurso	Suportado no Java SE 5 e superior	Suportado no Java SE 6 e superior
Anotações EntityManager API (Opcional: Também é possível usar os arquivos XML)	X	X
Java Persistence API (JPA): Carregador JPA, carregador de cliente do JPA e atualizador baseado em tempo do JPA	X	X
Despejo com base em memória (usa MemoryPoolMXBean)	X	X
Agentes de instrumentação: <ul style="list-style-type: none"><li>• <i>wxsizeagent.jar</i>: Aumenta a exatidão das métricas de mapa de bytes usados.</li><li>• <i>ogagent.jar</i>: Aumenta o desempenho de entidades de acesso a campo.</li></ul>	X	X
Console da web para monitoramento		X

## Considerações sobre o Java EE

Conforme você se prepara para integrar o WebSphere eXtreme Scale em um ambiente Java Platform, Enterprise Edition, considere determinados itens, como versões, opções de configuração, requisitos e limitações e implementação e gerenciamento de aplicativo.

### Executando Aplicativos eXtreme Scale em um Ambiente Java EE

Um aplicativo Java EE pode se conectar a um aplicativo eXtreme Scale remoto. Além disso, o ambiente do WebSphere Application Server suporta o início de um servidor eXtreme Scale conforme um aplicativo é iniciado no servidor de aplicativos.

Se você utilizar um arquivo XML para criar uma instância de ObjectGrid, e o arquivo XML estiver no módulo do arquivo enterprise archive (EAR), acesse o arquivo utilizando o método `getClass().getClassLoader().getResource("META-INF/objGrid.xml")` para obter um objeto URL para utilizar para criar uma instância de ObjectGrid. Substitua o nome do arquivo XML que você está utilizando na chamada de método.

É possível utilizar beans de inicialização para um aplicativo para autoinicializar uma instância do ObjectGrid quando o aplicativo for iniciado e para destruir a instância quando o aplicativo for parado. Um bean de inicialização é um bean de sessão stateless com um local remoto `com.ibm.websphere.startupservice.AppStartUpHome` e uma interface remota `com.ibm.websphere.startupservice.AppStartUp`. A interface remota possui dois métodos: o método `start` e o método `stop`. Utilize o método `start` para autoinicializar a instância e o método `stop` para destruir a instância. O aplicativo usa o método `ObjectGridManager.getObjectGrid` para manter uma referência à instância. Consulte as informações sobre como acessar um ObjectGrid com o `ObjectGridManager` no *Guia de Programação* para obter mais informações.

### Utilizando Carregadores de Classes

Quando módulos aplicativos que utilizam carregadores de classes diferentes compartilharem uma única instância de ObjectGrid em um aplicativo Java EE, verifique se os objetos estão armazenados no eXtreme Scale e se os plug-ins para o produto estão em um carregador comum no aplicativo.

### Gerenciando o Ciclo de Vida de Instâncias do ObjectGrid em um Servlet

Para gerenciar o ciclo de vida de uma instância do ObjectGrid em um servlet, é possível usar o método `init` para criar a instância e o método `destroy` para remover a instância. Se a instância estiver armazenada em cache, ela será recuperada e manipulada no código do servlet. Consulte as informações sobre como acessar um ObjectGrid com a interface `ObjectGridManager` no *Guia de Programação* para obter mais informações.

## Convenções de Diretório

As seguintes convenções de diretório devem ser usadas em todo o documento para referenciar diretórios especiais, como `wxs_install_root` e `wxs_home`. Acesse esses diretórios durante vários cenários diferentes, inclusive durante a instalação e durante o uso das ferramentas de linha de comandos.



### **wxs\_install\_root**

O diretório *wxs\_install\_root* é o diretório-raiz no qual os arquivos do produto WebSphere eXtreme Scale são instalados. O diretório *wxs\_install\_root* pode ser o diretório no qual o archive de teste é extraído ou o diretório no qual o produto WebSphere eXtreme Scale é instalado.

- Exemplo ao extrair o teste:

**Exemplo:** /opt/IBM/WebSphere/eXtremeScale

- Exemplo quando o WebSphere eXtreme Scale é instalado em um diretório independente:

**Exemplo:** /opt/IBM/eXtremeScale

- Exemplo quando o WebSphere eXtreme Scale é integrado com o WebSphere Application Server:

**Exemplo:** /opt/IBM/WebSphere/AppServer

### **wxs\_home**

O diretório *wxs\_home* é o diretório raiz das bibliotecas, amostras e componentes do produto WebSphere eXtreme Scale. Esse diretório é o mesmo que o diretório *wxs\_install\_root* quando o teste é extraído. Para instalações independentes, o diretório *wxs\_home* é o subdiretório ObjectGrid no diretório *wxs\_install\_root*. Para instalações que estão integradas ao WebSphere Application Server, esse diretório é o diretório optionalLibraries/ObjectGrid dentro do diretório *wxs\_install\_root*.

- Exemplo ao extrair o teste:

**Exemplo:** /opt/IBM/WebSphere/eXtremeScale

- Exemplo quando o WebSphere eXtreme Scale é instalado em um diretório independente:

**Exemplo:** /opt/IBM/eXtremeScale/ObjectGrid

- Exemplo quando o WebSphere eXtreme Scale é integrado com o WebSphere Application Server:

**Exemplo:** /opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid

### **was\_root**

O diretório *was\_root* é o diretório-raiz de uma instalação do WebSphere Application Server:

**Exemplo:** /opt/IBM/WebSphere/AppServer

### **restservice\_home**

O diretório *restservice\_home* é o diretório no qual as bibliotecas e amostras do serviço de dados REST do WebSphere eXtreme Scale estão localizadas. Este diretório é denominado restservice e é um subdiretório sob o diretório *wxs\_home*.

- Exemplo para implementações independentes:

**Exemplo:** /opt/IBM/WebSphere/eXtremeScale/ObjectGrid/restservice

- Exemplo para implementações integradas do WebSphere Application Server:

**Exemplo:** /opt/IBM/WebSphere/AppServer/optionalLibraries/ObjectGrid/restservice

### **tomcat\_root**

O *tomcat\_root* é o diretório-raiz da instalação do Apache Tomcat.

**Exemplo:** /opt/tomcat5.5

**wasce\_root**

O *wasce\_root* é o diretório-raiz da instalação do WebSphere Application Server Community Edition.

**Exemplo:** /opt/IBM/WebSphere/AppServerCE

**java\_home**

*java\_home* é o diretório-raiz de uma instalação de Java Runtime Environment (JRE).

**Exemplo:** /opt/IBM/WebSphere/eXtremeScale/java

**samples\_home**

O *samples\_home* é o diretório no qual os arquivos de amostra usados para os tutoriais são extraídos.

**Exemplo:** /wxs-samples/

**dvd\_root**

O diretório *dvd\_root* é o diretório-raiz do DVD que contém o produto.

**Exemplo:** dvd\_root/docs/

**equinox\_root**

O diretório *equinox\_root* é o diretório raiz da instalação da estrutura do Eclipse Equinox OSGi.

**Exemplo:**/opt/equinox

**user\_home**

O diretório *user\_home* é o local no qual arquivos de usuário são armazenados, tais como perfis de segurança.

**Windows** c:\Documents and Settings\*user\_name*

**UNIX** /home/*user\_name*

---

## Instalando o WebSphere eXtreme Scale com o Assistente de Instalação

É possível usar o assistente de instalação para instalar o WebSphere eXtreme Scale para configurações independentes ou do WebSphere Application Server.

### Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client com WebSphere Application Server

É possível instalar o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client em um ambiente no qual o WebSphere Application Server or WebSphere Application Server Network Deployment esteja instalado. É possível utilizar os recursos existentes do WebSphere Application Server or WebSphere Application Server Network Deployment para aprimorar os aplicativos do eXtreme Scale.

#### Antes de Iniciar

- Instale WebSphere Application Server ou WebSphere Application Server Network Deployment. Consulte Instalando o Ambiente de Serviço do Aplicativo para obter mais informações.
- Com base em qual versão você instalar, Versão 6.1 ou Versão 7.0, aplique o fix pack mais recente para WebSphere Application Server ou WebSphere Application Server Network Deployment para atualizar seu nível do produto. Consulte os Fix Packs mais Recentes para o WebSphere Application Server para obter mais informações.

- Verifique se o diretório de instalação de destino não contém uma instalação existente do WebSphere eXtreme Scale ou do WebSphere eXtreme Scale Client.
- Pare todos os processos em execução no ambiente do WebSphere Application Server ou do WebSphere Application Server Network Deployment. Consulte Utilitários de Linha de Comandos para obter informações adicionais sobre os comandos **stopManager**, **stopNode** e **stopServer**.

**CUIDADO:**

**Certifique-se de que quaisquer processos em execução sejam interrompidos. Se os processos em execução não forem interrompidos, a instalação continuará, criando resultados imprevisíveis e deixando a instalação em um estado indeterminado em algumas plataformas.**

- Se estiver instalando apenas o cliente, é possível usar o DVD para instalar o cliente ou fazer download do WebSphere eXtreme Scale Client para a plataforma específica a partir da seção de downloads no site de Suporte.

**Importante:** Quando você instala o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client, ele deve estar no mesmo diretório no qual você instalou o WebSphere Application Server. Por exemplo, se você instalou o WebSphere Application Server em `C:\was_root`, também deverá escolher `C:\was_root` como o diretório de destino para sua instalação do WebSphere eXtreme Scale ou do WebSphere eXtreme Scale Client.

## Sobre Esta Tarefa

Integre o eXtreme Scale com o WebSphere Application Server ou WebSphere Application Server Network Deployment para aplicar os recursos do eXtreme Scale nos aplicativos do Java Platform, Enterprise Edition. Os aplicativos Java EE hospedam grades de dados e acessam as grades de dados usando uma conexão do cliente.

## Procedimento

1. Use o assistente para concluir a instalação.
  - Execute o script a seguir para iniciar o assistente para a instalação completa do WebSphere eXtreme Scale. É possível escolher instalar apenas o cliente ou o servidor e o cliente:

```

- Linux UNIX dvd_root/install
- Windows dvd_root\install.bat

```

- Execute o script a seguir para iniciar o assistente para a instalação do WebSphere eXtreme Scale Client. Os arquivos de instalação estão no arquivo zip que você transferiu por download a partir da seção de downloads no site de Suporte:

```

- Linux UNIX root/WXS_Client/install
- Windows root\WXS_Client\install.bat

```

**Atenção:** Se usar convenções de nomenclatura uniformes (UNC) para identificar caminhos de arquivo em seu comando de instalação, nem todos os itens que você antecipa a instalação poderão ser instalados após a execução do comando. Para evitar problemas, mapeie o caminho de arquivo para uma unidade de rede. Execute o comando **install** na unidade mapeada. Usar uma unidade de rede mapeada assegura que todos os itens sejam instalados.

2. Siga os prompts no assistente.

O painel de recursos opcionais lista os recursos a partir dos quais é possível escolher instalar. Porém, os recursos não poderão ser incluídos incrementalmente no ambiente do produto depois que o produto for instalado. Se você escolher não instalar um recurso com a instalação inicial do produto, deverá desinstalar e reinstalar o produto para incluir o recurso.

O painel Aumento de Perfil lista os perfis existentes que podem ser selecionados para alterar com os recursos do eXtreme Scale. Porém, se você selecionar perfis existentes que já estão em uso, um painel de aviso será exibido. Para continuar com a instalação, pare os servidores que estão configurados nos perfis ou clique em **Voltar** para remover os perfis da sua seleção.

## Resultados

**Windows** Se estiver instalando o WebSphere eXtreme Scale Client no Windows, será possível ver o seguinte texto nos resultados da instalação:

Sucesso: A instalação do seguinte produto foi bem-sucedida:  
 WebSphere eXtreme Scale Client. Algumas etapas de configuração têm erros.  
 Para obter mais informações, consulte o seguinte arquivo de log:  
 <WebSphere Application Server install root>\logs\wxs\_client\install\log.txt"  
 Revise o log de instalação (log.txt) e o log de aumento do gerenciador de implementação

Se vir uma falha com o arquivo `iscdeploy.sh`, é possível ignorar o erro. Este erro não causa nenhum problema.

## O que Fazer Depois

- Se você estiver executando o WebSphere Application Server Versão 6.1 ou Versão 7.0, é possível usar o plug-in Profile Management Tool ou o comando **manageprofiles**. Para obter informações adicionais, consulte “Criando e Alterando Perfis para o WebSphere eXtreme Scale” na página 182.
- Verifique a instalação. Para obter informações adicionais, consulte “Verificando a Instalação” na página 204.
- Inicie a configuração do seu WebSphere eXtreme Scale ou a instalação do WebSphere eXtreme Scale Client. Para obter mais informações, consulte “Executando as Primeiras Etapas Após a Instalação” na página 205.

## Arquivos de Tempo de Execução para o WebSphere eXtreme Scale Integrado ao WebSphere Application Server

Os arquivos Java archive (JAR) são incluídos na instalação. É possível ver os arquivos JAR que são incluídos e o local para o qual eles estão instalados.

*Tabela 5. Arquivos de Tempo de Execução para o WebSphere eXtreme Scale. A seguinte tabela lista os arquivos Java archive (JAR) que são incluídos na instalação. O local da instalação é relativo ao diretório `wxs_home` que você escolher durante a instalação.*

Nome do Arquivo	Ambiente	Local da instalação	Descrição
wxsdynacache.jar	Cliente e Provedor	lib	O arquivo wxsdynacache.jar contém as classes necessárias a serem usadas com o provedor de cache dinâmico.
wsubjectgrid.jar	Local e cliente	lib	O wsubjectgrid.jar contém os tempos de execução local, do cliente e do servidor do eXtreme Scale.
ogagent.jar	Local, cliente e servidor	lib	O arquivo ogagent.jar contém as classes do tempo de execução que são necessárias para executar o agente de instrumentação Java que é usado com a API do EntityManager.

*Tabela 5. Arquivos de Tempo de Execução para o WebSphere eXtreme Scale (continuação). A seguinte tabela lista os arquivos Java archive (JAR) que são incluídos na instalação. O local da instalação é relativo ao diretório wxs\_home que você escolher durante a instalação.*

Nome do Arquivo	Ambiente	Local da instalação	Descrição
ogsip.jar	Servidor	lib	O arquivo ogsip.jar contém o tempo de execução de gerenciamento de sessões SIP (Session Initiation Protocol) do eXtreme Scale que é compatível com o WebSphere Application Server Versão 6.1.x.
sessionobjectgrid.jar	Cliente e Provedor	lib	O arquivo sessionobjectgrid.jar contém o tempo de execução de gerenciamento de sessões eXtreme Scale HTTP.
sessionobjectgridsip.jar	Servidor	lib	O arquivo sessionobjectgridsip.jar contém o tempo de execução de gerenciamento de sessões SIP do eXtreme Scale que é compatível com o WebSphere Application Server Versão 7.x.
wsogclient.jar	Local e cliente	lib	O arquivo wsogclient.jar é instalado quando você utiliza um ambiente contendo o WebSphere Application Server Versão 6.0.2 e posterior. Esse arquivo contém apenas os ambientes de tempo de execução local e do cliente.
wssizeagent.jar	Local, cliente e servidor	lib	O arquivo wssizeagent.jar é usado para fornecer informações mais precisas de dimensionamento de entrada de cache ao usar o Java Runtime Environment (JRE) Versão 1.5 ou mais recente.
oghibernate-cache.jar	Cliente e Provedor	optionalLibraries/ObjectGrid	O arquivo oghibernate-cache.jar contém o plug-in de cache de nível 2 do eXtreme Scale para JBoss Hibernate.
ogspring.jar	Local, cliente e servidor	optionalLibraries/ObjectGrid	O arquivo ogspring.jar contém classes de suporte para a integração de estrutura SpringSource Spring.
xsadmin.jar	Funções de Utilitário	optionalLibraries/ObjectGrid	O arquivo xsadmin.jar contém o utilitário de amostra de administração eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Cliente e Provedor	optionalLibraries/ObjectGrid/ endorsed	Este conjunto de arquivos inclui o tempo de execução Object Request Broker (ORB) que é usado para executar aplicativos em processos Java SE.
wxshyperic.jar	Funções de Utilitário	optionalLibraries/ObjectGrid/ hyperic/lib	O plug-in de detecção do servidor WebSphere eXtreme Scale para o agente de monitoramento SpringSource Hyperic.
restservice.ear	Cliente	optionalLibraries/ObjectGrid/ restservice/lib	O arquivo restservice.ear contém o archive corporativo de aplicativo de serviço de dados eXtreme Scale REST para ambientes WebSphere Application Server.
restservice.war	Cliente	optionalLibraries/ObjectGrid/ restservice/lib	O arquivo restservice.war contém o arquivamento Web de serviço de dados eXtreme Scale REST para servidores de aplicativos adquiridos de outro fornecedor.
splicerlistener.jar	Funções de Utilitário	optionalLibraries/ObjectGrid/ session/lib	O arquivo splicerlistener.jar contém o utilitário splicer para o filtro do gerenciador de sessões HTTP do eXtreme Scale.
splicer.jar	Funções de Utilitário	optionalLibraries/ObjectGrid/ legacy/session/lib	O splicer.jar contém o utilitário splicer Versão 7.0 para o filtro do gerenciador de sessões HTTP do eXtreme Scale.

*Tabela 6. Arquivos de Tempo de Execução para o WebSphere eXtreme Scale Client. A seguinte tabela lista os arquivos Java archive (JAR) que são incluídos na instalação. O local da instalação é relativo ao diretório wxs\_home que você escolher durante a instalação.*

Nome do Arquivo	Ambiente	Local da instalação	Descrição
wxsdynacache.jar	Cliente e Provedor	lib	O arquivo wxsdynacache.jar contém as classes necessárias a serem usadas com o provedor de cache dinâmico.
ogagent.jar	Local, cliente e servidor	lib	O arquivo ogagent.jar contém as classes do tempo de execução que são necessárias para executar o agente de instrumentação Java que é usado com a API do EntityManager.

*Tabela 6. Arquivos de Tempo de Execução para o WebSphere eXtreme Scale Client (continuação). A seguinte tabela lista os arquivos Java archive (JAR) que são incluídos na instalação. O local da instalação é relativo ao diretório `wxs_home` que você escolher durante a instalação.*

Nome do Arquivo	Ambiente	Local da instalação	Descrição
ogsip.jar	Servidor	lib	O arquivo ogsip.jar contém o tempo de execução de gerenciamento de sessões SIP (Session Initiation Protocol) do eXtreme Scale que é compatível com o WebSphere Application Server Versão 6.1.x.
sessionobjectgrid.jar	Cliente e Provedor	lib	O arquivo sessionobjectgrid.jar contém o tempo de execução de gerenciamento de sessões eXtreme Scale HTTP.
sessionobjectgridsip.jar	Servidor	lib	O arquivo sessionobjectgridsip.jar contém o tempo de execução de gerenciamento de sessões SIP do eXtreme Scale que é compatível com o WebSphere Application Server Versão 7.x.
wsogclient.jar	Local e cliente	lib	O arquivo wsogclient.jar é instalado quando você utiliza um ambiente contendo o WebSphere Application Server Versão 6.0.2 e posterior. Esse arquivo contém apenas os ambientes de tempo de execução local e do cliente.
wssizeagent.jar	Local, cliente e servidor	lib	O arquivo wssizeagent.jar é usado para fornecer informações mais precisas de dimensionamento de entrada de cache ao usar o Java Runtime Environment (JRE) Versão 1.5 ou mais recente.
oghibernate-cache.jar	Cliente e Provedor	optionalLibraries/ObjectGrid	O arquivo oghibernate-cache.jar contém o plug-in de cache de nível 2 do eXtreme Scale para JBoss Hibernate.
ogspring.jar	Local, cliente e servidor	optionalLibraries/ObjectGrid	O arquivo ogspring.jar contém classes de suporte para a integração de estrutura SpringSource Spring.
xsadmin.jar	Funções de Utilitário	optionalLibraries/ObjectGrid	O arquivo xsadmin.jar contém o utilitário de amostra de administração eXtreme Scale.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Cliente e Provedor	optionalLibraries/ObjectGrid/endorsed	Este conjunto de arquivos inclui o tempo de execução Object Request Broker (ORB) que é usado para executar aplicativos em processos Java SE.
wxshyperic.jar	Funções de Utilitário	optionalLibraries/ObjectGrid/hyperic/lib	O plug-in de detecção do servidor WebSphere eXtreme Scale para o agente de monitoramento SpringSource Hyperic.
restservice.ear	Cliente	optionalLibraries/ObjectGrid/restservice/lib	O arquivo restservice.ear contém o archive corporativo de aplicativo de serviço de dados eXtreme Scale REST para ambientes WebSphere Application Server.
restservice.war	Cliente	optionalLibraries/ObjectGrid/restservice/lib	O arquivo restservice.war contém o arquivamento Web de serviço de dados eXtreme Scale REST para servidores de aplicativos adquiridos de outro fornecedor.
splicerlistener.jar	Funções de Utilitário	optionalLibraries/ObjectGrid/session/lib	O arquivo splicerlistener.jar contém o utilitário splicer para o filtro do gerenciador de sessões HTTP do eXtreme Scale.
splicer.jar	Funções de Utilitário	optionalLibraries/ObjectGrid/legacy/session/lib	O splicer.jar contém o utilitário splicer Versão 7.0 para o filtro do gerenciador de sessões HTTP do eXtreme Scale.

## Uso do Plug-in Installation Factory para Criação e Instalação de Pacotes Customizados

Utilize o plug-in do IBM® Installation Factory para WebSphere eXtreme Scale para criar um pacote de instalação customizado (CIP) ou um pacote de instalação integrado (IIP). Um CIP contém um único pacote de instalação do produto e vários recursos opcionais. Um IIP combina um ou mais pacotes de instalação em um único fluxo de trabalho de instalação projetado.



## Antes de Iniciar

Antes de poder criar e instalar pacotes customizados para o eXtreme Scale, primeiro é necessário fazer o download dos seguintes produtos:

- IBM Installation Factory para WebSphere Application Server
- Plug-in do IBM Installation Factory plug-in para o WebSphere eXtreme Scale

## Sobre Esta Tarefa

Utilizando o Installation Factory, é possível criar um CIP ao combinar um único componente de produto com pacotes de manutenção, scripts de customização e outros arquivos. Ao criar um IIP, você agrega componentes individuais ou pacotes de instalação em um único pacote de instalação.

### Arquivo de Definição da Construção:

Um arquivo de definição de construção é um documento XML que especifica como construir e instalar um pacote de instalação customizado (CIP) ou um pacote de instalação integrado (IIP). O IBM Installation Factory para WebSphere eXtreme Scale lê os detalhes do pacote do arquivo de definição de construção para gerar um CIP ou um IIP.

Para poder criar um CIP ou um IIP, é necessário criar um arquivo de definição de construção para cada pacote customizado. O arquivo de definição de construção descreve quais componentes do produto ou pacotes de instalação devem ser instalados, o local do CIP ou do IIP, os pacotes de manutenção a serem incluídos, os scripts de instalação e outros arquivos escolhidos. Também é possível especificar no arquivo de definição de construção do IIP a ordem na qual o Installation Factory instalará cada pacote de instalação.

O assistente de Definição de Construção demonstra o processo de criação de um arquivo de definição de construção. Também é possível utilizar o assistente para modificar um arquivo de definição de construção existente. Cada painel do assistente de Definição de Construção solicita informações sobre um pacote customizado, como a identificação do pacote, o local de instalação da definição de construção e o local de instalação do pacote customizado. Todas essas informações são salvas no novo arquivo de definição de construção, ou modificadas e salvas em um arquivo de definição de construção existente. Para obter informações adicionais, consulte os painéis do Assistente de Definição de Construção do CIP e os painéis do Assistente de Definição de Construção do IIP.

Para criar apenas o arquivo de definição de construção, é possível utilizar a ferramenta da interface da linha de comandos para gerar o pacote customizado fora da GUI. Consulte “Instalação Silenciosa de um CIP ou IIP” na página 173 para obter informações adicionais.

### Criando um Arquivo de Definição de Construção e Gerando um CIP:

O plug-in do IBM Installation Factory para WebSphere eXtreme Scale gera um pacote de instalação customizado (CIP) de acordo com os detalhes especificados no arquivo de definição de construção. A definição de construção especifica o pacote do produto a ser instalado, o local do CIP, os pacotes de manutenção a serem incluídos na instalação, os arquivos de script de instalação e quaisquer arquivos adicionais a serem incluídos no CIP.



## Sobre Esta Tarefa

É possível usar o assistente de definição de Construção para criar um arquivo de definições de construção e gerar um CIP.

### Procedimento

1. Execute o seguinte script a partir do diretório `IF_HOME/bin` para iniciar o Installation Factory:

- `UNIX` `Linux` `ifgui.sh`
- `Windows` `ifgui.bat`

Clique no ícone **Nova Definição de Construção**.

2. Selecione o produto a ser incluído no arquivo de definições de construção e clique em **Concluir** para iniciar o assistente de definição de Construção.
3. Siga os prompts no assistente.

No painel Scripts de Instalação e Desinstalação, clique em **Incluir Scripts...** para preencher a tabela com todos os scripts de instalação customizados. Digite o local dos arquivos de script e limpe a caixa de opção para continuar se uma mensagem de erro for exibida. A operação é interrompida por padrão. Clique em **OK** para retornar ao painel.

### Resultados

Você criou e customizou o arquivo de definição de construção e gerou o CIP se tiver optado por trabalhar no modo conectado.

Se o assistente de definição de Construção não fornecer a opção de gerar o CIP a partir do arquivo de definições de construção, você ainda pode gerá-lo através da execução do script `ifcli.sh|bat` no diretório `IF_HOME/bin`.

### O que Fazer Depois

Instale o CIP. Consulte “Instalando um CIP” para obter mais informações.

*Instalando um CIP:*

Simplifique o processo de instalação do produto instalando um pacote de instalação customizado (CIP). Um CIP é uma imagem de instalação de produto único que pode incluir um ou mais pacotes de manutenção, scripts de configuração e outros arquivos.

### Antes de Iniciar

Para poder instalar um CIP, é necessário criar um arquivo de definição de construção para especificar quais opções serão incluídas no CIP. Consulte o “Criando um Arquivo de Definição de Construção e Gerando um CIP” na página 166 para obter informações adicionais.

## Sobre Esta Tarefa

Um CIP combina e instala um único componente do produto com pacotes de manutenção, scripts de customização e outros arquivos.

## Procedimento

1. Pare todos os processos que estão sendo executados na estação de trabalho que você está preparando para instalação. Para parar o gerenciador de implementação, execute o seguinte script:

- `Linux` `UNIX` `profile_root/bin/stopManager.sh`
- `Windows` `profile_root\bin\stopManager.bat`

Para parar os nós, execute o seguinte script:

- `Linux` `UNIX` `profile_root/bin/stopNode.sh`
- `Windows` `profile_root\bin\stopNode.bat`

2. Execute o seguinte script para iniciar a instalação:

- `Linux` `UNIX` `CIP_home/bin/install`
- `Windows` `CIP_home\bin\install.bat`

3. Siga os prompts no assistente para concluir a instalação.

O painel de recursos opcionais lista os recursos a partir dos quais é possível escolher instalar. Porém, os recursos não poderão ser incluídos incrementalmente no ambiente do produto depois que o produto for instalado. Se você escolher não instalar um recurso com a instalação inicial do produto, deverá desinstalar e reinstalar o produto para incluir o recurso.

O painel Aumento de Perfil lista os perfis existentes que podem ser selecionados para alterar com os recursos do eXtreme Scale. Porém, se você selecionar perfis existentes que já estão em uso, um painel de aviso será exibido. Para continuar com a instalação, pare os servidores que estão configurados nos perfis ou clique em **Voltar** para remover os perfis da sua seleção.

## Resultados

Você instalou com êxito o CIP.

## O que Fazer Depois

Se você estiver executando o WebSphere Application Server Versão 6.1 ou Versão 7.0, é possível usar o plug-in da Profile Management Tool ou o comando **manageprofiles** para criar e alterar perfis. Consulte “Criando e Alterando Perfis para o WebSphere eXtreme Scale” na página 182 para obter informações adicionais.

Se você alterar perfis do eXtreme Scale durante o processo de instalação, poderá implementar os aplicativos, iniciar um serviço de catálogo e iniciar os contêineres no seu ambiente do WebSphere Application Server. Consulte “Configurando o WebSphere eXtreme Scale com o WebSphere Application Server” na página 252 para obter mais informações.

*Instalando um CIP para Aplicar Manutenção a uma Instalação do Produto Existente:*

Você pode aplicar pacotes de manutenção a uma instalação existente do produto instalando um CIP (Pacote de Instalação Customizada). O processo de aplicar a manutenção a uma instalação existente com uma CIP é normalmente referida como *instalação slip*.

## Antes de Iniciar

Crie um arquivo de definições de construção para especificar quais opções incluir no CIP. Consulte o “Criando um Arquivo de Definição de Construção e Gerando um CIP” na página 166 para obter informações adicionais.

## Sobre Esta Tarefa

Ao aplicar a manutenção com um CIP que contém um pacote de atualizações, um fix pack ou ambos, todos os relatórios de análise do programa autorizado instalados anteriormente (APAR) são desinstalados pelo assistente. Se o CIP estiver no mesmo nível do produto, as APARs anteriormente instaladas permanecem somente se elas forem empacotadas no CIP. Para aplicar com sucesso a manutenção em uma instalação existente, você deve incluir os recursos instalados no CIP.

## Procedimento

1. Pare todos os processos que estão sendo executados na estação de trabalho que você está preparando para instalação. Para parar o gerenciador de implementação, execute o seguinte script:

- `Linux` `UNIX` `profile_root/bin/stopManager.sh`
- `Windows` `profile_root\bin\stopManager.bat`

Para parar os nós, execute o seguinte script:

- `Linux` `UNIX` `profile_root\bin\stopNode.sh`
- `Windows` `profile_root\bin\stopNode.bat`

2. Execute o seguinte script para iniciar a instalação:

- `Linux` `UNIX` `CIP_home/bin/install`
- `Windows` `CIP_home\bin\install.bat`

3. Siga os prompts no assistente para concluir a instalação.

O resumo da visualização da instalação relaciona a versão do produto resultante e todos os recursos aplicáveis e correções temporárias. Em seguida, o assistente aplica com êxito a manutenção e atualiza os recursos do produto.

## Resultados

Os arquivos binários do produto são copiados para o diretório `was_root/properties/version/nif/backup`. É possível usar o IBM Update Installer para desinstalar a atualização e restaurar sua estação de trabalho. Consulte o “Desinstalando Atualizações de CIP de uma Instalação Existente do Produto” para obter informações adicionais.

*Desinstalando Atualizações de CIP de uma Instalação Existente do Produto:*

É possível remover atualizações de CIP de uma instalação de produto existente sem remover o produto todo. Use o IBM Update Installer Versão 7.0.0.4 para desinstalar quaisquer atualizações do CIP. Esta tarefa também é chamada de uma *desinstalação slip*.

## Antes de Iniciar

Você precisa ter pelo menos uma cópia existente do produto instalada no sistema.

## Procedimento

1. Faça Download da Versão 7.0.0.4 do Update Installer a partir do seguinte site FTP:  
`ftp://ftp.software.ibm.com/software/websphere/cw/process_server/FEP/UPDI/7004`
2. Instale o Update Installer. Consulte Instalando o Update Installer para WebSphere Software no Centro de Informações do WebSphere Application Server para obter informações adicionais.
3. Desinstale todos os fix packs, pacotes de atualizações ou correções temporárias que você incluiu em seu ambiente após a instalação do CIP.
4. Desinstale todas as correções temporárias que tiver incluído na instalação slip. Este processo é igual a desinstalar um fix pack simples ou pacote de atualizações. Porém, a manutenção que foi incluída no CIP agora está incluída em uma única operação.
5. Desinstale o CIP utilizando o Update Installer. Os níveis de manutenção retornam ao estado pré-atualização e o CIP é denotado pelo identificador de CIP que é incluído como prefixo no nome do arquivo. O exemplo a seguir mostra como um CIP pode ser exibido de forma diferente de outros pacotes de manutenção regulares no painel de seleção do pacote de manutenção:

### CIP

`com.ibm.ws.cip.7000.wxs.primary.ext.pak`

## Resultados

Você removeu com êxito as atualizações de CIP de uma instalação existente do produto.

## Criando um Arquivo de Definição de Construção e Gerando um IIP:

O plug-in IBM Installation Factory para WebSphere eXtreme Scale gera um IIP baseado nas propriedades fornecidas pelo arquivo de definições de construção. O arquivo de definições de construção contém informações como quais pacotes de instalação incluir no IIP, a ordem em que o Installation Factory instala cada pacote e o local do IIP.

## Sobre Esta Tarefa

É possível usar o assistente de definição de Construção para criar um arquivo de definições de construção e gerar um IIP.

## Procedimento

1. Execute o seguinte script a partir do diretório `IF_HOME/bin` para iniciar o Installation Factory:
  - `UNIX Linux ifgui.sh`
  - `Windows ifgui.bat`
2. Clique no ícone **Criar Novo Pacote de Instalação Integrado** para iniciar o assistente de Definição de Construção.
3. Siga os prompts no assistente.
  - a. No painel Construir o IIP, selecione um pacote de instalação suportado na lista, e clique em **Incluir Instalador** para incluir o pacote de instalação no IIP. Um painel com o nome do pacote, identificador do pacote e as propriedades do pacote é exibido. Para visualizar informações específicas

sobre o pacote selecionado, clique em **Visualizar Informações do Pacote de Instalação**. Clique em **Modificar** para digitar o caminho de diretório para o pacote de instalação de cada sistema operacional. Se estiver incluindo um pacote de instalação para WebSphere Extended Deployment, selecione a caixa de opção, que fornece a opção para usar o mesmo pacote para todos os sistemas operacionais suportados. Clique em **OK** e retorne ao painel Construir o IIP. Uma chamada é criada por padrão.

- Para modificar o caminho do diretório para um pacote de instalação, selecione o pacote a partir dos Pacotes de Instalação usados na lista do IIP e clique em **Modificar**.
  - Para modificar uma chamada, selecione a chamada e clique em **Modificar**. Especifique o local de instalação padrão da chamada em cada sistema operacional. Especifique o local para o arquivo de resposta se selecionar uma instalação silenciosa como o modo de instalação padrão.
  - Clique em **Incluir Chamada** para incluir uma contribuição de chamada no pacote de instalação. É exibido um painel a partir do qual é possível especificar propriedades para a chamada.
  - Clique em **Remover** para remover os pacotes de instalação ou as chamadas.
4. Revise o resumo das seleções, selecione a opção **Salvar arquivo de definição de construção e gerar pacote de instalação integrado** e clique em **Concluir**. Alternativamente, é possível salvar o arquivo de definições de construção sem gerar o IIP. Com essa opção, você realmente gera o IIP fora do assistente executando o script `ifcli.bat | ifcli.sh` a partir do diretório `IF_home/bin/`.

## Resultados

Você criou e customizou o arquivo de definições de construção para um IIP.

## O que Fazer Depois

Instale o IIP.

*Instalando um IIP:*

Utilize o plug-in do IBM Installation Factory para WebSphere eXtreme Scale para instalar um pacote de instalação integrado (IIP). Um IIP combina um ou mais pacotes de instalação em um único fluxo de trabalho que você projeta.

## Antes de Iniciar

Para poder instalar um CIP, é necessário criar um arquivo de definição de construção para especificar quais opções serão incluídas no CIP. Consulte o “Criando um Arquivo de Definição de Construção e Gerando um IIP” na página 170 para obter informações adicionais.

## Sobre Esta Tarefa

Um IIP pode incluir um ou mais pacotes de instalação geralmente disponíveis, um ou mais CIPs e outros arquivos e diretórios opcionais. Ao instalar o IIP, você agrega vários pacotes de instalação, ou *contribuições*, em um único pacote, e depois instala as contribuições em uma ordem específica para concluir a instalação de ponta a ponta.

## Procedimento

1. Execute o seguinte script para iniciar o assistente:
  - `Linux` `UNIX` `IIP_home/bin/install`
  - `Windows` `IIP_home\bin\install.bat`
2. Clique em **Sobre** no painel Boas-vindas para visualizar os detalhes do IIP, como o identificador do pacote, os sistemas operacionais suportados e os pacotes de instalação incluídos.

**Optional:** Para modificar as opções de instalação de cada pacote, clique em **Modificar**.

**Optional:** Dois botões **Visualizar Log** são exibidos no painel do assistente. Para visualizar o log de cada pacote, clique no botão **Visualizar Log** exibido ao lado da tabela que lista os pacotes de instalação. Para visualizar os detalhes gerais do log do IIP, clique no botão **Visualizar Log** exibido ao lado das informações de status.

3. Selecione os pacotes de instalação a serem executados e clique em **Instalar**. É exibida uma lista de todas as contribuições na ordem de chamada que o IIP contém. Para designar que chamadas de contribuição não devem ser executadas durante a instalação, desmarque a caixa de opção localizada próxima ao campo **Nome da instalação**.

## Resultados

Você instalou com êxito um IIP.

*Modificando um Arquivo de Definição de Construção Existente de um IIP:*

É possível editar ou incluir nas propriedades de um IIP para customizar ainda mais a instalação.

## Sobre Esta Tarefa

Para alterar as propriedades de um IIP, modifique o arquivo de definição de construção existente.

## Procedimento

1. Execute o seguinte script a partir do diretório `IF_HOME/bin` para iniciar o Installation Factory:
  - `UNIX` `Linux` `ifgui.sh`
  - `Windows` `ifgui.bat`
2. Clique no ícone **Abrir Definição de Construção** e selecione o arquivo de definições de construção que deseja modificar.
3. Selecione as propriedades específicas do IIP que deseja modificar. A lista a seguir contém as possíveis modificações que podem ser feitas:
  - Alterar a seleção de modo atual. No modo conectado, crie a definição de construção que utilizará e, opcionalmente, gere o IIP, a partir da estação de trabalho atual. No modo desconectado, crie o arquivo de definição de construção a ser utilizado em outra estação de trabalho.
  - Incluir ou remover os sistemas operacionais existentes suportados pelo IIP.
  - Editar o identificador e a versão existentes do IIP.
  - Editar o local de destino do arquivo de definição de construção.

- Editar o local de destino do IIP.
- Mudar se um assistente de instalação deve ser exibido para o IIP. O assistente fornece informações sobre o IIP e as opções de instalação quando o IIP é executado.
- Incluir, remover e editar os pacotes de instalação contidos no IIP.

**Importante:** Se você tiver incluído um sistema operacional suportado e não tiver atualizado as propriedades do pacote de instalação no IIP, será exibida uma mensagem de aviso informando que as contribuições selecionadas não contêm os pacotes de instalação identificados de todos os sistemas operacionais suportados pelo IIP. Clique em **Sim** para continuar ou em **Não** para editar o pacote de instalação.

4. Revise o resumo das seleções, selecione **Salvar arquivo de definição de construção e gerar pacote de instalação integrado** e clique em **Concluir**.

#### **Instalação Silenciosa de um CIP ou IIP:**

É possível instalar silenciosamente um pacote de instalação customizado (CIP) ou um pacote de instalação integrado (IIP) para o produto usando um arquivo de resposta completo, que é configurado especificamente para atender suas necessidades, ou parâmetros passados para a linha de comandos.

#### **Antes de Iniciar**

Crie o arquivo de definições de construção para o CIP ou IIP. Consulte o “Criando um Arquivo de Definição de Construção e Gerando um CIP” na página 166 para obter informações adicionais.

#### **Sobre Esta Tarefa**

Uma instalação silenciosa utiliza o mesmo programa que a versão de interface gráfica com o usuário (GUI) utiliza. Entretanto, no lugar de exibir uma interface de assistente, a instalação silenciosa lê todas as suas respostas de um arquivo que você customiza ou dos parâmetros que você transmite para a linha de comandos. Se estiver instalando silenciosamente um IIP, é possível chamar uma contribuição com uma combinação de opções especificadas diretamente na linha de comandos, bem como opções que você especifica em um arquivo de resposta. No entanto, as opções de contribuição transmitidas à linha de comandos farão com que o instalador do IIP ignore todas as opções especificadas em um arquivo de resposta específico da contribuição. Consulte o Instalando um IIP Silenciosamente para obter mais informações.

**Nota:** É necessário especificar o nome completo do arquivo de resposta. Especificar o caminho relativo faz com que a instalação falhe sem nenhuma indicação de que ocorreu um erro.

#### **Procedimento**

1. Opcional: Se você optar por instalar o CIP ou IIP usando um arquivo de resposta, primeiro customize o arquivo.
  - a. Copie o arquivo de resposta `wxssetup.response.txt` do DVD do produto para a unidade de disco.
  - b. Abra e edite o arquivo de resposta no editor de texto de sua escolha. O arquivo inclui comentários para ajudá-lo no processo de configuração e deve incluir estes parâmetros:



- O contrato de licença
- O local da instalação do produto

**Dica:** O instalador usa o local que você seleciona para sua instalação para determinar onde sua instância do WebSphere Application Server está instalada. Se você instalar em um nó com múltiplas instâncias do WebSphere Application Server, defina claramente seu local.

c. Execute o script a seguir para iniciar seu arquivo de resposta customizado.

- `Linux` `UNIX` `install -options /absolute_path/response_file.txt -silent`
- `Windows` `install.bat -options C:\drive_path\response_file.txt -silent`

2. Opcional: Se você optar por instalar o CIP ou IIP passando determinados parâmetros para a linha de comandos, execute o script a seguir para iniciar a instalação:

- `Linux` `UNIX` `install -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=install_location`
- `Windows` `install.bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=install_location`

em que *install\_location* é o local de sua instalação do WebSphere Application Server existente.

3. Revise os registros de erros resultantes ou uma falha na instalação.

## Resultados

Você instalou silenciosamente o CIP ou IIP.

## O que Fazer Depois

Se você estiver executando o WebSphere Application Server Versão 6.1 ou Versão 7.0, é possível usar o plug-in da Profile Management Tool ou o comando **manageprofiles** para criar e alterar perfis.

Se você alterar perfis do eXtreme Scale durante o processo de instalação, poderá implementar os aplicativos, iniciar um serviço de catálogo e iniciar os contêineres no seu ambiente do WebSphere Application Server. Consulte “Configurando o WebSphere eXtreme Scale com o WebSphere Application Server” na página 252 para obter mais informações.

*Arquivo wxssetup.response.txt:*

É possível usar um arquivo completo de resposta para instalar o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client silenciosamente.

### CUIDADO:

Não inclua barras à direita, como / ou \, no final dos caminhos do local de instalação. Esses caminhos são especificados com o atributo installLocation. A inclusão de uma barra no final do local de instalação pode fazer com que a instalação falhe. Por exemplo, o caminho a seguir pode fazer com que a instalação falhe:

```
-OPT installLocation="/usr/IBM/WebSphere/eXtremeScale/"
```

### O caminho deve ser especificado como:

```
-OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
```

### Arquivo de resposta para instalação completa do WebSphere eXtreme Scale

```
#####  
#  
# Arquivo de Opções IBM WebSphere eXtreme Scale V7.1.1 InstallShield  
#  
# Nome do assistente: Install  
# Origem do assistente: setup.jar  
#  
# Esse arquivo pode ser utilizado para configurar a Instalação com as opções  
# especificadas abaixo quando o assistente for executado com a opção da  
# linha de comandos "-options". Leia a  
# documentação de cada configuração para obter informações sobre como alterar  
# esse valor.  
# Delimite todos os valores com um par simples de aspas duplas.  
#  
# Um uso comum de um arquivo de opções é executar o assistente no modo silencioso.  
# Isso permite que o autor do arquivo de opções especifique as configurações do  
# assistente sem ter que executar o assistente no modo gráfico ou de console.  
# Para utilizar esse arquivo de opções na execução de modo silencioso, utilize os  
# seguintes argumentos de linha de comandos ao executar o assistente:  
#  
#   -options "D:\installImage\WXS\wxssetup.response" -silent  
#  
# Observe que o nome do arquivo de resposta qualificado deve ser  
# utilizado.  
#  
#####  
  
#####  
#  
# Aceitação de Licença  
#  
# Valores Válidos:  
# true - Aceita a licença. Instalará o produto.  
# false - Recusa a licença. Não ocorrerá a instalação.  
#  
# Se não ocorrer nenhuma instalação, ela será registrada em um arquivo de log  
# temporário no diretório temporário do usuário.  
#  
# Ao alterar a propriedade silentInstallLicenseAcceptance neste arquivo de  
# resposta para "true", você concorda que revisou e concordou com os termos  
# do Contrato de Licença do Programa Internacional IBM que acompanha este  
# programa, que está localizado em  
# CD_ROOT\XD\wxs.primary.pak\repository\legal\license.xs. Se  
# você não concordar com esses termos, não altere o valor ou, de  
# alguma outra forma,  
# faça o download, instale, copie, acesse ou utilize o programa e  
# devolva imediatamente  
# o programa e a prova de titularidade à parte de quem você o  
# adquiriu para  
# obter um reembolso do valor pago.  
#  
-OPT silentInstallLicenseAcceptance="false"
```

```

#####
# Verificação de Pré-requisito Não de Bloqueio
#
# Se desejar desativar a verificação de pré-requisito não de bloqueio,
# remova o comentário da seguinte linha. Isso notificará o instalador
# para continuar a instalação e registrará os avisos mesmo que a
# verificação de pré-requisito tenha falhado.
#
#-OPT disableNonBlockingPrereqChecking="true"

#####
#
# Local da Instalação
#
# O local de instalação do produto. Especifique um diretório válido no qual o
# produto deve ser instalado. Se o diretório contiver espaços, coloque-o entre
# aspas duplas conforme mostrado no exemplo do Windows abaixo. Observe que
# espaços no local de instalação são suportados apenas nos sistemas operacionais
# Windows.O comprimento máximo do caminho é 60 caracteres para Windows.
#
# Abaixo está uma lista dos locais de instalação padrão para cada sistema
# operacional suportado quando você estiver instalando como um usuário root.
# Por padrão, neste arquivo de resposta, é utilizado o local de instalação do
# Windows. Se você desejar utilizar o local da instalação padrão para outro
# sistema operacional, remova o comentário da entrada do local da instalação
# padrão apropriada (removendo '#') e comente a linha(incluindo '#') na entrada
# do sistema operacional Windows abaixo.
#
# O local da instalação é utilizado para determinar se o WebSphere eXtreme Scale
# deve ser instalado como uma implementação independente ou se deve ser integrado
# a uma instalação existente do WebSphere Application Server.
#
# Se o local especificado for uma instalação existente do WebSphere Application
# Server ou WebSphere Network Deployment, o eXtreme Scale será integrado ao
# WebSphere Application Server existente. Se o local especificado for
# um diretório novo ou vazio, o WebSphere eXtreme Scale será instalado como
# uma implementação independente.
#
# Nota: Se o local da instalação especificado contiver uma instalação anterior
# do WebSphere eXtreme Scale, WebSphere eXtended Deployment DataGrid ou
# ObjectGrid, a instalação falhará.
#
# Local de Instalação Padrão do AIX:
#
# -OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
#
# Local de Instalação Padrão do HP-UX, Solaris ou Linux:
#
# -OPT installLocation="/opt/IBM/WebSphere/eXtremeScale"
#
#
# Local de Instalação Padrão do Windows:
#
#-OPT installLocation="C:\Program Files\IBM\WebSphere\eXtremeScale"

#
# Se você estiver instalando como um usuário não raiz no Unix ou um não
# administrador no Windows, os seguintes locais de instalação padrão serão
# sugeridos. Certifique-se de que você tenha permissões de gravação para o
# local da instalação escolhido.
#
# Local de Instalação Padrão do AIX:
#
# -OPT installLocation="/IBM/WebSphere/eXtremeScale"

```

```

#
# Local de Instalação Padrão do HP-UX, Solaris ou Linux:
#
# -OPT installLocation="<user's home>/IBM/WebSphere/eXtremeScale"
#
# Local de Instalação Padrão do Windows:
#
# -OPT installLocation="C:\IBM\WebSphere\eXtremeScale"

#####
# Instalação de Recursos Opcionais
#
# Especifique quais dos recursos opcionais deseja instalar configurando cada
# recurso desejado como "true". Configure quaisquer recursos opcionais que
# não queira instalar como "false".
#
# As opções selectServer, selectClient, selectPF e selectXSStreamQuery são
# válidas apenas quando a opção installLocation acima contém uma instalação
# do WebSphere Application Server. As opções são ignoradas em uma instalação
# independente do WebSphere eXtreme Scale.
#
# Na instalação independente do WebSphere eXtreme Scale, o cliente e o servidor
# do eXtreme Scale são instalados automaticamente.
As opções de recurso para a
# instalação independente do eXtreme Scale são selectXSConsoleOther e
# selectXSStreamQueryOther.

#
# Esta opção, quando selecionada, instala os componentes que são necessários
# para a execução de servidores WebSphere eXtreme Scale e do provedor de
# serviço de cache dinâmico eXtreme Scale.
# Se essa opção estiver selecionada, o WebSphere eXtreme Scale Client
# deve também ser selecionado ao não ser comentado e configurado para um
# valor de "true". Caso contrário, a instalação silenciosa FALHARÁ.
#
-OPT selectServer="true"

#
# Esta opção, quando selecionada, instala os componentes que são necessários
# para a execução de aplicativos clientes WebSphere eXtreme Scale. Se a opção
# do servidor for selecionada acima, essa opção também deverá ser selecionada
# por meio da remoção do comentário e configurada para um valor "true" ou a
# instalação silenciosa FALHARÁ.
#
-OPT selectClient="true"

#
# Esta opção, quando selecionada, instala os componentes que são necessários
# para a execução do WebSphere eXtreme Scale Console. Se esta opção for
# selecionada, o local de instalação especificado acima deve ser um diretório
# novo ou vazio porque a opção do console é válida apenas para a implementação
# independente do WebSphere eXtreme Scale.
# Para instalar esta opção, a linha de opção a seguir não deve ser comentada
# e deve ser configurada para um valor de "true".
-OPT selectXSConsoleOther="false"

#
# As opções a seguir, se selecionadas, instalarão uma funcionalidade REPROVADA.
#
# Esta opção seleciona o WebSphere Partition Facility para instalação.
# Esta funcionalidade está REPROVADA. Para a instalação dessa opção, a seguinte
# linha de opção deverá ter seu comentário removido e ser configurada para um
# valor "true".
#
-OPT selectPF="false"

```

```

#
# Esta opção seleciona o WebSphere eXtreme Scale StreamQuery for WAS para
# instalação. Esta funcionalidade está REPROVADA. Para a instalação dessa
# opção,a linha de opção a seguir deverá ter seu comentário removido e ser
# configurada para um valor "true".Se essa opção estiver selecionada,
# o WebSphere eXtreme Scale Client deve também ser selecionado ao não ser
# comentado e configurado para um valor de "true".Caso contrário, a instalação
# silenciosa FALHARÁ.
#
#-OPT selectXSStreamQuery="false"

#
# Esta opção seleciona o WebSphere eXtreme Scale StreamQuery for J2SE para
# instalação. Esta funcionalidade está REPROVADA. Para a instalação dessa
# opção,a linha de opção a seguir deverá ter seu comentário removido e ser
# configurada para um valor "true".
# Se essa opção estiver selecionada, o WebSphere eXtreme Scale Client
# deve também ser selecionado ao não ser comentado e configurado para um valor
# de "true".Caso contrário, a instalação silenciosa FALHARÁ.
#
#-OPT selectXSStreamQueryOther="false"

#####
# Lista de Perfis para Aumento
#
# Especifique qual dos perfis existentes deseja aumentar ou remover o comentário
# da linha para aumentar cada perfil existente detectado pela instalação.
#
# Para especificar vários perfis, utilize uma vírgula para separar diferentes
# nomes de perfis.
# Por exemplo, "AppSrv01,Dmgr01,Custom01". A lista não deve conter espaços.
#
-OPT profileAugmentList=""

#####
# Controle de Rastreamento
#
# O formato de saída de rastreamento pode ser controlado por meio da opção
# -OPT traceFormat=ALL
#
# As opções para o formato são 'text' e 'XML'. Por padrão, ambos os formatos
# serão produzidos em dois arquivos de rastreamento diferentes.
#
# Se apenas um formato for necessário, utilize a opção traceFormat para
# especificar um, como a seguir:
#
# Valores Válidos:
#
# text - As linhas no arquivo de rastreamento estarão em um formato de
# texto simples para facilitar a
# capacidade de leitura.
# XML - As linhas no arquivo de rastreamento estarão no formato XML de
# criação de log Java padrão,
# que pode ser visualizado com o uso de qualquer editor de
# texto ou XML ou com o uso da ferramenta Chainsaw da Apache na
# seguinte URL:
# (http://logging.apache.org/log4j/docs/chainsaw.html).
#
# A quantidade de informações de rastreamento capturadas pode ser controlada
# utilizando a opção:
# -OPT traceLevel=INFO
#
# Valores Válidos:
#
# Rastreamento Numérico

```

```

# Nível      Nível  Descrição
# -----
# OFF        0  Nenhum arquivo de rastreo é produzido
# SEVERE     1  Apenas erros graves são gerados para o arquivo de rastreo
# WARNING    2  Mensagens referentes a exceções não fatais e avisos são
#             incluídas no arquivo de rastreo
# INFO       3  As mensagens informativas são incluídas no arquivo de
# rastreo
#             (esse é o nível de rastreo padrão)
# CONFIG     4  As mensagens relacionadas à configuração são incluídas no
# arquivo
# de rastreo
# FINE       5  Rastrear chamadas de método para métodos públicos
# FINER      6  Rastrear chamadas de método para métodos não-públicos,
# exceto
#             getters e setters
# FINEST     7  Rastrear todas as chamadas de método,
# a entrada/saída de rastreo
# incluirão
#             parâmetros e valor de retorno

```

### Arquivo de resposta para instalação do WebSphere eXtreme Scale Client

```

#####
#
# Arquivo de Opções IBM WebSphere eXtreme Scale Client V7.1.1 InstallShield
#
# Nome do assistente: Install
# Origem do assistente: setup.jar
#
# Esse arquivo pode ser utilizado para configurar a Instalação
# com as opções especificadas abaixo
# quando o assistente for executado com a opção da linha de comandos
# "-options". Leia a
# documentação de cada configuração para obter informações sobre como
# alterar esse valor.
# Delimite todos os valores com um par simples de aspas duplas.
#
# Um uso comum de um arquivo de opções é executar o assistente no
# modo silencioso. Isso permite que o autor do arquivo de opções
# especifique as configurações do assistente sem ter que executar o
# assistente no modo gráfico ou de console. Para utilizar esse
# arquivo de opções na execução de modo silencioso, utilize os
# seguintes argumentos de linha de comandos ao executar o assistente:
#
#   -options "D:\installImage\WXS_Client\wxsetup.response" -silent
#
# Observe que o nome do arquivo de resposta qualificado deve ser
# utilizado.
#
#####

#####
#
# Aceitação de Licença
#
# Valores Válidos:
# true - Aceita a licença. Instalará o produto.
# false - Recusa a licença. Não ocorrerá a instalação.
#
# Se não ocorrer nenhuma instalação, ela será registrada em um arquivo de log
# temporário no
# diretório temporário do usuário.
#
# Ao alterar a propriedade silentInstallLicenseAcceptance neste arquivo de
# resposta para "true", você concorda que revisou e concordou com os termos
# do Contrato de Licença do Programa Internacional IBM que acompanha este

```

```

# programa, que está localizado em
# CD_ROOT\WXS_Cleint\wxs.client.primary.pak\repository\legal.xs.client\license.xs.
# Se você não concordar com esses termos, não altere o valor ou, de outra forma,
# faça o download, instale, copie, acesse ou utilize o programa e
# devolva imediatamente
# o programa e a prova de titularidade à parte de quem você o
# adquiriu para
# obter um reembolso do valor pago.
#
-OPT silentInstallLicenseAcceptance="false"

#####
# Verificação de Pré-requisito Não de Bloqueio
#
# Se desejar desativar a verificação de pré-requisito não de bloqueio, remova o
# comentário
# da seguinte linha. Isso notificará o instalador para continuar
# a instalação e registrará os avisos mesmo que a verificação de pré-requisito
# tenha falhado.
#
-OPT disableNonBlockingPrereqChecking="true"

#####
#
# Local da Instalação
#
# O local de instalação do produto. Especifique um diretório válido no qual o
# produto deve ser instalado. Se o diretório contiver espaços, coloque-o entre
# aspas duplas conforme mostrado no exemplo do Windows abaixo. Observe que
# espaços no local de instalação são suportados apenas nos sistemas operacionais
# Windows.
# O comprimento máximo do caminho é 60 caracteres para Windows.
#
# Abaixo está uma lista dos locais de instalação padrão para cada sistema
# operacional suportado quando você estiver instalando como um usuário root.
# Por padrão, neste arquivo de resposta, é utilizado o local de instalação do
# Windows. Se você desejar utilizar o local da instalação padrão para outro
# sistema operacional, remova o comentário da entrada do local da instalação
# padrão apropriada (removendo '#') e comente a linha(incluindo '#') na entrada
# do sistema operacional Windows abaixo.
#
# O local da instalação é utilizado para determinar se o WebSphere eXtreme Scale
# deve ser instalado como uma implementação independente ou se deve ser integrado
# a uma instalação existente do WebSphere Application Server.
#
# Se o local especificado for uma instalação existente do WebSphere Application
# Server ou WebSphere Network Deployment, o eXtreme Scale será integrado ao
# WebSphere Application Server existente. Se o local especificado for
# um diretório novo ou vazio, o WebSphere eXtreme Scale será instalado como
# uma implementação independente.
#
# Nota: Se o local da instalação especificado contiver uma instalação anterior
# do WebSphere eXtreme Scale, WebSphere eXtended Deployment DataGrid ou
# ObjectGrid, a instalação falhará.
#
# Local de Instalação Padrão do AIX:
#
# -OPT installLocation="/usr/IBM/WebSphere/eXtremeScale"
#
# Local de Instalação Padrão do HP-UX, Solaris ou Linux:
#
# -OPT installLocation="/opt/IBM/WebSphere/eXtremeScale"
#
#
# Local de Instalação Padrão do Windows:

```



```

#
-OPT installLocation="C:\Program Files\IBM\WebSphere\eXtremeScale"

#
# Se você estiver instalando como um usuário não raiz no Unix ou um não
# administrador no Windows, os seguintes locais de instalação padrão serão
# sugeridos. Certifique-se de que você tenha permissões de gravação para o
# local da instalação escolhido.
#
# Local de Instalação Padrão do AIX:
#
# -OPT installLocation="<user's home>/IBM/WebSphere/eXtremeScale"
#
# Local de Instalação Padrão do HP-UX, Solaris ou Linux:
#
# -OPT installLocation="<user's home>/IBM/WebSphere/eXtremeScale"
#
# Local de Instalação Padrão do Windows:
#
# -OPT installLocation="C:\IBM\WebSphere\eXtremeScale"

#####
# Lista de Perfis para Aumento
#
# Especifique qual dos perfis existentes deseja aumentar ou remover o comentário
# da linha para aumentar cada perfil existente detectado pela instalação.
#
# Para especificar vários perfis, utilize uma vírgula para separar diferentes
# nomes de perfis.
# Por exemplo, "AppSrv01,Dmgr01,Custom01". A lista não deve conter espaços.
#
-OPT profileAugmentList=""

#####
# Controle de Rastreamento
#
# O formato de saída de rastreamento pode ser controlado por meio da opção
# -OPT traceFormat=ALL
#
# As opções para o formato são 'text' e 'XML'. Por padrão, ambos os formatos
# serão produzidos em dois arquivos de rastreamento diferentes.
#
# Se apenas um formato for necessário, utilize a opção traceFormat para
# especificar um, como a seguir:
#
# Valores Válidos:
#
# text - As linhas no arquivo de rastreamento estarão em um formato de texto
# simples para facilitar a capacidade de leitura.
# XML - As linhas no arquivo de rastreamento estarão no formato XML de criação
# de log Java padrão,
# que pode ser visualizado com o uso de qualquer editor de texto ou
# XML ou com o uso da ferramenta Chainsaw da Apache na seguinte URL:
# (http://logging.apache.org/log4j/docs/chainsaw.html).
#
# A quantidade de informações de rastreamento capturadas pode ser controlada
# utilizando a opção:
# -OPT traceLevel=INFO
#
# Valores Válidos:
#
# Rastreamento Numérico
# Nível Nível Descrição
# -----
# OFF 0 Nenhum arquivo de rastreamento é produzido

```

```

# SEVERE      1 Apenas erros graves são gerados para o arquivo de rastreo
# WARNING    2 Mensagens referentes a exceções não fatais e avisos são
#             incluídas no arquivo de rastreo
# INFO       3 As mensagens informativas são incluídas no arquivo de
# rastreo
#             (esse é o nível de rastreo padrão)
# CONFIG     4 As mensagens relacionadas à configuração são incluídas no
# arquivo de rastreo
# FINE       5 Rastrear chamadas de método para métodos públicos
# FINER      6 Rastrear chamadas de método para métodos não-públicos,
# exceto
#             getters e setters
# FINEST     7 Rastrear todas as chamadas de método, a entrada/saída de
# rastreo incluirão
#             parâmetros e valor de retorno

```

## Criando e Alterando Perfis para o WebSphere eXtreme Scale

Depois de instalar o produto, crie tipos exclusivos de perfis e aumente os perfis existentes para o WebSphere eXtreme Scale.

### Antes de Iniciar

Instale o WebSphere eXtreme Scale. Consulte “Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client com WebSphere Application Server” na página 161 para obter mais informações.

O aumento de perfis para uso com WebSphere eXtreme Scale é opcional, mas é necessário nos seguintes cenários de uso:

- Para iniciar automaticamente um serviço de catálogo ou contêiner em um processo do WebSphere Application Server. Sem o aumento dos perfis do servidor, os servidores podem ser iniciados programaticamente apenas usando a API do ServerFactory ou como processos separados usando o script **startOgServer**.
- Para utilizar Performance Monitoring Infrastructure (PMI) para monitorar métricas de WebSphere eXtreme Scale.
- Para exibir a versão do WebSphere eXtreme Scale no console administrativo do WebSphere Application Server.

### Sobre Esta Tarefa

#### Executando com o WebSphere Application Server Versão 6.1 ou 7.0

Se o seu ambiente contiver o WebSphere Application Server Versão 6.1 ou Versão 7.0, poderá usar o plug-in da Profile Management Tool ou o comando **manageprofiles** para criar e alterar perfis.

### O que Fazer Depois

Dependendo de qual tarefa você escolher concluir, ative o Console de Primeiras Etapas para obter assistência sobre a configuração e o teste do ambiente do produto. O console do First Steps está no diretório *wxs\_install\_root\firststeps\wxs\firststeps.bat*. Você também pode criar ou aumentar perfis adicionais repetindo qualquer uma das tarefas anteriores.

#### Usando a Interface Gráfica com o Usuário para Criar Perfis:

Use a interface gráfica com o usuário (GUI) fornecida pelo plug-in Profile Management Tool para criar perfis para o WebSphere eXtreme Scale. Um perfil é um conjunto de arquivos que definem o ambiente de tempo de execução.

### Antes de Iniciar

Não é possível usar a GUI para aumentar perfis no cenário a seguir:

- **Instalações de 64 bits do WebSphere Application Server:**

O Profile Management Tool não existe para instalações de 64 bits do WebSphere Application Server. Use o script **manageprofiles** a partir da linha de comandos para essas instalações.

### Sobre Esta Tarefa

Para utilizar os recursos do produto, o plug-in Profile Management Tool ativa a GUI para ajudar você a configurar perfis, como perfil do WebSphere Application Server, um perfil de gerenciador de implementação, um perfil de célula e um perfil customizado. Os perfis podem ser aumentados durante ou após a instalação do WebSphere eXtreme Scale.

### Procedimento

Use a GUI do Profile Management Tool para criar perfis. Escolha uma das seguintes opções para iniciar o assistente:

- Selecione **Profile Management Tool** no console do First Steps.
- Acesse a Ferramenta Gerenciamento de Perfil a partir do menu **Iniciar**.
- Execute o script `./pmt.sh|bat` a partir do diretório `install_root/bin/ProfileManagement`.

### O que Fazer Depois

É possível criar perfis adicionais ou alterar perfis existentes. Para reiniciar o Profile Management Tool, execute o comando `./pmt.sh|bat` no diretório `was_root/bin/ProfileManagement` ou selecione **Profile Management Tool** no console do First Steps.

Inicie um serviço de catálogo, inicie os contêineres e configure as portas TCP no seu ambiente do WebSphere Application Server. Consulte “Configurando o WebSphere eXtreme Scale com o WebSphere Application Server” na página 252 para obter mais informações.

### Usando a Interface Gráfica com o Usuário para Alterar Perfis:

Depois de instalar o produto, poderá alterar um perfil existente para torná-lo compatível com o WebSphere eXtreme Scale.

### Sobre Esta Tarefa

Ao alterar um perfil existente, poderá mudar o perfil ao aplicar um modelo de mudança específico do produto. Por exemplo, os servidores do WebSphere eXtreme Scale não são iniciados automaticamente a menos que o perfil do servidor seja alterado com o modelo `xs_augment`.

- Altere o perfil com o modelo `xs_augment` se você instalou o cliente ou o cliente e o servidor do eXtreme Scale.

- Aumente o perfil com o modelo `pf_augment` apenas se você tiver instalado o recurso de particionamento.
- Aplique ambos os modelos se o seu ambiente contiver o cliente e o recurso de particionamento do eXtreme Scale.

### Procedimento

Use a GUI do Profile Management Tool para alterar os perfis para o eXtreme Scale. Escolha uma das seguintes opções para iniciar o assistente:

- Selecione **Profile Management Tool** no console do First Steps.
- Acesse a Ferramenta Gerenciamento de Perfil a partir do menu **Iniciar**.
- Execute o script `./pmt.sh|bat` a partir do diretório `was_root/bin/ProfileManagement`.

### O que Fazer Depois

Você pode alterar perfis adicionais. Para reiniciar o Profile Management Tool, execute o comando `./pmt.sh|bat` no diretório `was_root/bin/ProfileManagement` ou selecione **Profile Management Tool** no console do First Steps.

Inicie um serviço de catálogo, inicie os contêineres e configure as portas TCP no seu ambiente do WebSphere Application Server. Consulte “Configurando o WebSphere eXtreme Scale com o WebSphere Application Server” na página 252 para obter mais informações.

### Comando `manageprofiles`:

É possível usar o utilitário `manageprofiles` para criar perfis com o modelo WebSphere eXtreme Scale e alterar ou retornar o estado inalterado dos perfis do servidor de aplicativos existentes com os modelos de alteração do eXtreme Scale. Para usar os recursos do produto, seu ambiente deve conter pelo menos um perfil alterado para o produto.

- Antes de poder criar e alterar os perfis, é necessário instalar o eXtreme Scale . Consulte “Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client com WebSphere Application Server” na página 161 para obter mais informações.

### Propósito

O comando `manageprofiles` cria o ambiente de execução para um processo do produto em um conjunto de arquivos chamado perfil. O perfil define o ambiente de tempo de execução. É possível executar as seguintes ações com o comando `manageprofiles`:

- Criar e alterar um perfil do gerenciador de implementação
- Criar e alterar um perfil customizado
- Criar e alterar um perfil do servidor de aplicativos independente
- Criar e alterar um perfil de célula
- Retornar o estado inalterado de qualquer tipo de perfil

Ao alterar um perfil existente, poderá mudar o perfil ao aplicar um modelo de mudança específico do produto.

- Altere o perfil com o modelo `xs_augment` se você instalou o cliente ou o cliente e o servidor do eXtreme Scale.

- Altere o perfil com o modelo `pf_augment` se você instalou apenas o recurso de particionamento.
- Aplique os dois modelos se o seu ambiente contiver o cliente e o recurso de particionamento do eXtreme Scale.

## Local

O arquivo de comando está no diretório `install_root/bin`.

## Uso

Para obter ajuda detalhada, use o parâmetro **-help**:

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/dmgr -help
```

Nas seguintes seções, cada tarefa que pode ser executada usando o comando **manageprofiles** junto com uma lista de parâmetros necessários é descrita. Para obter detalhes sobre os parâmetros opcionais a serem especificados para cada tarefa, consulte o comando **manageprofiles** no Centro de Informações do WebSphere Application Server.

## Criar um Perfil do Gerenciador de Implementação

É possível usar o comando **manageprofiles** para criar um perfil do gerenciador de implementação. O gerenciador de implementação administra os servidores de aplicativos que são federados na célula.

### Parâmetros

#### **-create**

Cria um perfil. (Obrigatório)

#### **-templatePath** *template\_path*

Especifica o caminho de arquivo para o modelo. (Obrigatório)

Utilize o seguinte formato:

```
-templatePath install_root/profileTemplates/template_type/dmgr
```

em que *template\_type* é `xs_augment` ou `pf_augment`.

### Exemplo

- Usando o modelo `xs_augment`:

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/dmgr
```

- Usando o modelo `pf_augment`:

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/dmgr
```

## Criar um perfil customizado

É possível usar o comando **manageprofiles** para Criar um perfil customizado. Um perfil customizado é um nó vazio que você customiza por meio do gerenciador de implementação para incluir servidores de aplicativos, clusters ou outros processos Java.

### Parâmetros

#### **-create**

Cria um perfil. (Obrigatório)

**-templatePath** *template\_path*  
Especifica o caminho de arquivo para o modelo. (Obrigatório)  
Utilize o seguinte formato:  
`-templatePath install_root/profileTemplates/template_type/managed`  
em que *template\_type* é `xs_augment` ou `pf_augment`.

### Exemplo

- Usando o modelo `xs_augment`:  
`./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/managed`
- Usando o modelo `pf_augment`:  
`./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/managed`

### Criar um Perfil de Servidor de Aplicativos Independente

É possível usar o comando **manageprofiles** para Criar um perfil do servidor de aplicativos independente.

#### Parâmetros

**-create**  
Cria um perfil. (Obrigatório)

**-templatePath** *template\_path*  
Especifica o caminho de arquivo para o modelo. (Obrigatório)  
Utilize o seguinte formato:  
`-templatePath install_root/profileTemplates/template_type/default`  
em que *template\_type* é `xs_augment` ou `pf_augment`.

### Exemplo

- Usando o modelo `xs_augment`:  
`./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/default`
- Usando o modelo `pf_augment`:  
`./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/default`

### Criar o Perfil de uma Célula

É possível usar o comando **manageprofiles** para Criar um perfil de célula que consiste de um gerenciador de implementação e de um servidor de aplicativos.

#### Parâmetros

Especifique os seguintes parâmetros no modelo do gerenciador de implementação:

**-create**  
Cria um perfil. (Obrigatório)

**-templatePath** *template\_path*  
Especifica o caminho de arquivo para o modelo. (Obrigatório)  
Utilize o seguinte formato:  
`-templatePath install_root/profileTemplates/template_type/cell/dmgr`  
em que *template\_type* é `xs_augment` ou `pf_augment`.

Especifique os seguintes parâmetros com o modelo do servidor de aplicativos:

**-create**

Cria um perfil. (Obrigatório)

**-templatePath** *template\_path*

Especifica o caminho de arquivo para o modelo. (Obrigatório)

Utilize o seguinte formato:

```
-templatePath install_root/profileTemplates/template_type/cell/default
```

em que *template\_type* é *xs\_augment* ou *pf\_augment*.

### Exemplo

- Usando o modelo *xs\_augment*:

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/cell/dmgr  
-nodeProfilePath install_root/profiles/AppSrv01 -cellName cell01dmgr -nodeName node01dmgr  
-appServerNodeName node01
```

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/xs_augment/cell/default  
-dmgrProfilePath install_root/profiles/Dmgr01 -portsFile  
install_root/profiles/Dmgr01/properties/portdef.props -nodePortsFile  
install_root/profiles/Dmgr01/properties/nodeportdef.props -cellName cell01dmgr  
-nodeName node01dmgr -appServerNodeName node01
```

- Usando o modelo *pf\_augment*:

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/cell/dmgr  
-nodeProfilePath install_root/profiles/AppSrv01 -cellName cell01dmgr -nodeName node01dmgr  
-appServerNodeName node01
```

```
./manageprofiles.sh|bat -create -templatePath install_root/profileTemplates/pf_augment/cell/default  
-dmgrProfilePath install_root/profiles/Dmgr01 -portsFile  
install_root/profiles/Dmgr01/properties/portdef.props -nodePortsFile  
install_root/profiles/Dmgr01/properties/nodeportdef.props -cellName cell01dmgr  
-nodeName node01dmgr -appServerNodeName node01
```

### Alterar um Perfil do Gerenciador de Implementação

É possível usar o comando **manageprofiles** para Alterar um perfil do gerenciador de implementação.

#### Parâmetros

**-augment**

Altera o perfil existente. (Obrigatório)

**-profileName**

Especifica o nome do perfil. (Obrigatório)

**-templatePath** *template\_path*

Especifica o caminho para os arquivos de modelo que estão localizados no diretório raiz da instalação. (Obrigatório)

Utilize o seguinte formato:

```
-templatePath install_root/profileTemplates/template_type/dmgr
```

em que *template\_type* é *xs\_augment* ou *pf\_augment*.

### Exemplo

- Usando o modelo *xs\_augment*:

```
./manageprofiles.sh|bat -augment -profileName profile01  
-templatePath install_root/profileTemplates/xs_augment/dmgr
```

- Usando o modelo *pf\_augment*:

```
./manageprofiles.sh|bat -augment -profileName profile01  
-templatePath install_root/profileTemplates/pf_augment/dmgr
```



## Alterar um Perfil Customizado

É possível usar o comando **manageprofiles** para Alterar um perfil customizado.

### Parâmetros

#### **-augment**

Altera o perfil existente. (Obrigatório)

#### **-profileName**

Especifica o nome do perfil. (Obrigatório)

#### **-templatePath** *template\_path*

Especifica o caminho para os arquivos de modelo que estão localizados no diretório raiz da instalação. (Obrigatório)

Utilize o seguinte formato:

```
-templatePath install_root/profileTemplates/template_type/managed
```

em que *template\_type* é *xs\_augment* ou *pf\_augment*.

### Exemplo

- Usando o modelo *xs\_augment*:

```
./manageprofiles.sh|bat -augment -profileName profile01  
-templatePath install_root/profileTemplates/xs_augment/managed
```

- Usando o modelo *pf\_augment*:

```
./manageprofiles.sh|bat -augment -profileName profile01  
-templatePath install_root/profileTemplates/pf_augment/managed
```

## Alterar um Perfil de Servidor de Aplicativos Independente

É possível usar o comando **manageprofiles** para Alterar um perfil do servidor de aplicativos independente.

### Parâmetros

#### **-augment**

Altera o perfil existente. (Obrigatório)

#### **-profileName**

Especifica o nome do perfil. (Obrigatório)

#### **-templatePath** *template\_path*

Especifica o caminho para os arquivos de modelo que estão localizados no diretório raiz da instalação. (Obrigatório)

Utilize o seguinte formato:

```
-templatePath install_root/profileTemplates/template_type/default
```

em que *template\_type* é *xs\_augment* ou *pf\_augment*.

### Exemplo

- Usando o modelo *xs\_augment*:

```
./manageprofiles.sh|bat -augment -profileName profile01  
-templatePath install_root/profileTemplates/xs_augment/default
```

- Usando o modelo *pf\_augment*:

```
./manageprofiles.sh|bat -augment -profileName profile01  
-templatePath install_root/profileTemplates/pf_augment/default
```

## Alterar o Perfil de uma Célula

É possível usar o comando `manageprofiles` para Alterar um perfil de célula.

### Parâmetros

Especifique os seguintes parâmetros para o perfil do gerenciador de implementação:

#### **-augment**

Altera o perfil existente. (Obrigatório)

#### **-profileName**

Especifica o nome do perfil. (Obrigatório)

#### **-templatePath** *template\_path*

Especifica o caminho para os arquivos de modelo que estão localizados no diretório raiz da instalação. (Obrigatório)

Utilize o seguinte formato:

```
-templatePath install_root/profileTemplates/template_type/cell/dmgr
```

em que *template\_type* é `xs_augment` ou `pf_augment`.

Especifique os seguintes parâmetros com o perfil do servidor de aplicativos:

#### **-augment**

Altera o perfil existente. (Obrigatório)

#### **-profileName**

Especifica o nome do perfil. (Obrigatório)

#### **-templatePath** *template\_path*

Especifica o caminho para os arquivos de modelo que estão localizados no diretório raiz da instalação. (Obrigatório)

Utilize o seguinte formato:

```
-templatePath install_root/profileTemplates/template_type/cell/default
```

em que *template\_type* é `xs_augment` ou `pf_augment`.

## Exemplo

- Usando o modelo `xs_augment`:

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath install_root  
/profileTemplates/xs_augment/cell/dmgr
```

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath install_root  
/profileTemplates/xs_augment/cell/default
```

- Usando o modelo `pf_augment`:

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath install_root  
/profileTemplates/pf_augment/cell/dmgr
```

```
./manageprofiles.sh|bat -augment -profileName profile01 -templatePath install_root  
/profileTemplates/pf_augment/cell/default
```

## Retornar o Estado Inalterado de um Perfil

Para reverter o aumento de um perfil, especifique o parâmetro `-ignoreStack` com o parâmetro `-templatePath` além de especificar os parâmetros requeridos `-unaugment` e `-profileName`.

## Parâmetros

### -unaugment

Retorna o estado inalterado de um perfil alterado anteriormente. (Obrigatório)

### -profileName

Especifica o nome do perfil. O parâmetro é emitido por padrão se nenhum valor for especificado. (Obrigatório)

### -templatePath *template\_path*

Especifica o caminho para os arquivos de modelo que estão localizados no diretório raiz da instalação. (Opcional)

Utilize o seguinte formato:

```
-templatePath install_root/profileTemplates/template_type/profile_type
```

em que *template\_type* é *xs\_augment* ou *pf\_augment* e *profile\_type* é um dos quatro tipos de perfil:

- *dmgr*: perfil do gerenciador de implementação
- *managed*: perfil customizado
- *default*: perfil do servidor de aplicativos independente
- *cell*: perfil de célula

### -ignoreStack

Usado com o parâmetro **-templatePath** para alterar um perfil específico que foi alterado. (Opcional)

## Exemplo

- Usando o modelo *xs\_augment*:

```
./manageprofiles.sh|bat -unaugment -profileName profile01 -ignoreStack  
-templatePath install_root/profileTemplates/xs_augment/profile_type
```

- Usando o modelo *pf\_augment*:

```
./manageprofiles.sh|bat -unaugment -profileName profile01 -ignoreStack  
-templatePath install_root/profileTemplates/pf_augment/profile_type
```

## Perfis Não-root:

Dá permissões a um usuário não-root para arquivos e diretórios para que o usuário não-root possa criar um perfil para o produto. O usuário não-root também pode alterar um perfil que foi criado por um usuário root, um usuário não-root diferente ou pelo mesmo usuário não-root.

Em um ambiente do WebSphere Application Server, usuários não raiz (não administrador) são limitados à capacidade de criar e usar perfis em seu ambiente. Dentro do plug-in do Profile Management tool, nomes exclusivos e valores de porta são desativados para usuários não raiz. O usuário não-root deve alterar os valores de campo padrão na ferramenta Gerenciamento de Perfil para o nome do perfil, o nome do nó, o nome da célula e as designações de porta. Considere designar aos usuários não-root um intervalo de valores para cada um dos campos. É possível designar responsabilidade para os usuários não-root para aderirem a seus intervalos de valor apropriados e para manterem a integridade de suas próprias definições.

O termo *instalador* refere-se a um usuário root ou não-root. Como um instalador, você pode conceder permissões de usuários não-root para criar e estabelecer seus próprios ambientes de produto. Por exemplo, um usuário não raiz pode criar um ambiente do produto para testar a implementação do aplicativo com um perfil que

o usuário possui. Tarefas específicas que você pode concluir para permitir a criação de perfil não-raiz incluem os seguintes itens:

- Criar um perfil e designar a propriedade do diretório de perfil para um usuário não-raiz, para que o usuário não-raiz possa iniciar o WebSphere Application Server para um perfil específico.
- Conceder permissão de gravação dos arquivos e diretórios apropriados para um usuário não-root, permitindo que o usuário não-root crie o perfil. Com essa tarefa, você pode criar um grupo para usuários autorizados a criar perfis ou pode fornecer aos usuários individuais a capacidade de criar perfis.
- Instalar os pacotes de manutenção para o produto que inclui o serviço necessário para os perfis existentes de propriedade de um usuário não-root. Como o instalador, você é o proprietário de qualquer novo arquivo criado pelo pacote de manutenção.

Para obter mais informações sobre a criação de perfis para usuários não raiz, consulte Criando Perfis para Usuários Não Raiz.

Como um instalador, você também pode conceder permissões para um usuário não-root alterar perfis. Por exemplo, um usuário não raiz pode aumentar um perfil que é criado por um instalador ou aumentar um perfil criado por ele mesmo. Siga o processo de aumento do usuário não raiz do WebSphere Application Server Network Deployment.

Entretanto, quando um usuário não raiz aumenta um perfil que é criado pelo instalador, o usuário não raiz não precisa criar os seguintes arquivos antes do aumento. Os seguintes arquivos foram estabelecidos durante o processo de criação de perfil:

- *was\_root/logs/manageprofiles.xml*
- *was\_root/properties/fsdb.xml*
- *was\_root/properties/profileRegistry.xml*

Quando um usuário não raiz aumenta um perfil que o usuário cria, o usuário não raiz deve modificar as permissões para os documentos que estão localizados dentro dos modelos de perfil do eXtreme Scale.

**Atenção:** Também é possível utilizar um perfil não raiz (não administrador) para o WebSphere eXtreme Scale em um ambiente independente fora do WebSphere Application Server. Você deve alterar o proprietário do diretório do ObjectGrid para o perfil não raiz. Em seguida, é possível efetuar login com esse perfil não raiz e operar o eXtreme Scale como faria normalmente para um perfil raiz (administrador).

## Instalando WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client Independente

É possível instalar o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client independente em um ambiente que não contenha o WebSphere Application Server ou o WebSphere Application Server Network Deployment.

### Antes de Iniciar

- Verifique se o diretório de instalação de destino está vazio ou não existe.

**Importante:** Se uma versão anterior do WebSphere eXtreme Scale ou do componente ObjectGrid existir no diretório especificado para instalar a Versão

7.1.1, o produto não será instalado. Por exemplo, você pode ter uma pasta *wxs\_install\_root/ObjectGrid* anteriormente existente. É possível selecionar um diretório de instalação diferente ou cancelar a instalação. Em seguida, remova a instalação anterior e execute o assistente novamente.

- Um IBM Runtime Environment é instalado como parte da instalação independente na pasta *wxs\_install\_root/java*.
- Se estiver instalando apenas o cliente: Faça download do WebSphere eXtreme Scale Client para a plataforma apropriada a partir do site de Suporte.

## Sobre Esta Tarefa

Ao instalar o produto como independente, instale o cliente e o servidor do WebSphere eXtreme Scale juntos. Com a instalação do WebSphere eXtreme Scale Client no modo independente, você está instalando um cliente para acessar os dados nas grades de dados. Os processos do servidor e do cliente, portanto, acessam todos os recursos necessários localmente. Também é possível integrar o WebSphere eXtreme Scale em aplicativos Java Platform, Standard Edition (J2SE) existentes utilizando scripts e arquivos Java archive (JAR).

**Atenção:** Também é possível utilizar um perfil não raiz (não administrador) para o WebSphere eXtreme Scale em um ambiente independente. Para utilizar um perfil não raiz, você deverá alterar o proprietário do diretório *ObjectGrid* para o perfil não raiz. Em seguida, é possível efetuar login com esse perfil não raiz e operar o eXtreme Scale como faria normalmente para um perfil raiz (administrador).

## Procedimento

1. Use o assistente para instalar o servidor e o cliente a partir do DVD.
  - Execute o seguinte script para iniciar o assistente para a instalação completa do WebSphere eXtreme Scale:
    - `Linux` `UNIX` `dvd_root/install`
    - `Windows` `dvd_root\install.bat`
  - Execute o script a seguir para iniciar o assistente para a instalação do WebSphere eXtreme Scale Client. Os arquivos de instalação estão no arquivo zip que você transferiu por download a partir do site de Suporte:
    - `Linux` `UNIX` `root/WXS_Client/install`
    - `Windows` `root\WXS_Client\install.bat`

**Atenção:** Se usar convenções de nomenclatura uniformes (UNC) para identificar caminhos de arquivo em seu comando de instalação, nem todos os itens que você antecipa a instalação poderão ser instalados após a execução do comando. Para evitar problemas, mapeie o caminho de arquivo para uma unidade de rede. Execute o comando **install** na unidade mapeada. Usar uma unidade de rede mapeada assegura que todos os itens sejam instalados.

2. Siga os prompts no assistente e clique em **Concluir**.

**Restrição:** O painel de recursos opcionais lista os recursos dos quais você pode selecionar para instalar. Porém, os recursos não poderão ser incluídos incrementalmente no ambiente do produto depois que o produto for instalado. Se você escolher não instalar um recurso com a instalação inicial do produto, deverá desinstalar e reinstalar o produto para incluir o recurso.

## Resultados

**Windows** Se estiver instalando o WebSphere eXtreme Scale Client no Windows, será possível ver o seguinte texto nos resultados da instalação:

```
Sucesso: A instalação do seguinte produto foi bem-sucedida:
WebSphere eXtreme Scale Client. Algumas etapas de configuração têm erros.
Para obter mais informações, consulte o seguinte arquivo de log:
<WebSphere Application Server install root>\logs\wxs_client\install\log.txt"
Revise o log de instalação (log.txt) e o log de aumento do
gerenciador de implementação
```

Se vir uma falha com o arquivo `iscdeploy.sh`, é possível ignorar o erro. Este erro não causa nenhum problema.

## O que Fazer Depois

- Verifique a instalação. Para obter informações adicionais, consulte “Verificando a Instalação” na página 204.
- Inicie a configuração do seu WebSphere eXtreme Scale ou a instalação do WebSphere eXtreme Scale Client. Para obter mais informações, consulte “Executando as Primeiras Etapas Após a Instalação” na página 205.

## Arquivos de Tempo de Execução para Instalação Independente do WebSphere eXtreme Scale

Os arquivos Java archive (JAR) são incluídos na instalação. É possível ver os arquivos JAR que são incluídos e o local para o qual estão instalados.

*Tabela 7. Arquivos de Tempo de Execução para Instalação Completa do WebSphere eXtreme Scale.* O WebSphere eXtreme Scale conta com processos do ObjectGrid e APIs relacionadas. A seguinte tabela lista os arquivos JAR incluídos na instalação. O local da instalação é relativo ao diretório `wxs_home` que você escolher durante a instalação.

Nome do Arquivo	Ambiente	Local da instalação	Descrição
wxsdynacache.jar	Cliente e Provedor	dynacache/lib	O arquivo wxsdynacache.jar contém as classes necessárias a serem usadas com o provedor de cache dinâmico. O arquivo é incluído automaticamente no ambiente de tempo de execução do servidor quando você utiliza os scripts fornecidos.
wxshyperic.jar	Funções de Utilitário	hyperic/lib	O plug-in de detecção do servidor WebSphere eXtreme Scale para o agente de monitoramento SpringSource Hyperic.
objectgrid.jar	Local, cliente e servidor	lib	O arquivo objectgrid.jar é um pacote configurável OSGi usado pelo ambiente de tempo de execução do servidor do Java SE Versão 5.0 e posterior. O arquivo é incluído automaticamente no ambiente de tempo de execução do servidor quando você utiliza os scripts fornecidos.
ogagent.jar	Local, cliente e servidor	lib	O arquivo ogagent.jar contém as classes do tempo de execução que são necessárias para executar o agente de instrumentação Java que é usado com a API do EntityManager.
ogclient.jar	Local e cliente	lib	O arquivo ogclient.jar é um pacote configurável OSGi que contém apenas os ambientes de tempo de local e do cliente. É possível usar este arquivo com o Java SE 5.0 e posterior.
ogspring.jar	Local, cliente e servidor	lib	O arquivo ogspring.jar contém classes de suporte para a integração de estrutura SpringSource Spring.
wsogclient.jar	Local e cliente	lib	O arquivo wsogclient.jar é instalado quando você utiliza um ambiente contendo o WebSphere Application Server Versão 6.0.2 e posterior. Esse arquivo contém apenas os ambientes de tempo de execução local e do cliente.
wxssizeagent.jar	Local, cliente e servidor	lib	O arquivo wxssizeagent.jar é usado para fornecer informações mais precisas de dimensionamento de entrada de cache ao usar o Java Runtime Environment (JRE) Versão 1.5 ou mais recente.

**Tabela 7. Arquivos de Tempo de Execução para Instalação Completa do WebSphere eXtreme Scale (continuação).** O WebSphere eXtreme Scale conta com processos do ObjectGrid e APIs relacionadas. A seguinte tabela lista os arquivos JAR incluídos na instalação. O local da instalação é relativo ao diretório *wxs\_home* que você escolher durante a instalação.

Nome do Arquivo	Ambiente	Local da instalação	Descrição
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Cliente e Provedor	lib/endorsed	Este conjunto de arquivos inclui o tempo de execução Object Request Broker (ORB) que é usado para executar aplicativos em processos Java SE.
restservice.ear	Cliente	restservice/lib	O arquivo restservice.ear contém o archive corporativo de aplicativo de serviço de dados eXtreme Scale REST para ambientes WebSphere Application Server.
restservice.war	Cliente	restservice/lib	O arquivo restservice.war contém o arquivamento Web de serviço de dados eXtreme Scale REST para servidores de aplicativos adquiridos de outro fornecedor.
xsadmin.jar	Funções de Utilitário	samples	O arquivo xsadmin.jar contém o utilitário de amostra de administração eXtreme Scale.
sessionobjectgrid.jar	Cliente e Provedor	session/lib	O arquivo sessionobjectgrid.jar contém o tempo de execução de gerenciamento de sessões eXtreme Scale HTTP.
splicerlistener.jar	Funções de Utilitário	session/lib	O arquivo splicerlistener.jar contém o utilitário splicer para o listener de sessão HTTP do eXtreme Scale Versão 7.1 e posterior.
xsgbean.jar	Servidor	wasce/lib	O arquivo xsgbean.jar contém o GBean para integrar servidores eXtreme Scale nos servidores de aplicativos WebSphere Application Server Community Edition.
splicer.jar	Funções de Utilitário	legacy/session/lib	O utilitário splicer para o filtro do gerenciador de sessões HTTP do WebSphere eXtreme Scale Versão 7.0.

**Tabela 8. Arquivos de Tempo de Execução para o WebSphere eXtreme Scale Client.** O WebSphere eXtreme Scale Client conta com processos do ObjectGrid e APIs relacionadas. A seguinte tabela lista os arquivos JAR incluídos na instalação. O local da instalação é relativo ao diretório *wxs\_home* que você escolher durante a instalação.

Nome do Arquivo	Ambiente	Local da instalação	Descrição
wxsdynacache.jar	Cliente e Provedor	dynacache/lib	O arquivo wxsdynacache.jar contém as classes necessárias a serem usadas com o provedor de cache dinâmico. O arquivo é incluído automaticamente no ambiente de tempo de execução do servidor quando você utiliza os scripts fornecidos.
wxshyperic.jar	Funções de Utilitário	hyperic/lib	O plug-in de detecção do servidor WebSphere eXtreme Scale para o agente de monitoramento SpringSource Hyperic.
ogagent.jar	Local, cliente e servidor	lib	O arquivo ogagent.jar contém as classes do tempo de execução que são necessárias para executar o agente de instrumentação Java que é usado com a API do EntityManager.
ogclient.jar	Local e cliente	lib	O arquivo ogclient.jar é um pacote configurável OSGi que contém apenas os ambientes de tempo de local e do cliente. Esse arquivo pode ser usado com o Java SE 5 e posterior.
ogspring.jar	Local, cliente e servidor	lib	O arquivo ogspring.jar contém classes de suporte para a integração de estrutura SpringSource Spring.
wsogclient.jar	Local e cliente	lib	O arquivo wsogclient.jar é instalado quando você utiliza um ambiente contendo o WebSphere Application Server Versão 6.0.2 e posterior. Esse arquivo contém apenas os ambientes de tempo de execução local e do cliente.
wxssizeagent.jar	Local, cliente e servidor	lib	O arquivo wxssizeagent.jar é usado para fornecer informações mais precisas de dimensionamento de entrada de cache ao usar o Java Runtime Environment (JRE) Versão 1.5 ou mais recente.
ibmcfw.jar ibmorb.jar ibmorbapi.jar	Cliente e Provedor	lib/endorsed	Este conjunto de arquivos inclui o tempo de execução Object Request Broker (ORB) que é usado para executar aplicativos em processos Java SE.
restservice.ear	Cliente	restservice/lib	O arquivo restservice.ear contém o archive corporativo de aplicativo de serviço de dados eXtreme Scale REST para ambientes WebSphere Application Server.
restservice.war	Cliente	restservice/lib	O arquivo restservice.war contém o arquivamento Web de serviço de dados eXtreme Scale REST para servidores de aplicativos adquiridos de outro fornecedor.



Tabela 8. Arquivos de Tempo de Execução para o WebSphere eXtreme Scale Client (continuação). O WebSphere eXtreme Scale Client conta com processos do ObjectGrid e APIs relacionadas. A seguinte tabela lista os arquivos JAR incluídos na instalação. O local da instalação é relativo ao diretório *wxs\_home* que você escolher durante a instalação.

Nome do Arquivo	Ambiente	Local da instalação	Descrição
xsadmin.jar	Funções de Utilitário	samples	O arquivo xsadmin.jar contém o utilitário de amostra de administração eXtreme Scale.
sessionobjectgrid.jar	Cliente e Provedor	session/lib	O arquivo sessionobjectgrid.jar contém o tempo de execução de gerenciamento de sessões eXtreme Scale HTTP.
splicerlistener.jar	Funções de Utilitário	session/lib	O arquivo splicerlistener.jar contém o utilitário splicer para o listener de sessão HTTP do eXtreme Scale Versão 7.1 e posterior.
splicer.jar	Funções de Utilitário	legacy/session/lib	O utilitário splicer para o filtro do gerenciador de sessões HTTP do WebSphere eXtreme Scale Versão 7.0.

## Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client no Modo Silencioso

Use um arquivo completo de resposta, que você configura especificamente para suas necessidades ou transmita parâmetros para a linha de comandos para instalar de forma silenciosa o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client.

### Antes de Iniciar

- Pare todos os processos em execução no ambiente do WebSphere Application Server ou do WebSphere Application Server Network Deployment. Consulte Utilitários de Linha de Comandos para obter informações adicionais sobre os comandos **stopManager**, **stopNode** e **stopServer**.

#### **CUIDADO:**

**Certifique-se de que quaisquer processos em execução sejam interrompidos. Se os processos em execução não forem interrompidos, a instalação continuará, criando resultados imprevisíveis e deixando a instalação em um estado indeterminado em algumas plataformas.**

- Verifique se o diretório de instalação de destino está vazio ou não existe.

**Importante:** Se uma versão anterior do WebSphere eXtreme Scale ou do componente ObjectGrid existir no diretório especificado para instalar a Versão 7.1.1, o produto não será instalado. Por exemplo, você pode ter uma pasta *wxs\_install\_root/ObjectGrid* anteriormente existente. É possível selecionar um diretório de instalação diferente ou cancelar a instalação. Em seguida, remova a instalação anterior e execute o assistente novamente.

### Sobre Esta Tarefa

Uma instalação silenciosa utiliza o mesmo programa que a versão de interface gráfica com o usuário (GUI) utiliza. Entretanto, no lugar de exibir uma interface de assistente, a instalação silenciosa lê todas as suas respostas de um arquivo que você customiza ou dos parâmetros que você transmite para a linha de comandos. Consulte um exemplo de um “Arquivo *wxssetup.response.txt*” na página 174, que inclui uma descrição de cada opção.

### Procedimento

1. Opcional: Se você escolher instalar o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client usando um arquivo de resposta, primeiro customize o arquivo *wxssetup.response.txt*.

**Lembre-se:** É necessário especificar o nome completo do arquivo de resposta. Especificar o caminho relativo faz com que a instalação falhe sem nenhuma indicação de que ocorreu um erro.

- a. Faça uma cópia do arquivo de resposta a ser customizado.

Para a instalação completa do WebSphere eXtreme Scale, copie o arquivo de resposta do DVD do produto para sua unidade de disco.

Para o WebSphere eXtreme Scale Client, descompacte o arquivo zip do WebSphere eXtreme Scale Client em seu disco rígido e localize o arquivo de resposta.

- b. Abra e edite o arquivo de resposta no editor de texto de sua escolha. O arquivo de resposta de exemplo anterior fornece detalhes sobre como especificar cada um dos parâmetros. É necessário especificar os seguintes parâmetros:

- O contrato de licença
- O diretório de instalação

**Dica:** Quando você instala o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client em um ambiente do WebSphere Application Server, o instalador usa o diretório de instalação para determinar onde a instância existente do WebSphere Application Server está instalada. Se você instalar em um nó que contenha várias instâncias do WebSphere Application Server, defina claramente o seu local.

- c. Execute o seguinte script para iniciar a instalação.

**Para a instalação completa do WebSphere eXtreme Scale:**

```
./install.sh|bat -options C:/drive_path/response_file.txt -silent
```

**Para a instalação do WebSphere eXtreme Scale Client:**

```
./WXS_Client/install.sh|bat -options C:/drive_path/response_file.txt -silent
```

Você também pode usar o arquivo de resposta quando executar uma instalação da GUI. O arquivo de resposta pode ser usado com a instalação da GUI para depurar problemas que ficam ocultos com a instalação silenciosa. Quando especifica o arquivo `wxssetup.response` para a GUI ou para instalações silenciosas, você deve usar o caminho completo. Execute o seguinte script para executar a instalação da GUI com seu arquivo de resposta:

- `Linux` `UNIX` `<install_home>/install.sh -options <full_install_path_required>/wxssetup.response`
- `Windows` `<install_home>\install.exe -options c:\<full_install_path_required>\wxssetup.response`

2. Opcional: Se você escolher instalar o eXtreme Scale ao transmitir determinados parâmetros para a linha de comandos, execute o seguinte script para iniciar a instalação:

**Para a instalação completa do WebSphere eXtreme Scale:**

```
./install.sh|bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=install_location
```

**Para a instalação do WebSphere eXtreme Scale Client:**

```
./WXS_Client/install.sh|bat -silent -OPT silentInstallLicenseAcceptance=true -OPT installLocation=install_location
```

## Arquivo de Resposta para Instalação Silenciosa

Especifique os parâmetros na linha de comandos para customizar e configurar a instalação do produto.

**Nota:** É necessário especificar o nome completo do arquivo de resposta. Especificar o caminho relativo faz com que a instalação falhe sem nenhuma indicação de que ocorreu um erro.

## Parâmetros

É possível transmitir os seguintes parâmetros durante a instalação da linha de comandos ou do arquivo de opções do produto:

### **-silent**

Suprime a interface gráfica com o usuário (GUI). Especifique o parâmetro **-options** para indicar que o instalador conclui a instalação de acordo com um arquivo de opções customizadas. Se você não especificar o parâmetro **-options**, os valores padrão serão usados no lugar.

#### **Exemplo de Uso**

```
./install.sh|bat -silent -options options_file.txt
```

### **-options path\_name/file\_name**

Especifica um arquivo de opções usado pelo instalador para executar uma instalação silenciosa. As propriedades na linha de comandos têm precedência.

#### **Exemplo de Uso**

```
./install.sh|bat -options c:/path_name/options_file.txt
```

### **-log #!file\_name @event\_type**

Gera um arquivo de log de instalação que registra os seguintes tipos de eventos:

- err
- wrn
- msg1
- msg2
- dbg
- TODOS

#### **Exemplo de Uso**

```
./install.sh|bat -log # !c:/temp/logfiles.txt @ALL
```

### **-is:log path\_name/file\_name**

Cria um arquivo de log que contém as procuras do Java Virtual Machine (JVM) do instalador enquanto tenta iniciar a GUI. O arquivo de log não é criado, a menos que seja especificado.

#### **Exemplo de Uso**

```
./install.sh|bat -is:log c:/logs/javalog.txt
```

### **-is:javaconsole**

Exibe uma janela do console durante o processo de instalação.

#### **Exemplo de Uso**

```
./install.sh|bat -is:javaconsole
```

### **-is:silent**

Suprime a janela de inicialização Java que é exibida quando o instalador é iniciado.

#### **Exemplo de Uso**

```
./install.sh|bat -is:silent
```

**-is:tempdir** *path\_name*

Especifica o diretório temporário que o instalador usa durante a instalação.

#### Exemplo de Uso

```
./install.sh|bat -is:tempdir c:/temp
```

---

## Instalando o Serviço de Dados REST

Este tópico descreve como instalar o serviço de dados REST do WebSphere eXtreme Scale em um servidor da Web.

### Antes de Iniciar

#### Requisitos de Software

O serviço de dados REST do WebSphere eXtreme Scale é um aplicativo da Web Java que pode ser implementado para qualquer servidor de aplicativos que suporte a especificação de servlet Java, Versão 2.3 e um ambiente de tempo de execução Java, Versão 5 ou posterior.

O software a seguir é necessário:

- Java Standard Edition 5 ou posterior
- Contêiner do servlet da Web, Versão 2.3 ou posterior, que inclui uma das opções a seguir:
  - WebSphere Application Server Versão 6.1.0.25 ou posterior
  - WebSphere Application Server Versão 7.0.0.5 ou posterior
  - WebSphere Community Edition Versão 2.1.1.3 ou posterior
  - Apache Tomcat Versão 5.5 ou posterior
- WebSphere eXtreme Scale, Versão 7.1 ou posterior, incluindo o teste.

### Sobre Esta Tarefa

O serviço de dados REST do WebSphere eXtreme Scale inclui um único arquivo `wxsrestservice.war`. O arquivo `wxsrestservice.war` inclui um único servlet que age como um gateway entre os aplicativos clientes do WCF Data Services ou qualquer outro cliente HTTP REST e uma grade de dados.

O serviço de dados REST inclui uma amostra que o permite criar rapidamente uma grade de dados e interagir com ela usando o cliente do eXtreme Scale ou o serviço de dados REST. Consulte o “Configurando Serviços de Dados REST” na página 350 para obter detalhes sobre como usar a amostra.

Quando o WebSphere eXtreme Scale 7.1 é instalado ou o teste do eXtreme Scale Versão 7.1 é extraído, os seguintes diretórios e arquivos são incluídos:

- `restservice_home/lib`
  - O diretório `lib` contém esses arquivos:
    - `wxsrestservice.ear` – O archive do aplicativo corporativo do serviço de dados REST para uso com WebSphere Application Server e WebSphere Application Server CE.
    - `wxsrestservice.war` – O módulo da Web de serviço de dados REST para uso com Apache Tomcat.
  - O arquivo `wxsrestservice.ear` inclui o arquivo `wxsrestservice.war` e os dois são estreitamente acoplados ao tempo de execução do WebSphere WebSphere

eXtreme Scale. Se for feito o upgrade do WebSphere eXtreme Scale para uma nova versão um se um fix pack for aplicado, o arquivo `wxsrestservice.war` ou o arquivo `wxsrestservice.ear` precisarão ter seu upgrade feito manualmente para a versão instalada neste diretório.

- `restservice_home/gettingstarted`

O diretório `gettingstarted` contém uma amostra simples que demonstra como usar o serviço de dados REST do WebSphere eXtreme Scale com uma grade de dados.

## Procedimento

Empacote e implemente o serviço de dados REST.

O serviço de dados REST é projetado como um módulo WAR autocontido. Para configurar o serviço de dados REST, você deve primeiro empacotar a configuração do serviço de dados REST e os arquivos de configuração opcionais do WebSphere eXtreme Scale em um arquivo ou diretório JAR. Essa compactação de aplicativo é referida pelo tempo de execução do servidor de contêiner de Web. O diagrama a seguir ilustra os arquivos usados pelo serviço de dados REST do eXtreme Scale. O diretório ou JAR de configuração do serviço REST deve conter o seguinte

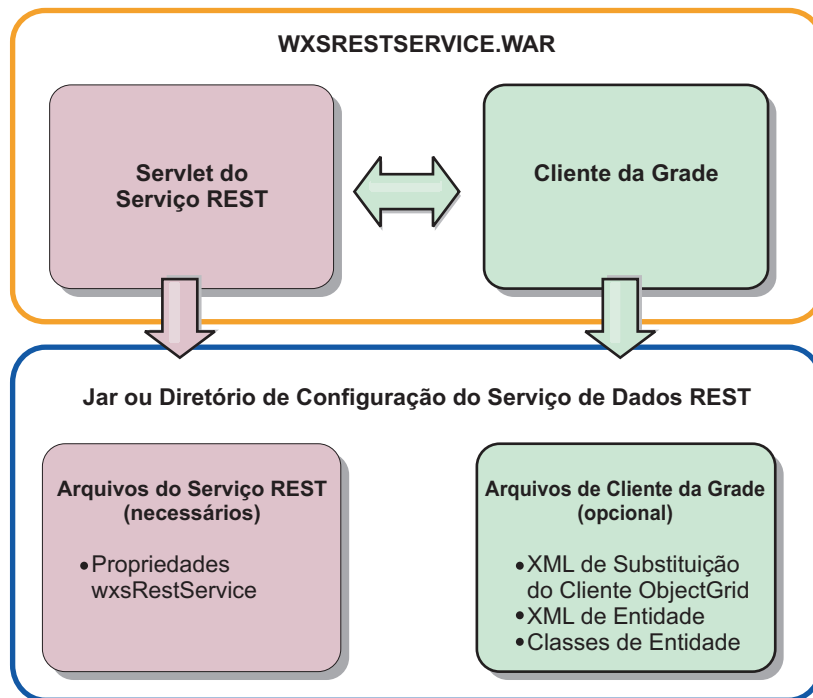


Figura 26. Arquivos do Serviço de Dados REST do WebSphere eXtreme Scale

arquivo:

`wxsRestService.properties`: O arquivo `wxsRestService.properties` inclui as opções de configuração para o serviço de dados REST. Isso inclui terminais de serviço de catálogo, nomes de ObjectGrid para expor, opções de rastreamento e mais. Consulte Arquivo de Propriedades do Serviço de Dados REST.

Os arquivos de cliente do ObjectGrid a seguir são opcionais:

- `META-INF/objectGridClient.xml`: O arquivo XML de substituição do cliente do ObjectGrid é usado para se conectar à grade de dados remota. Por padrão, esse arquivo não é necessário. Se este arquivo não estiver presente, o serviço REST usará a configuração do servidor, desativando o cache próximo.

O nome do arquivo pode ser substituído com o uso da propriedade de configuração do serviço de dados REST do objectGridClientXML. Se fornecido, esse arquivo XML deverá incluir:

1. Qualquer ObjectGrid que desejar expor ao serviço de dados REST.
  2. Inclua uma referência para o arquivo XML do descritor de entidade associado a cada configuração de ObjectGrid.
- *META-INF/arquivos XML do descritor de entidades*: Um ou mais arquivos XML do descritor de entidades serão necessários apenas se o cliente precisar substituir a definição de entidade do cliente. O arquivo XML do descritor de entidade deve ser utilizado junto com o arquivo descritor XML de substituição de cliente do ObjectGrid.
  - **Classes de entidade** Classes de entidade anotadas ou um arquivo XML de descritor de entidade podem ser utilizados para descrever os metadados da entidade. O serviço REST exigirá classes de entidades no caminho da classe apenas se os servidores eXtreme Scale forem configurados com classes de metadados de entidades e um descritor XML de entidade de substituição do cliente não for usado.

Um exemplo com o arquivo de configuração requerido mínimo, no qual as entidades são definidas em XML nos servidores:

```
restserviceconfig.jar:  
wxsRestService.properties
```

O arquivo de propriedades contém:

```
catalogServiceEndpoints=localhost:2809  
objectGridNames=NorthwindGrid
```

Um exemplo com uma entidade, arquivos XML de substituição e classes de entidade:

```
restserviceconfig.jar:  
wxsRestService.properties
```

O arquivo de propriedades contém:

```
catalogServiceEndpoints=localhost:2809  
objectGridNames=NorthwindGrid  
com/acme/entities/Customer.class  
META-INF/objectGridClient.xml
```

O arquivo XML do descritor de ObjectGrid cliente contém:

```
<objectGrid  
name="CustomerGrid" entityMetadataXMLFile="emd.xml"/>  
META-INF/emd.xml
```

O arquivo XML do descritor de metadados da entidade contém:

```
<entity  
class-name="com.acme.entities.Customer" name="Customer"/>
```

---

## Instalando a Estrutura do Eclipse Equinox OSGi com o Eclipse Gemini para Clientes e Servidores

Se desejar implementar o WebSphere eXtreme Scale na estrutura do OSGi, você deverá configurar o Ambiente do Eclipse Equinox.

### Sobre Esta Tarefa

A tarefa requer fazer o download e instalar a estrutura do blueprint, que permite configurar posteriormente o JavaBeans e expô-los como serviços. O uso de serviços é importante porque é possível expor plug-ins como serviços OSGi, de modo que eles possam ser usados pelo ambiente de tempo de execução do eXtreme Scale. O

produto suporta dois contêineres blueprint dentro da estrutura OSGi principal do Eclipse Equinox: Eclipse Gemini e Aries Apache. Use este procedimento para configurar o contêiner do Eclipse Gemini.

## Procedimento

1. Faça download do Eclipse Equinox SDK Versão 3.6.1 ou posterior a partir do website do Eclipse. Crie um diretório para a estrutura do Equinox, por exemplo: `/opt/equinox`. Essas instruções referenciam esse diretório como `equinox_root`. Extraia o arquivo compactado no diretório `equinox_root`.
2. Faça download do arquivo compactado `gemini-blueprint incubation 1.0.0` a partir do Website Eclipse. Extraia o conteúdo do arquivo em um diretório temporário e copie os seguintes arquivos extraídos para o diretório `equinox_root/plugins`:  

```
dist/gemini-blueprint-core-1.0.0.jar
dist/gemini-blueprint-extender-1.0.0.jar
dist/gemini-blueprint-io-1.0.0.jar
```
3. Faça download do Spring Framework Versão 3.0.5 a partir da página da web SpringSource a seguir: <http://www.springsource.com/download/community>. Extraia-o em um diretório temporário e copie os seguintes arquivos extraídos para o diretório `equinox_root/plugins`:  

```
org.springframework.aop-3.0.5.RELEASE.jar
org.springframework.asm-3.0.5.RELEASE.jar
org.springframework.beans-3.0.5.RELEASE.jar
org.springframework.context-3.0.5.RELEASE.jar
org.springframework.core-3.0.5.RELEASE.jar
org.springframework.expression-3.0.5.RELEASE.jar
```
4. Faça download do arquivo Java archive (JAR) AOP Alliance a partir da página da web do SpringSource. Copie o arquivo `com.springsource.org.aopalliance-1.0.0.jar` para o diretório `equinox_root/plugins`.
5. Faça download do arquivo JAR Apache Commons Logging 1.1.1 a partir da página da web do SpringSource. Copie o arquivo `com.springsource.org.apache.commons.logging-1.1.1.jar` para o diretório `equinox_root/plugins`.
6. Faça download do cliente de linha de comandos Luminis OSGi Configuration Admin. Use esse pacote configurável para gerenciar as configurações administrativas do OSGi. É possível fazer o download do arquivo JAR a partir da seguinte página da web: <https://opensource.luminis.net/wiki/display/SITE/OSGi+Configuration+Admin+command+line+client>. Copie o arquivo `net.luminis.cmc-0.2.5.jar` para o diretório `equinox_root/plugins`.
7. Faça download do pacote configurável do arquivo de instalação Apache Felix Versão 3.0.2 a partir da seguinte página da web: <http://felix.apache.org/site/index.html>. Copie o arquivo `org.apache.felix.fileinstall-3.0.2.jar` para o diretório `equinox_root/plugins`.
8. Crie um diretório de configuração dentro do diretório `equinox_root/plugins`, por exemplo:  

```
mkdir equinox_root/plugins/configuration
```
9. Crie o arquivo `config.ini` a seguir no diretório `equinox_root/plugins/configuration`, substituindo `equinox_root` com o caminho absoluto para seu diretório `equinox_root` e removendo todos os espaços à direita após a barra invertida em cada linha. Você deve incluir uma linha em branco no final do arquivo; por exemplo:  

```
osgi.noShutdown=true
osgi.java.profile.bootdelegation=none
org.osgi.framework.bootdelegation=none
eclipse.ignoreApp=true
osgi.bundles=\\
```



```

org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \
org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \
org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \
com.springsource.org.apache.commons.logging-1.1.1.jar@1:start, \
com.springsource.org.aopalliance-1.0.0.jar@1:start, \
org.springframework.aop-3.0.5.RELEASE.jar@1:start, \
org.springframework.asm-3.0.5.RELEASE.jar@1:start, \
org.springframework.beans-3.0.5.RELEASE.jar@1:start, \
org.springframework.context-3.0.5.RELEASE.jar@1:start, \
org.springframework.core-3.0.5.RELEASE.jar@1:start, \
org.springframework.expression-3.0.5.RELEASE.jar@1:start, \
org.apache.felix.fileinstall-3.0.2.jar@1:start, \
net.luminis.cmc-0.2.5.jar@1:start, \
gemini-blueprint-core-1.0.0.jar@1:start, \
gemini-blueprint-extender-1.0.0.jar@1:start, \
gemini-blueprint-io-1.0.0.jar@1:start

```

Se já tiver configurado o ambiente, será possível limpar o repositório de plug-in do Equinox ao remover o seguinte diretório: `equinox_root\plugins\configuration\org.eclipse.osgi`.

10. Execute os seguintes comandos para iniciar o console do equinox.

Se você estiver executando uma versão diferente do Equinox, o nome do seu arquivo JAR será diferente do nome no exemplo a seguir:

```
java -jar plugins\org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
```

## Instalando Pacotes Configuráveis do eXtreme Scale

O WebSphere eXtreme Scale inclui pacotes configuráveis que podem ser instalados em uma estrutura do Eclipse Equinox OSGi. Esses pacotes configuráveis são necessários para iniciar servidores do eXtreme Scale ou usar clientes do eXtreme Scale no OSGi.

### Antes de Iniciar

Esta tarefa assume que os produtos a seguir foram instalados:

- Estrutura do Eclipse Equinox OSGi
- Cliente ou servidor independente do eXtreme Scale

### Sobre Esta Tarefa

O eXtreme Scale inclui dois pacotes configuráveis. Apenas um dos pacotes configuráveis a seguir é requerido em uma estrutura do OSGi:

#### **objectgrid.jar**

O pacote configurável do servidor é o arquivo `objectgrid.jar` e é instalado com a instalação do servidor independente do eXtreme Scale e é requerido para executar servidores do eXtreme Scale e também pode ser usado para executar clientes do eXtreme Scale ou caches na memória locais. O ID do pacote configurável para o arquivo `objectgrid.jar` é `com.ibm.websphere.xs.server_<version>`, em que a versão está no formato: `<Version>.<Release>.<Modification>`. Por exemplo, o pacote configurável do servidor para o eXtreme Scale versão 7.1.1 é `com.ibm.websphere.xs.server_7.1.1`.

#### **ogclient.jar**

O pacote configurável do `ogclient.jar` é instalado com as instalações independentes e de cliente do eXtreme Scale e é usado para executar clientes do eXtreme Scale ou caches na memória locais. O ID do pacote configurável para o arquivo `ogclient.jar` é `com.ibm.websphere.xs.client_<version>`, em que a versão está no formato: `<Version>_<Release>_<Modification>`. Por exemplo, o pacote configurável do cliente para o eXtreme Scale Versão 7.1.1 é `com.ibm.websphere.xs.client_7.1.1`.

Para obter mais informações sobre como desenvolver plug-ins do eXtreme Scale, consulte o tópico APIs e Plug-ins do Sistema.

## Procedimento

Para instalar o pacote configurável do cliente ou servidor do eXtreme Scale na estrutura do Eclipse Equinox OSGi usando o console do OSGi:

1. Inicie a estrutura do Eclipse Equinox com o console ativado; por exemplo:  

```
java_home/bin/java -jar <equinox_root>/plugins/  
org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
```
2. Instale o pacote configurável do cliente ou servidor do eXtreme Scale no console do Equinox:  

```
osgi> install file:///<path to bundle>
```
3. O Equinox exibe o ID do pacote configurável para o pacote configurável recém-instalado:  

```
Bundle id is 25
```
4. Inicie o pacote configurável no console do Equinox, em que <id> é o ID do pacote configurável designado quando o pacote configurável foi instalado:  

```
osgi> start <id>
```
5. Recupere o status de serviço no console do Equinox para verificar se o pacote configurável foi iniciado; por exemplo:  

```
osgi> ss
```

Quando o pacote configurável foi iniciado com êxito, o pacote configurável exibe o estado ACTIVE; por exemplo:

```
25      ACTIVE      com.ibm.websphere.xs.server_7.1.1
```

Instale o pacote configurável do cliente ou servidor do eXtreme Scale na estrutura do Eclipse Equinox OSGi usando o arquivo config.ini:

6. Copie o pacote configurável do cliente ou servidor do eXtreme Scale (objectgrid.jar ou ogclient.jar) do <wxs\_install\_root>/ObjectGrid/lib para o diretório de plug-ins do Eclipse Equinox; por exemplo: <equinox\_root>/plugins
7. Edite o arquivo de configuração config.ini do Eclipse Equinox e inclua o pacote configurável na propriedade osgi.bundles; por exemplo:  

```
osgi.bundles=\  
org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \  
org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \  
org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \  
objectgrid.jar@1:start
```

**Importante:** Verifique se existe uma linha em branco após o nome do último pacote configurável. Cada pacote configurável é separado por uma vírgula.

8. Inicie a estrutura do Eclipse Equinox com o console ativado; por exemplo:  

```
java_home/bin/java -jar <equinox_root>/plugins/  
org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
```
9. Recupere o status do serviço no console do Equinox para verificar se o pacote configurável foi iniciado:  

```
osgi> ss
```

Quando o pacote configurável tiver iniciado com sucesso, o pacote configurável exibirá o estado ACTIVE; por exemplo:

```
25      ACTIVE      com.ibm.websphere.xs.server_7.1.1
```

## Resultados

O pacote configurável do servidor ou cliente do eXtreme Scale é instalado e iniciado em sua estrutura do Eclipse Equinox OSGi.

---

## Verificando a Instalação

Depois que o assistente de instalação for concluído, será possível verificar a instalação ao consultar os vários aspectos da instalação.

### Procedimento

- **Para uma instalação independente ou uma instalação que é integrada ao WebSphere Application Server:**

Use um dos seguintes métodos para verificar se sua instalação foi concluída com êxito:

- Execute o comando de informações de versão para WebSphere eXtreme Scale:  
`was_root/lib/> java -jar wsobjectgrid.jar version`

O número da versão do nome do produto e o número de construção são exibidos como resultado.

- Verifique os arquivos de propriedades para obter o número da versão adequado.

- Arquivos de assinatura: Os arquivos de assinaturas estão no diretório `was_root/properties/version`. Se um fix pack foi instalado, arquivos adicionais `fxtg` também serão incluídos. A seguir há alguns exemplos de nomes de arquivos de assinatura:

```
WebSphere_eXtreme_Scale.7.1.1..swtag  
WebSphere_eXtreme_Scale.7.1.0.2.fxtag  
WebSphere_eXtreme_Scale.7.1.0.3.fxtag
```

- Arquivo do produto WebSphere eXtreme Scale:

O arquivo do produto está no diretório `was_root/properties/version`. Procure pelo arquivo `WXS.product`. A seguir há um exemplo do conteúdo desse arquivo:

```
<?xml version="1.0" encoding="UTF-8"?>  
  <!DOCTYPE product SYSTEM "product.dtd">  
  <product name="IBM WebSphere eXtreme Scale">  
    <id>WXS</id>  
    <version>7.1.1.0</version>  
    <build-info  
      date="8/5/11"  
      level="a1132.68720"/>  
  </product>
```

- Verifique se os arquivos do tempo de execução estão instalados. As listas dos arquivos de tempo de execução para cada tipo de instalação são documentadas nos tópicos a seguir:

- “Arquivos de Tempo de Execução para Instalação Independente do WebSphere eXtreme Scale” na página 193
- “Arquivos de Tempo de Execução para o WebSphere eXtreme Scale Integrado ao WebSphere Application Server” na página 163

- **Para uma instalação integrada ao WebSphere Application Server, há as seguintes formas adicionais para verificar se a instalação foi concluída com êxito:**

- Execute o comando de informações de versão para WebSphere Application Server:

```
was_root/bin/> versionInfo.sh|.bat
```

A saída exibe uma lista dos produtos instalados, incluindo diretórios de instalação, produtos instalados, versões, o nível de construção, a data da construção, e assim por diante.

**Dica:** Inclua o parâmetro **-maintenancePackages** para ver detalhes adicionais:

```
was_root/bin/> versionInfo.sh|.bat -maintenancePackages
```

- Verifique o painel Bem-Vindo para o console administrativo WebSphere Application Server. Acesse <http://localhost:9060/ibm/console>. Efetue login no console. A versão do WebSphere eXtreme Scale é exibida no painel Bem-Vindo.

- Use o console do First Steps para aumentar a instalação do WebSphere Application Server com o WebSphere eXtreme Scale:

```
was_root/firststeps/WXS> firststeps.sh|.bat
```

Para obter informações adicionais, consulte “Criando e Alterando Perfis para o WebSphere eXtreme Scale” na página 182.

## O que Fazer Depois

Se observar que a instalação não foi concluída conforme o esperado, você deverá resolver o problema da instalação. Para obter informações adicionais, consulte “Instalação de Resolução de Problemas” na página 206.

---

## Executando as Primeiras Etapas Após a Instalação

Depois de concluir e verificar a instalação, é possível começar a usar o WebSphere eXtreme Scale para criar sua grade de dados.

### Procedimento

1. Atualize sua instalação ao aplicar a manutenção

**Mais informações:** “Atualizando Servidores eXtreme Scale” na página 209.

2. Se estiver usando o WebSphere eXtreme Scale pela primeira vez, será possível usar as informações de Introdução para aprender mais sobre como usar o produto.

**Mais informações:** Capítulo 1, “Introdução”, na página 1

3. Configure o produto. Crie propriedades e arquivos XML para definir a configuração para grades de dados, servidores e clientes. Também é possível configurar a integração de cache ou de banco de dados, serviços de dados REST ou os plug-ins OSGi.

**Mais informações:** Capítulo 6, “Configurando”, na página 219

4. Desenvolva um aplicativo que acesse a grade de dados.

**Mais informações:** Desenvolvendo Aplicativos

5. Inicie e administre os servidores de contêiner e de catálogo com seus arquivos de configuração e o aplicativo de grade de dados.

**Mais informações:** Capítulo 7, “Administrando”, na página 387

6. Monitore o desempenho de sua configuração com as várias ferramentas de monitoramento.

**Mais informações:** Capítulo 8, “Monitoramento”, na página 433

---

## Instalação de Resolução de Problemas

Use estas informações para resolver problemas com sua instalação.

### Procedimento

- **Problema:** Quando você executa o comando de instalação a partir de um computador remoto, como \\mymachine\downloads\, a seguinte mensagem é exibida: CMD.EXE foi iniciado com o caminho acima como o diretório atual. Caminhos UNC não são suportados. Padronizando para o diretório Windows. Como resultado, a instalação não é concluída corretamente.  
**Solução:** Mapeie o computador remoto para uma unidade de rede. Por exemplo, no Windows, é possível clicar com o botão direito em **Meu Computador**, escolher **Mapear Unidade de Rede** e incluir o caminho das convenções de nomenclatura uniformes (UNC) no computador remoto. Em seguida, é possível executar o script de instalação a partir da unidade de rede com êxito, por exemplo: y:\mymachine\downloads\WXS\install.bat.
- **Problema:** A instalação não é concluída com êxito.  
**Solução:** Verifique os arquivos de log para ver onde a instalação falhou. Quando a instalação não é concluída com êxito, os logs estão no diretório `wxs_install_root/logs/wxs`.
- **Problema:** Uma falha catastrófica ocorre durante a instalação.  
**Solução:** Verifique os arquivos de log para ver onde a instalação falhou. Quando a instalação falha quando estiver parcialmente concluída, os logs geralmente podem estar localizados no diretório `user_root/wxs_install_logs/`.
- **Windows Problema:** Se estiver instalando o WebSphere eXtreme Scale Client no Windows, o seguinte texto poderá ser exibido nos resultados da instalação:  
Sucesso: A instalação do seguinte produto foi bem-sucedida:  
WebSphere eXtreme Scale Client. Algumas etapas de configuração têm erros.  
Para obter mais informações, consulte o seguinte arquivo de log:  
<WebSphere Application Server install root>\logs\wxs\_client\install\log.txt"  
Revise o log de instalação (log.txt) e o log de aumento do gerenciador de implementação  
.  
  
**Solução:** Se uma falha for vista com o arquivo `iscdeploy.sh`, o erro poderá ser ignorado. Este erro não causa nenhum problema.

---

## Desinstalando o WebSphere eXtreme Scale

Para remover o WebSphere eXtreme Scale do seu ambiente, é possível usar o assistente ou desinstalar silenciosamente o produto.

### Antes de Iniciar

**Atenção:** O desinstalador remove todos os arquivos binários e toda a manutenção, como fix packs e correções temporárias, ao mesmo tempo.

### Procedimento

1. Pare todos os processos que executam o eXtreme Scale.

#### CUIDADO:

**Certifique-se de que quaisquer processos em execução sejam interrompidos. Se os processos em execução não forem interrompidos, a desinstalação continuará, criando resultados imprevisíveis e deixando a desinstalação em um estado indeterminado em algumas plataformas.**

- Se você instalou o eXtreme Scale independente, leia sobre como parar servidores independentes para parar processos.
  - Se você instalou o eXtreme Scale com uma instalação existente do WebSphere Application Server, leia sobre utilitários de linha de comandos para obter mais informações sobre como parar processos do WebSphere Application Server.
  - Se você estiver executando o console da web, use o comando `stopConsoleServer` para parar o servidor do console da web. O script `stopConsoleServer` está no diretório `wxs_install_root/ObjectGrid/bin`. Se esse servidor não for parado antes de executar a desinstalação, o processo será parado automaticamente durante o processo de desinstalação.
2. Desinstale o produto. É possível executar a desinstalação em uma GUI ou silenciosamente.

**Nota:** Ao especificar o arquivo `wxssetup.response` do arquivo de resposta para instalações ou desinstalações silenciosas ou da GUI, o caminho completo deve sempre ser especificado. O arquivo de resposta é opcional para a desinstalação da GUI.

- **Para executar uma desinstalação usando a GUI:**

- `Linux` `UNIX` `<install_home>/uninstall_wxs/uninstall`
- `Windows` `<install_home>\uninstall_wxs\uninstall.exe`

Se você quiser executar a desinstalação usando a GUI e o arquivo `wxssetup.response`, use um dos comandos a seguir:

- `Linux` `UNIX`  
`<install_home>/uninstall_wxs/uninstall -options  
 <full_install_path_required>/wxssetup.response`
- `Windows`  
`<install_home>\uninstall_wxs\uninstall.exe -options  
 <full_install_path_required>\wxssetup.response`

- **Para executar a desinstalação silenciosamente usando o script `wxssetup.response` do arquivo de resposta:**

- `Linux` `UNIX`  
`<install_home>/uninstall_wxs/uninstall -options  
 <full_install_path_required>/wxssetup.response -silent`
- `Windows`  
`<install_home>\uninstall_wxs\uninstall.exe -options  
 <full_install_path_required>\wxssetup.response -silent`

## Resultados

Você removeu o eXtreme Scale do seu ambiente.





---

## Capítulo 5. Fazendo Upgrade e Migrando o WebSphere eXtreme Scale



É possível migrar para a Versão 7.1.1 a partir de versões anteriores e também aplicar pacotes de manutenção na Versão 7.1.1. Para evitar indisponibilidades, você deve considerar a ordem na qual as atualizações são aplicadas nos servidores em sua configuração.

- Para fazer upgrade de uma instalação da Versão 7.1.0.x, consulte o “Atualizando Servidores eXtreme Scale” e o “Usando o Update Installer para instalar os pacotes de manutenção” na página 213.
- Para fazer upgrade de uma instalação da Versão 7.0.x, consulte o “Atualizando Servidores eXtreme Scale” e o “Migrando para o WebSphere eXtreme Scale Versão 7.1.1” na página 212.

---

### Atualizando Servidores eXtreme Scale

É possível atualizar o WebSphere eXtreme Scale para uma nova versão, ou aplicando a manutenção ou instalando uma nova versão, sem interromper o serviço.

#### Antes de Iniciar

Você deve ter o arquivo binário para a liberação ou manutenção de versão principal que deseja aplicar. Informações mais recentes sobre liberações e pacotes de manutenção disponíveis podem ser obtidas a partir do Portal de Suporte IBM para WebSphere eXtreme Scale.

#### Sobre Esta Tarefa

Para fazer upgrade sem a interrupção do serviço, primeiro faça upgrade de seus servidores de catálogo. Em seguida, faça upgrade dos servidores de contêineres e dos clientes.

#### Procedimento

1. Faça upgrade da camada de serviço de catálogo, repetindo as etapas a seguir para cada servidor de catálogos na grade de dados. Faça upgrade da camada do serviço de catálogo antes de fazer upgrade de quaisquer servidores de contêiner ou de clientes. Servidores de catálogos individuais podem interoperar com compatibilidade de versão, de modo que os upgrades possam ser aplicados em um servidor de catálogos por vez sem interromper o serviço.

- a. Verifique o status do quorum funcional. Execute o seguinte comando:

```
xsadmin -quorumStatus  
xscmd -c showQuorumStatus
```

Este resultado indica que todos os servidores de catálogos estão conectados.

- b. Se estiver usando replicação multimestre entre dois domínios de serviço de catálogo, desfaça o link entre os dois domínios do serviço de catálogo enquanto estiver fazendo upgrade dos servidores de catálogos.

```
xsadmin -ch host -p 1099 -dismissLink domain_name
```

**7.1.1+**

```
xscmd -c dismissLink -cep host:2809 -fd domain_name
```

É necessário apenas executar este comando a partir de um dos domínios do serviço de catálogo para remover o link entre os dois domínios do serviço de catálogo.

- c. Encerre um dos servidores de catálogos. É possível usar o comando **stop0gserver**, o comando **xscmd -c teardown** ou encerrar o servidor de aplicativos que está executando o serviço de catálogo no WebSphere Application Server. Não há nenhum requisito referente à ordem na qual os servidores de catálogos são interrompidos, mas encerrar o servidor de catálogos principal por último reduz a rotatividade. Para determinar qual servidor de catálogos é o principal, procure pela mensagem CWOBJ8106 nos arquivos de log. Sob condições normais, o quorum é mantido quando um servidor de catálogos é encerrado, mas é uma melhor prática consultar o status do quorum após cada encerramento com o comando **xscmd -c showQuorumStatus**.

Se você usar o comando **xscmd -c teardown**, será possível filtrar os nomes de servidor. O comando **stop0gServer** requer um nome de servidor exato ou uma lista de nomes de servidores em paralelo a serem inseridos. Você deve agrupar o processo de encerramento em vez de chamar o processo de parada ou de derrubada para vários servidores em paralelo. Ao agrupar os servidores a serem encerrados, a grade de dados poderá reagir aos servidores que estiverem sendo encerrados ao mover os shards ao redor da grade de dados. É possível usar um dos seguintes comandos para encerrar seus servidores:

Uma lista específica de servidores pode ser fornecida para parar com o comando **stop0gServer** ou **xscmd -c teardown**:

```
stop0gServer <server_name>[,<server_name>]
```

```
xsadmin -teardown <server_name>[,<server_name>]
```

#### 7.1.1+

```
xscmd -c teardown -sl <server_name>[,<server_name>]
```

Com os exemplos anteriores, os comandos **stop0gServer** ou **xscmd -c teardown** concluem as mesmas tarefas de encerramento. No entanto, é possível filtrar os servidores para parar com o comando **xscmd -c teardown**. Consulte o “Parando Servidores Corretamente com o Utilitário **xscmd**” na página 400 para obter mais informações sobre como filtrar os servidores por zona ou nome do host. O comando **teardown** filtra os servidores correspondentes e pergunta se os servidores selecionados estão corretos.

- d. Instale as atualizações no servidor de catálogos. É possível migrar o servidor de catálogos para uma nova liberação principal do produto ou aplicar um pacote de manutenção. Consulte os tópicos a seguir para obter mais informações:
- Para migrar de uma instalação da Versão 7.0.x : “Migrando para o WebSphere eXtreme Scale Versão 7.1.1” na página 212
  - Para migrar de uma instalação da Versão 7.1.0.x : “Usando o Update Installer para instalar os pacotes de manutenção” na página 213
- e. Reinicie o servidor de catálogos.

Se estiver usando um ambiente independente, consulte o “Iniciando um Serviço de Catálogo Independente” na página 387 para obter mais informações. Se estiver usando um ambiente do WebSphere Application Server, consulte o “Iniciando e Parando Servidores em um Ambiente do WebSphere Application Server” na página 401 para obter mais informações.

O servidor de catálogos é executado no modo de compatibilidade até que todos os servidores de catálogos sejam movidos para o mesmo nível. Normalmente o modo de compatibilidade aplica-se às migrações da liberação principal já que novas funções não estão disponíveis nos servidores que não são migrados. Não existe nenhuma restrição sobre quanto tempo os servidores de catálogo podem ser executados no modo de compatibilidade, porém a melhor prática é migrar todos os servidores de catálogos para o mesmo nível o mais breve possível.

- f. Aplique as atualizações nos servidores de catálogos restantes em sua configuração.
2. Faça upgrade dos servidores de contêiner, repetindo as etapas a seguir para cada servidor de contêiner na grade de dados. É possível fazer upgrade dos servidores de contêiner em qualquer ordem. No entanto, considere atualizar primeiro os servidores e depois os clientes, caso esteja usando novas funções no upgrade.
    - a. Pare os servidores de contêiner que deseja fazer upgrade. É possível parar a camada do servidor de contêiner em grupos com o comando **stopOgserver** ou o comando **teardown**. Enviando em lote operações de derrubada e executando operações de início do servidor em paralelo, o mecanismo de posicionamento pode mover shards em grupos maiores.  
`xsadmin -teardown -fz DefaultZone`

#### 7.1.1+

```
xscmd -c teardown -z DefaultZone
```

```
Connecting to Catalog service at localhost:1099
```

```
Processing filter options for Server teardown
```

```
The following servers will be torn down:
```

```
container00  
container01  
container02  
container03  
container04
```

```
Do you want to tear down the listed servers? (Y/N)
```

- b. Instale as atualizações nos servidores de contêiner. É possível migrar os servidores de contêiner para uma nova liberação principal do produto ou aplicar um pacote de manutenção. Consulte os tópicos a seguir para obter mais informações:
    - Para migrar de uma instalação da Versão 7.0.x : “Migrando para o WebSphere eXtreme Scale Versão 7.1.1” na página 212
    - Para migrar de uma instalação da Versão 7.1.0.x : “Usando o Update Installer para instalar os pacotes de manutenção” na página 213
  - c. Reinicie seus servidores de contêiner.
  - d. Faça upgrade de quaisquer servidores de contêiner restantes em sua configuração.
3. Se estiver usando replicação multimestre, reconecte os domínios do serviço de catálogo. Use o comando **xscmd -c establishLink** para reconectar a domínios do serviço de catálogo. **7.1.1+**  
`xsadmin -ch host -p 1099 -establishLink dname fdHostA:2809,fdHostB:2809`  
`xscmd -c establishLink -cep host:2809 -fd dname -fe fdHostA:2809,fdHostB:2809`

## O que Fazer Depois

Também é possível usar estas etapas para reverter para uma versão mais antiga ou para desinstalar pacotes de manutenção. No entanto, se você reverter para a Versão 7.1.0 quando estiver usando replicação multimestre, a replicação bidirecional poderá não funcionar corretamente quando ao restabelecer os links. Nesta situação, reinicie os domínios de serviço de catálogo e vincule novamente os domínios de serviço de catálogo ao comando **establishLink**.

---

## Migrando para o WebSphere eXtreme Scale Versão 7.1.1

Com o instalador do WebSphere eXtreme Scale, não é possível fazer upgrade ou modificar uma instalação anterior. É necessário desinstalar a versão anterior antes de instalar a nova. Não é necessário migrar seus arquivos de configuração porque eles são compatíveis com versões anteriores. No entanto, se você tiver alterado qualquer um dos arquivos de script que são enviados com o produto, será necessário re replicar essas mudanças para os arquivos de script atualizados.

### Antes de Iniciar

Verifique se seus sistemas atendem aos requisitos mínimos para as versões do produto que você pretende migrar e instalar. Consulte “Requisitos de Hardware e Software” na página 48 para obter mais informações.

### Sobre Esta Tarefa

Mesclre quaisquer arquivos de script de produto modificados com os novos arquivos de script do produto no diretório `/bin` para manter as alterações.

**Dica:** Se você não modificou os arquivos de script que estão instalados com o produto, não será necessário concluir as seguintes etapas de migração. Em vez disso, é possível fazer upgrade para a Versão 7.1.1 ao desinstalar a versão anterior e instalar a nova versão no mesmo diretório.

### Procedimento

1. Pare todos os processos que usam o eXtreme Scale.
  - Leia sobre como parar servidores independentes para parar todos os processos que estão em execução no ambiente independente do eXtreme Scale.
  - Leia sobre utilitários de linha de comandos para parar todos os processos que estão em execução em seu ambiente do WebSphere Application Server ou WebSphere Application Server Network Deployment.
2. Salve quaisquer scripts modificados a partir do diretório de instalação atual em um diretório temporário.
3. Desinstale o produto.
4. Instale o eXtreme Scale Versão 7.1.1. Consulte “Instalando o WebSphere eXtreme Scale com o Assistente de Instalação” na página 161 para obter informações adicionais.
5. Mesclre suas alterações a partir dos arquivos no diretório temporário para os novos arquivos de script do produto no diretório `/bin`.
6. Inicie todos os processos do eXtreme Scale para começar a usar o produto. Consulte Capítulo 7, “Administrando”, na página 387 para obter mais informações.

---

## Usando o Update Installer para instalar os pacotes de manutenção

Use o IBM Update Installer para atualizar o ambiente do WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client com vários tipos de manutenção, como correções temporárias, fix packs e pacotes de atualização.

### Sobre Esta Tarefa

Use o IBM Update Installer para instalar e aplicar vários tipos de pacotes de manutenção para o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client. Como o Update Installer executa manutenção regular, é necessário usar a versão mais atual da ferramenta.

### Procedimento

1. Pare todos os processos que estão em execução no seu ambiente.
  - Para parar todos os processos que estão em execução no seu ambiente do eXtreme Scale independente, consulte o “Parando Servidores Independentes” na página 397.
  - Para parar todos os processos que estão em execução no seu ambiente do WebSphere Application Server independente, consulte Utilitários da Linha de Comandos .
2. Faça download da versão mais recente do Update Installer. Consulte Correções recomendadas para obter informações adicionais.
3. Instale o Update Installer. Consulte Instalando o Update Installer para WebSphere Software no Centro de Informações do WebSphere Application Server para obter informações adicionais.
4. Faça download no diretório *updi\_root/maintenance* dos pacotes de manutenção que deseja instalar. Consulte o Site de Suporte para obter mais informações.
5. Utilize o Update Installer para instalar a correção temporária, fix pack ou pacote de atualizações. É possível instalar o pacote de manutenção executando a interface gráfica com o usuário (GUI) ou executando o Update Installer no modo silencioso.

Execute o seguinte comando a partir do diretório *updi\_root* para iniciar a GUI:

- `Linux` `UNIX` `update.sh`
- `Windows` `update.bat`

Execute o seguinte comando a partir do diretório *updi\_root* para executar o Update Installer no modo silencioso:

- `Linux` `UNIX` `./update.sh -silent -options responsefile/file_name`
- `Windows` `update.bat -silent -options responsefile\file_name`

Se o processo de instalação falhar, consulte o arquivo de log temporário, que está no diretório *updi\_root/logs/update/tmp*. O Update Installer cria o diretório *install\_root/logs/update/maintenance\_package.install* no qual os arquivos de log de instalação estão localizados.

---

## Ferramenta xsadmin para a Migração de Ferramenta xscmd

Em liberações anteriores, a ferramenta **xsadmin** é um utilitário de linha de comandos de amostra para monitorar o estado do ambiente. A ferramenta **xscmd** foi apresentada como uma ferramenta de linha de comandos administrativa e de monitoramento oficialmente suportada. Se a ferramenta **xsadmin** era usada anteriormente, considere migrar seus comandos para a nova ferramenta **xscmd**.

## Equivalentes de Comandos xsadmin e xscmd

Tabela 9. Argumentos para os Comandos Equivalentes xsadmin e xscmd. Alguns comandos xscmd possuem um formato curto e um formato longo. O comandos de formato curto possuem um traço (-) e os comandos no formato longo possuem dois traços (--). Os dois formatos podem ser usados intermitentemente.

Argumento de Linha de Comandos xsadmin	Comando Equivalente xscmd	Parâmetros do Comando xscmd
-bp	<ul style="list-style-type: none"> <li>-cep hostname:listener_port</li> <li>--catalogEndpoint hostname:listener_port</li> </ul>	n/d
-ch	<ul style="list-style-type: none"> <li>-cephostname:listener_port</li> <li>--catalogEndpoint hostname:listener_port</li> </ul>	n/d
-clear	-c clearGrid	-g, -ms, -v, -m, (-cep)
-containers	<ul style="list-style-type: none"> <li>-c listCoreGroups</li> <li>-c listCoreGroupMembers -cg core_group</li> </ul>	-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms
-continuous	<ul style="list-style-type: none"> <li>-cnt</li> <li>--continuous</li> </ul>	n/d
-coregroups	<ul style="list-style-type: none"> <li>-c listCoreGroups</li> <li>-c listCoreGroupMembers -cg core_group</li> </ul>	n/d
-dismissLink <catalog_service_domain>	-c dismissLink	<ul style="list-style-type: none"> <li>-fd &lt;foreignCatalogServiceDomain&gt;</li> <li>--foreignCatalogServiceDomain &lt;foreignCatalogServiceDomain&gt;</li> </ul>
-dmgr	n/a - esse argumento é determinada automaticamente com o xscmd	n/d
-empties	arg específico para um novo comando	n/d
-establishLink <foreign_domain_name> <host1:port1,host2:port2...>	-c establishLink	<ul style="list-style-type: none"> <li>-fd &lt;foreignCatalogServiceDomain&gt; -fe &lt;host1:port1,host2:port2...&gt;</li> <li>--foreignCatalogServiceDomain &lt;foreignCatalogServiceDomain&gt; -foreignEndpoints &lt;host1:port1,host2:port2...&gt;</li> </ul>
-fc	<ul style="list-style-type: none"> <li>-ct</li> <li>--container</li> </ul>	n/d
-fh	<ul style="list-style-type: none"> <li>-hf</li> <li>--hostFilter</li> </ul>	n/d
-fm	<ul style="list-style-type: none"> <li>-m</li> <li>--map</li> </ul>	n/d
-fnp	<ul style="list-style-type: none"> <li>-snp</li> <li>--serversWithNoPrimaries</li> </ul>	n/d
-fp	<ul style="list-style-type: none"> <li>-p</li> <li>--partitionId</li> </ul>	n/d
-fs	<ul style="list-style-type: none"> <li>-s</li> <li>--server</li> </ul>	n/d
-fst	<ul style="list-style-type: none"> <li>-st &lt;shard_type&gt;</li> <li>--shardType &lt;shard_type&gt;</li> </ul> <p>Shard values: P=primary A=asyncReplica S=syncReplica</p>	n/d
-fz	<ul style="list-style-type: none"> <li>-z</li> <li>--zone</li> </ul>	n/d
-force	arg específico para um novo comando	
-g	<ul style="list-style-type: none"> <li>-g</li> <li>--objectGrid</li> </ul>	n/d
-getstatsspec	-c getStatsSpec	n/d
-getTraceSpec	-c getTraceSpec	n/d

Tabela 9. Argumentos para os Comandos Equivalentes `xsadmin` e `xscmd` (continuação). Alguns comandos `xscmd` possuem um formato curto e um formato longo. O comandos de formato curto possuem um traço (-) e os comandos no formato longo possuem dois traços (--). Os dois formatos podem ser usados intermitentemente.

Argumento de Linha de Comandos <code>xsadmin</code>	Comando Equivalente <code>xscmd</code>	Parâmetros do Comando <code>xscmd</code>
<code>-h</code>	É possível executar a ajuda com ou sem um nome de comando específico: <ul style="list-style-type: none"> <li>• <code>-h</code></li> <li>• <code>--help</code></li> <li>• <code>-h &lt;command_name&gt;</code></li> <li>• <code>--help &lt;command_name&gt;</code></li> </ul>	n/d
<code>-hosts</code>	<code>-c listHosts</code>	<code>-g, -ms, -st, -c, -s, -hf, -z</code>
<code>-jmxUrl</code>	<ul style="list-style-type: none"> <li>• <code>-cep hostname:listener_port</code></li> <li>• <code>--catalogEndpoint hostname:listener_port</code></li> </ul>	n/d
<code>-l</code>	<code>-c listObjectGridNames</code>	n/d
<code>-m</code>	<ul style="list-style-type: none"> <li>• <code>-ms</code></li> <li>• <code>--mapSet</code></li> </ul>	n/d
<code>-mapsizes</code>	<code>-c showMapSizes</code>	<code>-g, -ms, -cnt, -i, [-ct, -z, -s, -hf, sht [P,A,S], -p]</code>
<code>-mbeanservers</code>	<code>-c listAllJMXAddresses</code>	n/d
<code>-overridequorum</code>	<code>-c overrideQuorum</code>	n/d
<code>-password</code>	<ul style="list-style-type: none"> <li>• <code>-pwd</code></li> <li>• <code>--password</code></li> </ul>	n/d
<code>-p</code>	<ul style="list-style-type: none"> <li>• <code>-cep hostname:listener_port</code></li> <li>• <code>--catalogEndpoint hostname:listener_port</code></li> </ul>	n/d
<code>-placementStatus</code>	<code>-c placementServiceStatus</code>	<code>-g, -ms</code>
<code>-primaries</code>	<code>-c showPlacement -sf P</code>	<code>-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms</code>
<code>-profile</code>	Para salvar as configurações de segurança atuais como um perfil de segurança: <ul style="list-style-type: none"> <li>• <code>-ssp profile_name</code></li> <li>• <code>--saveSecProfile profile_name</code></li> </ul> Para usar um perfil de segurança especificado: <ul style="list-style-type: none"> <li>• <code>-sp profile_name</code></li> <li>• <code>--securityProfile profile_name</code></li> </ul>	
<code>-quorumstatus</code>	<code>-c showQuorumStatus</code>	n/d
<code>-releaseShard &lt;container_server_name&gt; &lt;objectgrid_name&gt; &lt;map_set_name&gt; &lt;partition_name&gt;</code>	<code>-c releaseShard</code>	<code>-c, -g, -ms, -p</code>
<code>-reserved</code>	<ul style="list-style-type: none"> <li>• <code>-sf [R,U]</code></li> <li>• <code>--shardFilter [R,U]</code></li> </ul> R=reserved, U=unassigned	n/d
<code>-reserveShard &lt;container_server_name&gt; &lt;objectgrid_name&gt; &lt;map_set_name&gt; &lt;partition_name&gt;</code>	<code>-c reserveShard</code>	<code>-c, -g, -ms, -p</code>
<code>-resumeBalancing &lt;objectgrid_name&gt; &lt;map_set_name&gt;</code>	<code>-c resumeBalancing</code>	<code>-g, -ms</code>
<code>-revisions</code>	<code>-c revisions</code>	<code>-s, -p, -g, -m</code>
<code>-routetable</code>	<code>-c routetable</code>	<code>-z, -hf, -p, -g, -ms</code>
<code>-settracespec &lt;trace_string&gt;</code>	<code>-c setTraceSpec</code>	<code>-spec &lt;trace_string&gt;</code>
<code>-swapShardWithPrimary &lt;container_server_name&gt; &lt;objectgrid_name&gt; &lt;map_set_name&gt; &lt;partition_name&gt;</code>	<code>-c swapShardWithPrimary</code>	<code>-c -g, -ms, -p</code>
<code>-setstatsspec &lt;stats_spec&gt;</code>	<code>-c setStatsSpec</code>	<code>-spec &lt;stats_spec&gt;</code>
<code>-suspendBalancing &lt;objectgrid_name&gt; &lt;map_set_name&gt;</code>	<code>-c suspendBalancing</code>	<code>-g, -ms</code>



Tabela 9. Argumentos para os Comandos Equivalentes `xsadmin` e `xscmd` (continuação). Alguns comandos `xscmd` possuem um formato curto e um formato longo. O comandos de formato curto possuem um traço (-) e os comandos no formato longo possuem dois traços (--). Os dois formatos podem ser usados intermitentemente.

Argumento de Linha de Comandos <code>xsadmin</code>	Comando Equivalente <code>xscmd</code>	Parâmetros do Comando <code>xscmd</code>
<code>-ssl</code>	<ul style="list-style-type: none"> <li><code>-ssl</code></li> <li><code>--enableSSL</code></li> </ul>	n/d
<code>-teardown</code>	<code>-c teardown</code>	<code>-f, , -st, -snp, -c, -s, -p, -hf, -z, -g, -ms, -m</code>
<code>-triggerPlacement</code>	<code>-c triggerPlacement</code>	<code>-g, -ms</code>
<code>-trustPass</code>	<ul style="list-style-type: none"> <li><code>-tsp</code></li> <li><code>--trustStorePassword</code></li> </ul>	n/d
<code>-trustPath</code>	<ul style="list-style-type: none"> <li><code>-ts</code></li> <li><code>--trustStore</code></li> </ul>	n/d
<code>-trustType</code>	<ul style="list-style-type: none"> <li><code>-tst</code></li> <li><code>--trustStoreType</code></li> </ul>	n/d
<code>-unassigned</code>	<code>-c showPlacement -sf U</code>	<code>-e, -I, , -st, -snp, -ct, -s, -p, -hf, -z, -g, -m, -ms</code>
<code>-username</code>	<ul style="list-style-type: none"> <li><code>-user</code></li> <li><code>--username</code></li> </ul>	n/d
<code>-v</code>	<ul style="list-style-type: none"> <li><code>-v</code></li> <li><code>--verbose</code></li> </ul>	n/d
<code>-xml</code>	<code>-c showPlacement</code>	n/d

## Propriedades descontinuadas e APIs

A seguinte lista de propriedades e APIs foram descontinuadas no release da Versão 7.1.1. Use a ação de migração recomendada para determinar como atualizar sua configuração.

### 7.1.1+ Itens Descontinuados na Versão 7.1.1

Tabela 10. Propriedades descontinuadas e APIs

Reprovação	Ação de migração recomendada
<p><b>Classe <code>com.ibm.websphere.objectgrid.plugins.builtins.TranPropListener</code></b> Esta classe era usada para propagar com êxito processos de confirmação de transação do ObjectGrid para outros servidores de aplicativos do WebSphere que hospedam a mesma instância do ObjectGrid, com base no nome do ObjectGrid.</p>	<p><b>7.1.1+</b> A interface <code>TranPropListener</code> interface foi substituída pela interface <code>JMSObjectGridEventListener</code>, que é uma implementação baseada em JMS da interface <code>ObjectGridEventListener</code>. Ele suporta uma quase invalidação de cache e replicação ponto a ponto do lado do cliente.</p>
<p><b>Classe <code>com.ibm.websphere.objectgrid.plugins.OptimisticCallback</code></b> Esta classe era usada para fornecer operações de comparação otimistas para os valores de um mapa.</p>	<p><b>7.1.1+</b> O plug-in <code>OptimisticCallback</code> foi substituído pela interface <code>ValueDataSerializer.Versionable</code>, que pode ser implementada quando o plug-in <code>DataSerializer</code> for usado com o modo de cópia <code>COPY_TO_BYTES</code> ou quando usar a anotação <code>@Version</code> com a API do <code>EntityManager</code>. Consulte a Documentação da API para obter mais informações.</p>
<p><b>Plug-ins <code>com.ibm.websphere.objectgrid.plugins.NoVersioningOptimisticCallback</code></b> Este plug-in era usado para bloqueio otimista sem fazer verificação de versão. Com este manipulador <code>OptimisticCallback</code> integrado, o carregador manipulava a verificação de versão, porém o bloqueio otimista era usado para assegurar que os dados confirmados fossem sempre retornados em uma leitura.</p>	<p><b>7.1.1+</b> A interface <code>NoVersioningOptimisticCallback</code> estende a interface <code>OptimisticCallback</code>. Portanto, use a estratégia de bloqueio pessimista com um isolamento de transação padrão de <code>READ_COMMITTED</code> ou inferior. Consulte Ajustando o Desempenho de Bloqueio para obter mais informações.</p>

Tabela 10. Propriedades descontinuadas e APIs (continuação)

Reprovação	Ação de migração recomendada
<b>Classe <code>com.ibm.websphere.objectgrid.plugins.ObjectTransformer</code></b> Este plug-in era usado para serializar, desserializar e copiar objetos no cache.	<b>7.1.1+</b> A interface <code>ObjectTransformer</code> foi substituída pelos plug-ins <code>DataSerializer</code> , que podem ser usados para armazenar dados arbitrários eficientemente no WebSphere eXtreme Scale para que as APIs do produto existentes possam ser interagir de modo eficiente com seus dados.
<b>Método <code>com.ibm.websphere.objectgrid.BackingMap.setMapEventListeners</code></b> Este método era usado para definir a lista de objetos <code>MapEventListener</code> .	<b>7.1.1+</b> Use os métodos <code>addMapEventListener(EventListener)</code> ou <code>removeMapEventListener(EventListener)</code> para incluir ou remover os listeners de eventos a partir de um mapa de apoio.
<b>Método <code>com.ibm.websphere.objectgrid.ObjectGrid.setEventListeners</code></b> Este método era usado para sobrescrever a lista atual de objetos <code>ObjectGridEventListener</code> e substituí-la pela lista fornecida de objetos <code>ObjectGridEventListeners</code> .	<b>7.1.1+</b> Use os métodos <code>addEventListener(EventListener)</code> ou <code>removeEventListener(EventListener)</code> para incluir ou remover os listeners de eventos ou os listeners de ciclo de vida da grade de dados.

### 7.1.1+ Recursos Estabilizados na Versão 7.1.1

Se um recurso for listado como estabilizado, a IBM atualmente não planeja descontinuar ou remover este recurso em uma liberação subsequente do produto; mas o investimento futuro será focado na função alternativa. Os usuários não precisam alterar quaisquer aplicativos e scripts existentes que usam uma função estabilizada; mas eles devem considerar o uso da alternativa estratégica para novos aplicativos.

Tabela 11. Propriedades descontinuadas e APIs

Recurso Estabilizado	Ação de migração recomendada
<b>xsadmin</b> O utilitário <code>xsadmin</code> é fornecido como uma amostra de como é possível criar utilitários customizados para sua implementação.	<b>7.1.1+</b> Use o utilitário <code>xscmd</code> para concluir tarefas administrativas no ambiente tais como: estabelecer links de replicação multimaster, substituir quorum e parar grupos de servidores com o comando <code>teardown</code> .

### Itens Descontinuados na Versão 7.1

Tabela 12. Propriedades descontinuadas e APIs

Reprovação	Ação de migração recomendada
<b>Propriedade da célula e do servidor <code>catalog.services.cluster</code>:</b> Esta propriedade customizada foi usada para definir um grupo de servidores de catálogo na configuração do WebSphere Application Server.	Essa propriedade customizada foi descontinuada a partir do release da Versão 7.1.  Crie um domínio do serviço de catálogo no console administrativo do WebSphere Application Server, que cria a mesma configuração que usando a propriedade customizada. Consulte o para obter informações adicionais.
<b>MBean e interface <code>CoreGroupServicesMBean</code></b>	Esse MBean foi descontinuado a partir do release da Versão 7.1.  Use <code>CatalogServiceManagementMBean</code> no lugar.
<b>Operação MBean <code>ServerMBean.updateTraceSpec()</code></b>	Esta operação foi descontinuada a partir do release da Versão 7.1.  Use o atributo <code>TraceSpec</code> no <code>DynamicServerMBean</code> no lugar.
<b>MBean <code>CoreGroupServicesMBean</code></b>	Esse MBean foi descontinuado a partir do release da Versão 7.1.  Use o MBean <code>CatalogServiceManagementMbean</code> no lugar.
<b>Exceção <code>ServiceUnavailableException</code></b>	Essa exceção foi descontinuada a partir do release da Versão 7.1.  Use a exceção <code>TargetNotAvailableException</code> no lugar.

Tabela 12. Propriedades descontinuadas e APIs (continuação)

Reprovação	Ação de migração recomendada
<b>Recurso de Particionamento (WPF):</b> O recurso de particionamento é um conjunto de APIs de programação que permitem que os aplicativos Java EE suportem armazenamento em cluster assimétrico.	Os recursos do WPF podem ser executados alternativamente no WebSphere eXtreme Scale.
<b>StreamQuery:</b> Uma consulta contínua dos dados transferidos armazenados nos mapas do ObjectGrid.	Nenhum
<b>Configuração de Grade Estática:</b> Uma topologia estática baseada em cluster que usa o arquivo XML de implementação de cluster.	Substituído com a topologia de implementação dinâmica melhorada para gerenciar grades de dados grandes.
<b>Propriedades do Sistema Reprovadas:</b> As propriedades do sistema para especificar os arquivos de propriedades do servidor e do cliente foram reprovadas.	<p>Ainda é possível usar esses argumentos, mas altere as propriedades do sistema para os novos valores.</p> <ul style="list-style-type: none"> <li>-<b>Dcom.ibm.websphere.objectgrid.CatalogServerProperties</b> A propriedade foi reprovada no WebSphere eXtreme Scale Versão 7.0. Use a propriedade -<b>Dobjectgrid.server.props.</b></li> <li>-<b>Dcom.ibm.websphere.objectgrid.ClientProperties</b> A propriedade foi reprovada no WebSphere eXtreme Scale Versão 7.0. Use a propriedade -<b>Dobjectgrid.client.props.</b></li> <li>-<b>Dobjectgrid.security.server.prop</b> A propriedade foi reprovada no WebSphere eXtreme Scale Versão 6.1.0.3. Use a propriedade -<b>Dobjectgrid.server.prop.</b></li> <li>-<b>serverSecurityFile</b> Este argumento foi reprovado no WebSphere eXtreme Scale Versão 6.1.0.3. Essa opção é transmitida no script start0gServer. Use a propriedade -<b>serverProps.</b></li> </ul>

---

## Capítulo 6. Configurando



É possível configurar WebSphere eXtreme Scale para execução em um ambiente independente ou pode configurar eXtreme Scale para execução em um ambiente com WebSphere Application Server ou WebSphere Application Server Network Deployment. Para uma implementação do WebSphere eXtreme Scale para selecionar as mudanças de configuração no lado do servidor da grade de dados, você deve reiniciar os processos para que essas mudanças entrem em vigor em vez de serem aplicadas dinamicamente. Porém, no lado do cliente, embora não seja possível alterar as definições de configuração para uma instância de cliente existente, você poderá criar um novo cliente com as configurações necessárias ao usar um arquivo XML ou fazer isso programaticamente. Ao criar um cliente, será possível substituir as configurações padrão fornecidas com a configuração do servidor atual.

---

### Métodos de Configuração

É possível configurar a maioria dos aspectos do produto com os arquivos XML e com os arquivos de propriedades. Também é possível usar métodos programáticos, incluindo interfaces de programação do aplicativo e do sistema, plug-ins, e beans gerenciados.

#### Sobre Esta Tarefa

Use os arquivos a seguir para criar uma configuração básica:

##### Arquivo de Propriedades do Servidor

Use o arquivo de propriedades do servidor para definir as configurações para os servidores de catálogos e de contêiner, tais como rastreamento, criação de log, segurança, portas e assim por diante. É possível transmitir um arquivo de propriedades do servidor para o script **startOgServer**, colocar o arquivo em seu caminho de classe ou definir o arquivo com as propriedades do sistema. Para

##### Arquivo de Propriedades do Cliente

Use o arquivo de propriedades do cliente para configurar propriedades em seus clientes, incluindo portas e configurações de segurança. É possível especificar o arquivo de propriedades do cliente para uso com uma propriedade de sistema, colocando o arquivo no caminho de classe ou usando o método `ClientClusterContext.getClientProperties`.

##### Arquivo XML descritor do ObjectGrid

O arquivo XML do descritor do ObjectGrid descreve a configuração da grade de dados e do mapa. Especifique o arquivo para uso com o script **startOgServer** para configurações independentes ou inclua o arquivo no módulo aplicativo para as configurações do WebSphere Application Server.

##### Arquivo Descritor XML de Política de Implementação

O arquivo XML da política de implementação controla o shard e o posicionamento dos dados nos diversos servidores de contêiner na configuração. Especifique o arquivo para uso com o script **startOgServer** para configurações independentes ou inclua o arquivo no módulo aplicativo para as configurações do WebSphere Application Server.

---

## Configurando as Grades de Dados

Use um arquivo XML do descritor de ObjectGrid para configurar as grades de dados, mapas de apoio, plug-ins, etc. Para configurar o WebSphere eXtreme Scale, utilize um arquivo XML descritor do ObjectGrid e a API do ObjectGrid. Para uma topologia distribuída, é necessário um arquivo descritor XML do ObjectGrid e um arquivo XML de política de implementação.

## Configurando Implementações Locais

Uma configuração do eXtreme Scale em memória local pode ser criada usando um arquivo descritor XML ou APIs do ObjectGrid.

### Sobre Esta Tarefa

Para criar uma implementação local, crie um arquivo descritor XML do ObjectGrid e, em seguida, passe o arquivo para os métodos `createObjectGrid` na interface `ObjectGridManager`.

Como alternativa, a implementação inteira também pode ser criada programaticamente com a interface `ObjectGridManager`.

### Procedimento

1. Crie um arquivo descritor XML do ObjectGrid.

O arquivo `companyGrid.xml` a seguir é um exemplo de um XML de descritor de ObjectGrid. As primeiras linhas do arquivo incluem o cabeçalho obrigatório de cada arquivo XML do ObjectGrid. O arquivo define uma instância de ObjectGrid denominada "CompanyGrid" e vários BackingMaps denominados "Customer," "Item," "OrderLine" e "Order."

#### arquivo `companyGrid.xml`

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">

  <objectGrids>
    <objectGrid name="CompanyGrid">
      <backingMap name="Customer"/>
      <backingMap name="Item" />
      <backingMap name="OrderLine" />
      <backingMap name="Order"/>
    </objectGrid>
  </objectGrids>

</objectGridConfig>
```

2. Transmite o arquivo XML para um dos métodos na `createObjectGrid` na interface `ObjectGridManager`.

A amostra de código a seguir valida o arquivo `companyGrid.xml` com relação ao esquema XML e cria a instância de ObjectGrid denominada "CompanyGrid." A instância do ObjectGrid recém-criada não é armazenada em cache.

```
ObjectGridManager objectGridManager = ObjectGridManagerFactory.getObjectGridManager();
ObjectGrid companyGrid = objectGridManager.createObjectGrid("CompanyGrid", new URL
("file:etc/test/companyGrid.xml"), true, false);
```

### O que Fazer Depois

Consulte o Criando Instâncias do ObjectGrid com a Interface `ObjectGridManager` para obter informações adicionais sobre como definir todos os mapas

programaticamente com os métodos `createObjectGrid` na interface `ObjectGridManager`.

## Ativando Evictors com a Configuração XML

Em vez de usar a interface `BackingMap` para programaticamente configurar os atributos de `BackingMap` a serem usados pelos Evictor TTL, é possível usar um arquivo XML para configurar cada instância de `BackingMap`. O código a seguir demonstra como configurar tais atributos para três mapas `BackingMap` diferentes:

### Antes de Iniciar

Antes de começar, decida sobre o tipo de evictor que será usado:

- **Evictor TTL baseado em tempo padrão:** O evictor padrão usa uma política de despejo tempo de vida (TTL) para cada instância `BackingMap`.
- **Um mecanismo de evictor conectável:** Evictors conectáveis geralmente usam uma política de despejo com base no número de entradas em vez de no tempo.

A maior parte das configurações de evictor deve ser feita antes de inicializar o `ObjectGrid`.

### Procedimento

- Para configurar o evictor TTL padrão, inclua o atributo `ttlEvictorType` no arquivo descritor XML do `ObjectGrid`.

O exemplo a seguir mostra que a instância `map1` de `BackingMap` usa um tipo de evictor `NONE TTL`. A instância `BackingMap map2` usa um tipo de evictor `TTL LAST_ACCESS_TIME` ou `LAST_UPDATE_TIME`. Especifique apenas uma ou outras dessas configurações. A instância `BackingMap map2` tem um valor de tempo de vida de 1800 segundos, ou 30 minutos. A instância `map3` de `BackingMap` é definida para utilizar um tipo de evictor `CREATION_TIME TTL` e tem um valor de `time-to-live` de 1200 segundos ou 20 minutos.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
<objectGrids>
  <objectGrid name="grid1">
    <backingMap name="map1" ttlEvictorType="NONE" />
    <backingMap name="map2" ttlEvictorType="LAST_ACCESS_TIME|LAST_UPDATE_TIME"
      timeToLive="1800" />
    <backingMap name="map3" ttlEvictorType="CREATION_TIME" timeToLive="1200" />
  </objectGrid>
</objectGrids>
```

Figura 27. Ativar o Evictor `TimeToLive` com XML

- Para configurar um evictor conectável, use o seguinte exemplo.

```

<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
<objectGrids>
  <objectGrid name="grid">
    <backingMap name="map1" ttlEvictorType="NONE" pluginCollectionRef="LRU" />
    <backingMap name="map2" ttlEvictorType="NONE" pluginCollectionRef="LFU" />
  </objectGrid>
</objectGrids>
<backingMapPluginCollections>
  <backingMapPluginCollection id="LRU">
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      <property name="maxSize" type="int" value="1000" description="set max size for each LRU queue" />
      <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
      <property name="numberOfLRUQueues" type="int" value="53" description="set number of LRU queues" />
    </bean>
  </backingMapPluginCollection>
  <backingMapPluginCollection id="LFU">
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LFUEvictor">
      <property name="maxSize" type="int" value="2000" description="set max size for each LFU heap" />
      <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
      <property name="numberOfHeaps" type="int" value="211" description="set number of LFU heaps" />
    </bean>
  </backingMapPluginCollection>
</backingMapPluginCollections>
</objectGridConfig>

```

Figura 28. Conectando um Evictor Usando XML

## Configurando uma Estratégia de Bloqueio

É possível definir uma estratégia otimista, pessimista ou de nenhum bloqueio em cada BackingMap na configuração do WebSphere eXtreme Scale.

### Sobre Esta Tarefa

Cada instância do BackingMap pode ser configurada para usar uma das seguintes estratégias de bloqueio:

1. Modo de Bloqueio Otimista
2. Modo de Bloqueio Pessimista
3. Nenhum

A estratégia de bloqueio padrão é OPTIMISTIC. Utilize o bloqueio optimistic quando os dados são alterados de maneira infrequente. Os bloqueios são mantidos apenas por uma curta duração enquanto os dados estão sendo lidos do cache e copiados para a transação. Quando o cache da transação é sincronizado com o cache principal, quaisquer objetos de cache que foram atualizados são verificados junto à versão original. Se a verificação falhar, então, ocorre o rollback da transação e o resultado é uma exceção OptimisticCollisionException.

A estratégia de bloqueio PESSIMISTIC adquire bloqueios para entradas de cache e deve ser utilizada quando os dados são alterados frequentemente. Sempre que uma entrada de cache é lida, um bloqueio é adquirido e mantido condicionalmente até que a transação seja concluída. A duração de alguns bloqueios pode ser ajustada utilizando níveis de isolamento de transação para a sessão.

Se o bloqueio não for necessário porque os dados nunca são atualizados ou são atualizados apenas durante períodos tranquilos, você pode desativar o bloqueio utilizando a estratégia de bloqueio NONE. Esta estratégia é muito rápida porque um gerenciador de bloqueio não é necessário. A estratégia de bloqueio NONE é ideal para tabelas de consulta ou mapas somente leitura.



Para obter mais informações sobre as estratégias de bloqueio, consulte Estratégias de Bloqueio informações sobre as estratégias de bloqueio *Visão Geral do Produto*.

## Procedimento

- **Configure uma estratégia de bloqueio otimista**

- Usando o método `setLockStrategy` programaticamente:

```
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
...
ObjectGrid og =
    ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("optimisticMap");
bm.setLockStrategy( LockStrategy.OPTIMISTIC );
```

- Usando o atributo `lockStrategy` no Arquivo XML descritor do ObjectGrid:

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="test">
            <backingMap name="optimisticMap"
                lockStrategy="OPTIMISTIC"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

- **Configure uma estratégia de bloqueio pessimista**

- Usando o método `setLockStrategy` programaticamente:

**especificar programaticamente  
estratégia pessimista**

```
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
...
ObjectGrid og =
    ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("pessimisticMap");
bm.setLockStrategy( LockStrategy.PESSIMISTIC );
```

- Usando o atributo `lockStrategy` no Arquivo XML descritor do ObjectGrid:

**Especificar estratégia pessimista usando XML**

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="test">
            <backingMap name="pessimisticMap"
                lockStrategy="PESSIMISTIC"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>
```

- **Configure uma estratégia de ausência de bloqueio**

- Usando o método `setLockStrategy` programaticamente:

```
import com.ibm.websphere.objectgrid.BackingMap;
import com.ibm.websphere.objectgrid.LockStrategy;
import com.ibm.websphere.objectgrid.ObjectGrid;
import com.ibm.websphere.objectgrid.ObjectGridManagerFactory;
```

```

...
ObjectGrid og =
    ObjectGridManagerFactory.getObjectGridManager().createObjectGrid("test");
BackingMap bm = og.defineMap("noLockingMap");
bm.setLockStrategy( LockStrategy.NONE );

```

- Usando o atributo lockStrategy no Arquivo XML descritor do ObjectGrid:

```

<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">

    <objectGrids>
        <objectGrid name="test">
            <backingMap name="noLockingMap"
                lockStrategy="NONE"/>
        </objectGrid>
    </objectGrids>
</objectGridConfig>

```

## O que Fazer Depois

Para evitar uma exceção `java.lang.IllegalStateException`, é necessário chamar o método `setLockStrategy` antes de chamar os métodos `initialize` ou `getSession` na instância `ObjectGrid`.

## Configurando Replicação Ponto a Ponto com o JMS

Mecanismo de replicação ponto a ponto baseado no Java Message Service (JMS) é usado em ambiente local e distribuído do WebSphere eXtreme Scale. O JMS é um processo de replicação núcleo a núcleo e permite que as atualizações de dados fluam entre os `ObjectGrids` e os `ObjectGrids` distribuídos. Por exemplo, com esse mecanismo, é possível mover as atualizações de dados de uma grade de dados distribuída do eXtreme Scale em um local do eXtreme Scale ou de uma grade para outra grade em um domínio de sistema diferente.

### Antes de Iniciar

O mecanismo de replicação ponto a ponto baseado em JMS é baseado no `ObjectGridEventListener` baseado em JMS integrado, `com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener`. Para obter informações detalhadas sobre a ativação do mecanismo de replicação ponto a ponto, consulte “Listener de Eventos da JMS” na página 228.

Consulte “Ativando o Mecanismo de Invalidação do Cliente” na página 291 para obter informações adicionais.

A seguir há um exemplo de uma configuração XML para ativar um mecanismo de replicação ponto a ponto em uma configuração do eXtreme Scale:

```

configuração da replicação ponto a ponto - exemplo de XML
<bean id="ObjectGridEventListener"
    className="com.ibm.websphere.objectgrid.plugins.JMSObjectGridEventListener">
    <property name="replicationRole" type="java.lang.String" value="DUAL_ROLES" description="" />
    <property name="replicationStrategy" type="java.lang.String" value="PUSH" description="" />
    <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String"
        value="defaultTCF" description="" />
    <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
    <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
    <property name="jms_userid" type="java.lang.String" value="" description="" />
    <property name="jms_password" type="java.lang.String" value="" description="" />
    <property name="jndi_properties" type="java.lang.String"
        value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;

```

```
java.naming.provider.url=tcp://localhost:61616;connectionFactoryNames=defaultTCF;
topic.defaultTopic=defaultTopic"
description="jndi properties" />
</bean>
```

## Distribuindo Mudanças entre JVMs Peers

Os objetos LogSequence e LogElement distribuem mudanças entre JVMs peer e comunicam as mudanças que ocorreram em uma transação do eXtreme Scale com um plug-in do ObjectGridEventListener.

Para obter informações adicionais sobre como a JMS (Java Message Service) pode ser usado para distribuir mudanças transacionais, consulte Distribuindo Transações.

Um pré-requisito é que a instância do ObjectGrid deve ser armazenada em cache pelo ObjectGridManager. Consulte Métodos createObjectGrid para obter mais informações. O valor booleano cacheInstance deve ser configurado como true.

Não é necessário que você implemente esse mecanismo. Há um mecanismo de replicação ponto a ponto integrado disponível para utilizar esta função. Consulte “Configurando Replicação Ponto a Ponto com o JMS” na página 224.

Os objetos fornecem um meio para que um aplicativo publique facilmente alterações que ocorreram em um ObjectGrid utilizando um transporte de mensagens para ObjectGrids peer nas Java Virtual Machines remotas e, então, aplica estas alterações em tais JVM. A classe LogSequenceTransformer é importante para ativar este suporte. Este artigo examina como escrever um listener usando um sistema de mensagens JVM (Java Message Service) para propagação das mensagens. Para este fim, o eXtreme Scale suporta a transmissão de LogSequences que resultam de uma consolidação transacional de um eXtreme Scale através dos membros do cluster do WebSphere Application Server com um plug-in fornecido pela IBM. Esta função não é ativada por padrão, mas pode ser configurada para ser operacional. Porém, quando o consumidor ou produtor não for um WebSphere Application Server, o uso de um sistema de mensagens JMS externo pode ser necessário.

## Implementando o Mecanismo

A classe LogSequenceTransformer e as APIs ObjectGridEventListener, LogSequence e LogElement permitem que qualquer publicação-e-assinatura confiável seja utilizada para distribuir as alterações e filtrar os mapas que você deseja distribuir. Os snippets neste tópico mostram como utilizar estas APIs com o JMS para criar um ObjectGrid ponto a ponto compartilhado por aplicativos que estão hospedados em um conjunto diverso de plataforma compartilhando um transporte de mensagens comum.

### Inicialize o plug-in

O ObjectGrid chama o método de inicialização do plug-in, parte do contrato da interface ObjectGridEventListener, quando o ObjectGrid inicia. O método de inicialização deve obter seus recursos JMS, incluindo conexões, sessões, e publicadores, e inicia o encadeamento que é o listener JMS.

Os exemplos a seguir mostram o método de inicialização:

```
exemplo do método de inicializar
public void initialize(Session session) {
    mySession = session;
```

```

myGrid = session.getObjectGrid();
try {
    if (mode == null) {
        throw new ObjectGridRuntimeException("No mode specified");
    }
    if (userid != null) {
        connection = topicConnectionFactory.createTopicConnection
(userid, password);
    } else
        connection = topicConnectionFactory.createTopicConnection();

    // need to start the connection to receive messages.
    connection.start();

    // the jms session is not transactional (false).
    jmsSession = connection.createTopicSession(false,
javax.jms.Session.AUTO_ACKNOWLEDGE);
    if(topic == null)
        if (topicName == null) {
            throw new ObjectGridRuntimeException("Topic not specified");
        } else {
            topic = jmsSession.createTopic(topicName);
        }
    publisher = jmsSession.createPublisher(topic);
    // iniciar o encadeamento do listener.
    listenerRunning = true;
    listenerThread = new Thread(this);
    listenerThread.start();
} catch (Throwable e) {
    throw new ObjectGridRuntimeException("Cannot initialize", e);
}
}

```

O código para iniciar o encadeamento usa um encadeamento Java 2 Platform, Standard Edition (Java SE). Se você estiver executando um WebSphere Application Server Versão 6.x ou um servidor enterprise WebSphere Application Server Version 5.x, use a interface de programação de aplicativos (API) do bean assíncrono para iniciar este encadeamento de daemon. Também é possível utilizar as APIs comuns. A seguir está um snippet de substituição de exemplo que mostra a mesma ação utilizando um gerenciador de trabalho:

```

// iniciar o encadeamento do listener.
listenerRunning = true;
workManager.startWork(this, true);

```

O plug-in também deve implementar a interface `Work` em vez da interface `Runnable`. Também é necessário incluir um método de liberação para configurar a variável `listenerRunning` como `false`. O plug-in deve ser fornecido com uma instância do gerenciador de trabalho em seu construtor ou por injeção, se estiver utilizando um contêiner IoC (Inversion of Control).

### Transmita as alterações

A seguir, está um método `transactionEnd` de amostra para a publicação das alterações locais que são feitas em um `ObjectGrid`. Esta amostra utiliza o JMS, embora seja possível utilizar qualquer transporte de mensagens que seja capaz de emitir mensagens de publicação-e-assinatura confiáveis.

```

Exemplo do método transactionEnd
// This method is synchronized to make sure the
// messages are published in the order the transaction
// were committed. If we started publishing the messages
// in parallel then the receivers could corrupt the Map
// as deletes may arrive before inserts etc.
public synchronized void transactionEnd(String txid, boolean isWriteThroughEnabled,

```

```

boolean committed,
    Collection changes) {
    try {
        // must be write through and committed.
        if (isWriteThroughEnabled && committed) {
            // write the sequences to a byte []
            ByteArrayOutputStream bos = new ByteArrayOutputStream();
            ObjectOutputStream oos = new ObjectOutputStream(bos);
            if (publishMaps.isEmpty()) {
                // serialize the whole collection
                LogSequenceTransformer.serialize(changes, oos, this, mode);
            } else {
                // filter LogSequences based on publishMaps contents
                Collection publishChanges = new ArrayList();
                Iterator iter = changes.iterator();
                while (iter.hasNext()) {
                    LogSequence ls = (LogSequence) iter.next();
                    if (publishMaps.contains(ls.getMapName())) {
                        publishChanges.add(ls);
                    }
                }
                LogSequenceTransformer.serialize(publishChanges, oos, this, mode);
            }
            // make an object message for the changes
            oos.flush();
            ObjectMessage om = jmsSession.createObjectMessage(bos.toByteArray());
            // set properties
            om.setStringProperty(PROP_TX, txid);
            om.setStringProperty(PROP_GRIDNAME, myGrid.getName());
            // transmit it.
            publisher.publish(om);
        }
    } catch (Throwable e) {
        throw new ObjectGridRuntimeException("Cannot push changes", e);
    }
}

```

Este método utiliza diversas variáveis de instância:

- Variável `jmsSession`: Uma sessão JMS utilizada para publicar mensagens. É criada quando o plug-in inicializa.
- Variável `mode`: O modo de distribuição.
- Variável `publishMaps`: Um conjunto que contém o nome de cada mapa com alterações para publicar. Se a variável está vazia, então, todos os mapas são publicados.
- Variável `publisher`: Um objeto `TopicPublisher` que é criado durante o método de inicialização de plug-in

### Receber e aplicar mensagens de atualização

A seguir está o método de execução. Este método é executado em um loop até que o aplicativo pare o loop. Cada iteração do loop tenta receber uma mensagem JMS e aplicá-la ao `ObjectGrid`.

#### Exemplo do método de execução de mensagem JMS

```

private synchronized boolean isListenerRunning() {
    return listenerRunning;
}

public void run () {
    try {
        System.out.println("Listener starting");
        // get a jms session for receiving the messages.
        // Non transactional.
        TopicSession myTopicSession;
        myTopicSession = connection.createTopicSession(false, javax.jms.
            Session.AUTO_ACKNOWLEDGE);

        // get a subscriber for the topic, true indicates don't receive
        // messages transmitted using publishers
    }
}

```

```

        // on this connection. Otherwise, we'd receive our own updates.
        TopicSubscriber subscriber = myTopicSession.createSubscriber(topic,
null, true);
        System.out.println("Listener started");
        while (isListenerRunning()) {
            ObjectMessage om = (ObjectMessage)subscriber.receive(2000);
            if (om != null) {
                // Use Session that was passed in on the initialize...
                // very important to use no write through here
                mySession.beginNoWriteThrough();
                byte[] raw = (byte[])om.getObject();
                ByteArrayInputStream bis = new ByteArrayInputStream(raw);
                ObjectInputStream ois = new ObjectInputStream(bis);
                // inflate the LogSequences
                Collection collection = LogSequenceTransformer.inflate(ois,
myGrid);
                Iterator iter = collection.iterator();
                while (iter.hasNext()) {
                    // process each Maps changes according to the mode when
                    // the LogSequence was serialized
                    LogSequence seq = (LogSequence)iter.next();
                    mySession.processLogSequence(seq);
                }
                mySession.commit();
            } // if there was a message
        } // loop while
        // stop the connection
        connection.close();
    } catch (IOException e) {
        System.out.println("IO Exception: " + e);
    } catch (JMSEException e) {
        System.out.println("JMS Exception: " + e);
    } catch (ObjectGridException e) {
        System.out.println("ObjectGrid exception: " + e);
        System.out.println("Caused by: " + e.getCause());
    } catch (Throwable e) {
        System.out.println("Exception : " + e);
    }
    System.out.println("Listener stopped");
}
}

```

## Listener de Eventos da JMS

O `JMSObjectGridEventListener` foi projetado para suportar invalidação de cache próximo do lado do cliente e um mecanismo de replicação ponto a ponto. Ele é uma implementação Java Message Service (JMS) da interface do `ObjectGridEventListener`.

O mecanismo de invalidação do cliente pode ser usado em um ambiente distribuído `doeXtreme Scale` para garantir que os dados de cache próximos do cliente sejam sincronizados com os servidores ou outros cliente. Sem essa função, o cache perto do cliente pode deixar os dados obsoletos. Entretanto, mesmo com essa invalidação de cliente baseada em JMS, é necessário levar em consideração a janela de sincronização para atualizar um cache perto do cliente devido ao atraso do tempo de execução ao publicar as atualizações.

O mecanismo de replicação ponto a ponto pode ser usado em ambientes local e distribuído do `eXtreme Scale`. Ele é um processo de replicação de núcleo para núcleo do `ObjectGrid` e permite o fluxo de atualizações de dados entre os `ObjectGrids` locais e distribuídos. Por exemplo, com esse mecanismo, é possível mover as atualizações de dados de uma grade distribuída para um `ObjectGrid` local ou de qualquer grade para outra grade em um domínio de sistema diferente.

O `JMSObjectGridEventListener` requer que o usuário configure informações de JMS e Java Naming and Directory Interface (JNDI) para obter os recursos do JMS necessários. Além disso, as propriedades relacionadas à replicação devem ser configuradas corretamente. Em um ambiente JEE, o JNDI deve estar disponível em ambos os contêineres da web e do Enterprise JavaBean (EJB). Nesse caso, a propriedade JNDI é opcional, a menos que você deseje obter os recursos JMS externos.

Esse listener de evento possui propriedades que podem ser configuradas com o XML ou com abordagens programáticas, que podem ser usadas apenas para invalidação do cliente, apenas replicação ponto a ponto, ou ambos. A maioria das propriedades é opcional para customizar o comportamento para obter a funcionalidade necessária.

Para obter mais informações, consulte a API do `JMSObjectGridEventListener`.

### **Estendendo o Plug-in do `JMSObjectGridEventListener`**

O plug-in do `JMSObjectGridEventListener` permite que as instâncias equivalentes do `ObjectGrid` recebam atualizações quando os dados na grade forem alterados ou despejados. Também permite que os clientes sejam notificados quando as entradas forem atualizadas ou despejadas de uma grade do eXtreme Scale. Esse tópico descreve como estender o plug-in do `JMSObjectGridEventListener` para permitir que os aplicativos sejam notificados quando uma mensagem JMS for recebida. Isso é mais útil ao usar a configuração do `CLIENT_SERVER_MODEL` para invalidação do cliente.

Durante a execução na função de receptor, o método `JMSObjectGridEventListener.onMessage` substituído é automaticamente chamado pelo tempo de execução do eXtreme Scale quando a instância do `JMSObjectGridEventListener` recebe atualizações de mensagem JMS da grade. Essas mensagens agrupam uma coleta de Objetos `LogSequence`. Os objetos `LogSequence` são transmitidos para o método `onMessage` e o aplicativo usa o `LogSequence` para identificar quais entradas de cache foram inseridas, excluídas, atualizadas ou invalidadas.

Para usar o ponto de extensão `onMessage`, os aplicativos executam as seguintes etapas.

1. Criar uma nova classe, estendendo a classe `JMSObjectGridEventListener`, substituindo o método `onMessage`.
2. Configurar o `JMSObjectGridEventListener` estendido da mesma forma que o `ObjectGridEventListener` para `ObjectGrid`.

A classe `JMSObjectGridEventListener` estendida é uma classe-filha da classe `JMSObjectGridEventListener` e só pode substituir dois métodos: o `initialize` (opcional) e o `onMessage`. Se uma classe-filha da classe `JMSObjectGridEventListener` precisar utilizar quaisquer artefatos de `ObjectGrid` como `ObjectGrid` ou `Session` no método `onMessage`, ela pegará esses artefatos no método de inicialização e os armazenará em cache como variáveis de instância. Além disso, no método `onMessage`, artefatos de `ObjectGrid` armazenados em cache podem ser utilizados para processar uma coleta de `LogSequences` transmitida.



**Nota:** O método de inicialização substituído deve invocar o método `super.initialize` para inicializar o `JMSObjectGridEventListener` pai corretamente.

A seguir há uma amostra de uma classe `JMSObjectGridEventListener` estendida.

```
package com.ibm.websphere.samples.objectgrid.jms.price;

import java.util.*;
import com.ibm.websphere.objectgrid.*;
import com.ibm.websphere.objectgrid.plugins.LogElement;
import com.ibm.websphere.objectgrid.plugins.LogSequence;
import com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener;

public class ExtendedJMSObjectGridEventListener extends JMSObjectGridEventListener{
    protected static boolean debug = true;

    /**
     * Essa é a grade associada a esse listener.
     */
    ObjectGrid grid;

    /**
     * Essa é a sessão associada a esse listener.
     */
    Session session;

    String objectGridType;

    public List receivedLogSequenceList = new ArrayList();

    /* (non-Javadoc)
     * @see com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener
     * #initialize(com.ibm.websphere.objectgrid.Session)
     */
    public void initialize(Session session) {
        // Note: if need to use any ObjectGrid artifact, this class need to get ObjectGrid
        // from the passed Session instance and get ObjectMap from session instance
        // for any transactional ObjectGrid map operation.

        super.initialize(session); // must invoke super's initialize method.
        this.session = session; // cache the session instance, in case need to
        // use it to perform map operation.
        this.grid = session.getObjectGrid(); // get ObjectGrid, in case need
        // to get ObjectGrid information.

        if (grid.getObjectGridType() == ObjectGrid.CLIENT)
            objectGridType = "CLIENT";
        else if (grid.getObjectGridType() == ObjectGrid.SERVER)
            objectGridType = "Server";

        if (debug)
            System.out.println("ExtendedJMSObjectGridEventListener[" +
                objectGridType + "].initialize() : grid = " + this.grid);
    }

    /* (non-Javadoc)
     * @see com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener
     * #onMessage(java.util.Collection)
     */
    protected void onMessage(Collection logSequences) {
        System.out.println("ExtendedJMSObjectGridEventListener[" +
            objectGridType + "].onMessage(): ");

        Iterator iter = logSequences.iterator();

        while (iter.hasNext()) {
            LogSequence seq = (LogSequence)iter.next();

            StringBuffer buffer = new StringBuffer();
            String mapName = seq.getMapName();
            int size = seq.size();
            buffer.append("\nLogSequence[mapName=" + mapName + ", size=" + size + ",
                objectGridType=" + objectGridType
                + "]: ");

            Iterator logElementIter = seq.getAllChanges();
            for (int i = seq.size() - 1; i >= 0; --i) {
                LogElement le = (LogElement) logElementIter.next();
            }
        }
    }
}
```

```

        buffer.append(1e.getType() + " -> key=" + 1e.getCacheEntry().getKey() + ", ");
    }
    buffer.append("\n");

    receivedLogSequenceList.add(buffer.toString());

    if (debug) {
        System.out.println("ExtendedJMSObjectGridEventListener["
+ objectGridType + "].onMessage(): " + buffer.toString());
    }
}

}

public String dumpReceivedLogSequenceList() {
    String result = "";
    int size = receivedLogSequenceList.size();
    result = result + "\nExtendedJMSObjectGridEventListener[" + objectGridType
+ "]: receivedLogSequenceList size = " + size + "\n";
    for (int i = 0; i < size; i++) {
        result = result + receivedLogSequenceList.get(i) + "\n";
    }
    return result;
}

public String toString() {
    return "ExtendedJMSObjectGridEventListener["
+ objectGridType + " - " + this.grid + "];"
}
}

```

## Configuração

A classe `JMSObjectGridEventListener` estendida deve ser configurada da mesma forma para o mecanismo de invalidação de cliente e de replicação ponto a ponto. A seguir há um exemplo de configuração XML.

```

<objectGrid name="PRICEGRID">
  <bean id="ObjectGridEventListener"
    className="com.ibm.websphere.samples.objectgrid.jms.
      price.ExtendedJMSObjectGridEventListener">
    <property name="invalidationModel" type="java.lang.String"
      value="CLIENT_SERVER_MODEL" description="" />
    <property name="invalidationStrategy" type="java.lang.String"
      value="INVALIDATE" description="" />
    <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String"
      value="jms/TCF" description="" />
    <property name="jms_topicJndiName" type="java.lang.String"
      value="GRID.PRICEGRID" description="" />
    <property name="jms_topicName" type="java.lang.String"
      value="GRID.PRICEGRID" description="" />
    <property name="jms_userid" type="java.lang.String" value="" description="" />
    <property name="jms_password" type="java.lang.String" value="" description="" />
  </bean>
  <backingMap name="PRICE" pluginCollectionRef="PRICE"></backingMap>
</objectGrid>

```

**Nota:** `ClassName` do bean `ObjectGridEventListener` é configurado com a classe `JMSObjectGridEventListener` estendida com as mesmas propriedades que o `JMSObjectGridEventListener` genérico.

---

## Configurando Políticas de Implementação

Use o arquivo XML do descritor de política de implementação e o arquivo XML do descritor de `objectgrid` para gerenciar uma topologia distribuída. A política de implementação é codificada como um arquivo XML que é fornecido para o servidor de contêiner. A política de implementação fornece informações sobre mapas, conjuntos de mapas, partições, réplicas e assim por diante. Ela também controla os comportamentos do posicionamento de shards.

## Configurando Implementações Distribuídas

Utilize o arquivo XML do descritor de política de implementação e o arquivo XML do descritor de ObjectGrid para gerenciar sua topologia.

A política de implementação é codificada como um arquivo XML que é fornecido para o servidor de contêiner do eXtreme Scale. O arquivo XML especifica as seguintes informações:

- Os mapas que pertencem a cada conjunto de mapas
- O número de partições
- O número de réplicas síncronas e assíncronas

A política de implementação também controla os seguintes comportamentos de colocação.

- O número mínimo de servidores de contêiner ativos antes de ocorrer o posicionamento
- A substituição automática de shards perdidos
- A disposição de cada shard de uma única partição em uma máquina diferente.

As informações de terminal não estão pré-configuradas no ambiente dinâmico. Não há nomes de servidor ou informações de topologia física localizadas na política de implementação. Todos os shards em uma grade de dados são colocados automaticamente nos servidores de contêiner pelo serviço de catálogo. O serviço de catálogo usa as restrições que são definidas pela política de implementação para gerenciar automaticamente a colocação de shard. Essa colocação automática de shard facilita a configuração de grades de dados grandes. Também é possível incluir servidores no seu ambiente, conforme necessário.

**Restrição:** Em um ambiente do WebSphere Application Server, um tamanho de grupo principal de mais de 50 membros não é suportado.

Um arquivo XML de política de implementação é passado para o servidor de contêiner durante a inicialização. Uma política de implementação deve ser usada junto com o arquivo XML de ObjectGrid. A política de implementação não é necessária para iniciar um servidor de contêiner, mas é recomendada. A política de implementação deve ser compatível com o arquivo XML do ObjectGrid que é utilizado com ela. Para cada elemento `objectgridDeployment` na política de implementação, você deve incluir um elemento `objectGrid` correspondente no arquivo XML de ObjectGrid. Os mapas no `objectgridDeployment` devem ser consistentes com os elementos `backingMap` localizados no XML do ObjectGrid. Cada `backingMap` deve ser referenciado dentro somente de um elemento `mapSet`.

No exemplo a seguir, o arquivo `companyGridDpReplication.xml` é destinado a formar um par com o arquivo `companyGrid.xml` correspondente.

```
companyGridDpReplication.xml
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

<objectgridDeployment objectgridName="CompanyGrid">
  <mapSet name="mapSet1" numberOfPartitions="11"
minSyncReplicas="1" maxSyncReplicas="1"
maxAsyncReplicas="0" numInitialContainers="4">
    <map ref="Customer" />
    <map ref="Item" />
    <map ref="OrderLine" />
    <map ref="Order" />
  </mapSet>
</objectgridDeployment>
</deploymentPolicy>
```

```

</mapSet>
</objectgridDeployment>

</deploymentPolicy>
companyGrid.xml
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">

<objectGrids>
<objectGrid name="CompanyGrid">
<backingMap name="Customer"/>
<backingMap name="Item" />
<backingMap name="OrderLine" />
<backingMap name="Order"/>
</objectGrid>
</objectGrids>

</objectGridConfig>

```

O arquivo `companyGridDpReplication.xml` tem um elemento `mapSet` que é dividido em 11 partições. Cada partição deve ter exatamente uma réplica síncrona. O número de réplicas síncronas é especificado pelos atributos `minSyncReplicas` e `maxSyncReplicas`. Como o atributo `minSyncReplicas` é configurado como 1, cada partição no elemento `mapSet` deve ter pelo menos uma réplica síncrona disponível para processar transações de gravação. Como o atributo `maxSyncReplicas` é configurado para 1, cada partição não pode exceder uma réplica síncrona. As partições nesse elemento `mapSet` não têm réplicas assíncronas.

O atributo `numInitialContainers` instrui o serviço de catálogo a adiar o posicionamento até quatro servidores de contêiner estarem disponíveis para suportar esta instância do ObjectGrid. O atributo `numInitialContainers` é ignorado após o número especificado de servidores de contêiner ter sido atingido.

**7.1.1+** Também é possível usar a propriedade `placementDeferralInterval` e o comando `xscmd -c suspendBalancing` para atrasar o posicionamento de shards nos servidores de contêiner.

Embora o arquivo `companyGridDpReplication.xml` seja um exemplo básico, uma política de implementação pode oferecer um controle total sobre seu ambiente.

## Topologia Distribuída

Caches consistentes distribuídos oferecem melhores desempenho, disponibilidade e escalabilidade que podem ser configurados.

O WebSphere eXtreme Scale equilibra automaticamente os servidores. É possível incluir servidores adicionais sem reiniciar o WebSphere eXtreme Scale. Incluir servidores adicionais sem precisar reiniciar o eXtreme Scale permite executar implementações simples e implementações que atingem terabytes em que milhares de servidores são necessários.

Essa topologia de implementação é flexível. Utilizando o serviço de catálogo, é possível incluir e remover servidores para utilizar melhor os recursos sem remover o cache inteiro. É possível os comandos `startOgServer` e `stopOgServer` para iniciar e parar os servidores de contêiner. Esses dois comandos requerem especificar a opção `-catalogServiceEndpoints`. Todos os clientes de topologia distribuída se comunicam com o serviço de catálogo através do Internet Interoperability Object Protocol (IIOP). Todos os clientes utilizam a interface ObjectGrid para comunicar-se com servidores.

O recurso de configuração dinâmica do WebSphere eXtreme Scale facilita incluir recursos no sistema. Os contêineres hospedam os dados e o serviço de catálogo permite que os clientes se comuniquem com a grade de servidores de contêineres. O serviço de catálogo encaminha solicitações, aloca espaço nos servidores de contêiner do host e gerencia o funcionamento e a disponibilidade do sistema geral. Os clientes se conectam a um serviço de catálogo, recuperam uma descrição da topologia do servidor de contêiner e, então, comunicam-se diretamente com cada servidor conforme necessário. Quando a topologia do servidor muda devido à inclusão de novos servidores, ou devido à falha de outros, o serviço de catálogo roteia automaticamente os pedidos do cliente para o servidor apropriado que hospeda os dados.

Um serviço de catálogo normalmente existe na sua própria grade do Java Virtual Machines. Um único servidor de catálogos pode gerenciar vários servidores. É possível iniciar um servidor de contêiner em uma JVM sozinho ou carregar o servidor de contêiner em uma JVM arbitrária com outros servidores de contêiner para diferentes servidores. Pode existir um cliente em alguma JVM e comunicar-se com um ou mais servidores. Um cliente também pode existir na mesma JVM que um servidor de contêiner.

Também é possível criar uma política de implementação programaticamente quando você está integrando um servidor de contêiner em um processo ou aplicativo Java existente. Para obter mais informações, consulte a documentação da API DeploymentPolicy.

## Controlando o Posicionamento de Shard com Zonas

Use a política de posicionamento para definir zonas. As zonas fornecem um controle sobre o posicionamento de shard no WebSphere eXtreme Scale. As zonas são um conceito lógico definido pelo usuário usado para representar agrupamentos lógicos de servidores físicos.

### Configurando Zonas para a Colocação de Réplica

O suporte à zona permite configurações sofisticadas para posicionamento de réplica nos datacenters. Com esse recurso, as grades de milhares de partições podem ser facilmente gerenciadas utilizando algumas regras de localização opcionais. Um datacenter pode estar em diferentes andares de um prédio, diferentes prédios ou até mesmo cidades diferentes ou outras distinções, conforme configurado com as regras de zonas.

### Flexibilidade de Zonas

É possível colocar shards em zonas. Esta função permite que você tenha mais controle sobre como o eXtreme Scale coloca shards em uma grade. A Java Virtual Machines que hospeda um servidor eXtreme Scale pode ser identificada com um identificador de zona. O arquivo de implementação agora pode incluir uma ou mais regras de zonas e estas regras de zonas estão associadas com um tipo shard. A seção a seguir fornece uma visão geral de uso da zona. Para obter mais informações, consulte as orientações sobre como controlar o posicionamento do shard com as zonas no *Guia de Administração*.

Controle de zonas de disposição de como o eXtreme Scale designa primários e réplicas para configurar topologias avançadas.

Um Java Virtual Machine pode ter vários contêineres, mas apenas 1 servidor. Um contêiner pode hospedar vários shards a partir de um único ObjectGrid.

Este recurso é útil para certificar-se de que as réplicas e os primários sejam colocados em diferentes locais ou zonas para obter uma melhor alta disponibilidade. Normalmente, o eXtreme Scale não coloca um shard primário e de réplica na Java Virtual Machines com o mesmo endereço IP. Esta regra simples normalmente evita que dois servidores eXtreme Scale sejam colocados no mesmo computador físico. Entretanto, pode ser necessário um mecanismo mais flexível. Por exemplo, você pode estar utilizando dois chassis blade e desejar que os shards primários sejam *divididos* em ambos os chassis e que a réplica para cada shard primário seja colocada no outro chassi a partir do primário.

*Dividido* significa que os shards primários são colocados em cada zona e a réplica para cada primário está localizada na zona oposta. O primário 0, por exemplo, poderia estar na zoneA, a réplica síncrona 0 poderia estar na zoneB. O primário 1 poderia estar na zoneB, e a réplica síncrona 1 poderia estar na zoneA.

O nome do chassi poderia ser o nome da zona neste caso. Alternativamente, você pode chamar as zonas de acordo com os andares em um prédio e utilizar as zonas para ter certeza de que os primários e as réplicas para os mesmos dados estejam em andares diferentes. Prédios e datacenter também são possíveis. Foram feitos testes em datacenters utilizando zonas como um mecanismo para garantir que os dados sejam adequadamente replicados entre os datacenters. Se você estiver utilizando o HTTP Session Manager para eXtreme Scale, também é possível utilizar zonas. Com este recurso, é possível implementar um aplicativo da Web único em três datacenters e garantir que as sessões HTTP para usuários sejam replicadas nos datacenters para que as sessões possam ser recuperadas, mesmo se um datacenter inteiro falhar.

O WebSphere eXtreme Scale está ciente da necessidade de gerenciar uma grade grande sobre vários datacenters. Isso pode assegurar que os primários e os backups da mesma partição estejam localizados em datacenters diferentes, se isto for necessário. Ele pode colocar todos os primários no datacenter 1 e todas as réplicas no datacenter 2, ou pode fazer um round robin com os primários e as réplicas em ambos os datacenters. As regras são flexíveis de modo que muitos cenários são possíveis. O eXtreme Scale também pode gerenciar milhares de servidores, que juntos com a disposição completamente automática e o reconhecimento do datacenter tornam tais grades grandes financeiramente suportáveis de um ponto de vista administrativo. Os administradores podem especificar o que desejam fazer de maneira simples e eficiente.

Como um administrador, utilize zonas de disposição para controlar onde os shards primários e de réplica são colocados, o que permite a configuração de topologias avançadas de alto desempenho e altamente disponíveis. É possível definir uma zona para qualquer agrupamento lógico de processos do eXtreme Scale, conforme observado acima: Estas zonas podem corresponder a locais de estação de trabalho físicas, tais como um datacenter, um andar de um datacenter ou um chassi blade. É possível dividir dados em zonas, o que fornece aumento de disponibilidade ou você dividir os primários e as réplicas em zonas separadas quando um hot standby é necessário.

### **Associando um Servidor eXtreme Scale com um Zona que não está utilizando o WebSphere Extended Deployment**

Se o eXtreme Scale for utilizado com Java Standard Edition ou um servidor de aplicativos que não seja baseado no WebSphere Extended Deployment versão 6.1, um JVM que é um contêiner de shard pode ser associado a uma zona se estiver utilizando as seguintes técnicas.

## Aplicativos utilizando o script startOgServer

O script startOgServer é utilizado para iniciar um aplicativo eXtreme Scale quando ele não está sendo integrado em um servidor existente. O parâmetro **-zone** é utilizado para especificar a zona a utilizar para todos os contêineres no servidores.

## Especificando a zona ao iniciar um contêiner utilizando APIs

### Associando Nós do WebSphere Extended Deployment com Zonas

Se estiver utilizando eXtreme Scale com aplicativos WebSphere Extended Deployment Java EE, será possível alavancar os grupos de nós do WebSphere Extended Deployment para colocar os servidores em zonas específicas.

No eXtreme Scale, um JVM só pode ser membro de uma única zona. Entretanto, o WebSphere permite que um nó faça parte de vários grupos de nós. É possível utilizar essa funcionalidade de zonas do eXtreme Scale se tiver certeza de que cada um de seus nós esteja em apenas um grupo de nós da zona.

Utilize a seguinte sintaxe para nomear o grupo de nós para declará-lo em uma zona: `ReplicationZone<UniqueSuffix>`. Servidores em execução em um nó que faz parte desse grupo de nós são incluídos nas zonas especificadas pelo nome do grupo de nós. A seguir está uma descrição de uma topologia de exemplo.

Primeiro, você configura 4 nós: node1, node2, node3 e node4, cada um com 2 servidores. Em seguida, você cria um grupo de nós denominado ReplicationZoneA e um grupo de nós denominado ReplicationZoneB. Em seguida, você inclui node1 e node2 em ReplicationZoneA e inclui node3 e node4 em ReplicationZoneB.

Quando os servidores no node1 e node2 são iniciados, eles se tornam parte de ReplicationZoneA; da mesma forma, os servidores em node3 e node4 se tornarão parte de ReplicationZoneB.

Uma JVM membro da grade verifica a associação da zona apenas na inicialização. A inclusão de um novo grupo de nós ou a mudança da associação têm impacto apenas nas JVMs recém iniciadas ou reiniciadas.

## Regras de Zonas

Uma partição do eXtreme Scale possui um shard primários e zero ou mais shards de réplica. Para este exemplo, considere a seguinte convenção de nomenclatura para estes shards. P é o shard primário, S é uma réplica assíncrona e A é uma réplica assíncrona. Uma regra de zona possui três componentes:

- Um nome de regra
- Uma lista de zonas
- Um sinalizador inclusivo ou exclusivo

O nome da zona para um contêiner pode ser especificado conforme descrito na documentação do “API do Servidor Integrado” na página 405. Uma regra de zona especifica o possíveis conjuntos de zonas nos quais um shard pode ser colocado. O sinalizador inclusivo indica que após um shard ser colocado na lista, então, todos os outros shards também são colocados em tal zona. Uma configuração exclusiva indica que cada shard para uma partição é colocada em uma zona diferente na lista de zonas. Por exemplo, utilizar uma configuração exclusiva significa que se houver três shards (um primário e duas réplicas síncronas), então, a lista de zonas deve ter três zonas.



Cada shard pode ser associado com uma regra de zona. Uma regra de zona pode ser compartilhada entre dois shards. Quando uma regra é compartilhada, então o sinalizador inclusivo ou exclusivo é estendido pelos shards de todos os tipos compartilhando uma regra única.

## Exemplos

A seguir, está um conjunto de exemplos mostrando diversos cenários e a configuração de implementação para implementar os cenários.

### Dividindo primários e réplicas entre zonas

É possível ter três chassis blade e desejar primários distribuídos para os três, com uma réplica síncrona única colocada em um chassi diferente do primário. Defina cada chassis como uma zona com nomes de chassi ALPHA, BETA e GAMMA. A seguir, está um exemplo de XML de implementação:

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
  http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
    <mapSet name="ms1" numberOfPartitions="37" minSyncReplicas="1"
      maxSyncReplicas="1" maxAsyncReplicas="0">
      <map ref="book" />
    <zoneMetadata>
      <shardMapping shard="P" zoneRuleRef="stripeZone"/>
      <shardMapping shard="S" zoneRuleRef="stripeZone"/>
      <zoneRule name="stripeZone" exclusivePlacement="true" >
        <zone name="ALPHA" />
        <zone name="BETA" />
        <zone name="GAMMA" />
      </zoneRule>
    </zoneMetadata>
  </mapSet>
</objectgridDeployment>
</deploymentPolicy>
```

Este XML de implementação contém uma grade chamada library com um Mapa único denominado book. Ele utiliza quatro partições com uma única réplica síncrona. A cláusula de metadados de zona mostra a definição de uma única regra de zona e a associação das regras de zona com os shards. Os shards primários e síncronos estão ambos associados com a regra de zona "stripeZone". A regra de zona possui as três zonas e utiliza a disposição exclusiva. Esta regra significa que se o primário para a partição 0 for colocado no ALPHA, então, a réplica para a partição 0 será colocada no BETA ou GAMMA. De maneira semelhante, os primários para outras partições são colocados em outras zonas e as réplicas serão colocadas.

### Réplica assíncrona em uma zona diferente da primária e réplica síncrona

Neste exemplo, existem dois prédios com uma alta conexão de latência entre eles. Você não deseja alta disponibilidade de perda de dados para todos os cenários. Entretanto, o impacto de desempenho da replicação síncrona entre os prédios o leva a um dilema. Você deseja um primário com réplica síncrona em um prédio e uma réplica assíncrona em outro prédio. Normalmente, as falhas são paralisações da JVM ou falhas de computador ao invés de problemas de larga escala. Com esta topologia, é possível sobreviver a falhas normais sem nenhuma perda de dados. A perda de um prédio é rara o suficiente que alguma perda de dados é aceitável neste caso. É possível criar duas zonas, uma para cada prédio. A seguir, está o arquivo XML de implementação:

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
```

```

<mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="1"
maxSyncReplicas="1" maxAsyncReplicas="1">
  <map ref="book" />
  <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="primarySync"/>
    <shardMapping shard="S" zoneRuleRef="primarySync"/>
    <shardMapping shard="A" zoneRuleRef="aysnc"/>
    <zoneRule name="primarySync" exclusivePlacement="false" >
      <zone name="B1dA" />
      <zone name="B1dB" />
    </zoneRule>
    <zoneRule name="aysnc" exclusivePlacement="true">
      <zone name="B1dA" />
      <zone name="B1dB" />
    </zoneRule>
  </zoneMetadata>
</mapSet>
</objectgridDeployment>
</deploymentPolicy>

```

O primário e a réplica síncrona compartilham uma regra de zona primarySync com uma configuração de sinalizador exclusivo de false. Portanto, após o primário ou síncrono ser colocado em uma zona, então o outro também é colocado na mesma zona. A réplica assíncrona utiliza uma segunda regra de zona com as mesmas zonas que a regra de zona primarySync, mas ela utiliza o atributo **exclusivePlacement** configurado como true. Este atributo indica que um shard não pode ser colocado em uma zona com outro shard a partir da mesma partição. Como resultado, a réplica assíncrona não é colocada na mesma zona que o primário e as réplicas síncronas.

### Colocar todos os primários em uma zona e todas réplicas em outra zona

Aqui, todos os primários estão em uma zona específica e todas as réplicas em uma zona diferente. Teremos um primário e uma réplica assíncrona única. Todas as réplicas estarão na zona A e os primários na B.

```

<?xml version="1.0" encoding="UTF-8"?>

<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=
  "http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

  <objectgridDeployment objectgridName="library">
    <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
maxSyncReplicas="0" maxAsyncReplicas="1">
      <map ref="book" />
      <zoneMetadata>
        <shardMapping shard="P" zoneRuleRef="primaryRule"/>
        <shardMapping shard="A" zoneRuleRef="replicaRule"/>
        <zoneRule name="primaryRule">
          <zone name="A" />
        </zoneRule>
        <zoneRule name="replicaRule">
          <zone name="B" />
        </zoneRule>
      </zoneMetadata>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>

```

Aqui, você pode ver duas regras, uma para os primários (P) e outra para a réplica (A).

### Zonas sobre Wide Area Networks (WAN)

Você pode desejar implementar um eXtreme Scale único em vários prédios ou datacenters com interconexões de rede mais lentas. Conexões de rede mais lentas levam a uma largura da banda menor e mais conexões de latência. A possibilidade de partições de rede também aumenta neste modo devido ao congestionamento de rede e outros fatores. O eXtreme Scale aborda este ambiente severo das seguintes maneiras.

## **Pulsção limitada entre zonas**

As Java Virtual Machines agrupadas em grupos principais realizam pulsção uma com a outra. Quando o serviço de catálogo organiza as Java Virtual Machines nos grupos, tais grupos não se estendem sobre zonas. Um líder neste grupo executa o push das informações de associação no serviço de catálogo. O serviço de catálogo verifica quaisquer falhas relatadas antes de tomar a ação. Ele faz isto ao tentar conectar-se às Java Virtual Machines suspeitas. Se o serviço de catálogo visualiza uma falsa detecção de falha, então, ele não executa nenhuma ação já que a partição do grupo principal irá resolver o problema em um curto período de tempo.

O serviço de catálogo também executar pulsção nos líderes do grupo principal periodicamente em uma taxa mas baixa para tratar o caso de isolamento do grupo principal.

## **Roteamento para Zonas Preferenciais**

Com o roteamento para zonas preferenciais, é possível definir como o WebSphere eXtreme Scale direciona as transações para zonas.

Você tem controle sobre onde os shards em uma grade de dados são colocados. Consulte “Configurando Zonas para a Colocação de Réplica” na página 234 para obter mais informações sobre alguns cenários básicos e como configurar sua política de implementação de acordo.

O roteamento preferencial fornece ao clientes do WebSphere eXtreme Scale a capacidade de especificar uma preferência para uma determinada zona ou conjunto de zonas. Como resultado, as transações do cliente são roteadas para zonas preferenciais antes de tentar rotear para qualquer outra zona.

## **Requisitos para Roteamento de Zona Preferencial**

Antes de tentar o roteamento preferencial por zona, certifique-se de que o aplicativo possa atender aos requisitos de seu cenário.

O posicionamento de partição por contêiner é necessário para usar roteamento preferencial por zona. Esta estratégia de posicionamento é uma boa opção para aplicativos que são dados de sessão de armazenamento no ObjectGrid. A estratégia de posicionamento de partição padrão para o WebSphere eXtreme Scale é partição fixa. As chaves são colocadas em hash no momento de consolidação de transação para determinar qual partição hospeda o par chave-valor do mapa ao usar o posicionamento de partição fixa.

O posicionamento por contêiner designa seus dados a uma partição aleatória quando a transação confirma o tempo por meio do objeto SessionHandle. Você deve poder reconstruir o objeto SessionHandle para recuperar seus dados a partir da grade de dados.

É possível usar zonas para ter mais controle sobre onde os shards primário e os shards de réplica são colocados em seu domínio. Usar diversas zonas em sua implementação é vantajoso quando seus dados estiverem em diversos locais físicos. Separar geograficamente os primários e réplicas é uma forma de assegurar que a perda catastrófica de um datacenter não afete a disponibilidade dos dados.

Quando os dados são espalhados em diversas zonas, é mais provável que os clientes também se espalhem pela topologia. O roteamento de clientes para sua zona ou datacenter local possui o benefício de desempenho óbvio de reduzir a

latência de rede. Roteie os clientes para zonas ou datacenters locais quando possível.

## Configuração de sua Topologia para Roteamento de Zona Preferencial

Considere o seguinte cenário. Você tem dois datacenters: Chicago e Londres. Para minimizar o tempo de resposta dos clientes, você deseja que os clientes leiam e gravem dados em seu datacenter local.

Os shards primários devem ser colocados em cada datacenter para que as transações possam ser gravadas localmente a partir de cada local. Os clientes também deve reconhecer as zonas para rotear para a zona local.

O posicionamento por contêiner localiza os novos shards primários em cada contêiner que é iniciado. As réplicas são posicionadas de acordo com as regras da zona e de posicionamento que são especificadas pela política de implementação. Por padrão, uma réplica é colocada em uma zona diferente de seu shard primário. Considere a política de implementação a seguir para este cenário.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="universe">
    <mapSet name="mapSet1" placementStrategy="PER_CONTAINER"
      numberOfPartitions="3" maxAsyncReplicas="1">
      <map ref="planet" />
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Cada contêiner que inicia com uma política de implementação recebe três novos primários. Cada primário possui uma réplica assíncrona. Inicie cada contêiner com o nome de zona adequado. Use o parâmetro **-zone** se estiver ativando seus contêineres com o script **startOgServer**.

Para um servidor de contêineres Chicago:

- **UNIX** **Linux**  

```
startOgServer.sh s1 -objectGridFile ../xml/universeGrid.xml
-deploymentPolicyFile ../xml/universeDp.xml
-catalogServiceEndPoints MyServer1.company.com:2809
-zone Chicago
```
- **Windows**  

```
startOgServer.bat s1 -objectGridFile ../xml/universeGrid.xml
-deploymentPolicyFile ../xml/universeDp.xml
-catalogServiceEndPoints MyServer1.company.com:2809
-zone Chicago
```

Se seus contêineres estiverem em execução no WebSphere Application Server, você deverá criar um grupo de nós e nomeá-lo com o prefixo **ReplicationZone**. Os servidores que estiverem em execução nos nós nestes grupos de nós são colocados na zona adequada. Por exemplo, os servidores em execução em um nó Chicago podem estar em um grupo de nós denominado **ReplicationZoneChicago**.

Consulte o “Configurando Zonas para a Colocação de Réplica” na página 234 para obter informações adicionais.

Os shards primários para a zona Chicago possuem réplicas na zona London. Os shards primários para a zona London possuem réplicas na zona Chicago.

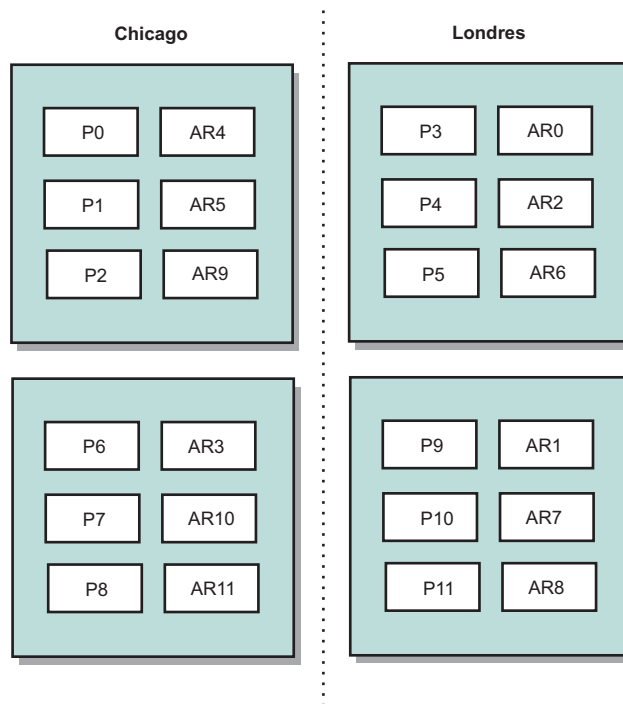


Figura 29. Primários e Réplicas em Zonas

Configure as zonas preferenciais para os clientes. Forneça um arquivo de propriedades do cliente na Java virtual machine (JVM) do cliente. Crie um arquivo denominado `objectGridClient.properties` e assegure-se de que esse arquivo esteja em seu caminho de classe.

Inclua a propriedade **preferZones** no arquivo. Consulte o valor da propriedade para a zona adequada. Os clientes em Chicago devem ter o seguinte valor no arquivo `objectGridClient.properties`:

```
preferZones=Chicago
```

O arquivo de propriedades para clientes London devem conter o seguinte valor:

```
preferZones=Londres
```

Esta propriedade instrui cada cliente para rotear transações à sua zona local se possível. A topologia replica assincronamente dados que são inseridos em um shard primário da zona local na zona estrangeira.

### Usando a Interface `SessionHandle` para Rotear para a Zona Local

A estratégia de posicionamento por contêiner não usa um algoritmo baseado em hash para determinar o local de seus pares de valores de chave na grade de dados. Você deve usar os objetos `SessionHandle` para assegurar que as transações sejam roteadas para o local correto quando usar esta estratégia de posicionamento. Quando uma transação é confirmada, um `SessionHandle` é limitado à sessão se uma ainda não tiver sido configurada. O objeto `SessionHandle` também pode ser limitado ao `Session` ao chamar o método `Session.getSessionHandle` antes de confirmar a transação. O fragmento de código a seguir mostra um `SessionHandle` sendo ligado antes de confirmar a transação.

```

Session ogSession = objectGrid.getSession();

// binding the SessionHandle
SessionHandle sessionHandle = ogSession.getSessionHandle();

ogSession.begin();
ObjectMap map = ogSession.getMap("planet");
map.insert("planet1", "mercury");

// tran is routed to partition specified by SessionHandle
ogSession.commit();

```

Suponha que o código anterior estava executando em um cliente em seu datacenter Chicago. O atributo **preferZones** é configurado para Chicago para este cliente. Como resultado, sua implementação rotearia as transações para uma das partições primárias na zona Chicago: partição 0, 1, 2, 6, 7 ou 8.

O objeto SessionHandle fornece um caminho de volta para a partição que está armazenando os dados confirmados. O objeto SessionHandle deve ser reutilizado ou reconstruído e configurado no Session para voltar para a partição que contém os dados confirmados.

```

ogSession.setSessionHandle(sessionHandle);
ogSession.begin();

// value returned will be "mercury "
String value = map.get("planet1");
ogSession.commit();

```

A transação neste código reutiliza o objeto SessionHandle que foi criado durante a transação insert. A transação get é então roteada para a partição que mantém os dados inseridos. Sem o objeto SessionHandle, a transação não pode recuperar os dados inseridos.

## Como as Falhas de Contêiner e Zonas Afetam o Roteamento Baseado em Zona

Geralmente, um cliente com a propriedade **preferZones** configurada roteia todas as transações para a zona ou zonas especificadas. No entanto, a perda de um contêiner resulta na promoção de um shard réplica para um shard primário. Um cliente que era roteado anteriormente para partições na zona local deve recuperar os dados inseridos anteriormente a partir da zona remota.

Considere o seguinte cenário. Um contêiner na zona Chicago foi perdido. Ele anteriormente continha primários para partições 0, 1 e 2. Os novos shards primários para estas partições são então colocados na zona London porque a zona London hospeda as réplicas para estas partições.

Qualquer cliente Chicago que estiver usando um SessionHandle que aponta para uma das partições com falha agora é roteado novamente para Londres. Os clientes Chicago que usam os novos objetos SessionHandle são roteados para os primários baseados em Chicago.

Da mesma forma, se a zona Chicago inteira for perdida, todas as réplicas na zona London serão promovidas para primárias. Neste cenário, todos os clientes Chicago roteiam suas transações para London.

## Definindo Zonas para Servidores de Contêiner

As zonas são coleções de servidores de contêiner. Um servidor de contêiner pode pertencer a apenas uma zona. Um servidor de contêiner é designado a uma zona quando ele é iniciado.

## Sobre Esta Tarefa

Suas zonas devem ser planejadas antes de iniciar os servidores de contêiner porque os servidores de contêiner definem seus membros da zona durante a inicialização. Se desejar alterar a associação de zona de um servidor de contêiner, você deverá reiniciar o servidor com as novas informações da zona.

## Procedimento

- **Defina zonas para servidores de contêiner independente.**

1. Use o parâmetro **-zone** do script **start0gServer** para especificar a zona de todos os contêineres no servidor iniciado. Para obter informações sobre como iniciar os servidores, consulte “Scriptstart0gServer” na página 393.
2. Também é possível definir uma zona para os nomes quando iniciar servidores de contêiner programaticamente com a API de servidor integrada. Para obter informações adicionais, consulte “Usando a API do Servidor Integrado para Iniciar e Parar Servidores” na página 402.

- **Defina zonas para os servidores de contêiner que estão em execução no WebSphere Application Server.**

Grupos de nós podem ser usados para colocar os servidores de contêiner em zonas específicas. Utilize a seguinte sintaxe para nomear seu grupo de nós para designá-lo a uma zona: `ReplicationZone<identifíer>`. Quando definir zonas na política de implementação, você deve nomear as zonas exatamente conforme os grupos de nós foram nomeados. O nome do grupo de nós e o nome da zona no arquivo do descritor XML de política de implementação devem ser idênticos.

**Importante:** O WebSphere Application Server permite que os nós estejam em diversos grupos de nós. Como os servidores de contêiner podem ser apenas uma zona, certifique-se de que seus nós estejam exatamente em um grupo de nós `ReplicationZone`.

Por exemplo, divida quatro nós em duas zonas, A e B.

1. Configure quatro nós: `node1`, `node2`, `node3` e `node4`, cada um tendo dois servidores.
2. Crie um grupo de nós denominado `ReplicationZoneA` e um grupo de nós denominado `ReplicationZoneB`.
3. Inclua `node1` e `node2` em `ReplicationZoneA` e inclua `node3` e `node4` em `ReplicationZoneB`.
4. Defina `ReplicationZoneA` e `ReplicationZoneB` em seu arquivo do descritor XML de política de implementação. Consulte o “Exemplo: Zonas em um Ambiente WebSphere Application Server” na página 246 para obter um exemplo.
5. Quando os servidores no `node1` e `node2` são iniciados, eles unem `ReplicationZoneA`, ou a zona A, na configuração do WebSphere eXtreme Scale. Os servidores em `node3` e `node4` unem `ReplicationZoneB`, como zona B, na configuração do WebSphere eXtreme Scale.

## Exemplo: Definições de Zona no Arquivo Descritor XML de Política de Implementação

É possível especificar zonas e regras de zona com o arquivo descritor XML de política de implementação.



## Exemplo: Shards Primários e de Réplica em Zonas Diferentes

Este exemplo coloca shards primários em uma zona e shards de réplica em uma zona diferente, com uma réplica assíncrona única. Todos os shards primários iniciam na zona DC1. Os shards de réplica iniciam na zona DC2.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
  ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
<objectgridDeployment objectgridName="library">
<mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
  maxSyncReplicas="0" maxAsyncReplicas="1">
  <map ref="book" />
  <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="primaryRule"/>
    <shardMapping shard="A" zoneRuleRef="replicaRule"/>
    <zoneRule name="primaryRule">
      <zone name="DC1" />
    </zoneRule>
    <zoneRule name="replicaRule">
    </zoneRule>
  </zoneMetadata>
</mapSet>
</objectgridDeployment>
</deploymentPolicy>
```

Uma réplica assíncrona é definida no elemento mapSet ms1. Portanto, existem dois shards para cada partição: um shard primário e um shard de réplica assíncrona. No elemento zoneMetadata, um elemento shardMapping é definido para cada shard: P para o primário e DC1 para a réplica assíncrona. O atributo primaryRule define a zona configurada para os shards primários, que é apenas a zona DC1, e essa regra deve ser usada para o posicionamento de shard primário. As réplicas assíncronas são colocadas na zona DC2.

No entanto, se a zona DC2 for perdida, os shards de réplica se tornarão indisponíveis. A perda ou a falha de um servidor de contêiner na zona DC1 pode resultar em perda de dados, embora uma réplica tenha sido especificada.

Para abordar essa possibilidade, é possível incluir ou uma zona ou incluir uma réplica, conforme descrito nas seguintes seções.

## Exemplo: Incluir uma Zona e Dividindo os Shards

O seguinte código configura uma nova zona:

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
  ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
<objectgridDeployment objectgridName="library">
<mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
  maxSyncReplicas="0" maxAsyncReplicas="1">
  <map ref="book" />
  <zoneMetadata>
    <shardMapping shard="P" zoneRuleRef="stripeRule"/>
    <shardMapping shard="A" zoneRuleRef="stripeRule"/>
    <zoneRule name="stripeRule" exclusivePlacement="true">
      <zone name="A" />
      <zone name="B" />
      <zone name="C" />
    </zoneRule>
```

```

    </zoneMetadata>
  </mapSet>
</objectgridDeployment>
</deploymentPolicy>

```

Um total de três zonas foram definidas neste código: A, B e C. Em vez de separar as regras de zona em primário e de réplica, uma regra de zona compartilhada denominada `stripeRule` é definida. Essa regra inclui todas as zonas, com o atributo `exclusivePlacement` configurado para `true`. A política de posicionamento do `eXtreme Scale` garante que os shards primários e de réplica estejam em zonas separadas. Essa divisão de posicionamento faz com que os shards primários e de réplica se propaguem pelas duas zonas de acordo com essa política. Incluir uma terceira zona C assegura que a perda de qualquer zona não resulte em perda de dados e ainda deixará os shards primários e de réplica para cada partição. Uma falha de zona resulta na perda de um shard primário, de um shard de réplica, ou nenhum deles. Qualquer shard perdido é substituído a partir do shard sobrevivente, colocando-o na outra zona sobrevivente.

### Exemplo: Incluir uma Réplica e Definir Diversos Datacenters

O cenário clássico de dois datacenters possui alta velocidade e redes de baixa latência em cada datacenter, porém possui alta latência entre os datacenters. As réplicas síncronas são usadas em cada datacenter onde a baixa latência minimiza o impacto da replicação nos tempos de resposta. A replicação assíncrona é usada entre os datacenters, portanto, a rede de alta latência não tem impacto no tempo de resposta.

```

<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
  ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
    <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="1"
      maxSyncReplicas="1" maxAsyncReplicas="1">
      <map ref="book" />
      <zoneMetadata>
        <shardMapping shard="P" zoneRuleRef="primarySync"/>
        <shardMapping shard="S" zoneRuleRef="primarySync"/>
        <shardMapping shard="A" zoneRuleRef="async"/>
        <zoneRule name="primarySync" exclusivePlacement="false" >
          <zone name="DC1" />
          <zone name="DC2" />
        </zoneRule>
        <zoneRule name="async" exclusivePlacement="true">
          <zone name="DC1" />
          <zone name="DC2" />
        </zoneRule>
      </zoneMetadata>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>

```

A réplica primária e a síncrona compartilham a regra `primarySync` com uma configuração de atributo `exclusivePlacement` de `false`. O atributo `exclusivePlacement` configurado para `false` cria uma configuração com os shards primário e de réplica síncrona de cada partição colocada na mesma zona. O shard de réplica assíncrono usa uma segunda regra de zona com a maioria das mesmas zonas como a regra de zona `primarySync`. No entanto, a réplica assíncrona usa o atributo `exclusivePlacement` configurado para `true`. O atributo `exclusivePlacement`, quando configurado como `true`, significa que um shard não pode ser colocado em uma zona com outro shard na mesma partição. Como resultado, o shard de réplica

assíncrono não é colocado na mesma zona que o shard de réplica primário ou síncrono. Há três shards por partição nesse mapSet: um primário, um síncrono e um assíncrono, de modo que há três elementos shardMapping, um para cada shard.

Se uma zona for perdida, todas as réplicas assíncronas serão perdidas e nenhuma é gerada novamente porque elas não possuem nenhuma zona separada. Se os shards primários e de réplica forem perdidos, a réplica assíncrona sobrevivente será promovida para o primário e uma nova réplica síncrona é criada na zona. Os primários e réplicas são divididos em cada zona.

Com a disposição exclusiva, cada shard possui sua própria zona: Você deve ter zonas suficientes para todos os shards que desejar colocar em suas próprias zonas. Se uma regra possuir uma zona, somente um shard poderá ser colocado na zona. Com duas zonas, é possível ter até dois shards na zona.

### Exemplo: Zonas em um Ambiente WebSphere Application Server

O seguinte código configura uma nova zona:

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy
  ../deploymentPolicy.xsd" xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="library">
    <mapSet name="ms1" numberOfPartitions="13" minSyncReplicas="0"
      maxSyncReplicas="0" maxAsyncReplicas="1">
      <map ref="book" />
      <zoneMetadata>
        <shardMapping shard="P" zoneRuleRef="stripeRule"/>
        <shardMapping shard="A" zoneRuleRef="stripeRule"/>
        <zoneRule name="stripeRule" exclusivePlacement="true">
          <zone name="ReplicationZoneA" />
          <zone name="ReplicationZoneB" />
          <zone name="ReplicationZoneC" />
        </zoneRule>
      </zoneMetadata>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Para este exemplo, três grupos de nós são definidos no ambiente do WebSphere Application Server: ReplicationZoneA, ReplicationZoneB e ReplicationZoneC. O nome do grupo de nós e o nome da zona no arquivo descritor XML de política de implementação devem ser idênticos e deve conter o texto ReplicationZone<identificador>. Este arquivo define uma configuração semelhante ao exemplo shards divididos, porém mostra a nomenclatura necessária para uma configuração do WebSphere Application Server .

### Visualizando Informações de Zona com o Utilitário xscmd

É possível usar o utilitário **xscmd** de amostra para visualizar informações sobre sua implementação de zona atual, incluindo dados de posicionamento de shard.

#### Antes de Iniciar

- Implemente uma grade de dados distribuída com diversos datacenters. Consulte o “Roteamento para Zonas Preferenciais” na página 239 para obter informações adicionais.

## Sobre Esta Tarefa

É possível determinar informações sobre sua configuração relacionadas às configurações de zona usando o utilitário **xscmd** fornecido com o produto.

## Procedimento

Use o utilitário **xscmd** para determinar informações sobre os shards de dados. Execute o seguinte comando:

```
xscmd -c showPlacement -z zone_name
```

## Exemplo

Também é possível executar um cenário simples usando a introdução de amostra: *wxs\_install\_root/ObjectGrid/gettingstarted*. Consulte o “Tutorial: Introdução ao WebSphere eXtreme Scale” na página 1 para obter informações adicionais.

1. Inicie um servidor de catálogos:  
runcat.bat
2. Determine o número de réplicas, as regras de zona, os contêineres e outras configurações necessárias, tais como o seguinte comando: startOgServer.bat  
serverA0 -objectgridFile xml\objectgrid.xml -deploymentPolicyFile xml\deployment.xml -zone zoneA
3. Também é possível parar os processos do contêiner para simular uma falha na grade de dados: stopOgServer.bat serverA0,serverA1,serverB0  
-catalogServiceEndpoints localhost:2809.

Se o servidor que contém o último shard de uma partição estiver parado, o eXtreme Scale alocará um novo shard primário. É possível verificar a perda de dados:

- O script **runclient** insere e lê um item em sua grade de dados.
  - O comando **xscmd -c showMapSizes** mostra o número de itens na grade de dados.
4. Mostre servidores de contêiner ativos com o comando a seguir:  
xscmd -c showPlacement -z zone\_name

---

## Configurando Servidores de Catálogo e Contêiner

O WebSphere eXtreme Scale possui dois tipos de servidores: servidores de catálogos e servidores de contêiner. Os servidores de catálogo controlam o posicionamento de shards e descobrem e monitoram os servidores de contêiner. Vários servidores de catálogos juntos compõem o serviço de catálogo. Um servidor de contêiner é um Java Virtual Machine (JVM) que armazena os dados do aplicativo na grade de dados.

## Sobre Esta Tarefa

Os servidores de catálogos e de contêiner podem iniciar nos processos do WebSphere Application Server, como processos do Java SE independentes ou ao integrar os servidores nos aplicativos Java SE. O modo com que os servidores de catálogos e de contêiner são configurados depende de sua topologia.

### Servidores de catálogos

- **Servidores de catálogos independentes:**

Configure servidores de catálogos independentes com um arquivo de propriedades do servidor. Controle o ciclo de vida de um servidor de catálogos com os scripts **start0gServer** e **stop0gServer** ou usando a API do servidor integrado.

- **Servidores de catálogos que iniciam no WebSphere Application Server:**  
Configure os servidores de catálogos que são executados no WebSphere Application Server com o console administrativo, tarefas administrativas e com o arquivo de propriedades do servidor do WebSphere Application Server. O ciclo de vida do servidor é controlado pelo ciclo de vida do processo dentro do WebSphere Application Server. Quando os processos iniciam e param no WebSphere Application Server, os servidores de catálogos que estiverem em execução nesses processos também iniciam ou param.

#### Servidores de Contêineres

- **Servidores de contêiner independentes:**  
Configure os servidores de contêiner independente com um arquivo de propriedades do servidor e um arquivo XML de política de implementação. Controle o ciclo de vida de um servidor de contêiner com os scripts **start0gServer** e **stop0gServer** ou usando a API do servidor integrado.
- **Servidores de contêiner que iniciam no WebSphere Application Server:**  
Configure os servidores de contêiner no WebSphere Application Server com um arquivo de propriedades do servidor e o com o arquivo XML da política de implementação que são integrados em um módulo aplicativo Java EE. O ciclo de vida dos servidores de contêiner é controlado pelo aplicativo. Os servidores de contêiner iniciam e param dentro do aplicativo.

Use os tópicos a seguir para configurar seus servidores de catálogos e contêiner:

## Melhor Prática: Armazenando em Cluster o Serviço de Catálogo com Domínios de Serviço de Catálogo

Quando estiver usando o serviço de catálogo, no mínimo dois servidores de catálogos são necessários para evitar um ponto de falha único. Dependendo do número de nós em seu ambiente, é possível criar configurações diferentes para assegurar que pelo menos dois servidores de catálogos estejam sempre em execução.

### Número de Servidores de Catálogos

A melhor prática para evitar um ponto de falha único para o domínio do serviço de catálogos é iniciar no mínimo três servidores de catálogos em três nós diferentes.

Se você estiver usando somente dois nós, configure os dois servidores de catálogo em cada um dos dois nós para um total de quatro processos do servidor de catálogos. Criar esta configuração assegura que quando apenas um dos nós for iniciado, os dois servidores de catálogos necessárias estejam em execução. É necessário iniciar pelo menos dois servidores de catálogo ao mesmo tempo. Quando os servidores de catálogos forem iniciados, eles procurarão por outros servidores de catálogos na configuração e não serão iniciados com êxito até que pelo menos um outro servidor de catálogos seja localizado.

## Exemplo: Iniciando Quatro Servidores de Catálogos para Dois Nós em um Ambiente Independente

O script a seguir inicia os servidores de catálogos cs0 e cs1 no nó host1 e inicia os servidores de catálogos cs2 e cs3 no nó host2.

```
./startOgServer.sh|bat cs0 -listenerPort 2809 -catalogServiceEndpoints  
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604  
-quorum true -jvmArgs -Xmx256m
```

```
./startOgServer.sh|bat cs1 -listenerPort 2810 -catalogServiceEndpoints  
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604  
-quorum true -jvmArgs -Xmx256m
```

```
./startOgServer.sh|bat cs2 -listenerPort 2809 -catalogServiceEndpoints  
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604  
-quorum true -jvmArgs -Xmx256m
```

```
./startOgServer.sh|bat cs3 -listenerPort 2810 -catalogServiceEndpoints  
cs0:host1:6601:6602,cs1:host1:6603:6604,cs2:host2:6601:6602,cs3:host2:6603:6604  
-quorum true -jvmArgs -Xmx256m
```

**Lembre-se:** Você deve usar a opção **-listenerPort** porque cada um dos servidores de catálogos em execução em um nó requer um número de porta exclusivo.

## Exemplo, Iniciando Diversos Servidores de Catálogos em um Ambiente do WebSphere Application Server

Os servidores de catálogos iniciam automaticamente em um ambiente do WebSphere Application Server. É possível definir diversos servidores de catálogo para iniciar ao criar um domínio de serviço de catálogo. Depois de especificar diversos terminais no domínio de serviço de catálogo, reinicie os servidores de aplicativos incluídos para que os servidores de catálogos iniciem simultaneamente.

- **WebSphere Application Server Network Deployment:** É possível escolher diversos servidores de aplicativos existentes na célula para que sejam membros de um domínio de serviço de catálogo.
- **WebSphere Application Server de Base:** É possível iniciar o serviço de catálogo em diversos nós independentes. Ao definir diversos perfis na mesma imagem de instalação com a ferramenta de gerenciamento de perfil, é possível criar um conjunto de nós independentes, com cada um possuindo portas exclusivas designadas. Em cada servidor de aplicativos, defina o serviço de catálogos. É possível especificar qualquer outros servidores de aplicativos ao incluir servidores remotos na configuração. Depois de criar esta configuração em todos os servidores independentes, é possível iniciar o conjunto de servidores de aplicativos base simultaneamente ao executar o script **startServer** ou ao usar um serviço do Windows para iniciar os servidores.

## Ajustando a Configuração do Intervalo de Pulsação para Detecção de Failover

É possível configurar a quantidade de tempo entre verificações do sistema para servidores com falha com a configuração do intervalo de pulsação.

### Sobre Esta Tarefa

A configuração de failover varia dependendo do tipo de ambiente que você está usando. Se você estiver utilizando um ambiente independente, é possível configurar o failover com a linha de comandos. Se você estiver usando um ambiente do WebSphere Application Server Network Deployment, é necessário



configurar o failover no console administrativo do WebSphere Application Server Network Deployment.

## Procedimento

- Configurar o failover para ambientes independentes.

É possível configurar intervalos de pulsação na linha de comandos ao usar o parâmetro **-heartbeat** no arquivo de script **startOgServer**. Configure esse parâmetro para um dos seguintes valores:

Tabela 13. Intervalos de Pulsações

Valor	Ação	Descrição
0	Típica (padrão)	Failovers são tipicamente detectados em 30 segundos.
-1	Agressiva	Failovers são tipicamente detectados em 5 segundos.
1	Moderada	Failovers são tipicamente detectado em 180 segundos.

Um intervalo de pulsação agressivo pode ser útil quando os processos e a rede estão estáveis. Se a rede ou os processos não são configurados de maneira ideal, as pulsações podem ser perdidas, o que pode resultar em uma falsa detecção de falhas.

- Configurar o failover para ambientes do WebSphere Application Server.

O WebSphere Application Server Network Deployment Versão 6.0.2 e superior pode ser configurado para permitir que o WebSphere eXtreme Scale execute failover muito rapidamente. O tempo de failover padrão para falhas "hard" é de aproximadamente 200 segundos. Uma falha de disco rígido é um travamento de computador ou de um servidor físico, uma desconexão de cabo de rede ou um erro do sistema operacional. As falhas causadas por travamentos de processo ou por falhas de software normalmente executam failover em menos de um segundo. A detecção de falha para falhas "soft" ocorre quando os soquetes de rede a partir de um processo inativo são fechados automaticamente pelo sistema operacional para o servidor que hospeda o processo.

### Configuração de pulsação do grupo principal

O WebSphere eXtreme Scale que executa um processo do WebSphere Application Server herda as características de failover a partir das configurações de grupo principal do servidor de aplicativos. As seguintes seções descrevem como definir as configurações de pulsação do grupo principal para versões diferentes do WebSphere Application Server Network Deployment:

#### – Atualize as configurações do grupo principal para o WebSphere Application Server Network Deployment Versão 6.x e 7.x:

O intervalo de pulsação pode ser especificado em segundos nas versões do WebSphere Application Server da Versão 6.0 a Versão 6.1.0.12, ou em milissegundos iniciando na Versão 6.1.0.13. Também é necessário especificar o número de pulsações perdidas. Esse valor indica quantas pulsações podem estar ausentes antes que um Java Virtual Machine (JVM) equivalente seja considerado uma falha. O tempo de detecção de falha "hard" é, aproximadamente, o produto do intervalo de pulsações e o número de pulsações ausentes.

Essas propriedades são especificadas usando as propriedades customizadas no grupo principal usando o console administrativo do WebSphere. Consulte Propriedades customizadas do grupo principal para obter detalhes da configuração. Estas propriedades devem ser especificadas para todos os grupos principais utilizados pelo aplicativo:



- O intervalo de pulsação é especificado usando a propriedade customizada IBM\_CS\_FD\_PERIOD\_SEC para segundos ou a propriedade customizada IBM\_CS\_FD\_PERIOD\_MILLIS para milissegundos (requer a Versão 6.1.0.13 ou posterior).
- O número de pulsações ausentes é especificado usando a propriedade customizada IBM\_CS\_FD\_CONSECUTIVE\_MISSED.

O valor padrão para a propriedade IBM\_CS\_FD\_PERIOD\_SEC é 20 e para a propriedade IBM\_CS\_FD\_CONSECUTIVE\_MISSED é 10. Se a propriedade IBM\_CS\_FD\_PERIOD\_MILLIS for especificada, ela substituirá qualquer conjunto de propriedades customizadas IBM\_CS\_FD\_PERIOD\_SEC. Os valores destas propriedades são valores de número inteiro positivo.

Use as seguintes configurações para alcançar um tempo de detecção de falha de 1500 milissegundos para os servidores WebSphere Application Server Network Deployment Versão 6.x:

- Configure IBM\_CS\_FD\_PERIOD\_MILLIS = 750 (WebSphere Application Server Network Deployment V6.1.0.13 e superior)
  - Configure IBM\_CS\_FD\_CONSECUTIVE\_MISSED = 2
- **Atualize as configurações do grupo principal para o WebSphere Application Server Network Deployment Versão 7.0:**

O WebSphere Application Server Network Deployment Versão 7.0 fornece duas configurações de grupo principal que podem ser ajustadas para aumentar ou diminuir a detecção de failover:

- **Período de transmissão de pulsação.** O padrão é 30000 milissegundos.
- **Período de tempo limite de pulsação.** O padrão é 180000 milissegundos.

Para obter mais detalhes sobre como alterar essas configurações, consulte o Centro de Informações do WebSphere Application Server Network Deployment: Configurações de Falha e Detecção de Descoberta.

Use as seguintes configurações para alcançar um tempo de detecção de falha de 1500 milissegundos para os servidores WebSphere Application Server Network Deployment Versão 7:

- Configure o período de transmissão de pulsação para 750 milissegundos.
- Configure o tempo limite da pulsação para 1500 milissegundos.

## O que Fazer Depois

Quando estas configurações são modificadas para fornecer tempos de failover curtos, há alguns problemas de ajuste de sistema a considerar. Primeiro, Java não é um ambiente em tempo real. É possível que os encadeamentos sejam atrasados se a JVM estiver experimentando longos tempos de coleta de lixo. Os encadeamentos também podem ser atrasados se a máquina que hospeda o JVM estiver sobrecarregada (devido ao próprio JVM ou outros processos que são executados na máquina). Se os encadeamentos forem atrasados, as pulsações talvez não sejam enviadas a tempo. No pior dos casos, elas podem ser atrasadas pelo tempo necessário de failover. Se os encadeamentos forem atrasados, ocorrerão falsas detecções de falhas. O sistema deve ser ativado e dimensionado para garantir que falsas detecções de falhas não aconteçam na produção. O teste de carregamento adequado é a melhor maneira de garantir isto.

**Nota:** A versão atual do eXtreme Scale suporta o WebSphere Real Time.

## Configurando o WebSphere eXtreme Scale com o WebSphere Application Server

É possível executar um serviço de catálogo e processos de servidor de contêiner no WebSphere Application Server. O processo para configurar esses servidores é diferente de uma configuração independente. O serviço de catálogo pode ser iniciado automaticamente nos servidores ou gerenciadores de implementação do WebSphere Application Server. O processo do contêiner é iniciado quando um aplicativo eXtreme Scale for implementado e iniciado no ambiente do WebSphere Application Server.

### Sobre Esta Tarefa

**Atenção:** Não coloque seus servidores de contêiner com servidores de catálogos em um ambiente de produção. Inclua o serviço de catálogo em vários processos do agente do nó ou em um servidor de aplicativos que não esteja hospedando um aplicativo eXtreme Scale.

### Configurando o Serviço de Catálogo no WebSphere Application Server

Os processos do serviço de catálogo podem ser executados no WebSphere Application Server. O ciclo de vida do servidor no WebSphere Application Server determina quando o serviço de catálogo é iniciado e parado.

### Procedimento

1. Escolha um ou mais processos do WebSphere Application Server a serem aumentados com o perfil do WebSphere eXtreme Scale. Consulte “Criando e Alterando Perfis para o WebSphere eXtreme Scale” na página 182 para obter informações adicionais. Se desejar que o serviço de catálogo seja iniciado automaticamente no WebSphere Application Server Network Deployment no gerenciador de implementação, instale o WebSphere eXtreme Scale no nó do gerenciador de implementação e aumente o perfil do gerenciador de implementação.
2. Configure os arquivos de propriedades do servidor para os processos do WebSphere Application Server e inclua-os no caminho de classe do nó. Consulte o Arquivo de Propriedades do Servidor para obter informações adicionais.
3. Configure um domínio de serviço de catálogo. O domínio de serviço de catálogo é um grupo de servidores de catálogos dentro do seu ambiente. Consulte “Criando Domínios do Serviço de Catálogo no WebSphere Application Server” para obter mais informações.
4. Inicie os processos do WebSphere Application Server que hospedam os servidores de catálogos. Consulte “Iniciando e Parando Servidores em um Ambiente do WebSphere Application Server” na página 401 para obter informações adicionais.

### Criando Domínios do Serviço de Catálogo no WebSphere Application Server:

Os domínios do serviço de catálogo definem um grupo de servidores de catálogo que gerenciam o posicionamento de shards e monitoram o funcionamento de servidores de contêiner em sua grade de dados.

### Antes de Iniciar

- Instale o WebSphere eXtreme Scale no WebSphere Application Server. Consulte “Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client com WebSphere Application Server” na página 161 para obter mais informações.

### Sobre Esta Tarefa

Ao criar um domínio do serviço de catálogo, você está definindo uma coleta altamente disponível de servidores de catálogo.

Esses servidores de catálogo podem ser executados no WebSphere Application Server dentro de uma única célula ou um grupo principal. O domínio do serviço de catálogo também pode definir um grupo remoto de servidores executados em processos SE Java diferentes ou outras células do WebSphere Application Server.

**Para servidores de catálogos que são executados em servidores de aplicativos existentes na célula:** Ao definir um domínio do serviço de catálogo que coloca servidores de catálogos nos servidores de aplicativos dentro da célula, os mecanismos do grupo principal do WebSphere Application Server são usados. O serviço de catálogo inicia automaticamente nos servidores de aplicativos na célula. Como resultado, os membros de um único domínio do serviço de catálogo não podem ampliar os limites de um grupo principal e um domínio do serviço de catálogo, portanto, não pode ampliar células. Entretanto, os servidores de contêiner e clientes do WebSphere eXtreme Scale podem estender células conectando-se a um servidor de catálogos nos limites da célula, tal como um domínio do serviço de catálogo independente ou um domínio de serviço de catálogo integrado em outra célula.

**Para servidores de catálogos remotos:** É possível conectar contêineres e clientes do WebSphere eXtreme Scale a um domínio do serviço de catálogo que está executando em outra célula do WebSphere Application Server ou que está executando como processos independentes. Como servidores de catálogos configurados remotamente não iniciam automaticamente na célula, você deve iniciar manualmente qualquer servidor de catálogos configurado remotamente. Quando configurar um domínio do serviço de catálogo remoto, o nome de domínio deve corresponder ao nome de domínio que você especificou ao iniciar os servidores de catálogos remotos. O nome de domínio do serviço de catálogo padrão para servidores de catálogos independentes é `DefaultDomain`. Especifique um nome de domínio do serviço de catálogo com o parâmetro **startOgServer** command **-domain**, um arquivo de propriedades de servidor ou com a API do servidor integrado. É necessário iniciar cada processo do servidor de catálogos remoto no domínio remoto com o mesmo nome de domínio. Consulte “Iniciando um Serviço de Catálogo Independente” na página 387 para obter informações adicionais sobre como iniciar servidores de catálogos.

**Atenção:** Não coloque os serviços de catálogo com servidores de contêiner do WebSphere eXtreme Scale em um ambiente de produção. Inclua o serviço de catálogo em vários processos do agente do nó ou em um servidor de aplicativos que não esteja hospedando um aplicativo WebSphere eXtreme Scale.

### Procedimento

1. Crie o domínio do serviço de catálogo.
  - a. No console administrativo do WebSphere Application Server, clique em **Administração do sistema > WebSphere eXtreme Scale > Domínios do serviço de catálogo > Novo**.

- b. Defina um nome, valor padrão e credenciais de autenticação JMX para o domínio do serviço de catálogo. Se estiver configurando terminais remotos para o domínio do serviço de catálogo, o nome do domínio do serviço de catálogo deverá corresponder ao nome do domínio de serviço de catálogo especificado ao iniciar os servidores de catálogos.
    - c. Inclua terminais do servidor de catálogos. É possível selecionar servidores de aplicativos existentes ou incluir servidores remotos que estão executando um serviço de catálogo.
  2. Teste a conexão com os servidores de catálogos em seu domínio de serviço de catálogo. Para servidores de aplicativos existentes, os servidores de catálogos iniciam quando o servidor de aplicativos associado é iniciado. Para servidores de aplicativos remotos, você deve iniciar os servidores manualmente usando o comando **startOgServer** ou a API do servidor integrado.
    - a. No console administrativo do WebSphere Application Server, clique em **Administração do sistema > WebSphere eXtreme Scale > Domínios do serviço de catálogo**.
    - b. Selecione o domínio do serviço de catálogo que deseja testar e clique em **Testar conexão**. Quando você clica neste botão, todos os terminais do domínio de serviço de catálogo definidos são consultados um a um, se algum terminal estiver disponível, retornará uma mensagem que indica que a conexão com o domínio do serviço de catálogo foi bem-sucedida.

*Tarefas Administrativas de Domínio do Serviço de Catálogo:*

É possível usar os idiomas de script Jacl ou Jython para gerenciar os domínios de serviço de catálogo em sua configuração do WebSphere Application Server.

### **Requisitos**

É necessário ter o WebSphere eXtreme Scale Client instalado em seu ambiente do WebSphere Application Server.

### **Lista de Todas as Tarefas Administrativas**

Para obter uma lista de todas as tarefas administrativas associadas aos domínios do serviço de catálogo, execute o seguinte comando com wsadmin:

```
wsadmin>$AdminTask help XSDomainManagement
```

### **Comandos**

As tarefas administrativas para os domínios do serviço de catálogo incluem os seguintes comandos:

- “createXSDomain” na página 255
- “deleteXSDomain” na página 258
- “getDefaultXSDomain” na página 258
- “listXSDomains” na página 259
- “modifyXSDomain” na página 259
- “testXSDomainConnection” na página 264
- “testXSServerConnection” na página 264

## createXSDomain

O comando **createXSDomain** registra um novo domínio do serviço de catálogo.

Tabela 14. Argumentos de Comando createXSDomain

Argumento	Descrição
<b>-name</b> (obrigatório)	Especifica o nome do domínio do serviço de catálogo que deseja criar.
<b>-default</b>	Especifica se o domínio do serviço de catálogo é o padrão para a célula. O valor padrão é true. (Booleano: configurar para true ou false)
<b>-properties</b>	Especifica as propriedades customizadas para o domínio do serviço de catálogo.

Tabela 15. Argumentos da Etapa defineDomainServers

Argumento	Descrição
<i>name_of_endpoint</i>	Especifica o nome do terminal do serviço de catálogo. <ul style="list-style-type: none"><li>• <b>Para servidores de aplicativos existentes:</b> O nome do terminal deve estar no seguinte formato: <i>cell_name\node_name\server_name</i></li><li>• <b>Para servidores remotos:</b> Especifica o nome do host do servidor remoto. É possível usar o mesmo nome para diversos terminais, mas os valores de porta do cliente devem ser exclusivos para cada terminal.</li></ul>
<i>custom_properties</i>	Especifica propriedades customizadas para o terminal do domínio do serviço de catálogo. Se não tiver nenhuma propriedade customizada, use um conjunto de aspas duplas ("") para este argumento.

Tabela 15. Argumentos da Etapa `defineDomainServers` (continuação)

Argumento	Descrição
<code>endpoint_ports</code>	<p>Especifica os números de porta para o terminal do domínio do serviço de catálogo. As portas devem ser especificadas na seguinte ordem:  <code>&lt;client_port&gt;,&lt;listener_port&gt;</code></p> <p><b>Porta do Cliente</b>  Especifica a porta que é usada para a comunicação entre os servidores de catálogos no domínio de serviço de catálogo. Este valor é necessário para os servidores de catálogos que estão em execução apenas nos processos do WebSphere Application Server e podem ser configurados para qualquer porta que não está sendo usada em outro lugar.</p> <p><b>Porta do Listener</b>  Especifica a porta que é usada para comunicação com clientes. Este valor é necessário para terminais remotos e deve corresponder ao valor usado quando o serviço de catálogo foi iniciado. A porta do listener é usada pelos clientes e contêineres se comunicarem com o serviço de catálogo.</p> <p><b>Para obter os terminais remotos do WebSphere eXtreme Scale</b> : Define a porta do listener do Object Request Broker (ORB) para os contêineres e clientes se comunicarem com o serviço de catálogo por meio do ORB. Para terminais do WebSphere Application Server, o valor da porta do listener é opcional porque o valor é herdado da configuração da porta <code>BOOTSTRAP_ADDRESS</code>.</p>

Tabela 16. Argumentos da Etapa `configureClientSecurity`

Argumento	Descrição
<code>-securityEnabled</code>	<p>Especifica que a segurança do cliente está ativada para o servidor de catálogos. O arquivo das propriedades de servidor que está associado ao servidor de catálogos selecionado deve ter a configuração <code>securityEnabled</code> correspondente no arquivo de propriedades de servidor. Se essas configurações não corresponderem, uma exceção ocorrerá. (Booleano: configurar para <code>true</code> ou <code>false</code>)</p>

Tabela 16. Argumentos da Etapa `configureClientSecurity` (continuação)

Argumento	Descrição
<code>-credentialAuthentication</code> (opcional)	<p>Indica se a autenticação da credencial é forçada ou suportada.</p> <p><b>Nunca</b> Nenhuma autenticação por certificado de cliente é forçada.</p> <p><b>Requerido</b> A autenticação de credencial é sempre forçada. Se o servidor não suportar autenticação de credencial, o cliente não poderá se conectar ao servidor.</p> <p><b>Suportado</b> (Padrão) A autenticação de credencial é forçada apenas se o cliente e o servidor suportarem a autenticação de credencial.</p>
<code>-authenticationRetryCount</code> (opcional)	<p>Especifica o número de vezes que a autenticação é tentada novamente se a credencial expirar.</p> <p>Se você não deseja tentar a autenticação novamente, configure o valor para 0. O valor padrão é 0.</p>
<code>-credentialGeneratorClass</code>	<p>Indicates the <code>com.ibm.websphere.objectgrid.security.plugins.builtins</code>. Classe de implementação <code>WSTokenCredentialGenerator</code>, para que o cliente recupere os tokens de segurança do encadeamento.</p>
<code>-credentialGeneratorProps</code>	<p>Especifica as propriedades para a classe de implementação <code>CredentialGenerator</code>. As propriedades são configuradas para o objeto com o método <code>setProperties(String)</code>. O valor da propriedades do gerador de credencial é usado apenas quando um valor for especificado para o campo <b>Classe do Gerador de Credencial</b>.</p>

#### Valor de retorno:

#### Uso de exemplo do modo em lote

O modo em lote requer formatação correta da entrada do comando. Considere usar o modo interativo para assegurar que os valores que você inserir sejam processados corretamente. Quando usar o modo em lote, você deverá definir os argumentos da etapa **`-defineDomainServers`** usando uma matriz de propriedades específica. Esta matriz de propriedades está no formato `name_of_endpoint custom_properties endpoint_ports`. O valor de `endpoint_ports` é uma lista de portas que devem ser especificadas na seguinte ordem: `<client_port>,<listener_port>`.

- Crie um domínio do serviço de catálogo de terminais remotos usando Jacl:

```
$AdminTask createXSDomain {-name
TestDomain -default true -defineDomainServers
{{xhost1.ibm.com "" ,2809}} -configureClientSecurity {-securityEnabled false
-credentialAuthentication Required -authenticationRetryCount 0 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1"}}
```

- Crie um domínio do serviço de catálogo de terminais remotos usando a sequência Jython:



```
AdminTask.createXSDomain('[-name TestDomain -default true
-defineDomainServers [[xhost1.ibm.com "" ,2809]
[xhost2.ibm.com "" ,2809]] -configureClientSecurity [-securityEnabled false
-credentialAuthentication Required -authenticationRetryCount 0 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1" ]')
```

- Crie um domínio de serviço de catálogo de terminais do servidor de aplicativos existentes usando Jacl:

```
$AdminTask createXSDomain {-name TestDomain -default true -defineDomainServers
{{cellName/nodeName/serverName "" 1109}}}
```

### Uso de exemplo do modo interativo

- Usando Jacl:

```
$AdminTask createXSDomain {-interactive}
```

- Usando a cadeia Jython:

```
AdminTask.createXSDomain ('[-interactive]')
```

### deleteXSDomain

O comando **deleteXSDomain** exclui um domínio do serviço de catálogo.

#### Parâmetros obrigatórios:

##### **-name**

Especifica o nome do domínio do serviço de catálogo a excluir.

#### Valor de retorno:

### Uso de exemplo do modo em lote

- Usando Jacl:

```
$AdminTask deleteXSDomain {-name TestDomain }
```

- Usando a cadeia Jython:

```
AdminTask.deleteXSDomain('[-name TestDomain ]')
```

### Uso de exemplo do modo interativo

- Usando Jacl:

```
$AdminTask deleteXSDomain {-interactive}
```

- Usando a cadeia Jython:

```
AdminTask.deleteXSDomain ('[-interactive]')
```

### getDefaultXSDomain

O comando **getDefaultXSDomain** retorna o domínio do serviço de catálogo padrão para a célula.

#### Parâmetros obrigatórios: Nenhum

**Valor de retorno:** O nome do domínio do serviço de catálogo padrão.

### Uso de exemplo do modo em lote

- Usando Jacl:

```
$AdminTask getDefaultXSDomain
```

- Usando a cadeia Jython:

```
AdminTask.getDefaultXSDomain
```

### Uso de exemplo do modo interativo

- Usando Jacl:  
`$AdminTask getDefaultXSDomain {-interactive}`
- Usando a cadeia Jython:  
`AdminTask.getDefaultXSDomain ('[-interactive]')`

### listXSDomains

O comando **listXSDomains** retorna uma lista dos domínios do serviço de catálogo existentes.

**Parâmetros obrigatórios:** Nenhum

**Valor de retorno:** Uma lista de todos os domínios do serviço de catálogo na célula.

### Uso de exemplo do modo em lote

- Usando Jacl:  
`$AdminTask listXSDomains`
- Usando a cadeia Jython:  
`AdminTask.listXSDomains`

### Uso de exemplo do modo interativo

- Usando Jacl:  
`$AdminTask listXSDomains {-interactive}`
- Usando a cadeia Jython:  
`AdminTask.listXSDomains ('[-interactive]')`

### modifyXSDomain

O comando **modifyXSDomain** modifica um domínio do serviço de catálogo existente.

O modo em lote requer formatação correta da entrada do comando. Considere usar o modo interativo para assegurar que os valores que você inserir sejam processados corretamente. Quando usar o modo em lote, você deve definir os argumentos da etapa **-modifyEndpoints**, **-addEndpoints** e **-removeEndpoints** usando uma matriz de propriedades específica. Esta matriz de propriedades está no formato *name\_of\_endpoint host\_name custom\_properties endpoint\_ports*. O valor de *endpoint\_ports* é uma lista de portas que devem ser especificadas na seguinte ordem: *<client\_port>,<listener\_port>*.

Tabela 17. Argumentos de Comando modifyXSDomain

Argumento	Descrição
<b>-name</b> (obrigatório)	Especifica o nome do domínio do serviço de catálogo que você deseja editar.
<b>-default</b>	Se configurado para true, especifica que o domínio do serviço de catálogo selecionado é o padrão para a célula. (Booleano)
<b>-properties</b>	Especifica as propriedades customizadas para o domínio do serviço de catálogo.

Tabela 18. Argumentos da Etapa modifyEndpoints

Argumento	Descrição
<p><i>name_of_endpoint</i></p>	<p>Especifica o nome do terminal do serviço de catálogo.</p> <ul style="list-style-type: none"> <li>• <b>Para servidores de aplicativos existentes:</b> O nome do terminal deve estar no seguinte formato: <i>cell_name\node_name\server_name</i></li> <li>• <b>Para servidores remotos:</b> Especifica o nome do host do servidor remoto. É possível usar o mesmo nome para diversos terminais, mas os valores da porta do listener devem ser exclusivos para cada terminal.</li> </ul>
<p><i>endpoint_ports</i></p>	<p>Especifica os números de porta para o terminal do domínio do serviço de catálogo. Os terminais devem ser especificados na seguinte ordem: <i>&lt;client_port&gt;,&lt;listener_port&gt;</i></p> <p><b>Porta do Cliente</b> Especifica a porta que é usada para a comunicação entre os servidores de catálogos no domínio de serviço de catálogo. Este valor é necessário para os servidores de catálogos que estão em execução apenas nos processos do WebSphere Application Server e podem ser configurados para qualquer porta que não está sendo usada em outro lugar.</p> <p><b>Porta do Listener</b> Especifica a porta que é usada para comunicação com clientes. Este valor é necessário para terminais remotos e deve corresponder ao valor usado quando o serviço de catálogo foi iniciado. A porta do listener é usada pelos clientes e contêineres se comunicarem com o serviço de catálogo.</p> <p><b>Para obter os terminais remotos do WebSphere eXtreme Scale :</b> Define a porta do listener do Object Request Broker (ORB) para os contêineres e clientes se comunicarem com o serviço de catálogo por meio do ORB. Para terminais do WebSphere Application Server, a especificação do valor da porta do listener é opcional porque o valor é herdado da configuração da porta <code>BOOTSTRAP_ADDRESS</code>.</p>

Tabela 19. Argumentos da Etapa `addEndpoints`

Argumento	Descrição
<i>name_of_endpoint</i>	<p>Especifica o nome do terminal do serviço de catálogo.</p> <ul style="list-style-type: none"> <li>• <b>Para servidores de aplicativos existentes:</b> O nome do terminal deve estar no seguinte formato: <i>cell_name\node_name\server_name</i></li> <li>• <b>Para servidores remotos:</b> Especifica o nome do host do servidor remoto. É possível usar o mesmo nome para diversos terminais, mas os valores da porta do listener devem ser exclusivos para cada terminal.</li> </ul>
<i>custom_properties</i>	<p>Especifica propriedades customizadas para o terminal do domínio do serviço de catálogo. Se não tiver nenhuma propriedade customizada, use um conjunto de aspas duplas ("") para este argumento.</p>

Tabela 19. Argumentos da Etapa addEndpoints (continuação)

Argumento	Descrição
<i>endpoint_ports</i>	<p>Especifica os números de porta para o terminal do domínio do serviço de catálogo. Os terminais devem ser especificados na seguinte ordem:  <code>&lt;client_port&gt;,&lt;listener_port&gt;</code></p> <p><b>Porta do Cliente</b>  Especifica a porta que é usada para a comunicação entre os servidores de catálogos no domínio de serviço de catálogo. Este valor é necessário para os servidores de catálogos que estão em execução apenas nos processos do WebSphere Application Server e podem ser configurados para qualquer porta que não está sendo usada em outro lugar.</p> <p><b>Porta do Listener</b>  Especifica a porta que é usada para comunicação com clientes. Este valor é necessário para terminais remotos e deve corresponder ao valor usado quando o serviço de catálogo foi iniciado. A porta do listener é usada pelos clientes e contêineres se comunicarem com o serviço de catálogo.</p> <p><b>Para obter os terminais remotos do WebSphere eXtreme Scale :</b> Define a porta do listener do Object Request Broker (ORB) para os contêineres e clientes se comunicarem com o serviço de catálogo por meio do ORB. Para terminais do WebSphere Application Server, a especificação do valor da porta do listener é opcional porque o valor é herdado da configuração da porta <code>BOOTSTRAP_ADDRESS</code>.</p>

Tabela 20. Argumentos da Etapa removeEndpoints

Argumento	Descrição
<i>name_of_endpoint</i>	Especifica o nome do terminal do serviço de catálogo a excluir.

Tabela 21. Argumentos da Etapa `configureClientSecurity`

Argumento	Descrição
<code>-securityEnabled</code>	Especifica que a segurança do cliente está ativada para o servidor de catálogos. O arquivo das propriedades de servidor que está associado ao servidor de catálogos selecionado deve ter a configuração <code>securityEnabled</code> correspondente no arquivo de propriedades de servidor. Se essas configurações não corresponderem, uma exceção ocorrerá. (Booleano: configurar para <code>true</code> ou <code>false</code> )
<code>-credentialAuthentication</code> (opcional)	Indica se a autenticação da credencial é forçada ou suportada.  <b>Nunca</b> Nenhuma autenticação por certificado de cliente é forçada.  <b>Requerido</b> A autenticação de credencial é sempre forçada. Se o servidor não suportar autenticação de credencial, o cliente não poderá se conectar ao servidor.  <b>Suportado</b> (Padrão) A autenticação de credencial é forçada apenas se o cliente e o servidor suportarem a autenticação de credencial.
<code>-authenticationRetryCount</code> (opcional)	Especifica o número de vezes que a autenticação é tentada novamente se a credencial expirar.  Se você não deseja tentar a autenticação novamente, configure o valor para 0. O valor padrão é 0.
<code>-credentialGeneratorClass</code>	Indicates the <code>com.ibm.websphere.objectgrid.security.plugins.builtins</code> . Classe de implementação <code>WSTokenCredentialGenerator</code> , para que o cliente recupere os tokens de segurança do encadeamento.
<code>-credentialGeneratorProps</code>	Especifica as propriedades para a classe de implementação <code>CredentialGenerator</code> . As propriedades são configuradas para o objeto com o método <code>setProperty(String)</code> . O valor da propriedades do gerador de credencial é usado apenas quando um valor for especificado para o campo <b>Classe do Gerador de Credencial</b> .

#### Valor de retorno:

#### Uso de exemplo do modo em lote

- Usando Jacl:
 

```
$AdminTask modifyXSDomain {-name TestDomain -default true -modifyEndpoints
  {{xhost1.ibm.com "" ,2809}} -addEndpoints {{xhost2.ibm.com "" ,2809}}
  -removeEndpoints {{xhost3.ibm.com}}
```
- Usando a cadeia Jython:
 

```
AdminTask.modifyXSDomain('[-name TestDomain
  -default false -modifyEndpoints [[xhost1.ibm.com "" ,2809]]
  -addEndpoints [[xhost3.ibm.com "" ,2809]]
  -removeEndpoints [[xhost2.ibm.com]]')
```

- Usando a especificação de segurança do cliente durante o comando de modificação:

```
$AdminTask modifyXSDomain {-name myDomain -default false
-configureClientSecurity {-securityEnabled true -
Supported -authenticationRetryCount 1 -credentialGeneratorClass
com.ibm.websphere.objectgrid.security.plugins.builtins.UserPasswordCredentialGenerator
-credentialGeneratorProps "manager manager1"}}
```

### Uso de exemplo do modo interativo

- Usando Jacl:

```
$AdminTask modifyXSDomain {-interactive}
```

- Usando a cadeia Jython:

```
AdminTask.modifyXSDomain ('[-interactive]')
```

### testXSDomainConnection

O comando **testXSDomainConnection** testa a conexão com um domínio do serviço de catálogo.

#### Parâmetros obrigatórios:

##### -name

Especifica o nome do domínio do serviço de catálogo para o qual testar a conexão.

#### Parâmetros Opcionais

##### -timeout

Especifica o período máximo de tempo a aguardar a conexão, em segundos.

**Valor de retorno:** Se uma conexão puder ser feita, retornará true, caso contrário, serão retornadas informações de erro de conexão.

### Uso de exemplo do modo em lote

- Usando Jacl:

```
$Admintask testXSDomainConnection
```

- Usando a cadeia Jython:

```
AdminTask.testXSDomainConnection
```

### Uso de exemplo do modo interativo

- Usando Jacl:

```
$AdminTask testXSDomainConnection {-interactive}
```

- Usando a cadeia Jython:

```
AdminTask.testXSDomainConnection ('[-interactive]')
```

### testXSServerConnection

O comando **testXSServerConnection** testa a conexão com um servidor de catálogos. Este comando funciona para servidores independentes e para servidores que fazem parte de um domínio do serviço de catálogo.

#### Parâmetros obrigatórios:

##### host

Especifica o host no qual reside o servidor de catálogos.



### **listenerPort**

Especifica a porta do listener para o servidor de catálogos.

### **Parâmetros Opcionais**

#### **timeout**

Especifica o período máximo de tempo a aguardar uma conexão para o servidor de catálogos, em segundos.

#### **domain**

Especifica o nome de um domínio do serviço de catálogo. Se você definir um valor para este parâmetro, as propriedades de segurança do cliente para o domínio de serviço de catálogo especificado serão usadas para testar a conexão. Caso contrário, ocorrerá uma procura para localizar o domínio de serviço de catálogo para o host especificado e a porta do listener. Se um domínio de serviço de catálogo for localizado, as propriedades de segurança do cliente que forem definidas para o domínio de serviço de catálogo serão usadas para testar o servidor. Caso contrário, nenhuma propriedade de segurança do cliente será usada durante o teste.

### **Valor de retorno:**

#### **Uso de exemplo do modo em lote**

- Usando Jacl:  
`$Admintask testXSSTestServerConnection {-host xhost1.ibm.com -listenerPort 2809}`
- Usando a cadeia Jython:  
`AdminTask.testXSSTestServerConnection('[-host xshost3.ibm.com -listenerPort 2809]')`

#### **Uso de exemplo do modo interativo**

- Usando Jacl:  
`$AdminTask testXSSTestServerConnection {-interactive}`
- Usando a cadeia Jython:  
`AdminTask.testXSSTestServerConnection ('[-interactive]')`

#### *Coleção de Domínio do Serviço de Catálogo:*

Use essa página para gerenciar os domínios de serviço de catálogo. Os domínios de serviço de catálogo definem um grupo de servidores de catálogos que gerenciam o posicionamento dos shards e monitora o funcionamento dos servidores de contêiner na grade de dados.

Para visualizar essa página do console administrativo, clique em **Administração do Sistema > WebSphere eXtreme Scale > Domínios do Serviço de Catálogo**. Para criar um novo domínio do serviço de catálogo, clique em **Novo**. Para excluir um domínio de serviço de catálogo, selecione o domínio de serviço de catálogo que deseja remover e clique em **Excluir**.

#### *Testar Conexão:*

Quando você clica no botão **Testar conexão**, todos os terminais do domínio de serviço de catálogo definidos são consultados um a um e, se algum terminal estiver disponível, retornará uma mensagem indicando que a conexão com o domínio de serviço de catálogo foi bem-sucedida. É possível usar este botão para testar se as informações de conexão e de segurança foram configuradas corretamente.


### *Definir Padrão:*

Define o domínio de serviço de catálogo que é usado como o padrão. Selecione um domínio de serviço de catálogo como o padrão e clique em **Configurar padrão**. Apenas um domínio de serviço de catálogo pode ser selecionado como o padrão.

### *Nome:*

Especifica o nome para o domínio de serviço de catálogo.

### *Padrão:*

Especifica qual domínio de serviço de catálogo na lista é o padrão. O domínio de serviço de catálogo padrão é indicado com o seguinte ícone:  .

### *Configurações do Domínio de Serviço de Catálogo:*

Use esta página para gerenciar as configurações para um domínio de serviço de catálogo específico. Os domínios de serviço de catálogo definem um grupo de servidores de catálogos que gerenciam o posicionamento dos shards e monitora o funcionamento dos servidores de contêiner na grade de dados. É possível definir um domínio de serviço de catálogo que esteja na mesma célula que o seu gerenciador de implementação. Também é possível definir domínios de serviço de catálogo remoto se a configuração do WebSphere eXtreme Scale estiver em uma célula diferente ou se sua grade de dados for composta de processos Java SE.

Para visualizar essa página do console administrativo, clique em **Administração do Sistema > WebSphere eXtreme Scale > Domínios do Serviço de Catálogo > catalog\_service\_domain\_name**.

### *Testar a Conexão:*

Quando você clica no botão **Testar conexão**, todos os terminais do domínio de serviço de catálogo definidos são consultados um a um e, se algum terminal estiver disponível, retornará uma mensagem indicando que a conexão com o domínio de serviço de catálogo foi bem-sucedida. É possível usar este botão para testar se as informações de conexão e de segurança foram configuradas corretamente.

### *Nome:*

Especifica o nome do domínio de serviço de catálogo.

*Ative este domínio de serviço de catálogo como o padrão, a menos que outro domínio de serviço de catálogo seja explicitamente especificado:*

Se você marcar essa caixa de seleção, o domínio de serviço de catálogo selecionado se tornará o domínio de serviço de catálogo padrão para a célula. Cada perfil do servidor na célula que for aumentado com o perfil do WebSphere eXtreme Scale pertence ao domínio de serviço de catálogo selecionado.

Para o WebSphere eXtreme Scale, todos os contêineres eXtreme Scale que estiverem integrados nos módulos do aplicativo Java EE se conectam ao domínio padrão. Os clientes podem se conectar ao domínio padrão usando a API

ServerFactory.getServerProperties().getCatalogServiceBootstrap() para recuperar os terminais de serviço de catálogo a serem usados quando chamar a API ObjectGridManager.connect().

Se o domínio padrão for alterado para apontar para um conjunto diferente de servidores de catálogos, todos os contêineres e clientes referenciarão o novo domínio depois que eles são reiniciados.

*Servidores de catálogos:*

Especifica uma lista de servidores de catálogos que pertencem a este domínio de serviço de catálogo.

Clique em **Novo** para incluir um servidor de catálogos na lista. Este servidor de catálogos já deverá existir na configuração do eXtreme Scale. Também é possível editar ou excluir um servidor da lista ao selecionar o terminal e em seguida, clicar em **Editar** ou um **Excluir**. Defina as seguintes propriedades para cada terminal do servidor de catálogos:

#### **Terminal de servidores de catálogo**

Especifica o nome do servidor de aplicativos existente ou do servidor remoto no qual o serviço de catálogo está em execução. Um domínio de serviço de catálogo não pode conter uma combinação de terminais de servidores de aplicativos existentes e de servidores remotos.

- **Servidor de aplicativos existente:** Especifica o caminho de um servidor de aplicativos, de um agente do nó ou de um gerenciador de implementação na célula. Um serviço de catálogo inicia automaticamente no servidor selecionado. Selecione na lista de servidores de aplicativos existentes. Todos os servidores de aplicativos definidos dentro do domínio de serviço de catálogo devem estar no mesmo grupo principal.
- **Servidor remoto:** Especifica o nome do host do servidor de catálogos remoto.

**Para os terminais remotos do WebSphere eXtreme Scale:** Especifica o nome do host do processo do servidor de catálogos remoto. Os servidores remotos devem ser iniciados com o script **startOgServer** ou com a API do servidor integrado.

#### **Porta do Cliente**

Especifica a porta que é usada para a comunicação entre os servidores de catálogos no domínio de serviço de catálogo. Este valor é necessário para os servidores de catálogos que estão em execução nos processos do WebSphere Application Server. É possível configurar o valor para qualquer porta que não estiver sendo usada por outro processo.




#### **Porta do Listener**

Especifica a porta que é usada para comunicação com clientes. Este valor é necessário para terminais remotos e deve corresponder ao valor usado quando o serviço de catálogo foi iniciado. A porta do listener é usada pelos clientes e contêineres se comunicarem com o serviço de catálogo.

**Para obter os terminais remotos do WebSphere eXtreme Scale :** Define a porta do listener do Object Request Broker (ORB) para os contêineres e clientes se comunicarem com o serviço de catálogo por meio do ORB. Para terminais do WebSphere Application Server, o valor da porta do listener é herdado da configuração da porta BOOTSTRAP\_ADDRESS.

#### **Status**

Tabela 22. Status do Terminal do Servidor de Catálogos

Ícone	Definição
	Desconhecido
	Iniciado
	Interrompido

*Propriedades de Segurança do Cliente:*

Use essa página para configurar a segurança do cliente para um domínio do serviço de catálogo. Essas configurações se aplicam a todos os servidores no domínio de serviço de catálogo. Essas propriedades podem ser substituídas pela especificação de um arquivo `splicer.properties` com a propriedade customizada `com.ibm.websphere.xs.sessionFilterProps` ou ao unir o arquivo EAR do aplicativo.

Para visualizar essa página do console administrativo, clique em **Administração do Sistema > WebSphere eXtreme Scale > Domínios de Serviço de Catálogo > catalog\_service\_domain\_name > Propriedades de Segurança do Cliente.**

*Ativar a Segurança do Cliente:*

Especifica que a segurança do cliente está ativada para o servidor de catálogos. O arquivo das propriedades de servidor que está associado ao servidor de catálogos selecionado deve ter a configuração **securityEnabled** correspondente no arquivo de propriedades de servidor. Se essas configurações não corresponderem, uma exceção ocorrerá.

*Autenticação de Credencial:*

Indica se a autenticação da credencial é forçada ou suportada.

**Nunca**

Nenhuma autenticação de credencial do cliente é imposta.

**Requerido**

A autenticação de credencial é sempre forçada. Se o servidor não suportar autenticação de credencial, o cliente não poderá se conectar ao servidor.

**Suportado**

A autenticação de credencial é imposta apenas se ambos cliente e servidor suportarem a autenticação de credencial.

*Contagem de Novas Tentativas de Autenticação:*

Especifica o número de vezes que a autenticação é tentada novamente se a credencial expirar.

Se você não deseja tentar a autenticação novamente, configure o valor para 0.

*Classe de Gerador de Credencial:*

Indica a classe de implementação `com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator`, de modo que o cliente recupere a credencial a partir do objeto `CredentialGenerator`.

É possível escolher entre duas classes de gerador de credencial predefinidas ou especificar um gerador de credencial customizado. Se escolher um gerador de credencial customizado, você deve indicar o nome da classe de gerador de credencial.

- `com.ibm.websphere.objectgrid.security.plugins.UserPasswordCredentialGenerator`
- `com.ibm.websphere.objectgrid.security.plugins.UserPasswordCredentialGenerator`
- Gerador de credencial customizado

*Tipo de Assunto:*

Especifica se está usando o responsável pela chamada J2EE ou o tipo `subject runAs J2EE`. Este valor deve ser especificado quando o gerador de credencial `WSTokenCredentialGenerator` for escolhido.

- **runAs:** O `subject` contém o principal do J2EE executado como uma identidade e do J2EE executado como uma credencial.
- **caller:** O `subject` contém o principal do responsável pela chamada J2EE e a credencial do responsável pela chamada J2EE.

*ID do usuário:*

Especifique um ID do usuário quando estiver usando a implementação do gerador de credenciais `UserPasswordCredentialGenerator`.

*Senha:*

Especifique uma senha quando estiver usando a implementação do gerador de credenciais `UserPasswordCredentialGenerator`.

*Propriedades do Gerador de Credencial:*

Especifica as propriedades para uma classe de implementação `CredentialGenerator` customizada. As propriedades são configuradas no objeto com o método `setProperty(String)`. O valor da propriedades do gerador de credencial é usado apenas quando um valor for especificado para o campo **Classe do Gerador de Credencial**.

*Propriedades Customizadas do Domínio de Serviço de Catálogo:*

É possível editar adicionalmente a configuração do domínio do serviço de catálogo ao definir as propriedades customizadas.

Para visualizar essa página do console administrativo, clique em **Administração do Sistema > WebSphere eXtreme Scale > Domínios do Serviço de Catálogo > Propriedades Customizadas**. Para criar uma nova propriedade customizada, clique em **Novo**.

*Nome:*

Especifica o nome da propriedade customizada para o domínio de serviço de catálogo.

*Valor:*

Especifica um valor para a propriedade customizada para o domínio de serviço de catálogo.

## **Configurando Servidores de Contêiner no WebSphere Application Server**

Configure os servidores de contêiner no WebSphere Application Server usando um arquivo de propriedades do servidor e o arquivo XML da política de implementação integrados em um módulo aplicativo Java EE. Os servidores de contêiner param e iniciam quando o aplicativo é interrompido e iniciado.

### **Antes de Iniciar**

Configure um domínio de serviço de catálogo. Consulte “Criando Domínios do Serviço de Catálogo no WebSphere Application Server” na página 252 para obter mais informações.

### **Sobre Esta Tarefa**

Para criar servidores de contêiner no WebSphere Application Server, os arquivos XML de configuração do WebSphere eXtreme Scale devem ser integrados para criar os servidores de contêiner dentro do módulo aplicativo.

### **Procedimento**

1. Identifique os servidores de aplicativos nos quais deseja implementar o aplicativo Java EE que contém as definições do servidor de contêiner do WebSphere eXtreme Scale. Verifique se os perfis do servidor de aplicativos de destino foram aumentados com o perfil do WebSphere eXtreme Scale. Em um ambiente de produção, não posicione os servidores que forem usados para servidores de contêiner junto com os servidores de catálogos. Consulte “Criando e Alterando Perfis para o WebSphere eXtreme Scale” na página 182 para obter informações adicionais.
2. Configure um arquivo de propriedades do servidor e inclua o arquivo de propriedades do servidor no caminho de classe para cada nó do servidor de aplicativos de destino. Consulte o Arquivo de Propriedades do Servidor para obter informações adicionais.
3. Inclua o arquivo descritor XML do ObjectGrid e o arquivo XML da política de implementação no módulo aplicativo. Consulte o “Configurando os Aplicativos do WebSphere Application Server para Iniciar Automaticamente os Servidores de Contêiner” para obter informações adicionais.

### **Configurando os Aplicativos do WebSphere Application Server para Iniciar Automaticamente os Servidores de Contêiner:**

Os servidores de contêiner em um ambiente do WebSphere Application Server iniciam automaticamente quando é iniciado um módulo que possui os arquivos XML do eXtreme Scale.

## Antes de Iniciar

WebSphere Application Server e WebSphere eXtreme Scale devem estar instalados e você deve estar apto a acessar o console administrativo do WebSphere Application Server.

## Sobre Esta Tarefa

Os aplicativos Java Platform, Enterprise Edition possuem regras de carregador de classe complexas que complicam muito o carregamento das classes quando uma grade de dados compartilhados é usada dentro de um servidor Java EE. Um aplicativo do Java EE normalmente é um arquivo Archive Corporativo (EAR). O arquivo EAR contém um ou mais módulos do Enterprise JavaBeans (EJB) ou archive web (WAR) implementados.

O WebSphere eXtreme Scale procura cada início de módulo e consulta arquivos XML do eXtreme Scale. Se o serviço de catálogo detectar que um módulo inicia com os arquivos XML, o servidor de aplicativos será registrado como um servidor de contêiner Java Virtual Machine (JVM). Ao registrar os servidores de contêiner com o serviço de catálogo, o mesmo aplicativo pode ser implementado em grades de dados de dados diferentes, mas não pode ser usado como uma grade de dados única pelo serviço de catálogo. O serviço de catálogo não é referente às células, grades ou grades dinâmicas. Uma única grade de dados pode se propagar por diversas células, se necessário.

## Procedimento

1. Compacte seu arquivo EAR para ter módulos que incluem os arquivos XML do eXtreme Scale na pasta META-INF. O WebSphere eXtreme Scale detecta a presença dos arquivos `objectGrid.xml` e `objectGridDeployment.xml` no META-INF dos módulos EJB e da WEB quando eles iniciam. Se apenas um arquivo `objectGrid.xml` for localizado, a JVM será assumida para ser o cliente. Caso contrário, assume-se que esta JVM atua como um contêiner para a grade de dados que é definida no arquivo `objectGridDeployment.xml`.

É necessário utilizar os nomes corretos para estes arquivos XML. Os nomes do arquivo fazem distinção entre maiúscula e minúscula. Se os arquivos não estiverem presentes, então o contêiner não será iniciado. É possível verificar no arquivo `systemout.log` se há mensagens que indicam que os shards estão sendo localizados. Um módulo EJB ou módulo WAR utilizando o eXtreme Scale deve ter arquivos XML do eXtreme Scale em seu diretório META-INF.

Os arquivos XML do eXtreme Scale incluem:

- Um arquivo descritor XML do ObjectGrid, denominado `objectGrid.xml`. Consulte o Arquivo XML descritor do ObjectGrid para obter informações adicionais.
- Um arquivo descritor XML de implementação chamado `objectGridDeployment.xml`. Consulte o Arquivo Descritor XML de Política de Implementação para obter informações adicionais.
- (Opcional) Um arquivo XML do descritor de metadados da entidade, se entidades forem usadas. O nome do arquivo `entity.xml` deve corresponder ao nome que é especificado no arquivo `objectGrid.xml`. Consulte o Arquivo Descritor XML de Metadados de Entidade para obter informações adicionais.

O tempo de execução detecta estes arquivos e, em seguida, entre em contato com o serviço de catálogo para informá-lo que outro contêiner está disponível para hospedar shards para esse eXtreme Scale.



**Dica:** Se seu aplicativo tiver entidades e você estiver planejando usar um servidor de contêiner, configure o valor **minSyncReplicas** para 0 no arquivo descritor XML de implementação. Caso contrário, é possível ver uma das mensagens a seguir no arquivo `SystemOut.log`, pois o posicionamento não pode ocorrer até que outro servidor seja iniciado para atender a política `minSyncReplica`:

CWPRJ1005E: Erro ao resolver associação de entidade. Entity=entity\_name, association=association\_name.

CW0BJ3013E: O repositório EntityMetadata não está disponível. Tempo Limite limite atingido ao tentar registrar a entidade: entity\_name.

## 2. Implementar e iniciar seu aplicativo.

O contêiner inicia automaticamente quando o módulo é iniciado. O serviço de catálogo inicia para colocar primários e réplicas de partição (shards) o quanto antes. Esta disposição ocorre imediatamente, a menos que você configure o ambiente para atrasar o posicionamento. Para obter informações adicionais, consulte “Controlando o Posicionamento” na página 419.

## O que Fazer Depois

Aplicativos dentro da mesma célula que os contêineres podem se conectar a estas grades de dados utilizando um método `ObjectGridManager.connect(null, null)` e, em seguida, chama o método `getObjectGrid(ccc, "object grid name")`. Os métodos `connect` ou `getObjectGrid` podem ser bloqueados até que os contêineres tenham colocado os shards, mas este bloqueio é um problema apenas quando a grade de dados está iniciando.

## ClassLoaders

Quaisquer plug-ins ou objetos armazenados em um eXtreme Scale são carregados em um determinado utilitário de carga de classes. Dois módulos EJB no mesmo EAR podem incluir esses objetos. Os objetos são os mesmos, porém são carregados usando `ClassLoaders` diferentes. Se o aplicativo A armazena um objeto `Person` em um mapa que é local para o servidor, o aplicativo B recebe um `ClassCastException` se ele tentar ler esse objeto. Esta exceção ocorre porque o aplicativo B carregou o objeto `Person` em um utilitário de carga de classe diferente.

Uma abordagem para resolver este problema é ter um módulo raiz que contém os plug-ins e objetos necessários que estão armazenados no eXtreme Scale. Cada módulo que utiliza o eXtreme Scale deve referenciar tal módulo para suas classes. Outra resolução é colocar estes objetos compartilhados em um arquivo JAR do utilitário que está em um carregador de classe comum compartilhado por ambos os módulos e aplicativos. Os objetos também podem ser colocados nas classes do `WebSphere` ou no diretório `lib/ext`, no entanto, esse posicionamento complica a implementação.

Os módulos EJB em um arquivo EAR normalmente compartilham o mesmo `ClassLoader` e não são afetados por este problema. Cada módulo WAR possui seu próprio `ClassLoader` e é afetado por este problema.

## Conectando-se apenas ao cliente da grade de dados

Se a propriedade **catalog.services.cluster** for definida na célula, nas propriedades customizadas de nó ou servidor, qualquer módulo no arquivo EAR poderá chamar o método `ObjectGridManager.connect(ServerFactory.getServerProperties().getCatalogServiceBootstrap(), null, null)` para

obter um `ClientClusterContext`. O módulo pode também chamar o método `ObjectGridManager.getObjectGrid(ccc, "grid name")` para obter uma referência para a grade de dados. Se qualquer objeto do aplicativo for armazenado em Mapas, verifique se esses objetos estão presentes em um `ClassLoader` comum.

Os clientes Java ou clientes fora da célula podem conectar-se usando a porta IIOP de autoinicialização do serviço de catálogo. No WebSphere Application Server, o gerenciador de implementação hospeda o serviço de catálogo por padrão. O cliente pode então obter um `ClientClusterContext` e a grade de dados nomeada.

### Entity Manager


Com o gerenciador de entidades, as tuplas são armazenadas nos mapas em vez de nos objetos de aplicativos, resultando em menos problemas do carregador de classe. Entretanto, os plug-ins podem ser um problema. Observe também que o arquivo descritor XML do `ObjectGrid` de substituição do cliente sempre é necessário ao se conectar a uma grade de dados que possui entidades definidas: `ObjectGridManager.connect("host:port[,host:port], null, objectGridOverride)` ou `ObjectGridManager.connect(null, objectGridOverride)`.

---

## Configurando o IBM eXtremeMemory e o IBM eXtremeIO

Ao configurar o eXtremeMemory, é possível armazenar os objetos na memória nativa em vez de armazenar no heap Java. Configurar o eXtremeMemory ativa o eXtremeIO como um novo mecanismo de transporte. Movendo objetos para fora do heap Java, é possível evitar pausas da coleta de lixo, levando ao desempenho mais constante e tempos de resposta atribuíveis.

### Antes de Iniciar

-  O eXtremeIO e o eXtremeMemory são suportados em sistemas Linux x86 de 64 bits que usam apenas um SDK de 64 bits.
- Você deve usar conjuntos de mapas que tenham todos os mapas configurados com os modos de cópia `COPY_TO_BYTES` ou `COPY_TO_BYTES_RAW`. Se algum mapa dentro do conjunto de mapas não for configurado usando um desses modos de cópia, os objetos serão armazenados no heap Java e o Object Request Broker (ORB) será usado.
- O eXtremeIO e o eXtremeMemory não podem ser usados nos seguintes cenários de configuração:
  - Quando estiver usando servidores de contêiner que estão em execução em um ambiente do WebSphere Application Server.
  - Quando estiver usando plug-ins do evictor customizados.
  - Quando estiver usando índices compostos.
  - Quando estiver usando os carregadores write-behind integrados.
  - Quando estiver usando a interface `ReplicationMapListener` para criar uma implementação de um listener de evento para mapas do lado do cliente que estiverem no modo de replicação.

### Sobre Esta Tarefa

A JVM conta com a heurística de uso para coletar, compactar e expandir a memória do processo. O coletor de lixo conclui estas operações. No entanto, executar a coleta de lixo tem um custo associado. O custo de executar a coleta de lixo aumenta conforme o tamanho do heap Java e o número de objetos no aumento da grade de dados. A JVM fornece heurísticas diferentes para diferentes

casos de uso e objetivos: rendimento ideal, tempo de pausa ideal, de geração, balanceado e em tempo real. Nenhuma heurística é perfeita. Uma única heurística não pode se adequar a todas as configurações possíveis.

O WebSphere eXtreme Scale usa armazenamento em cache de dados, com mapas distribuídos que possuem entradas com um ciclo de vida bem conhecido. Este ciclo de vida inclui as seguintes operações: GET, INSERT, DELETE e UPDATE. Usando estes ciclos de vida de mapa bem conhecidos, o eXtremeMemory e o eXtremeIO podem usar memória de forma mais eficiente do que as heurísticas de uso da JVM.

O diagrama a seguir mostra como o uso de eXtremeMemory leva a tempos de resposta relativos mais consistentes no ambiente. Conforme os tempos de resposta relativos atingem o percentual mais alto, as solicitações que estão usando o eXtremeMemory têm os tempos de resposta relativos inferiores. O diagrama mostra os percentis de 95 a 100.

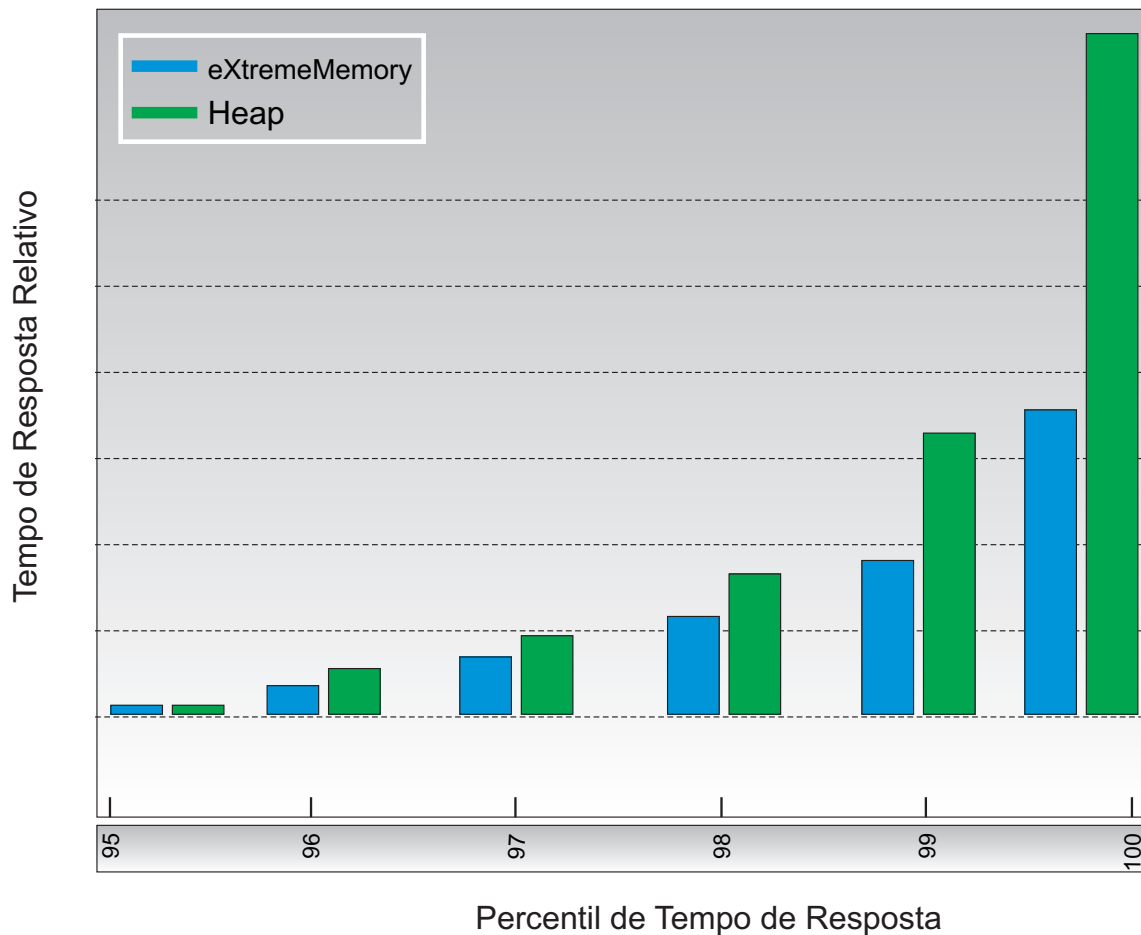


Figura 30. Comparação de eXtremeMemory e Tempos de Resposta de Armazenamento Heap

Quando você estiver usando o eXtremeMemory, o eXtremeIO é usado para comunicação entre os servidores de contêiner. Os objetos são serializados em bytes

no servidor de contêiner. Para ativar o eXtremeIO e eXtremeMemory, configure as propriedades do servidor necessárias em todos os servidores de contêiner na grade de dados e reinicie os servidores.

## Procedimento

1. Opcional: Determine o valor da propriedade **maxXMSize** apropriado a ser usado.
  - a. Em sua configuração existente, determine o tamanho por entrada. Execute o comando **xscmd -c showMapSizes** para determinar este tamanho.
  - b. Calcule o valor de **maxXMSize**. Para obter tamanho total máximo de entradas (*maximum\_total\_size*), multiplique a *size\_per\_entry* \* *maximum\_number\_of\_entries*. Use no máximo 60% de **maxXMSize** para a conta para processamento de metadados. Multiplique *maximum\_total\_size* \* 1.65 para obter o valor de **maxXMSize**.
2. Atualize o arquivo de propriedades do servidor para cada servidor de contêiner na configuração para ativar o novo transporte. As seguintes propriedades do servidor ativam o novo transporte:

### Propriedades Obrigatórias

#### 7.1.1+ enableXM

Quando configurado para true, ativa o IBM eXtremeMemory no servidor e configura o servidor para usar o IBM eXtremeIO para replicação síncrona e assíncrona. As entradas de cache são armazenadas na memória nativa em vez de serem armazenadas no heap Java. Todos os servidores de contêiner na grade de dados devem usar o mesmo valor para a propriedade **enableXM**.

Padrão: false

### Propriedades Sugeridas

#### 7.1.1+ maxXMSize

Define a quantidade máxima de memória, em megabytes, usada pelo servidor para armazenamento do eXtremeMemory.

Padrão: 25% do total de memória no sistema

### Propriedades Opcionais

#### 7.1.1+ maxXIONetworkThreads

Configura o número máximo de encadeamentos para alocar no conjunto de encadeamentos de rede de transporte eXtremeIO.

Padrão: 50

#### 7.1.1+ minXIONetworkThreads

Configura o número mínimo de encadeamentos para alocar no conjunto de encadeamentos de rede de transporte eXtremeIO.

Padrão: 50

#### 7.1.1+ maxXIOWorkerThreads

Configura o número máximo de encadeamentos para alocar no conjunto de encadeamentos de processamento de solicitação de transporte eXtremeIO.

Padrão: 128

#### 7.1.1+ minXIOWorkerThreads

Configura o número mínimo de encadeamentos para alocar no conjunto de encadeamentos de processamento de solicitação de transporte eXtremeIO.

Padrão: 128

**7.1.1+ xioChannel.xioContainerTCPNonSecure.Port**

Especifica o número de porta listener não segura do eXtremeIO no servidor. Se esse valor não for configurado, uma porta temporária será usada. Esta propriedade é usada somente quando a propriedade **transportType** é configurada como TCP/IP.

**7.1.1+ xioChannel.xioContainerTCPSecure.Port**

Especifica o número de porta SSL do eXtremeIO no servidor. Esta propriedade é usada somente quando a propriedade **transportType** é configurada como SSL-Supported ou SSL-Required.

3. Reinicie os servidores de contêiner para começar a usar o novo mecanismo de transporte. Para obter mais informações, consulte “Iniciando e Parando Servidores Independentes” na página 387 e “Iniciando e Parando Servidores em um Ambiente do WebSphere Application Server” na página 401.

---

## Configurando Diversas Topologias do Datacenter

Com a replicação assíncrona multimestre, um conjunto de domínios de serviço de catálogo é vinculado. Os domínios de serviço de catálogo conectados são então sincronizado usando a replicação sobre os links. É possível definir os links usando arquivos de propriedades, no tempo de execução com programas Java Management Extensions (JMX) ou com utilitários de linha de comandos. O conjunto de links atuais para um domínio é armazenado no serviço de catálogo. É possível incluir e remover links sem reiniciar o domínio de serviço de catálogo que hospeda a grade de dados.

### Antes de Iniciar

- Consulte “Planejando Diversas Topologias do Datacenter” na página 35 para obter mais informações sobre as topologias de replicação multimestre e considerações de design. É possível configurar links entre domínios do serviço de catálogo com o arquivo de propriedades do servidor para formar a topologia durante a inicialização do servidor. Também é possível configurar links no tempo de execução.
- Se estiver usando os carregadores na sua topologia de replicação multimestre, você deve planejar como manterá os dados precisos entre os datacenters. As abordagens que podem ser usadas variam dependendo da topologia que você está usando. Para obter informações adicionais, consulte “Considerações Sobre o Carregador em uma Topologia Multimestre” na página 40.

### Procedimento

- Defina links no arquivo de propriedades para o servidor de catálogos de cada domínio de serviço de catálogo na topologia, para fins de autoinicialização. Consulte Arquivo de Propriedades do Servidor para obter mais informações sobre como definir esse arquivo para o servidor de catálogos.

**Importante:** Os nomes da propriedade fazem distinção entre maiúsculas e minúsculas.

#### Nome do Domínio Local

Especifique o nome do domínio de serviço de catálogo para o servidor de catálogos atual, como domínio A:

```
domainName=A
```

#### Uma lista opcional de nomes de domínio externo:

Especifique os nomes dos domínios de serviço de catálogo ao qual você deseja vincular na topologia de replicação multimestre, como o domínio B:

```
foreignDomains=B
```

#### **Uma lista opcional de terminais para os nomes de domínio externo:**

Especifica as informações de conexão para os servidores de catálogos dos domínios externos, como domínio B:

```
B.endPoints=hostB1:2809, hostB2:2809
```

Se um domínio externo tiver diversos servidores de catálogo, especifique todos eles.

- Use o utilitário **xscmd** ou a programação JMX para incluir ou remover links no tempo de execução.

Os links para um domínio são mantidos no serviço de catálogo na memória replicada. Este conjunto de links pode ser alterado a qualquer momento pelo administrador sem exigir o reinício deste domínio ou de qualquer outro domínio. O utilitário **xscmd** inclui várias opções para trabalhar com links.

O utilitário **xscmd** se conecta a um serviço de catálogo e, assim, a um domínio de serviço de catálogo único. Portanto, o utilitário **xscmd** pode ser usado para criar e destruir links entre o domínio ao qual ele se conecta e qualquer outro domínio.

Use a linha de comandos para criar um link, por exemplo:

```
xscmd -c establishLink -cep host:2809 -fd dname -fe fdHostA:2809,fdHostB:2809
```

O comando estabelece um novo link entre o domínio local e o domínio externo denominado dname. O serviço de catálogo dname está em execução no fdHostA:2809 e no fdHostB:2809. O domínio de serviço de catálogo local possui um host e uma porta do listener de serviço de catálogo igual a host:2809.

Especifique todos os terminais de serviço de catálogo a partir do domínio externo para que a conectividade de tolerância a falhas com o domínio seja possível. Não use um par host:port único para o serviço de catálogo do domínio do serviço de catálogo externo.

É possível usar qualquer JVM do serviço de catálogo local com **xscmd** e usar a opção **-cep**. Se o servidor de catálogos for hospedado em um gerenciador de implementação do WebSphere Application Server, a porta geralmente será 9809.

As portas especificadas para o domínio externo não são portas JMX. Elas são as portas usuais que você usaria para clientes do eXtreme Scale.

Depois de o comando para incluir um novo link ser emitido, o serviço de catálogo instrui todos os contêineres sob seu gerenciamento a começar a replicar para o domínio externo. Um link não é necessário em ambos os lados. Ele é necessário apenas para criar um link em um lado.

Use a linha de comandos para remover um link, por exemplo:

```
xscmd -c dismissLink -cep host:2809 -fd dname
```

O comando conecta ao serviço de catálogo para um domínio e o instrui a parar de replicar para um domínio específico. Um link precisa ser indeferido apenas de um lado.

## **Link entre Dois Domínios de Serviço de Catálogo**

Suponha que você queira definir uma configuração com dois domínios envolvendo os domínios A e B do serviço de catálogo.

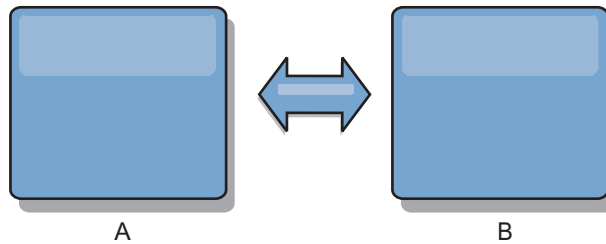


Figura 31. Link Entre Domínios de Serviço de Catálogo

Aqui é apresentado o arquivo de propriedades do servidor para o servidor de catálogos no domínio A:

```
domainName=A
foreignDomains=B
B.endPoints=hostB1:2809, hostB2:2809
```

Aqui é apresentado o arquivo de propriedades do servidor para o servidor de catálogos no domínio B. Observe a semelhança entre os dois arquivos de propriedades.

```
domainName=B
foreignDomains=A
A.endPoints=hostA1:2809,hostA2:2809
```

Depois que os dois domínios forem iniciados, quaisquer grades de dados com as seguintes características são replicadas entre os domínios.

- Ter um serviço de catálogo privado com um nome de domínio exclusivo
- Ter o mesmo nome de grade de dados que outras grades no domínio
- Ter o mesmo número de partições de outras grades de dados no domínio
- Ser uma grade de dados FIXED\_PARTITION (grades de dados PER\_CONTAINER não podem ser replicadas)
- Ter o mesmo número de partições (pode o não ter o mesmo número e tipos de réplicas)
- Ter os mesmos tipos de dados sendo replicados de outras grades de dados no domínio
- Ter o mesmo nome do conjunto de mapas, nomes de mapas e modelos de mapas dinâmicos de outras grades de dados no domínio

A política de replicação de um domínio de serviço de catálogo é ignorada.

O exemplo anterior mostra como configurar cada domínio para ter um link para o outro domínio, mas é necessário apenas definir um link em uma direção. Este fato é especialmente útil em topologias hub e spoke, permitindo uma configuração muito mais simples. O arquivo de propriedades do hub não requer atualizações conforme os spokes são incluídos e cada arquivo do spoke precisa apenas incluir informações do hub. De forma semelhante, uma topologia em anel requer que cada domínio tenha apenas um link para o domínio anterior e o próximo no anel.

Exemplo: Topologia Hub e Spoke



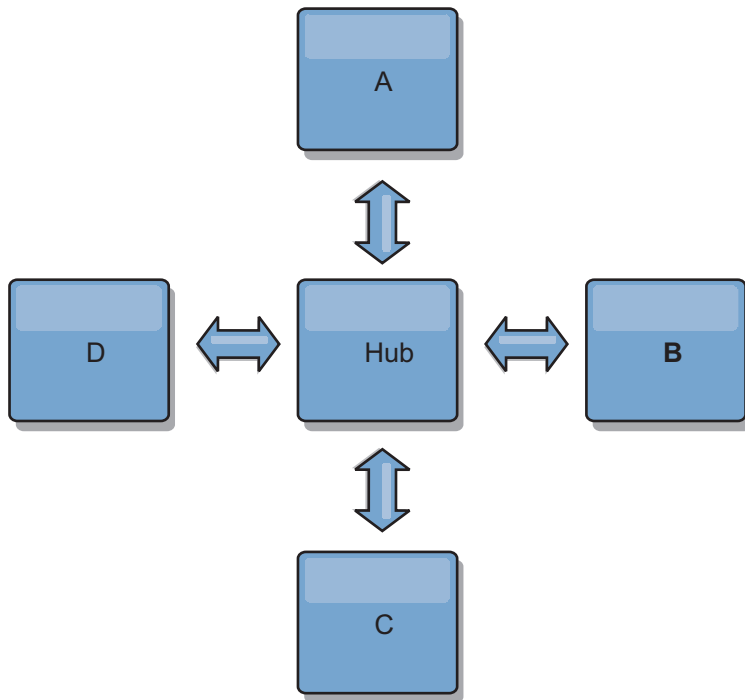


Figura 32. Topologia Hub e Spoke

O hub e quatro spokes (domínios A, B, C e D) possuem arquivos de propriedades do servidor como os seguintes exemplos.

```
domainName=Hub
```

O spoke A possui as seguintes propriedades do servidor:

```
domainName=A
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

O spoke B possui as seguintes propriedades do servidor:

```
domainName=B
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

O spoke C possui as seguintes propriedades do servidor:

```
domainName=C
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

O spoke D possui as seguintes propriedades:

```
domainName=D
foreignDomains=Hub
Hub.endPoints=hostH1:2809, hostH2:2809
```

## O que Fazer Depois

É possível fornecer um árbitro de colisão customizado para resolver colisões entre os domínios de serviço de catálogo. Consulte o Desenvolvendo Árbitros Customizados para a Replicação Multimestre para obter informações adicionais.

---

## Configurando Portas

O WebSphere eXtreme Scale é um cache distribuído que requer a abertura de portas para se comunicar com a pilha Object Request Broker (ORB) e Protocolo de Controle de Transmissões (TCP) entre Java Virtual Machine (JVM) e outros servidores.

### Configurando Portas no Modo Independente

É possível configurar as portas necessárias para servidores e clientes em uma implementação do eXtreme Scale usando parâmetros da linha de comandos, arquivos de propriedades ou programaticamente. A maioria dos exemplos incluídos nas seções a seguir descreve parâmetros da linha de comandos para o script **startOgServer**. As opções de configuração equivalentes também podem ser configuradas nos arquivos de propriedades usando a API do servidor integrado ou a API do cliente.

#### Procedimento

##### 1. Iniciar os Terminais do Serviço de Catálogo

O WebSphere eXtreme Scale usa o IIOP para comunicação entre Java virtual machines. As JVMs do serviço de catálogo são os únicos processos que requerem uma configuração explícita das portas para os serviços IIOP e para as portas de serviços de grupo. Outros processos alocam portas dinamicamente.

A porta do cliente e a porta equivalente são usadas para a comunicação entre serviços de catálogo em um domínio do serviço de catálogo. Para especificar a porta do cliente e porta equivalente, use a seguinte opção da linha de comandos:

**-catalogServiceEndpoints <serverName:hostName:clientPort:peerPort>**

No contêiner, referencia o host e a porta do Object Request Broker (ORB) no serviço de catálogo. Cada atributo é definido conforme a seguir:

##### **serverName**

Especifica o nome para identificar o processo que você está ativando.

##### **hostName**

Especifica o nome do host para o computador onde o servidor é ativado.

##### **clientPort**

Especifica a porta que é usada para comunicação de serviço de catálogo peer.

##### **peerPort**

Este valor é o mesmo que haManagerPort. Especifica a porta que é usada para comunicação de serviço de catálogo peer.

O exemplo a seguir inicia o servidor de catálogos cs1, que está no mesmo domínio de serviço de catálogo que os servidores cs2 e cs3:

```
startOgServer.bat|sh cs1 -catalogServiceEndpoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
```

Os terminais de serviço de catálogo também podem ser configurados com a propriedade de servidor catalogServiceEndpoints. A porta do listener do Object Request Broker (ORB) é usada para comunicação entre os serviços de catálogo, em um domínio de serviço de catálogo, e para comunicação entre serviços de

catálogos e servidores de contêiner e clientes. Para especificar a porta do listener e o host do listener, use as seguintes opções da linha de comandos:

**-listenerHost <nome do host>**

Especifica o nome do host ao qual o Object Request Broker (ORB) se conecta para comunicação com o Internet Inter-ORB Protocol (IIOP). O valor deve ser um nome de domínio ou endereço IP completo. Se sua configuração envolver diversas placas de rede, configure o host e a porta do listener para permitir que o Object Request Broker na JVM conheça o endereço IP para o qual se ligar. Se você não especificar qual endereço IP usar, sintomas tais como tempos limites de conexão, falhas de API incomuns e clientes que parecem ser interrompidos podem ocorrer.**Padrão:** localhost

**-listenerPort <porta>**

Especifica o número da porta ao qual o Object Request Broker (ORB) se conecta. Esta configuração define que contêineres e clientes se comunicam com o serviço de catálogo por meio do ORB. No WebSphere Application Server, a listenerPort é herdada pela configuração de porta BOOTSTRAP\_ADDRESS. Esta propriedade se aplica ao servidor de contêiner e ao serviço de catálogo.**Padrão:** 2809

A porta do listener e o host do listener também podem ser configurados usando as propriedades de servidor listenerHost e listenerPort.

A porta de serviço JMX é usada para comunicação a partir de clientes JMX. Para especificar a porta de serviço JMX, use a seguinte opção da linha de comandos:

**-JMXServicePort <porta>**

Especifica o número da porta na qual o servidor MBean recebe a comunicação com Java Management Extensions (JMX). Você deve usar um número da porta diferente para cada JVM na configuração. Se desejar usar JMX/RMI, especifique explicitamente **JMXServicePort** e o número da porta, mesmo se desejar usar o valor de porta padrão. Esta propriedade se aplica ao servidor de contêiner e ao serviço de catálogo.

**Padrão:** 1099

A porta de serviço JMX também pode ser configurada usando a propriedade de servidor JMXServicePort. Quando a segurança está ativada, uma porta Secure Socket Layer (SSL) também é necessária. Para especificar a porta SSL, use a seguinte opção da linha de comandos:

```
-jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>
```

```
./startOgServer.sh cs1 -listenerHost hostA -listenerPort 2809 -catalogServiceEndPoints  
cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

*Figura 33. Exemplo do Uso da Linha de Comandos.* Inicie o primeiro servidor de catálogos no hostA. A seguir há um exemplo do comando:

Inicie o segundo servidor de catálogos no hostB. A seguir há um exemplo do comando:

```
./startOgServer.sh cs2 -listenerHost hostB -listenerPort 2809  
-catalogServiceEndPoints cs1:hostA:6601:6611,cs2:hostB:6601:6611
```

2. Inicie os terminais do servidor de contêiner.

O seguinte comando inicia uma JVM de contêiner para uso com o serviço de catálogo de exemplo:

```
./startOgServer.sh c0  
-catalogServiceEndPoints hostA:2809,hostB:2809
```

O servidor de contêiner Java Virtual Machines usa duas portas. A porta do gerenciador de HA é usada para comunicação interna entre os servidores de contêiner e servidores de catálogos equivalentes. A porta do listener é usada para comunicação IIOP entre servidores de contêiner, servidores de catálogos e clientes peer. O host do listener é usado para ligar o ORB a um adaptador de rede específico. Se não for especificada, ambas as portas serão selecionadas dinamicamente. No entanto, se desejar configurar portas explicitamente, tal como em um ambiente de firewall, será possível usar uma opção da linha de comandos para especificar a porta do ORB. Para especificar a porta do listener e o host do listener, use as seguintes opções da linha de comandos:

```
-listenerHost <nome_do_host>  
-listenerPort <porta>
```

A porta do listener e o host do listener também podem ser configurados usando a propriedade do servidor `listenerHost` e `listenerPort`.

Para especificar a porta do gerenciador JMX, use a seguinte opção da linha de comandos:

**-haManagerPort <porta>**

Sinônimo de porta de peer. Especifica o número da porta usada pelo gerenciador de alta disponibilidade. Se essa propriedade não for configurada, o serviço de catálogo gera uma porta disponível automaticamente. Essa propriedade se aplica ao servidor de contêiner e ao serviço de catálogo. (Necessário somente para ambientes do WebSphere Application Server.)

A porta do gerenciador de HA também pode ser configurada usando a propriedade de servidor `HAManagerPort`.

Quando a segurança está ativada, uma porta Secure Socket Layer (SSL) também é necessária. Para especificar a porta SSL, use a seguinte opção da linha de comandos:

```
-jvmArgs -Dcom.ibm.CSI.SSLPort=<sslPort>
```

### 3. Inicie os terminais do cliente.

Os clientes devem conhecer somente os terminais do listener do serviço de catálogo. Os clientes recuperam terminais para o servidor de contêiner Java Virtual Machines, que são os Java Virtual Machines que mantém os dados, automaticamente a partir do serviço de catálogo. Para conectar-se ao serviço de catálogo no exemplo anterior, o cliente deve transmitir a lista a seguir de pares `host:port` para a API de conexão:

```
hostA:2809,hostB:2809
```

O cliente também pode receber os retornos de chamadas a partir dos servidores de contêiner ao usar a API do DataGrid. Esses retornos de chamada se comunicam usando o IIOP com a porta do listener do ORB. Para especificar a porta e o adaptador de rede para receber retornos de chamada, configure as propriedades `listenerHost` e `listenerPort` no arquivo de propriedades do cliente.

Quando a segurança está ativada, uma porta Secure Socket Layer (SSL) também é necessária. Para especificar a porta SSL, use a seguinte propriedade de sistema ao iniciar o processo do cliente:

```
-Dcom.ibm.CSI.SSLPort=<sslPort>
```

## Configurando Portas em um Ambiente do WebSphere Application Server

Os serviços de catálogo, os servidores de contêiner e clientes do WebSphere eXtreme Scale, quando estão em execução nos processos do WebSphere Application Server, use portas e serviços já definidos para o processo.

### Sobre Esta Tarefa

As seções a seguir explicam os detalhes relacionados ao uso de portas em sua implementação.

#### 1. Terminais do Serviço de Catálogo

Os serviços de catálogo do WebSphere eXtreme Scale são executados em qualquer processo do WebSphere Application Server e são configurados usando o console administrativo ou usando as tarefas administrativas. Todas as portas são herdadas pelo processo, exceto a porta do cliente que é configurada explicitamente. Para obter detalhes sobre quais portas são usadas pelo serviço de catálogo, consulte o “Planejamento para Portas de Rede” na página 62. Para obter detalhes sobre como configurar um domínio do serviço de catálogo, consulte Serviço de Catálogo de Alta Disponibilidade.

#### 2. Terminais do Servidor de Contêiner

Os servidores de contêiner do WebSphere eXtreme Scale são hospedados nos módulos do Java EE. Os servidores de contêiner usam as portas definidas para o processo do servidor de aplicativos. Para obter detalhes sobre quais portas são usadas pelo serviço de contêiner, consulte “Planejamento para Portas de Rede” na página 62. Para obter detalhes sobre como iniciar um contêiner dentro de um módulo Java EE como um Enterprise JavaBeans™ (EJB) ou módulo da web, consulte “Configurando os Aplicativos do WebSphere Application Server para Iniciar Automaticamente os Servidores de Contêiner” na página 270.

#### 3. Terminais do Cliente

Os clientes do WebSphere eXtreme Scale são hospedados nos módulos da web ou EJB do Java EE.

Os clientes se conectam programaticamente ao domínio do serviço de catálogo usando a API `ObjectGridManager.connect()`. Ao se conectar a um domínio do serviço de catálogo hospedado dentro da mesma célula, a conexão do cliente localizará automaticamente o domínio do serviço de catálogo padrão usando a seguinte chamada de API no `ObjectGridManager`:

```
connect(securityProps, overrideObjectGridXML)
```

Se o domínio do serviço de catálogo padrão for hospedado remotamente (fora da célula), os terminais de serviço de catálogo deverão ser especificados usando o seguinte método na API `ObjectGridManager`:

```
connect(catalogServerAddresses, securityProps, overrideObjectGridXml)
```

Se o domínio do serviço de catálogo padrão estiver definido na célula, a API `CatalogServerProperties` poderá ser usada para recuperar os endereços do servidor de catálogo. A tarefa administrativa `XSDomainManagement` também pode ser usada para recuperar quaisquer terminais de domínio do serviço de catálogo configurados.

## Servidores com Diversas Placas de Rede

É possível executar processos do eXtreme Scale em um servidor com mais de uma placa de rede.

Se um servidor ou cliente estiver em execução em um servidor que contém mais de uma placa de rede, você deverá especificar a porta de rede e o nome do host na configuração do eXtreme Scale para ligar a uma placa especificada. Se essa configuração não for especificada, o tempo de execução do eXtreme Scale escolherá automaticamente uma, o que pode resultar em falhas de conexão ou diminuir o desempenho.

Para servidores de catálogos ou de contêiner, você deve configurar o host e a porta do listener de uma das seguintes maneiras:

- propriedades do servidor
- Parâmetro da linha de comandos no script `startOgServer.sh | bat`.

Para clientes, a linha de comandos não pode ser usada e as propriedades do cliente devem ser usadas.

---

## Configurando Transportes

Os transportes permitem a troca de objetos e dados entre os processos do servidor diferente em sua configuração.

### Sobre Esta Tarefa

O mecanismo de transporte principal é o Object Request Broker (ORB). Este mecanismo armazena em cache as entradas no heap Java.

**7.1.1+** Usar o ORB como o mecanismo de transporte é necessário nos seguintes cenários de configuração:

- Quando estiver usando um sistema diferente do Linux x86 de 64 bits.
- Quando estiver usando servidores de contêiner que estão em execução em um ambiente do WebSphere Application Server.
- Quando estiver usando os plug-ins evictor ou índices compostos.

**7.1.1+** Se estiver usando o eXtremeMemory, um novo transporte chamado eXtremeIO é usado. Com o eXtremeMemory, as entradas de cache são armazenadas na memória nativa. A memória nativa não passa pela coleta de lixo, o que proporciona um desempenho mais constante e tempos de resposta previsíveis. Os objetos são serializados em bytes no servidor de contêiner. Para obter informações adicionais, consulte “Configurando o IBM eXtremeMemory e o IBM eXtremeIO” na página 273.

## Configurando Object Request Brokers

O Object Request Broker (ORB) é usado pelo WebSphere eXtreme Scale para se comunicar por meio da pilha TCP. Use o arquivo `orb.properties` para transmitir as propriedades que são usadas pelo ORB para modificar o comportamento de transporte da grade de dados. Nenhuma ação é necessária para usar o ORB fornecido pelo WebSphere eXtreme Scale ou WebSphere Application Server para seus servidores WebSphere eXtreme Scale.

### Configurando o Object Request Broker em um Ambiente do WebSphere Application Server

É possível usar o WebSphere eXtreme Scale com aplicativos que usam o Object Request Broker (ORB) diretamente em ambientes do WebSphere Application Server ou do WebSphere Application Server Network Deployment.

## Procedimento

1. Nomeie seus servidores de aplicativos apropriadamente.  
Você não pode ter servidores em um ambiente do WebSphere Application Server com o mesmo nome quando os servidores estiverem utilizando o ORB para se comunicarem uns com os outros. É possível resolver essa restrição especificando a propriedade de sistema `-Dcom.ibm.websphere.orb.uniqueServerName=true` para os processos que têm o mesmo nome. Por exemplo, quando servidores com o nome `server1` são usados como um domínio de serviço de catálogo em cada nó ou quando diversos agentes de nó são usados para formar um domínio de serviço de catálogo.
2. Ajuste as propriedades do ORB dentro da configuração do WebSphere Application Server.  
Consulte “Propriedades do ORB” na página 482 para obter mais informações sobre as propriedades que podem ser ajustadas. Dependendo da propriedade, é possível alterar uma configuração no console administrativo ou no arquivo `was_rootproperties/orb.properties`.
3. Se estiver usando diversas placas da interface de rede, o valor `ORB_LISTENER_ADDRESS` deverá ser configurado no painel portas no console administrativo do WebSphere Application Server. Repita essa etapa para cada servidor de aplicativos na configuração.
  - a. Para um servidor de aplicativos, clique em **Servidores > Servidores de Aplicativos > *server\_name***. Em Comunicações, clique em **Portas**. O painel Portas é exibido para o servidor especificado.
  - b. Clique em **Detalhes** e edite o valor `ORB_LISTENER_ADDRESS`.
  - c. Insira o endereço IP no campo **Host**. Este valor deve ser um endereço privado para um ambiente de diversas interfaces de rede.  
  
**Nota:** Os nomes de host DNS não são suportados para o valor `ORB_LISTENER_ADDRESS`.
  - d. Insira o número da porta no campo **Porta**. O número da porta especifica a porta para a qual o serviço é configurado para aceitar solicitações de clientes. O valor da porta é usado com o nome do host.

## O que Fazer Depois

**7.1.1+** É possível usar a ferramenta **wxsLogAnalyzer** para verificar as definições do ORB em seu ambiente. Consulte o “Analisando Dados de Log e de Rastreo” na página 529 para obter informações adicionais.

## Configurando o Object Request Broker com Processos do WebSphere eXtreme Scale Independentes

É possível usar o WebSphere eXtreme Scale com aplicativos que usam o Object Request Broker (ORB) diretamente em ambientes que não contêm o WebSphere Application Server ou o WebSphere Application Server Network Deployment.

## Antes de Iniciar

Se você usar o ORB dentro do mesmo processo do eXtreme Scale ao executar aplicativos ou outros componentes e estruturas que não estejam incluídos com o eXtreme Scale, poderá ser necessário concluir tarefas adicionais para garantir que o eXtreme Scale seja executado corretamente no seu ambiente.



## Sobre Esta Tarefa

Inclua a propriedade **ObjectGridInitializer** no arquivo `orb.properties` para inicializar o uso do ORB em seu ambiente. Use o ORB para ativar a comunicação entre os processos do eXtreme Scale e outros processos que estão em seu ambiente.

### Procedimento

1. A instalação independente não inclui um arquivo `orb.properties`. O arquivo `orb.properties` deve ser colocado no diretório `java/jre/lib`. Para obter descrições das propriedades e configurações, consulte “Propriedades do ORB” na página 482.
2. No arquivo `orb.properties`, digite a seguinte linha e salve suas mudanças:  
`org.omg.PortableInterceptor.ORBInitializerClass.com.ibm.ws.objectgrid.corba.ObjectGridInitializer`

### Resultados

O eXtreme Scale inicializa corretamente o ORB e coexiste com outros aplicativos para os quais o ORB está ativado.

Para usar uma versão customizada do ORB com o eXtreme Scale, consulte “Configurando um Object Request Broker Customizado”.

### O que Fazer Depois

**7.1.1+** É possível usar a ferramenta **xsLogAnalyzer** para verificar as definições de ORB em seu ambiente. Consulte o “Analisando Dados de Log e de Rastreamento” na página 529 para obter informações adicionais.

### Configurando um Object Request Broker Customizado

O WebSphere eXtreme Scale usa o Object Request Broker (ORB) para ativar a comunicação entre processos. Nenhuma ação é necessária para usar o Object Request Broker (ORB) fornecido pelo WebSphere eXtreme Scale ou WebSphere Application Server para seus servidores WebSphere eXtreme Scale. Um esforço pequeno é necessário para usar os mesmos ORBs para seus clientes do WebSphere eXtreme Scale. Se, em vez disso, for necessário usar um ORB customizado, o ORB fornecido com o IBM SDK será uma boa opção, embora você deva configurar o ORB. Os ORBs de outros fornecedores podem ser usados, também com configuração.

### Antes de Iniciar

Determine se está usando o ORB fornecido com o WebSphere eXtreme Scale ou o WebSphere Application Server, o ORB fornecido com o IBM SDK ou um ORB de fornecedor externo.

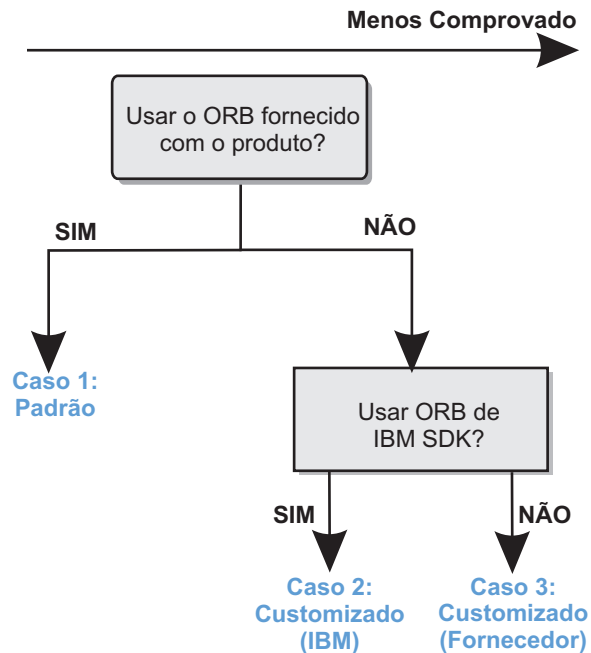


Figura 34. Escolhendo um ORB

É possível tomar decisões separadas para os processos do servidor do WebSphere eXtreme Scale e para os processos do cliente do WebSphere eXtreme Scale. Apesar de o eXtreme Scale suportar Developer Kits da maioria dos fornecedores, recomenda-se que você use o ORB que é fornecido com o eXtreme Scale para seus processos do servidor e do cliente. O eXtreme Scale não suporta o ORB que é fornecido com o Sun Microsystems Java Development Kit (JDK).

### Sobre Esta Tarefa

Familiarize-se com a configuração que é necessária para usar o ORB de sua escolha.

#### Caso 1: ORB Padrão

- Para seus processos do servidor do WebSphere eXtreme Scale, nenhuma configuração é necessária para usar o ORB fornecido com o WebSphere eXtreme Scale ou o WebSphere Application Server.
- Para seus processos do cliente do WebSphere eXtreme Scale, uma configuração mínima de caminho de classe é necessária para usar o ORB fornecido com o WebSphere eXtreme Scale ou o WebSphere Application Server.

#### Caso 2: ORB Customizado (IBM)

Para configurar seus processos do cliente do WebSphere eXtreme Scale para usar o ORB fornecido com o IBM SDK, consulte as instruções posteriormente neste tópico. É possível usar o ORB IBM se estiver usando o IBM SDK ou outro kit de desenvolvimento. É possível usar o IBM SDK Versão 5 ou posterior.

#### Caso 3: ORB Customizado (Fornecedor externo)

Usar um ORB de fornecedor para seus processos do cliente do WebSphere eXtreme Scale é a última opção testada. Quaisquer problemas que você tiver ao usar os ORBs a partir de fornecedores de software independentes deverão ser reproduzíveis com o IBM ORB e compatíveis com o JRE antes de entrar em contato com o suporte.

O ORB fornecido com o Sun Microsystems Java Development Kit (JDK) não é suportado.

### Procedimento

- Configure seus processos do cliente para usar um dos ORBs padrão (**Caso 1**). Use o seguinte argumento da JVM :

```
-jvmArgs -Djava.endorsed.dirs=default_ORB_directory${pathSeparator}JRE_HOME/lib/endorsed
```

O diretório padrão é: *wxs\_home/lib/endorsed*. Também pode ser necessário atualizar as seguintes propriedades no arquivo *orb.properties*:

```
org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB  
org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton
```

- Configure os processos do cliente ou do servidor para usarem o IBM SDK, Versão 5 (**Caso 2**).
  1. Copie os arquivos do Java archive (JAR) do ORB para um diretório vazio, ou para o *custom\_ORB\_directory*.
    - *ibmorb.jar*
    - *ibmorbapi.jar*
  2. Especifique o diretório *custom\_ORB\_directory* como um diretório endossado nos scripts que iniciam o comando Java.

**Dica:** Se seus comandos Java já especificarem um diretório endossado, uma outra opção será colocar o diretório *custom\_ORB\_directory* sob o diretório endossado existente. Colocando o diretório *custom\_ORB\_directory* sob o diretório endossado existente, a atualização dos scripts não é necessária. Se você decidir atualizar os scripts de qualquer forma, certifique-se de incluir o diretório *custom\_ORB\_directory* como um prefixo para seu argumento `-Djava.endorsed.dirs=` existente, em vez de substituir completamente o argumento existente.

- Atualize scripts para um ambiente independente do eXtreme Scale.  
Edite o caminho para a variável *OBJECTGRID\_ENDORSED\_DIRS* no arquivo *setupCmdLine.bat|sh* para especificar o *custom\_ORB\_directory*. Salve as alterações.
- Atualize scripts quando o eXtreme Scale estiver integrado em um ambiente do WebSphere Application Server.  
Inclua a seguinte propriedade de sistema e os parâmetros no script *startOgServer*:

```
-jvmArgs -Djava.endorsed.dirs=custom_ORB_directory
```
- Atualize os scripts customizados que você usa para iniciar um processo do aplicativo cliente ou um processo do servidor.

```
-Djava.endorsed.dirs=custom_ORB_directory
```

---

## Configurando Clientes

É possível configurar o WebSphere eXtreme Scale para execução em um ambiente independente, ou é possível configurar o eXtreme Scale para execução em um ambiente com o WebSphere Application Server. Para uma implementação do WebSphere eXtreme Scale para selecionar mudanças de configuração no lado da grade do servidor, você deve reiniciar os processos para que essas mudanças entrem em vigor em vez de serem aplicadas dinamicamente. No entanto, no lado do cliente, embora não seja possível alterar as definições de configuração para uma instância de cliente existente, uma nova instância do cliente pode ser criada com as configurações necessárias ao usar um arquivo XML ou fazer isso

programaticamente. Ao criar um cliente, será possível substituir as configurações padrão fornecidas com a configuração do servidor atual.

É possível configurar um cliente do eXtreme Scale das seguintes maneiras, sendo que cada uma delas pode ser realizada com um arquivo XML de substituição do cliente ou programaticamente:

- Configuração XML
- Configuração Programática
- Configuração da Estrutura Spring
- Desativando o Cache Local

É possível substituir os seguintes plug-ins em um cliente:

- **Plug-ins do ObjectGrid**
  - Plug-in TransactionCallback
  - Plug-in ObjectGridEventListener
- **Plug-ins do BackingMap**
  - Plug-in Evictor
  - Plug-in MapEventListener
  - Atributo numberOfBuckets
  - Atributo ttlEvictorType
  - Atributo timeToLive

## Configurando Clientes com a Configuração XML

É possível usar um arquivo XML de configuração do ObjectGrid para alterar as configurações no lado do cliente.

### Sobre Esta Tarefa

Para alterar as configurações em um cliente WebSphere eXtreme Scale, você deve criar um arquivo XML do ObjectGrid cuja estrutura é semelhante ao arquivo que foi usado para o servidor de contêiner.

As seguintes configurações podem ser substituídas no cliente:

1. Crie uma instância ObjectGrid específica do cliente.
2. Copie o arquivo XML ObjectGrid que foi utilizado para abrir o servidor.
3. Edite o novo arquivo a ser customizado para o lado do cliente:
  - Para configurar ou atualizar qualquer um dos atributos no cliente, especifique um novo valor ou altere o valor existente.
  - Para remover um plug-in do cliente, use a cadeia vazia como o valor para o atributo className.
  - Para alterar um plug-in existente, especifique um novo valor para o atributo className.
  - Também é possível incluir qualquer plug-in suportado para uma substituição de cliente: TRANSACTION\_CALLBACK, OBJECTGRID\_EVENT\_LISTENER, EVICTOR, MAP\_EVENT\_LISTENER.
4. Crie um cliente com o arquivo XML de substituição de cliente recém criado.

### Procedimento

1. Crie um arquivo XML de configuração do ObjectGrid para o cliente cuja estrutura é semelhante ao arquivo que foi usado para o servidor de contêiner.

Suponha que o seguinte arquivo XML foi emparelhado com um arquivo XML de política de implementação, e esses arquivos foram usados para iniciar um servidor de contêiner.

**companyGridServerSide.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">

  <objectGrids>
    <objectGrid name="CompanyGrid">
      <bean id="TransactionCallback"
        className="com.company.MyTxCallback" />
      <bean id="ObjectGridEventListener"
        className="com.company.MyOgEventListener" />
      <backingMap name="Customer"
        pluginCollectionRef="customerPlugins"/>
      <backingMap name="Item" />
      <backingMap name="OrderLine" numberOfBuckets="1049"
        timeToLive="1600" ttlEvictorType="LAST_ACCESS_TIME" />
      <backingMap name="Order" lockStrategy="PESSIMISTIC"
        pluginCollectionRef="orderPlugins" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="customerPlugins">
      <bean id="Evictor"
        className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" />
      <bean id="MapEventListener"
        className="com.company.MyMapEventListener" />
    </backingMapPluginCollection>
    <backingMapPluginCollection id="orderPlugins">
      <bean id="MapIndexPlugin"
        className="com.company.MyMapIndexPlugin" />
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Em um servidor de contêiner, a instância do ObjectGrid denominada CompanyGrid se comporta conforme o definido pelo arquivo companyGridServerSide.xml. Por padrão, o cliente CompanyGrid tem as mesmas configurações que a instância CompanyGrid em execução no servidor. O seguinte arquivo XML ObjectGrid pode ser usado para especificar alguns dos atributos e plug-ins no cliente CompanyGrid.

**companyGridClientSide.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">

  <objectGrids>
    <objectGrid name="CompanyGrid">
      <bean id="TransactionCallback"
        className="com.company.MyClientTxCallback" />
      <bean id="ObjectGridEventListener" className="" />
      <backingMap name="Customer" numberOfBuckets="1429"
        pluginCollectionRef="customerPlugins"/>
      <backingMap name="Item" />
      <backingMap name="OrderLine" numberOfBuckets="701"
        timeToLive="800" ttlEvictorType="LAST_ACCESS_TIME" />
      <backingMap name="Order" lockStrategy="PESSIMISTIC"
        pluginCollectionRef="orderPlugins" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="customerPlugins">
      <bean id="Evictor"
        className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" />
      <bean id="MapEventListener" className="" />
    </backingMapPluginCollection>
    <backingMapPluginCollection id="orderPlugins">
      <bean id="MapIndexPlugin" />
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

```

        className="com.company.MyMapIndexPlugin" />
    </backingMapPluginCollection>
</backingMapPluginCollections>
</objectGridConfig>

```

A seguir há um resumo das substituições que são definidas:

- TransactionCallback no cliente é com.company.MyClientTxCallback em vez da configuração do lado do servidor de com.company.MyTxCallback.
- O cliente não tem um plug-in ObjectGridEventListener porque o valor className é a cadeia vazia.
- O cliente configura numberOfBuckets como 1429 para Customer backingMap, retém seu plug-in Evictor e remove o plug-in MapEventListener.
- Os atributos numberOfBuckets e timeToLive de OrderLine backingMap mudaram
- Embora um atributo lockStrategy diferente seja especificado, não há nenhum efeito, pois o atributo lockStrategy não é suportado para uma substituição de cliente.

## 2. Crie o cliente usando o arquivo XML.

Para criar o cliente CompanyGrid usando o arquivo companyGridClientSide.xml, passe o arquivo ObjectGrid XML como uma URL para um dos métodos de conexão na interface ObjectGridManager:

```

ObjectGridManager ogManager =
    ObjectGridManagerFactory.ObjectGridManager();
ClientClusterContext clientClusterContext = ogManager.connect
("MyServer1.company.com:2809", null, new URL(
    "file:xml/companyGridClientSide.xml"));

```

## Ativando o Mecanismo de Invalidação do Cliente

Em um ambiente do WebSphere eXtreme Scale distribuído, o lado do cliente possui um cache local por padrão ao utilizar a estratégia de bloqueio otimista ou quando o bloqueio está desativado. O cache perto tem seus próprios dados locais armazenados em cache. Se um eXtreme Scale cliente confirmar uma atualização, a atualização chegará próxima do cache e do servidor. Entretanto, outros eXtreme Scale clientes não recebem as informações de atualização e poderão ter dados desatualizados.

### Cache Local

Os aplicativos devem ficar cientes desse problema de dados obsoletos no cliente do eXtreme Scale. É possível usar a classe com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener ObjectGridEventListener baseada no Java Message Service (JMS) integrada para ativar o mecanismo de invalidação do cliente dentro de um ambiente distribuído do eXtreme Scale.

O mecanismo de invalidação do cliente é a solução para o problema de dados antigos no cache local do cliente no ambiente do eXtreme Scale distribuído. Esse mecanismo garante que o cache local ao cliente esteja em sincronia com os servidores ou outros clientes. Contudo, mesmo com esse mecanismo de invalidação do cliente baseado em JMS, o cache local do cliente não é atualizado imediatamente. Um atraso ocorre quando o tempo de execução do eXtreme Scale publica atualizações.

Dois modelos estão disponíveis para o mecanismo de invalidação do cliente em um ambiente do eXtreme Scale distribuído:

- Modelo cliente-servidor: Neste modelo, todos os processos do servidor assumem a função de publicador e publicam todas as alterações na transação para o destino do JMS designado. Todos os processos do cliente estão nas funções de receptor e recebem todas as alterações na transação do destino do JMS designado.
- Modelo cliente como dupla função: Neste modelo, os processos do servidor não têm nada a fazer com o destino do JMS. Todos os processos do cliente assumem ambas as funções do JMS, publicador e receptor. As alterações na transação que ocorrem no cliente são publicadas no destino do JMS e todos os clientes recebem tais alterações.

Para obter mais informações, leia sobre “Listener de Eventos da JMS” na página 228.

## Modelo de Cliente-servidor

Em um modelo cliente/servidor, os servidores exercem a função de publicador JMS e o cliente a função de receptor JMS.

### Exemplo de XML de modelo cliente/servidor

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="AgentObjectGrid">
      <bean id="ObjectGridEventListener"
        className="com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener">
        <property name="invalidationModel" type="java.lang.String" value="CLIENT_SERVER_MODEL"
description="" />
        <property name="invalidationStrategy" type="java.lang.String" value="PUSH" description="" />
        <property name="mapsToPublish" type="java.lang.String" value="agent;profile;pessimisticMap" description="" />
        <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String" value="defaultTCF" description="" />
        <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_userid" type="java.lang.String" value="" description="" />
        <property name="jms_password" type="java.lang.String" value="" description="" />
        <property name="jndi_properties" type="java.lang.String"
          value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;
java.naming.provider.url=
tcp://localhost:61616;connectionFactoryNames=defaultTCF;topic.defaultTopic=defaultTopic"
description="jndi properties" />
      </bean>

      <backingMap name="agent" readOnly="false" pluginCollectionRef="agent" preloadMode="false"
lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
timeToLive="28800" />
      <backingMap name="profile" readOnly="false" pluginCollectionRef="profile" preloadMode="false"
lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
timeToLive="2700" />
      <backingMap name="pessimisticMap" readOnly="false" pluginCollectionRef="pessimisticMap" preloadMode="false"
lockStrategy="PESSIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
timeToLive="2700" />
      <backingMap name="excludedMap1" readOnly="false" pluginCollectionRef="excludedMap1" preloadMode="false"
lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
timeToLive="2700" />
      <backingMap name="excludedMap2" readOnly="false" pluginCollectionRef="excludedMap2" preloadMode="false"
lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
timeToLive="2700" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="agent">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.AgentObjectTransformer" />
    </backingMapPluginCollection>
    <backingMapPluginCollection id="profile">
      <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.ProfileObjectTransformer" />
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
        <property name="maxSize" type="int" value="2000" description="set max size for LRU evictor" />
        <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
        <property name="numberOfLRUQueues" type="int" value="50" description="set number of LRU queues" />
      </bean>
    </backingMapPluginCollection>

    <backingMapPluginCollection id="pessimisticMap" />
    <backingMapPluginCollection id="excludedMap1" />
  </backingMapPluginCollections>
</objectGridConfig>
```



```

    <backingMapPluginCollection id="excludedMap2" />
</backingMapPluginCollections>

</objectGridConfig>

```

## Cliente como Modelo de Funções Duplas

No modelo cliente como dupla função, cada cliente assume ambas as funções do JMS, publicador e receptor. O cliente publica cada mudança da transação confirmada para um destino JMS designado e recebe todas as alterações transacionais confirmadas de outros clientes. O servidor não tem nada a fazer com o JMS neste modelo.

### Exemplo de XML de modelo de função dupla

```

<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="AgentObjectGrid">
      <bean id="ObjectGridEventListener"
        className="com.ibm.websphere.objectgrid.plugins.builtins.JMSObjectGridEventListener">
        <property name="invalidationModel" type="java.lang.String" value="CLIENT_AS_DUAL_ROLES_MODEL" description="" />
        <property name="invalidationStrategy" type="java.lang.String" value="PUSH" description="" />
        <property name="mapsToPublish" type="java.lang.String" value="agent;profile;pessimisticMap" description="" />
        <property name="jms_topicConnectionFactoryJndiName" type="java.lang.String" value="defaultTCF" description="" />
        <property name="jms_topicJndiName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_topicName" type="java.lang.String" value="defaultTopic" description="" />
        <property name="jms_userid" type="java.lang.String" value="" description="" />
        <property name="jms_password" type="java.lang.String" value="" description="" />
        <property name="jndi_properties" type="java.lang.String"
          value="java.naming.factory.initial=org.apache.activemq.jndi.ActiveMQInitialContextFactory;java.naming.provider.url=
          tcp://localhost:61616;connectionFactoryNames=defaultTCF;topic.defaultTopic=defaultTopic"
          description="jndi properties" />
        </bean>

        <backingMap name="agent" readOnly="false" pluginCollectionRef="agent" preloadMode="false"
          lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
          timeToLive="28800" />
        <backingMap name="profile" readOnly="false" pluginCollectionRef="profile" preloadMode="false"
          lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
          timeToLive="2700" />
        <backingMap name="pessimisticMap" readOnly="false" pluginCollectionRef="pessimisticMap" preloadMode="false"
          lockStrategy="PESSIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
          timeToLive="2700" />
        <backingMap name="excludedMap1" readOnly="false" pluginCollectionRef="excludedMap1" preloadMode="false"
          lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
          timeToLive="2700" />
        <backingMap name="excludedMap2" readOnly="false" pluginCollectionRef="excludedMap2" preloadMode="false"
          lockStrategy="OPTIMISTIC" copyMode="COPY_ON_READ_AND_COMMIT" ttlEvictorType="LAST_ACCESS_TIME"
          timeToLive="2700" />
      </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
      <backingMapPluginCollection id="agent">
        <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.AgentObjectTransformer" />
      </backingMapPluginCollection>
      <backingMapPluginCollection id="profile">
        <bean id="ObjectTransformer" className="com.ibm.ws.objectgrid.test.scenario.ProfileObjectTransformer" />
        <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
          <property name="maxSize" type="int" value="2000" description="set max size for LRU evictor" />
          <property name="sleepTime" type="int" value="15" description="evictor thread sleep time" />
          <property name="numberOfLRUQueues" type="int" value="50" description="set number of LRU queues" />
        </bean>
      </backingMapPluginCollection>

      <backingMapPluginCollection id="pessimisticMap" />
      <backingMapPluginCollection id="excludedMap1" />
      <backingMapPluginCollection id="excludedMap2" />
    </backingMapPluginCollections>
  </objectGridConfig>

```

## Configurando Valores de Tempo Limite de Nova Tentativa de Solicitação

Com mapas confiáveis, é possível fornecer um valor de tempo limite de nova tentativa em milissegundos para o WebSphere eXtreme Scale para solicitações de transação.

## Sobre Esta Tarefa

É possível configurar o valor de tempo limite no arquivo de propriedades do cliente ou em uma sessão. O valor de sessão substitui a configuração de propriedades do cliente. Se o valor estiver configurado como maior que zero, a solicitação será tentada até que a condição de tempo limite seja atendida ou até ocorrer uma falha permanente. Uma falha permanente pode ser uma exceção `DuplicateKeyException`. Um valor de zero indica a configuração do modo fail-fast e o eXtreme Scale não tenta a transação novamente após qualquer tipo de transação.

Durante o tempo de execução, o valor de tempo limite da transação é usado com o valor de tempo limite de nova tentativa, garantindo que o tempo limite de nova tentativa não exceda o tempo limite da transação.

Há dois tipos de transações: Transações de confirmação automática e as transações que usam os métodos `begin` e `commit` explícitos. As exceções válidas para nova tentativa diferem entre estes dois tipos de transações:

- Para transações que são chamadas dentro de uma sessão, as transações são tentadas novamente para as exceções `SystemException` CORBA e `TargetNotAvailable` do eXtreme Scale.
- Para transações de confirmação automática, as transações são tentadas novamente para `SystemException` CORBA e exceções de disponibilidade do eXtreme Scale. Essas exceções incluem as exceções `ReplicationVotedToRollbackTransactionException`, `TargetNotAvailable` e `AvailabilityException`.

As falhas do aplicativo ou outras falhas permanentes são retornadas imediatamente e o cliente não tenta a transação novamente. Essas falhas permanentes incluem as exceções `DuplicateKeyException` e `KeyNotFoundException`. Use a configuração fail-fast para retornar todas as exceções sem tentar as transações novamente depois de quaisquer exceções.

### As exceções nas quais o cliente tenta a transação novamente são:

- `ReplicationVotedToRollbackTransactionException` (somente na auto-consolidação)
- `TargetNotAvailable`
- `org.omg.CORBA.SystemException`
- `AvailabilityException` (somente na auto-consolidação)
- `LockTimeoutException` (somente na auto-consolidação)
- `UnavailableServiceException` (somente na auto-consolidação)

### Exceções permanentes, onde a transação não será tentada novamente:

- `DuplicateKeyException`
- `KeyNotFoundException`
- `LoaderException`
- `TransactionAffinityException`
- `LockDeadlockException`
- `OptimisticCollisionException`

## Procedimento

- Configure o valor de tempo limite em um arquivo de propriedades do cliente.

Para configurar o valor `requestRetryTimeout` em um cliente, inclua ou modifique a propriedade `requestRetryTimeout` no Arquivo de Propriedades do Cliente. As propriedades do cliente estão no arquivo `objectGridClient.properties` pelo padrão. A propriedade `requestRetryTimeout` é configurada em milissegundos. Configure o valor maior que zero para que o pedido seja tentado novamente em exceções para as quais uma nova tentativa está disponível. Configure o valor como 0 para falhar sem as novas tentativas em exceções. Para usar o comportamento padrão, remova a propriedade ou configure o valor como -1. Um exemplo do valor no arquivo `objectGridClient.properties` é o seguinte:

```
requestRetryTimeout = 30000
```

O valor `requestRetryTimeout` é especificado em milissegundos. No exemplo, se o valor for usado em uma instância de `ObjectGrid`, o valor de `requestRetryTimeout` será de 30 segundos.

- Configure o valor de tempo limite programaticamente.

Para configurar as propriedades do cliente programaticamente, primeiro crie um arquivo de propriedades do cliente em um `<local>` apropriado para seu aplicativo. No exemplo a seguir, o arquivo de propriedades do cliente refere-se ao fragmento de `objectGridClient.properties` na seção anterior. Depois de se conectar à instância `ObjectGridManager`, configure as propriedades do cliente, conforme descrito. Em seguida, quando possuir uma instância do `ObjectGrid`, a instância possuirá as propriedades do cliente que foram definidas no arquivo. Se você alterar o arquivo de propriedades do cliente, será necessário toda vez obter explicitamente uma nova instância de `ObjectGrid`.

```
ObjectGridManager manager= ObjectGridManagerFactory.getObjectGridManager();
String objectGridName = "testObjectGrid";
URL clientXML = null;
ClientContext ccc = manager.connect("localhost:2809", null,
clientXML);
File file = new File("<location>/objectGridClient.properties");
URL url = file.toURI().toURL();
ccc.setClientProperties(objectGridName, url);
ObjectGrid objectGrid = ogManager.getObjectGrid(ccc, objectGridName);
```

- Configure o arquivo de substituição durante uma confirmação de sessão.

Para configurar o tempo limite de nova tentativa de solicitação em uma sessão ou para substituir a propriedade do cliente `requestRetryTimeout`, chame o método `setRequestRetryTimeout(long)` na interface `Session`.

```
Session sessionA = objectGrid.getSession();
sessionA.setRequestRetryTimeout(30000);
ObjectMap mapA = sessionA.getMap("payroll");
String key = "key:" + j;
mapA.insert(key, "valueA");
```

Esta sessão agora usa um valor de `requestRetryTimeout` igual a 30000 ms ou 30 segundos, independentemente do valor que está configurado no arquivo de propriedades do cliente. Para obter informações adicionais sobre a interface da sessão, consulte [Uso de Sessões para Acessar Dados na Grade](#).

---

## Configurando a Integração de Cache

O WebSphere eXtreme Scale pode integrar-se com outros produtos relacionados ao armazenamento em cache. Também é possível usar o provedor de cache dinâmico do WebSphere eXtreme Scale para plugar o WebSphere eXtreme Scale no componente de cache dinâmico no WebSphere Application Server. Outra extensão para o WebSphere Application Server é o gerenciador de sessões HTTP do WebSphere eXtreme Scale, que pode ajudar a armazenar em cache as sessões HTTP.

## Configurando os Gerenciadores de Sessão HTTP

O gerenciador de sessões HTTP fornece recursos de replicação de sessão para um aplicativo associado. O gerenciador de sessões trabalha com o contêiner da web para criar e gerenciar os ciclos de vida de sessões HTTP que estão associadas ao aplicativo.

### Configurando o Gerenciador de Sessões HTTP com WebSphere Application Server

Enquanto o WebSphere Application Server fornece a função de gerenciamento de sessões, o desempenho diminui conforme o número de solicitações aumenta. O WebSphere eXtreme Scale vem com uma implementação de gerenciamento de sessões que fornece opções de replicação de sessão, alta disponibilidade, melhor escalabilidade e configuração mais firme.

#### Antes de Iniciar

- O WebSphere eXtreme Scale deve ser instalado na célula do seu WebSphere Application Server ou WebSphere Application Server Network Deployment para usar o gerenciador de sessões do eXtreme Scale. Consulte “Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client com WebSphere Application Server” na página 161 para obter mais informações.
- A segurança global deve ser ativada no console administrativo do WebSphere Application Server, se os servidores de catálogo em seu domínio de serviço de catálogo tiver o Secure Sockets Layer (SSL) ativado e desejar usar o SSL para um domínio de serviço de catálogo com SSL suportado. Você requer SSL para um servidor de catálogos configurando o atributo `transportType` como `SSL-Required` no Arquivo de Propriedades do Servidor. Para obter informações adicionais sobre como configurar a segurança global, consulte Configurações de Segurança Global.

#### Sobre Esta Tarefa

O gerenciador de sessão HTTP WebSphere eXtreme Scale suporta servidores integrados e remotos para armazenamento em cache.

##### • Cenário Integrado

No cenário integrado, os servidores WebSphere eXtreme Scale são colocados no mesmo processo onde os servlets são executados. O gerenciador de sessões pode ser comunicar diretamente com a instância do ObjectGrid local, evitando atrasos de rede caros.

Se você estiver usando o WebSphere Application Server, coloque os arquivos `wxs_home/session/samples/objectGrid.xml` e `wxs_home/session/samples/objectGridDeployment.xml` fornecidos nos diretórios META-INF dos seus arquivos Web Archive (WAR). O eXtreme Scale detecta automaticamente esses arquivos quando o aplicativo é iniciado e inicia automaticamente os contêineres do eXtreme Scale no mesmo processo do gerenciador de sessões.

É possível modificar o arquivo `objectGridDeployment.xml` dependendo se você deseja usar a replicação síncrona ou assíncrona e de quantas réplicas você deseja configurar.

##### • Cenário dos Servidores Remotos

No cenário de servidores remotos, os servidores de contêiner são executados em processos diferentes dos servlets. O gerenciador de sessões se comunica com um servidor de contêiner remoto. Para usar um servidor de contêiner remoto, conectado à rede, o gerenciador de sessões deve ser configurado com os nomes de host e números de porta do domínio do serviço de catálogo. O gerenciador

de sessões, então, usa uma conexão do cliente do eXtreme Scale para se comunicar com o servidor de catálogos e com os servidores de contêiner.

Se os servidores de contêiner forem iniciados em processos independentes, inicie os contêineres do eXtreme Scale com os arquivos `objectGridStandAlone.xml` e `objectGridDeploymentStandAlone.xml` que são fornecidos no diretório de amostras do gerenciador de sessões.

## Procedimento

1. Una seu aplicativo para que ele possa utilizar o gerenciador de sessões. Para usar o gerenciador de sessões, é necessário incluir as declarações de filtro apropriadas nos descritores de implementação da Web para o aplicativo. Além disso, os parâmetros de configuração do gerenciador de sessões são passados no gerenciador de sessões no formato de parâmetros de inicialização de contexto do servlet nos descritores de implementação. Há várias maneiras pelas quais é possível introduzir essas informações no seu aplicativo:

- **Combinação Automática com WebSphere Application Server**

É possível configurar seu aplicativo para usar o gerenciador de sessões HTTP do WebSphere eXtreme Scale quando instalar seu aplicativo. Pode também editar o aplicativo ou a configuração do servidor para usar o gerenciador de sessões HTTP do WebSphere eXtreme Scale. Consulte “Unindo Automaticamente Aplicativos para Gerenciamento de Sessões HTTP no WebSphere Application Server” na página 300 para obter mais informações.

- **Combinação Automática do Aplicativo com Propriedades Customizadas**

Não é necessário conectar seus aplicativos manualmente quando o aplicativo está em execução no WebSphere Application Server ou no WebSphere Application Server Network Deployment.

Inclua uma propriedade customizada para uma célula ou um servidor para configurar o arquivo `splicer.properties` para todos os aplicativos da Web nesse escopo. Use as seguintes etapas para configurar a propriedade customizada:

- a. No console administrativo WebSphere Application Server, navegue até o caminho correto para onde deseja configurar a propriedade customizada para indicar o local do arquivo `splicer.properties`.
  - Para configurar a propriedade customizada para todos os aplicativos ou um aplicativo específico, clique em **Administração do Sistema > Célula > Propriedades Customizadas**.
  - Para configurar a propriedade customizada para aplicar a todos os aplicativos em um servidor de aplicativos específico, clique em **Servidor de Aplicativos > <server\_name> > Administração > Propriedades Customizadas**. O nome da propriedade é `com.ibm.websphere.xs.sessionFilterProps` e seu valor é o local do arquivo `splicer.properties` que seus aplicativos requerem. Um caminho de exemplo para o local de um arquivo é o seguinte:  
`/opt/splicer.properties`.
- b. Inclua a propriedade customizada `com.ibm.websphere.xs.sessionFilterProps`. O valor desta propriedade customizada fornece o local do arquivo `splicer.properties` a ser editado. O arquivo existe no gerenciador de implementação. Se desejar indicar o arquivo `splicer.properties` para um aplicativo específico com uma propriedade customizada no nível da célula, insira o nome da propriedade customizada

como:<*application\_name*>, com `ibm.websphere.xs.sessionFilterProps`, em que *application\_name* indica o nome do aplicativo ao qual deseja aplicar a propriedade customizada.

**Importante:** Assegure que o arquivo `splicer.properties` atualizado esteja no mesmo caminho em todos os nós que contêm um servidor de aplicativos que hospeda o aplicativo ou os aplicativos que estão sendo unidos para replicação de sessão.

A célula, o servidor e o escopo do aplicativo são escopos disponíveis e ficam disponíveis apenas durante a execução em um gerenciador de implementação. Se um escopo diferente for necessário, combine manualmente seus aplicativos da web.

**Lembre-se:** Além disso, observe que a opção de combinação automática funciona apenas se todos os nós que executam o aplicativo contiverem o arquivo `splicer.properties` no mesmo caminho. Para ambientes mistos que contêm nós Windows e UNIX, esta opção não é possível, portanto, você deve conectar o aplicativo manualmente.

- **Combine o aplicativo com o script `addObjectGridFilter`**

Use o script de linha de comandos fornecido junto com o eXtreme Scale para unir um aplicativo com as declarações de filtro e com as configuração no formato de parâmetros de inicialização de contexto do servlet. Para uma implementação do WebSphere Application Server, este script está localizado em `<was_home>/optionalLibraries/ObjectGrid/session/bin/addObjectGridFilter.bat/sh`. Para um implementação independente, o script está em `WXS_HOME/ObjectGrid/session/bin/addObjectGridFilter.sh/bat`. O script **`addObjectGridFilter`** usa dois parâmetros:

- Aplicativo - caminho absoluto para o archive corporativo para ser unido
- Caminho absoluto para o arquivo `splicer.properties` que contém várias propriedades de configuração.

O formato de uso deste script é o seguinte:

**Windows**

```
addObjectGridFilter.bat [ear_file] [splicer_properties_file]
```

**UNIX**

```
addObjectGridFilter.sh [ear_file] [splicer_properties_file]
```

**UNIX**

**Exemplo do uso do eXtreme Scale instalado no WebSphere Application Server no Unix:**

- `cd was_home/optionalLibraries/ObjectGrid/session/bin`
- `addObjectGridFilter.sh /tmp/mySessionTest.ear was_root/optionalLibraries/ObjectGrid/session/samples/splicer.properties`

**UNIX**

**Exemplo do uso do eXtreme Scale instalado em um diretório independente no UNIX:**

- `cd was_root/session/bin`
- `addObjectGridFilter.sh /tmp/mySessionTest.ear was_root/session/samples/splicer.properties`

O filtro do servlet unido mantém os padrões para os valores de configuração. É possível substituir estes valores-padrão com opções de configuração especificados no arquivo de propriedades no segundo argumento. Para obter uma lista dos parâmetros que podem ser utilizados, consulte “Parâmetros de inicialização do contexto do servlet” na página 316.



É possível modificar e usar o arquivo `splicer.properties` de amostra que é fornecido com a instalação do eXtreme Scale. Também é possível usar o script `addObjectGridServlets`, que insere o gerenciador de sessões estendendo cada servlet. No entanto, o script recomendado é o `addObjectGridFilter`.

- **Combine manualmente o aplicativo com o script de construção Ant**

O WebSphere eXtreme Scale é fornecido com um arquivo `build.xml` que pode ser usado pelo Apache Ant, que é incluído na pasta `was_root/bin` de uma instalação WebSphere Application Server. É possível modificar o arquivo `build.xml` para alterar as propriedades de configuração do gerenciador de sessões. As propriedades de configuração são idênticas aos nomes de propriedades no arquivo `splicer.properties`. Depois de modificar o arquivo `build.xml`, chame o processo Ant executando o seguinte comando:

- `UNIX` `ant.sh, ws_ant.sh`
- `Windows` `ant.bat, ws_ant.bat`

(UNIX) ou (Windows).

- **Atualize manualmente o descritor da Web**

Edite o arquivo `web.xml` que é empacotado com o aplicativo da Web para incorporar a declaração do filtro, seu mapeamento de servlet e os parâmetros de inicialização de contexto do servlet. Não use este método porque ele é propenso a erros.

Para obter uma lista dos parâmetros que podem ser utilizados, consulte “Parâmetros de inicialização do contexto do servlet” na página 316.

2. Implemente o aplicativo. Implemente o aplicativo com seu conjunto normal de etapas para um servidor ou cluster. Após implementar o aplicativo, é possível iniciar o aplicativo.
3. Acesse o aplicativo. É possível acessar o aplicativo, que interage com o gerenciador de sessões e o WebSphere eXtreme Scale.

## O que Fazer Depois

É possível alterar a maioria dos atributos de conexão para o gerenciador de sessões quando você instrumenta seu aplicativo para utilizar o gerenciador de sessões. Esses atributos incluem: replicação síncrona ou assíncrona, tamanho da tabela de sessão em memória, e assim por diante. Não considerando os atributos que podem ser alterados no momento da instrumentação do aplicativo, os únicos outros atributos de configuração que podem ser alterados após a implementação do aplicativo são os atributos que estão relacionados à topologia em cluster do servidor WebSphere eXtreme Scale e a maneira pela qual seus clientes (gerenciadores de sessões) se conectam a eles.

**Comportamento de um cenário remoto:** Se a grade de dados inteira que hospeda os dados da sessão de aplicativo estiver inacessível a partir do cliente do contêiner da web, o cliente usará no lugar o contêiner da web de base no WebSphere Application Server para o gerenciamento de sessões. A grade de dados pode estar inatingível nos cenários a seguir:

- Um problema de rede entre o contêiner da web e os servidores de contêiner remotos.
- Os processos do servidor de contêiner remoto foram interrompidos.

O número de referências de sessão mantidas na memória, especificado pelo parâmetro `sessionTableSize`, ainda é mantido quando as sessões são armazenadas



no contêiner da web de base. As sessões usadas menos recentemente são invalidadas a partir do cache de sessão do contêiner da web quando o valor **sessionTableSize** é excedido. Se a grade de dados remota se tornar disponível, as sessões que foram invalidadas a partir do cache de contêiner da web poderão recuperar os dados da grade de dados remota e carregar os dados em uma nova sessão. Se a grade de dados remota inteira não estiver disponível e a sessão for invalidada a partir do cache de sessão, os dados da sessão do usuário serão perdidos. Devido a esse problema, não encerre a grade de dados remota de produção inteira quando o sistema estiver sendo executado sob carga.

### **Unindo Automaticamente Aplicativos para Gerenciamento de Sessões HTTP no WebSphere Application Server:**

É possível configurar o aplicativo WebSphere Application Server para persistir as sessões para uma grade de dados. Esta grade de dados pode estar em um servidor de contêiner integrado executado no WebSphere Application Server ou pode estar em uma grade de dados remota.

#### **Antes de Iniciar**

Antes de alterar a configuração no WebSphere Application Server, você deve ter:

- O nome da grade de dados da sessão que você deseja usar. Consulte “Configurando o Gerenciador de Sessões HTTP com WebSphere Application Server” na página 296 para obter informações sobre como criar uma grade de dados da sessão.
- Se o serviço de catálogo que você deseja usar para gerenciar suas sessões estiver fora da célula na qual está instalando o aplicativo da sessão, será necessário criar um domínio do serviço de catálogo. Consulte o “Criando Domínios do Serviço de Catálogo no WebSphere Application Server” na página 252 para obter informações adicionais.
- Se você estiver configurando um domínio de serviço de catálogo, talvez precise ativar a segurança do cliente no domínio de serviço de catálogo se os servidores de contêiner precisarem de autenticação. Essas configurações informam o tempo de execução qual implementação CredentialGenerator deve ser usada. Esta implementação gera uma credencial para passar para a grade de dados remota. Consulte “Configurando a Segurança do Cliente em um Domínio de Serviço de Catálogo” na página 518 para obter informações adicionais sobre a configuração dessas definições.
- Segurança global ativada no console administrativo do WebSphere Application Server, se desejar suportar um dos cenários a seguir:
  - Os servidores de catálogos em seu domínio de serviço de catálogo têm Secure Sockets Layer (SSL) ativado.
  - Você deseja usar SSL para um domínio de serviço de catálogo com SSL suportado.

Você requer SSL para um servidor de catálogos configurando o atributo **transportType** como SSL-Required no Arquivo de Propriedades do Servidor. Para obter informações adicionais sobre como configurar a segurança global, consulte Configurações de Segurança Global.

- Se você estiver usando a Versão 7.1.0.3 ou posterior, será possível persistir sessões que usam a gravação de URL ou cookies como um mecanismo de rastreamento de sessão para a grade de dados. Para liberações anteriores da Versão 7.1.0.3, não é possível persistir sessões que usam a gravação de URL como um mecanismo de rastreamento de sessão. Para ativar a persistência de sessões que

usam a regravação de URL, configure a propriedade **useURLEncoding** para true no arquivo `splicer.properties` após unir automaticamente o aplicativo.

- **7.1.1+** Quando você estiver unindo automaticamente aplicativos para o gerenciamento de sessões HTTP no WebSphere Application Server, todos os servidores de aplicativos que hospedam o aplicativo da web terão a propriedade customizada do contêiner da web **HttpSessionIdReuse** configurada como true. Esta propriedade permite que sessões que falharam a partir de um servidor de aplicativos para outro ou foram invalidadas a partir do cache de sessão na memória em um cenário remoto, preservem seu ID de sessão entre as solicitações. Se você não desejar este comportamento, configure a propriedade customizada do contêiner da web como false em todos os servidores de aplicativos aplicáveis antes de configurar o gerenciamento de sessões para os aplicativos. Para obter informações adicionais sobre esta propriedade customizada, consulte “Resolvendo Problemas da Integração de Cache” na página 535.

### Procedimento

- **Para configurar o gerenciamento de sessões quando estiver instalando o aplicativo, conclua as seguintes etapas:**
  1. No console administrativo do WebSphere Application Server, clique em **Aplicativos > Novo aplicativo > Novo Aplicativo Corporativo**. Escolha o caminho **Detalhado** para criar o aplicativo e concluir as etapas iniciais do assistente.
  2. Na etapa **Configurações de gerenciamento de sessão do eXtreme Scale** do assistente, configure a grade de dados que deseja usar. Escolha a **Grade de dados do eXtreme Scale remota** ou a **Grade de dados do eXtreme Scale integrada**.
    - Para a opção **Grade de dados do eXtreme Scale remota**, escolha o domínio do serviço de catálogo que gerencia a grade de dados da sessão e escolha uma grade de dados na lista de grades de dados da sessão ativa.
    - Para a opção **Grade de dados do eXtreme Scale integrada**, escolha a configuração padrão do ObjectGrid ou especifique o local específico dos arquivos de configuração do ObjectGrid.
  3. Conclua as etapas do assistente para concluir a instalação do aplicativo.

É possível também instalar o aplicativo com um script `wsadmin`. No exemplo a seguir, o parâmetro **-SessionManagement** cria a mesma configuração que é possível criar no console administrativo:

#### Para a configuração remota de grade de dados eXtreme Scale:

```
AdminApp.install('C:/A.ear', '[ -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname A -edition 8.0
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
off -noprocessEmbeddedConfig -filepermission .*.dll=755#.*.so=755#.*.a=755#.*.sl=755
-buildVersion Unknown -noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -nouseAutoLink -SessionManagement [[true
XSRemoteSessionManagement cs0!:grid0]]
-MapWebModToVH [[MicroWebApp microwebapp.war,WEB-INF/web.xml default_host] [MicroSipApp
microsipapp.war,WEB-INF/web.xml default_host] [MicroDG1App microdglapp.war,WEB-INF/web.xml
default_host] [MicroDG2App microdgg2app.war,WEB-INF/web.xml default_host] [MicroSip2App
microsip2app.war,WEB-INF/web.xml default_host]]')
```

#### Para o cenário integrado do eXtreme Scale com configuração padrão:

```
AdminApp.install('C:/A.ear', '[ -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname A -edition 8.0
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
off -noprocessEmbeddedConfig -filepermission .*.dll=755#.*.so=755#.*.a=755#.*.sl=755
```

```
-buildVersion Unknown -noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -noseAutoLink -SessionManagement [[true
XSRemoteSessionManagement :!: :!:default]] -MapWebModToVH [[MicroWebApp microwebapp.war,
WEB-INF/web.xml default_host] [MicroSipApp
microsipapp.war,WEB-INF/web.xml default_host] [MicroDG1App microdglapp.war,WEB-INF/web.xml
default_host] [MicroDG2App microdgd2app.war,WEB-INF/web.xml default_host] [MicroSip2App
microsip2app.war,WEB-INF/web.xml default_host]]]')
```

### Para o cenário integrado do eXtreme Scale com uma configuração customizada:

```
AdminApp.install('C:/A.ear', '[ -nopreCompileJSPs -distributeApp
-nouseMetaDataFromBinary -nodeployejb -appname A -edition 8.0
-createMBeansForResources -noreloadEnabled -nodeployws -validateinstall
off -noprocessEmbeddedConfig -filepermission *.*.dll=755#.*.so=755#.*.a=755#.*.sl=755
-buildVersion Unknown -noallowDispatchRemoteInclude -noallowServiceRemoteInclude
-asyncRequestDispatchType DISABLED -noseAutoLink -SessionManagement [[true
XSRemoteSessionManagement :!: :!:custom!:c:\XS\objectgrid.xml!:c:\XS\objectgriddeployment.xml]]
-MapWebModToVH [[MicroWebApp microwebapp.war,WEB-INF/web.xml default_host] [MicroSipApp
microsipapp.war,WEB-INF/web.xml default_host] [MicroDG1App microdglapp.war,WEB-INF/web.xml
default_host] [MicroDG2App microdgd2app.war,WEB-INF/web.xml default_host] [MicroSip2App
microsip2app.war,WEB-INF/web.xml default_host]]]')
```

- Para configurar o gerenciamento de sessões em um aplicativo existente no console administrativo do WebSphere Application Server:

1. No console administrativo do WebSphere Application Server, clique em **Aplicativos > Tipos de Aplicativo > Aplicativos Corporativos do WebSphere > application\_name > Propriedades do módulo da Web > Gerenciamento de Sessões > Configurações do Gerenciamento de Sessões do eXtreme Scale.**
2. Atualize os campos para ativar a persistência de sessão para uma grade de dados.

Também é possível atualizar o aplicativo com um script wsadmin. No exemplo a seguir, o parâmetro **-SessionManagement** cria a mesma configuração que é possível criar no console administrativo:

### Para a configuração remota de grade de dados eXtreme Scale:

```
AdminApp.edit('DefaultApplication', '[-SessionManagement[[true
XSRemoteSessionManagement cs0!:grid0]]')
```

Os caracteres **!:!** que são transmitidos são usados como delimitadores.

Os valores que são transmitidos são:

```
catalogServiceName!:gridName
```

### Para o cenário integrado do eXtreme Scale com configuração padrão:

```
AdminApp.edit('DefaultApplication', '[-SessionManagement[[true
XSEmbeddedSessionManagement :!: :!:default]]')
```

Os caracteres **!:!** que são transmitidos são usados como delimitadores.

Os valores que são transmitidos são:

```
catalogServiceName!:gridName!:default!:
absolutePath_to_objectGridXmlfile!:absolutePath_to_DeploymentXmlfile
```

### Para o cenário integrado do eXtreme Scale com uma configuração customizada:

```
AdminApp.edit('DefaultApplication', '[-SessionManagement[[true
XSEmbeddedSessionManagement
:!: :!:custom!:c:\XS\objectgrid.xml!:c:\XS\objectgriddeployment.xml]]')
```

Os caracteres **!:!** que são transmitidos são usados como delimitadores.

Os valores que são transmitidos são:

```
catalogServiceName!:gridName!:custom!:
absolutePath_to_objectGridXmlfile!:absolutePath_to_DeploymentXmlfile
```

Quando você salva as mudanças, o aplicativo usa a grade de dados configurada para a persistência de sessão no dispositivo.

- **Para configurar o gerenciamento de sessões em um servidor existente:**

1. No console administrativo do WebSphere Application Server, clique em **Servidores > Tipos de Servidor > Servidores de Aplicativos do WebSphere > server\_name > Gerenciamento de Sessões > Configurações do Gerenciamento de Sessões do eXtreme Scale.**
2. Atualize os campos para ativar a persistência de sessão.

Também é possível configurar o gerenciamento de sessões em um servidor existente com os seguintes comandos de ferramenta wsadmin:

**Para a configuração remota de grade de dados eXtreme Scale:**

```
AdminTask.configureServerSessionManagement(['-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSRemoteSessionManagement -XSRemoteSessionManagement
[-catalogService cs0 -csGridName grid0]'])
```

**Para a configuração integrada do eXtreme Scale:**

- A configuração padrão, se você estiver usando os arquivos XML padrão:

```
AdminTask.configureServerSessionManagement(['-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement [-embeddedGridType default -objectGridXML -objectGridDeploymentXML ]'])
```

- A configuração customizada, se você estiver usando arquivos XML customizados:

```
AdminTask.configureServerSessionManagement(['-nodeName IBM-C77EE220EB6Node01 -serverName server1
-enableSessionManagement true -sessionManagementType XSEmbeddedSessionManagement
-XSEmbeddedSessionManagement
[-embeddedGridType custom -objectGridXML c:\XS\objectgrid.xml -objectGridDeploymentXML
c:\XS\objectgriddeployment.xml]'])
```

Quando você salva as mudanças, o servidor usa a grade de dados configurada para a persistência de sessão com os aplicativos que estão em execução no servidor.

- Se desejar editar outros aspectos da configuração de sessão HTTP, será possível editar o arquivo `splicer.properties`. É possível obter o local do caminho do arquivo `splicer.properties` localizando a propriedade customizada **sessionFilterProps**. Se você configurou a persistência da sessão no nível do servidor, o nome da propriedade customizada é:

`:com.ibm.websphere.xs.sessionFilterProps`. Se você configurou a persistência da sessão no nível do aplicativo, o nome da propriedade customizada é: `<application_name>,com.ibm.websphere.xs.sessionFilterProps`. Estas propriedades customizadas podem estar em um dos seguintes locais:

- Em um ambiente do WebSphere Application Server Network Deployment: Altere o arquivo `splicer.properties` no caminho do perfil de gerenciador de implementação.
- Em um ambiente independente do WebSphere Application Server: Uma propriedade customizada no servidor de aplicativos

É possível abrir o arquivo indicado, fazer mudanças e sincronizar os nós de forma que o arquivo de propriedades atualizado seja propagado para os outros nós na configuração. Todos os nós de servidor de aplicativos requerem que o arquivo `splicer.properties` esteja no caminho especificado para persistir as sessões corretamente.

**Atenção:** Se desejar ativar a persistência de sessões que usam a regravação de URL, configure a propriedade **useURLEncoding** como `true` no arquivo `splicer.properties`.

Para obter informações adicionais sobre as propriedades no arquivo `splicer.properties`, consulte “Arquivo `splicer.properties`” na página 318.

## Resultados

Você configurou o gerenciador de sessões HTTP para persistir as sessões para uma grade de dados. As entradas são removidas da grade de dados quando as sessões expiram. Consulte Configurações de Gerenciamento de Sessões para obter informações adicionais sobre como atualizar o valor do tempo limite da sessão no console administrativo do WebSphere Application Server.

*Configurações de Gerenciamento de Sessão do eXtreme Scale:*

É possível configurar os aplicativos do WebSphere Application Server para usar o WebSphere eXtreme Scale ou um WebSphere DataPower XC10 Appliance para persistência de sessão.

Essas configurações podem ser editadas no assistente de instalação do aplicativo corporativo ou na página de detalhes do aplicativo ou do servidor:

- Versão 6.1: **Aplicativos** > **Instalar Novo Aplicativo**
- Versão 6.1: **Aplicativos** > **Aplicativos Corporativos** > *application\_name*
- Versão 6.1: **Servidores** > **Servidores de Aplicativos** > *server\_name* > **Configurações do Contêiner da Web** > **Gerenciamento de Sessões**
- Versão 7.0: **Aplicativos** > **Novo Aplicativo** > **Novo Aplicativo Corporativo**, e escolha o caminho Detalhado para criar o aplicativo.
- Versão 7.0: **Aplicativos** > **Tipos de Aplicativo** > **Aplicativos Corporativos do WebSphere** > *application\_name* > **Propriedades do Módulo da Web** > **Gerenciamento de Sessões** > **Configurações de Gerenciamento de Sessões**
- Versão 7.0: **Servidores** > **Tipos de Servidor** > **Servidores de Aplicativos do WebSphere** > *server\_name* > **Configurações do Contêiner** > **Configurações de Gerenciamento de Sessão**

*Ativar o Gerenciamento de Sessões:*

Permite que o gerenciamento de sessões use o WebSphere eXtreme Scale integrado, a grade de dados remota ou um WebSphere DataPower XC10 Appliance para persistência de sessão.

*Gerenciar Persistência de Sessão por:*

Especifica como a persistência de sessão é gerenciada. Escolha uma das seguintes opções:

- WebSphere DataPower XC10 Appliance
- Grade de dados do eXtreme Scale remota
- Grade de dados do eXtreme Scale integrada

As configurações restantes que podem ser definidas dependem do mecanismo de persistência de sessão escolhido.

*Configurações Específicas do WebSphere DataPower XC10 Appliance:*

As seguintes configurações são específicas para configurar o WebSphere DataPower XC10 Appliance para persistência de sessão.

*IP ou Nome do Host do WebSphere DataPower XC10 Appliance:*

Especifica o IP ou o nome do host do dispositivo a ser usado para persistir as sessões.

*Credenciais Administrativas do IBM WebSphere DataPower XC10 Appliance:*

Especifica o **Nome de Usuário** e a **Senha** usados para efetuar login na interface com o usuário do DataPower XC10 Appliance. Clique em **Testar Conexão...** para testar a conexão com seu dispositivo.

*Preferência de Persistência de Sessão:*

Especifica a grade de dados na qual as sessões são persistidas. Escolha uma das seguintes opções:

- **Persistir sessões em uma nova grade de dados no IBM WebSphere DataPower XC10 Appliance.** É possível especificar um **Nome da grade de dados**.
- **Persistir sessões em uma grade de dados existente no IBM WebSphere DataPower XC10 Appliance.** É possível inserir ou procurar por um **Nome da grade de dados existente**.

*Configuração da Grade de Dados do eXtreme Scale Remota:*

As seguintes definições são específicas para configurar a grade do eXtreme Scale remota para persistência de sessão.

*Domínio de Serviço de Catálogo que Gerencia a Grade de Dados de Sessão Remota:*

Especifica o domínio de serviço de catálogo que deseja usar para gerenciar suas sessões.

Se nenhum domínio de serviço de catálogo for exibido ou se desejar criar um novo domínio de serviço de catálogo, clique em **Administração do Sistema > WebSphere eXtreme Scale > Domínios de Serviço de Catálogo**.

*Grade de Dados Remota na qual as Informações de Sessão São Armazenadas:*

Especifica o nome da grade de dados no domínio de serviço de catálogo no qual deseja armazenar suas informações de sessão. A lista de grades remotas ativas é preenchida quando um serviço de catálogo é selecionado. A grade de dados remota já deve existir na configuração do eXtreme Scale.

*Configuração da Grade de Dados do eXtreme Scale Integrada:*

As seguintes especificações são próprias para definir uma configuração do eXtreme Scale integrado. No cenário do eXtreme Scale integrado, os processos do eXtreme Scale são hospedados pelos processos do WebSphere Application Server.

*Configuração da Grade de Dados Integrada do eXtreme Scale:*

- **Usar a configuração do ObjectGrid padrão**
- **Especifique os arquivos de configuração do ObjectGrid customizados**

**Caminho completo para o arquivo `objectgrid.xml` para copiar na configuração**

Especifica o caminho completo para o arquivo `objectgrid.xml` para a configuração que deseja usar.



### Caminho completo para o arquivo `objectgriddeployment.xml` para copiar na configuração

Especifica o caminho completo para o arquivo `objectgriddeployment.xml` para a configuração que deseja usar.

## Usando o WebSphere eXtreme Scale para Gerenciamento de Sessão SIP

É possível usar o WebSphere eXtreme Scale como um mecanismo de replicação Session Initiation Protocol (SIP) como uma alternativa confiável ao data replication service (DRS) para replicação de sessão de SIP.

### Configuração do Gerenciamento de Sessão de SIP

Para usar o WebSphere eXtreme Scale como o mecanismo de replicação SIP, configure a propriedade personalizada `com.ibm.sip.ha.replicator.type`. No console administrativo, selecione **Servidores de aplicativos > *my\_application\_server* > contêiner de SIP > Propriedades personalizadas** para cada servidor para incluir a propriedade personalizada. Digite `com.ibm.sip.ha.replicator.type` para o Nome e `OBJECTGRID` para o Valor.

Utilize as seguintes propriedades para customizar o comportamento do ObjectGrid utilizado para armazenar as sessões de SIP. No console administrativo, clique em **Servidores de aplicativos > *my\_application\_server* > Contêiner de SIP > Propriedades personalizadas** para cada servidor para incluir a propriedade personalizada. Digite o **Nome** e **Valor**. Cada servidor deve ter o mesmo conjunto de propriedades para funcionar apropriadamente.

Tabela 23. Propriedades Personalizadas para Gerenciamento de Sessão SIP com ObjectGrid

Propriedade	Valor	Padrão
<code>com.ibm.sip.ha.replicator.type</code>	OBJECTGRID: use ObjectGrid como armazenamento de sessão SIP	
<code>min.synchronous.replicas</code>	Número mínimo de réplicas síncronas	0
<code>max.synchronous.replicas</code>	Número máximo de réplicas síncronas	0
<code>max.asynchronous.replicas</code>	Número máximo de réplicas assíncronas	1
<code>auto.replace.lost.shards</code>	Consulte "Configurando Implementações Distribuídas" na página 232 para obter informações adicionais.	true
<code>development.mode</code>	<ul style="list-style-type: none"><li>• true - permite que as réplicas sejam ativadas no mesmo nó das primárias</li><li>• false - as réplicas devem estar em um nó diferente do das primárias</li></ul>	false

## Configurando o Gerenciador de Sessões HTTP com WebSphere Portal

É possível persistir as sessões HTTP a partir do WebSphere Portal em uma grade de dados.

### Antes de Iniciar

O ambiente do WebSphere eXtreme Scale e do WebSphere Portal devem atender aos seguintes requisitos:

- A maneira como você instala o WebSphere eXtreme Scale depende do seu cenário de implementação. É possível executar os servidores de contêiner, que hospeda as grades de dados, dentro ou fora da célula WebSphere Application Server:



- Se você estiver executando os servidores de contêiner na célula WebSphere Application Server (**cenário integrado**): Instale ambos, o servidor e o cliente WebSphere eXtreme Scale em seus nós WebSphere Application Server e WebSphere Portal.
- Se você estiver executando servidores de contêiner fora da célula WebSphere Application Server (**cenário remoto**): Instale o WebSphere eXtreme Scale Client nos seus nós WebSphere Application Server e WebSphere Portal.

Consulte “Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client com WebSphere Application Server” na página 161 para obter mais informações.

- WebSphere Portal Versão 7 ou posterior.
- Os portlets customizados devem ser configurados no WebSphere Portal. Os portlets administrativos fornecidos com o WebSphere Portal não podem ser atualmente integrados com grades de dados.

## Sobre Esta Tarefa

Apresentar o WebSphere eXtreme Scale em um ambiente a um ambiente WebSphere Portal pode ser benéfico nos seguintes cenários:

**Importante:** Embora os seguintes cenários apresentem benefícios, o uso de processador aumentado na camada WebSphere Portal pode ser resultante da introdução de WebSphere eXtreme Scale no ambiente.

- **Quando a persistência de sessão for necessária.**

Por exemplo, se os dados da sessão a partir dos portlets customizados tiverem que permanecer disponíveis durante uma falha do WebSphere Portal Server, será possível persistir nas sessões HTTP para a grade de dados WebSphere eXtreme Scale. As réplicas de dados entre vários servidores, aumentando a disponibilidade de dados.

- **Em uma topologia de diversos datacenters.**

Se a sua topologia ultrapassar os diversos datacenters em diferentes locais físicos, será possível persistir nas sessões WebSphere Portal HTTP para a grade de dados WebSphere eXtreme Scale. As sessões são replicadas nas grades de dados dos datacenters. Se um datacenter falhar, as sessões serão revertidas para outro datacenter que tem uma cópia dos dados da grade de dados.

- **Para diminuir os requisitos de memória na camada WebSphere Portal Server.**

Ao fazer transferência dos dados da sessão para uma camada remota dos servidores de contêiner, um subconjunto de sessões ficam nos servidores WebSphere Portal. Esta transferência de dados reduz os requisitos de memória na camada do WebSphere Portal Server.

## Procedimento

1. Divida o aplicativo wps WebSphere Portal e qualquer portlet customizado para permitir que as sessões sejam armazenadas na grade de dados.  
É possível combinar o aplicativo configurando o gerenciamento de sessões HTTP ao implementar o aplicativo ou é possível usar as propriedades customizadas para combinar automaticamente os seus aplicativos. Consulte “Configurando o Gerenciador de Sessões HTTP com WebSphere Application Server” na página 296 para obter informações adicionais sobre a combinação do aplicativo.
2. Se você estiver usando o cenário remoto, no qual os servidores de contêiner estão fora do WebSphere Application Server, inicie explicitamente os

contêineres eXtreme Scale remoto para os cenários de persistência de sessão HTTP remota. Inicie os contêineres com os arquivos de configuração XS/ObjectGrid/session/samples/objectGridStandAlone.xml e objectGridDeploymentStandAlone.xml. Por exemplo, você pode usar o seguinte comando:

```
startOgServer.sh xsContainer1 -catalogServiceEndpoints <host>:<port>  
-objectgridFile XS/ObjectGrid/session/samples/objectGridStandAlone.xml -deploymentPolicyFile  
XS/ObjectGrid/session/samples/objectGridDeploymentStandAlone.xml
```

Para obter informações adicionais sobre o início dos servidores de contêiner, consulte “Iniciando Servidores de Contêiner” na página 390. Se você estiver usando um cenário integrado, consulte “Configurando Servidores de Contêiner no WebSphere Application Server” na página 270 para obter informações adicionais sobre a configuração e o início dos servidores de contêiner.

3. Reinicie os servidores do WebSphere Portal. Consulte WebSphere Portal Versão 7: Iniciando e Parando Servidores, Gerenciadores de Implementação e Agentes de Nó para obter informações adicionais.

## Resultados

É possível acessar os dados de sessão WebSphere Portal Server e HTTP para que os portlets customizados configurados sejam persistidos na grade de dados.

Se a grade de dados inteira que está hospedando os dados da sessão do aplicativo estiver inacessível a partir do cliente do contêiner da Web, o cliente usará no lugar o contêiner da web base do WebSphere Application Server para o gerenciamento de sessão. A grade de dados pode estar inatingível nos cenários a seguir:

- Um problema de rede entre o contêiner da web e os servidores de contêiner remotos.
- Os processos do servidor de contêiner remoto foram interrompidos.

O número de referências de sessão mantidas na memória, especificado pelo parâmetro **sessionTableSize**, ainda é mantido quando as sessões são armazenadas no contêiner da web de base. As sessões usadas menos recentemente são invalidadas a partir do cache de sessão do contêiner da web quando o valor **sessionTableSize** é excedido. Se a grade de dados remota se tornar disponível, as sessões que foram invalidadas a partir do cache de contêiner da web poderão recuperar os dados da grade de dados remota e carregar os dados em uma nova sessão. Se a grade de dados remota inteira não estiver disponível e a sessão for invalidada no cache de sessão, os dados de sessão do usuário serão perdidos. Devido a esse problema, você não deve encerrar a grade de dados remota de produção inteira quando o sistema estiver sendo executado sob carga.

## Configurando o Gerenciador de Sessões HTTP para Vários Servidores de Aplicativos

O WebSphere eXtreme Scale é fornecido com uma implementação de gerenciamento de sessões que substitui o gerenciador de sessões padrão para um contêiner da web. Esta implementação fornece replicação de sessão, alta disponibilidade, melhor escalabilidade e opções de configuração. É possível ativar o gerenciador de replicação de sessão do WebSphere eXtreme Scale e a inicialização de contêiner de ObjectGrid integrada genérica.

## Sobre Esta Tarefa

É possível usar o gerenciador de sessões HTTP com outros servidores de aplicativos que não estejam executando o WebSphere Application Server, como

WebSphere Application Server Community Edition. Para configurar outros servidores de aplicativos para usar a grade de dados, você deve unir seu aplicativo e incorporar os arquivos Java archive (JAR) do WebSphere eXtreme Scale no seu aplicativo.

## Procedimento

1. Una seu aplicativo para que ele possa utilizar o gerenciador de sessões. Para usar o gerenciador de sessões, você deve incluir as declarações de filtro apropriadas nos descritores de implementação da Web para o aplicativo. Além disso, os parâmetros de configuração do gerenciador de sessões são passados no gerenciador de sessões no formato de parâmetros de inicialização de contexto do servlet nos descritores de implementação. Há três formas nas quais é possível introduzir essas informações no seu aplicativo:

- Script **addObjectGridFilter**:

Use o script de linha de comandos fornecido junto com o eXtreme Scale para unir um aplicativo com as declarações de filtro e com as configuração no formato de parâmetros de inicialização de contexto do servlet. O script `wxs_home/session/bin/addObjectGridFilter.sh|bat` utiliza dois parâmetros: o caminho absoluto para o arquivo de archive corporativo (EAR) ou o arquivo web archive (WAR) que você deseja unir e o caminho absoluto para o arquivo de propriedades do splicer que contém várias propriedades de configuração. O formato de uso deste script é o seguinte:

**Windows**

```
addObjectGridFilter.bat <ear_or_war_file> <splicer_properties_file>
```

**UNIX**

```
addObjectGridFilter.sh <ear_or_war_file> <splicer_properties_file>
```

**UNIX**

Exemplo usando o eXtreme Scale instalado em um diretório independente no UNIX:

- a. `cd wxs_home/session/bin`
- b. `addObjectGridFilter.sh /tmp/mySessionTest.ear wxs_home/session/samples/splicer.properties`

O filtro do servlet que é unido mantém os padrões para os valores de configuração. É possível substituir estes valores-padrão com opções de configuração especificados no arquivo de propriedades no segundo argumento. Para obter uma lista dos parâmetros que podem ser utilizados, consulte “Parâmetros de inicialização do contexto do servlet” na página 316.

É possível modificar e usar o arquivo `splicer.properties` de amostra que é fornecido com a instalação do eXtreme Scale. Também é possível usar o script **addObjectGridServlets**, que insere o gerenciador de sessões estendendo cada servlet. No entanto, o script recomendado é o **addObjectGridFilter**.

- Script de construção Ant:

O WebSphere eXtreme Scale é fornecido com um arquivo `build.xml` que pode ser usado pelo Apache Ant, que é incluído na pasta `was_root/bin` de uma instalação WebSphere Application Server. É possível modificar o arquivo `build.xml` para alterar as propriedades de configuração do gerenciador de sessões. As propriedades de configuração são idênticas aos nomes de propriedades no arquivo `splicer.properties`. Depois que o arquivo `build.xml` tiver sido modificado, chame o processo Ant executando `ant.sh`, `ws_ant.sh` (UNIX) ou `ant.bat`, `ws_ant.bat` (Windows).

- Atualize o descritor da web manualmente:

Edite o arquivo `web.xml` que é empacotado com o aplicativo da Web para incorporar a declaração do filtro, seu mapeamento de servlet e os parâmetros de inicialização de contexto do servlet. Não use este método porque ele é propenso a erros.

Para obter uma lista dos parâmetros que podem ser utilizados, consulte “Parâmetros de inicialização do contexto do servlet” na página 316.

2. Incorpore os arquivos JAR do gerenciador de replicação de sessão do WebSphere eXtreme Scale ao seu aplicativo. É possível integrar os arquivos ao diretório `WEB-INF/lib` do módulo aplicativo ou ao caminho de classe do servidor de aplicativos. Os arquivos JAR necessários variam dependendo do tipo de contêineres que você está usando:
  - Servidores de contêiner remotos: `ogclient.jar` e `sessionobjectgrid.jar`
  - Servidores de contêiner integrados: `objectgrid.jar` e `sessionobjectgrid.jar`
3. Opcional: Se os servidores de contêiner remotos forem usados, inicie os servidores de contêiner. Veja detalhes em “Iniciando Servidores de Contêiner” na página 390.
4. Implemente o aplicativo. Implemente o aplicativo com seu conjunto normal de etapas para um servidor ou cluster. Após implementar o aplicativo, é possível iniciar o aplicativo.
5. Acesse o aplicativo. É possível acessar o aplicativo, que interage com o gerenciador de sessões e o WebSphere eXtreme Scale.

## O que Fazer Depois

É possível alterar a maioria dos atributos de conexão para o gerenciador de sessões quando você instrumenta seu aplicativo para utilizar o gerenciador de sessões. Esses atributos incluem variações para o tipo de replicação (síncrona ou assíncrona), para o tamanho da tabela de sessão na memória e assim por diante. Não considerando os atributos que podem ser alterados no momento da instrumentação do aplicativo, os únicos outros atributos de configuração que podem ser alterados após a implementação do aplicativo são os atributos que estão relacionados à topologia em cluster do servidor WebSphere eXtreme Scale e a maneira pela qual seus clientes (gerenciadores de sessões) se conectam a eles.

Comportamento do cenário remoto: Se a grade de dados inteira que está hospedando os dados da sessão de aplicativo ficar inacessível a partir do cliente do contêiner da web, o cliente em vez disso usará o contêiner da web de base do servidor de aplicativos para gerenciamento de sessões. A grade de dados pode estar inatingível nos cenários a seguir:

- Um problema de rede entre o contêiner da web e os servidores de contêiner remotos.
- Os processos do servidor de contêiner remoto foram interrompidos.

O número de referências de sessão mantidas na memória, especificado pelo parâmetro `sessionTableSize`, ainda é mantido quando as sessões são armazenadas no contêiner da web de base. As sessões usadas menos recentemente são invalidadas a partir do cache de sessão do contêiner da web quando o valor `sessionTableSize` é excedido. Se a grade de dados remota se tornar disponível, as sessões que foram invalidadas a partir do cache de contêiner da web poderão recuperar os dados da grade de dados remota e carregar os dados em uma nova sessão. Se a grade de dados remota inteira não estiver disponível e a sessão for invalidada a partir do cache de sessão, os dados da sessão do usuário serão perdidos. Devido a esse problema, não encerre a grade de dados remota de produção inteira quando o sistema estiver sendo executado sob carga.

## Arquivos XML para Configuração do Gerenciador de Sessões HTTP

Quando você inicia um servidor de contêiner que armazena dados da sessão HTTP, é possível usar os arquivos XML padrão ou você pode especificar arquivos XML customizados. Estes arquivos criam nomes de ObjectGrid específicos, número de réplicas, etc.

### Local dos Arquivos de Amostra

Esses arquivos XML são compactados no diretório *wxs\_install\_root/ObjectGrid/session/samples* para uma instalação independente ou no diretório *was\_root/optionalLibraries/ObjectGrid/session/samples* para o WebSphere eXtreme Scale instalado em uma célula do WebSphere Application Server .

### Pacote XML Integrado

Se estiver configurando um cenário integrado, o servidor de contêiner iniciará na camada do contêiner da web. Use o arquivo *objectGrid.xml* e o arquivo *objectGridDeployment.xml*, os quais são fornecidos por padrão. É possível atualizar estes arquivos para customizar o comportamento do gerenciador de sessões HTTP.

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd" xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="session" txTimeout="30">
      <bean id="ObjectGridEventListener" className="com.ibm.ws.xs.sessionmanager.SessionHandleManager"/>
      <backingMap name="objectgridSessionMetadata" pluginCollectionRef="objectgridSessionMetadata" readOnly="false"
        lockStrategy="PESSIMISTIC" ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="NO_COPY"/>
      <backingMap name="objectgridSessionAttribute.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
        ttlEvictorType="NONE" copyMode="NO_COPY"/>
      <backingMap name="objectgridSessionTTL.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
        ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="NO_COPY"/>
    </objectGrid>
  </objectGrids>
  <backingMapPluginCollections>
    <backingMapPluginCollection id="objectgridSessionMetadata">
      <bean id="MapEventListener" className="com.ibm.ws.xs.sessionmanager.MetadataMapListener"/>
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

Figura 35. Arquivo *objectGrid.xml*

### Os valores que podem ser alterados:

#### Atributo de nome do ObjectGrid

O valor deve corresponder aos valores a seguir em outros arquivos de configuração:

- A propriedade **objectGridName** no arquivo *splicer.properties* que é usado para unir o aplicativo da Web.
- O atributo **objectgridName** no arquivo *objectGridDeployment.xml*.

Se você tiver diversos aplicativos, e desejar que os dados da sessão sejam armazenados em diferentes grades de dados, esses aplicativos deverão ter diferentes valores de atributo de nome do ObjectGrid.

#### 7.1.1+ Atributo **txTimeout** do ObjectGrid

Este valor determina quantos segundos uma transação pode ser aberta antes que o servidor de contêiner acione a transação para atingir o tempo limite. O padrão é 30 segundos e pode ser alterado, dependendo do ambiente. Se a persistência de sessão HTTP for configurada com o valor do parâmetro de inicialização de contexto de servlet **replicationInterval**

configurado como maior que zero, as transações serão colocadas em lote em um encadeamento. O comprimento da transação é determinado pelo tamanho e número de sessões HTTP. O valor padrão é 10 segundos e 30 segundos é o tempo suficiente para essas transações concluírem. Se a propriedade **replicationInterval** for configurada como 0, uma transação normalmente será iniciada quando um aplicativo da web recuperar um objeto HttpSession válido. A transação é confirmada no final da solicitação de aplicativo da web. Se o seu ambiente possuir solicitações que demoram mais do que 30 segundos, configure este valor adequadamente.

**Os valores que não podem ser alterados:**

#### **ObjectGridEventListener**

A linha do ObjectGridEventListener não pode ser alterada e é usada internamente.

#### **objectgridSessionMetadata**

A linha de objectgridSessionMetadata se refere ao mapa no qual os metadados da sessão HTTP são armazenados. Há uma entrada para cada sessão HTTP armazenada na grade de dados neste mapa.

#### **objectgridSessionTTL.\***

Este valor não pode ser alterado e destina-se para uso futuro.

#### **objectgridSessionAttribute.\***

O texto objectgridSessionAttribute.\* define um mapa dinâmico. Este valor é usado para criar o mapa no qual atributos de sessão HTTP são armazenados quando o parâmetro **fragmentedSession** é configurado como true no arquivo splicer.properties. Este mapa dinâmico é chamado de objectgridSessionAttribute. Um outro mapa é criado com base neste modelo chamado objectgridSessionAttributeEvicted, o qual armazena sessões que atingiram o tempo limite, mas o contêiner da web não foi invalidado.

A linha **MapEventListener** é interna e não pode ser modificada

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="session">
    <mapSet name="sessionMapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="0"
maxAsyncReplicas="1" developmentMode="false" placementStrategy="PER_CONTAINER">
      <map ref="objectgridSessionMetadata"/>
      <map ref="objectgridSessionAttribute.*"/>
      <map ref="objectgridSessionTTL.*"/>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Figura 36. Arquivo objectGridDeployment.xml

**Os valores que podem ser alterados:**

#### **Atributo de nome do ObjectGrid**

O valor deve corresponder aos valores a seguir em outros arquivos de configuração:

- A propriedade **objectGridName** no arquivo splicer.properties que é usado para unir o aplicativo da Web.
- O atributo **name** do ObjectGrid no arquivo objectGrid.xml.



Se você tiver diversos aplicativos, e desejar que os dados da sessão sejam armazenados em diferentes grades de dados, esses aplicativos deverão ter diferentes valores de atributo de nome do ObjectGrid.

#### **Atributos do elemento mapSet**

É possível alterar todas as propriedades de mapSet, exceto para o atributo placementStrategy.

**Nome** Pode ser atualizado para qualquer valor.

#### **numberOfPartitions**

Especifica o número de partições primárias que são iniciadas em cada servidor que está hospedando o aplicativo da Web. Conforme você inclui partições, os dados se tornam mais espalhados no caso de um failover. O valor padrão é de 5 partições e é adequado para a maioria dos aplicativos.

#### **minSyncReplicas, maxSyncReplicas e maxAsyncReplicas**

Especifica o número e o tipo de réplicas que armazenam os dados da sessão HTTP. O padrão é 1 réplica assíncrona, o qual é adequado para a maioria dos aplicativos. A replicação síncrona ocorre durante o caminho do pedido, que pode aumentar os tempos de resposta para seu aplicativo da Web.

#### **developmentMode**

Informa o serviço de posicionamento do eXtreme Scale se os shards de réplica para uma partição podem ser posicionados no mesmo nó que seu shard primário. É possível configurar o valor para true em um ambiente de desenvolvimento, mas desative esta função em um ambiente de produção, pois uma falha do nó pode causar a perda de dados da sessão.

#### **placementStrategy**

Não altere o valor deste atributo.

O restante do arquivo faz referência aos mesmos nomes do mapa que no arquivo objectGrid.xml. Esses nomes não podem ser alterados.

#### **Valores que não podem ser alterados:**

- O atributo placementStrategy no elemento mapSet.

#### **Pacote XML Remoto**

Quando você está usando o modo remoto, no qual os contêineres são executados como processos independentes, você deve usar o arquivo objectGridStandAlone.xml e o arquivo objectGridDeploymentStandAlone.xml para iniciar os processos. É possível atualizar esses arquivos para modificar a configuração.



```

<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
<objectGrids>
<objectGrid name="session" txTimeout="30">
<bean id="ObjectGridEventListener" className="com.ibm.ws.xs.sessionmanager.SessionHandleManager"/>
<backingMap name="objectgridSessionMetadata" pluginCollectionRef="objectgridSessionMetadata"
readOnly="false" lockStrategy="PESSIMISTIC" ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600"
copyMode="COPY_TO_BYTES"/>
<backingMap name="objectgridSessionAttribute.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
ttlEvictorType="NONE" copyMode="COPY_TO_BYTES"/>
<backingMap name="objectgridSessionTTL.*" template="true" readOnly="false" lockStrategy="PESSIMISTIC"
ttlEvictorType="LAST_ACCESS_TIME" timeToLive="3600" copyMode="COPY_TO_BYTES"/>
</objectGrid>
</objectGrids>
<backingMapPluginCollections>
<backingMapPluginCollection id="objectgridSessionMetadata">
<bean id="MapEventListener" className="com.ibm.ws.xs.sessionmanager.MetadataMapListener"/>
</backingMapPluginCollection>
</backingMapPluginCollections>
</objectGridConfig>

```

Figura 37. Arquivo *objectGridStandAlone.xml*

### Os valores que podem ser alterados:

#### Atributo de nome do ObjectGrid

O valor deve corresponder aos valores a seguir em outros arquivos de configuração:

- A propriedade **objectGridName** no arquivo *splicer.properties* que é usado para unir o aplicativo da Web.
- O atributo **objectgridName** no arquivo *objectGridStandAlone.xml*.

Se você tiver diversos aplicativos, e desejar que os dados da sessão sejam armazenados em diferentes grades de dados, esses aplicativos deverão ter diferentes valores de atributo de nome do ObjectGrid.

#### 7.1.1+ Atributo txTimeout do ObjectGrid

Este valor determina quantos segundos uma transação pode ser aberta antes que o servidor de contêiner acione a transação para atingir o tempo limite. O padrão é 30 segundos e pode ser alterado, dependendo do ambiente. Se a persistência de sessão HTTP for configurada com o valor do parâmetro de inicialização de contexto de servlet **replicationInterval** configurado como maior que zero, as transações serão colocadas em lote em um encadeamento. O comprimento da transação é determinado pelo tamanho e número de sessões HTTP. O valor padrão é 10 segundos e 30 segundos é o tempo suficiente para essas transações concluírem. Se a propriedade **replicationInterval** for configurada como 0, uma transação normalmente será iniciada quando um aplicativo da web recuperar um objeto *HttpSession* válido. A transação é confirmada no final da solicitação de aplicativo da web. Se o seu ambiente possuir solicitações que demoram mais do que 30 segundos, configure este valor adequadamente.

### Os valores que não podem ser alterados:

#### ObjectGridEventListener

A linha do *ObjectGridEventListener* não pode ser alterada e é usada internamente.

#### objectgridSessionMetadata

A linha de *objectgridSessionMetadata* se refere ao mapa no qual os metadados da sessão HTTP são armazenados. Há uma entrada para cada sessão HTTP armazenada na grade de dados neste mapa.

#### objectgridSessionTTL.\*

Este valor não pode ser alterado e destina-se para uso futuro.

### **objectgridSessionAttribute.\***

O texto `objectgridSessionAttribute.*` define um mapa dinâmico. Este valor é usado para criar o mapa no qual atributos de sessão HTTP são armazenados quando o parâmetro `fragmentedSession` é configurado como `true` no arquivo `splicer.properties`. Este mapa dinâmico é chamado de `objectgridSessionAttribute`. Um outro mapa é criado com base neste modelo chamado `objectgridSessionAttributeEvicted`, o qual armazena sessões que atingiram o tempo limite, mas o contêiner da web não foi invalidado.

A linha `MetadataMapListener` é interna e não pode ser modificada.

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">

  <objectgridDeployment objectgridName="session">
    <mapSet name="sessionMapSet" numberOfPartitions="5" minSyncReplicas="0" maxSyncReplicas="0"
      maxAsyncReplicas="1" developmentMode="false" placementStrategy="PER_CONTAINER">
      <map ref="objectgridSessionMetadata"/>
      <map ref="objectgridSessionAttribute.*"/>
      <map ref="objectgridSessionTTL.*"/>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

Figura 38. Arquivo `objectGridDeploymentStandAlone.xml`

### **Os valores que podem ser alterados:**

#### **Atributo `objectgridName`**

O valor deve corresponder aos valores a seguir em outros arquivos de configuração:

- A propriedade `objectGridName` no arquivo `splicer.properties` que é usado para unir o aplicativo da Web.
- O atributos `name` do `ObjectGrid` no arquivo `objectGrid.xml`.

Se tiver diversos aplicativos, e desejar que os dados da sessão sejam armazenados em diferentes grades de dados, esses aplicativos deverão ter diferentes valores de atributo de nome do `ObjectGrid`.

#### **Atributos do elemento `mapSet`**

É possível alterar todas as propriedades de `mapSet`, exceto para o atributo `placementStrategy`.

**Nome** Pode ser atualizado para qualquer valor.

#### **`numberOfPartitions`**

Especifica o número de partições primárias que são iniciadas em cada servidor que está hospedando o aplicativo da Web. Conforme você inclui partições, os dados se tornam mais espalhados no caso de um failover. O valor padrão é de 5 partições e é adequado para a maioria dos aplicativos.

#### **`minSyncReplicas`, `maxSyncReplicas` e `maxAsyncReplicas`**

Especifica o número e o tipo de réplicas que armazenam os dados da sessão HTTP. O padrão é 1 réplica assíncrona, o qual é adequado para a maioria dos aplicativos. A replicação síncrona

ocorre durante o caminho do pedido, que pode aumentar os tempos de resposta para seu aplicativo da Web.

#### **developmentMode**

Informa o serviço de posicionamento do eXtreme Scale se os shards de réplica para uma partição podem ser posicionados no mesmo nó que seu shard primário. É possível configurar o valor para true em um ambiente de desenvolvimento, mas desative esta função em um ambiente de produção, pois uma falha do nó pode causar a perda de dados da sessão.

#### **placementStrategy**

Não altere o valor deste atributo.

O restante do arquivo faz referência aos mesmos nomes do mapa que no arquivo objectGrid.xml. Esses nomes não podem ser alterados.

#### **Valores que não podem ser alterados:**

- O atributo placementStrategy no elemento mapSet.

### **Parâmetros de inicialização do contexto do servlet**

A lista de parâmetros de inicialização de contexto de servlet a seguir pode ser especificada no arquivo splicer properties conforme requerido no método de conexão escolhido.

#### **Parâmetros**

##### **objectGridType**

Um valor de sequência igual a REMOTE ou EMBEDDED. O padrão é REMOTE.

Se ele estiver configurado como REMOTE, os dados da sessão serão armazenados fora do servidor no qual o aplicativo da web está em execução.

Se ele estiver configurado como EMBEDDED, um contêiner do eXtreme Scale integrado iniciará no processo do servidor de aplicativos no qual o aplicativo da web está em execução.

##### **objectGridName**

Um valor de cadeia que define o nome da instância de ObjectGrid usada para um aplicativo da Web particular. O nome padrão é session.

Esta propriedade deve refletir o objectGridName nos arquivos XML do ObjectGrid e de implementação usados para iniciar os servidores de contêiner do eXtreme Scale.

##### **catalogHostPort**

O servidor de catálogos pode ser contatado para obter uma instância de ObjectGrid do lado do cliente. O valor deve ser no formato host:port<,host:port>. O host é o host listener no qual o servidor de catálogos está em execução. A porta é a porta do listener para esse processo do servidor de catálogos. Esta lista pode ser arbitrariamente longa e é usada apenas para autoinicialização. O primeiro endereço viável é usado. Ele é opcional dentro do WebSphere Application Server se a propriedade **catalog.services.cluster** é configurada.

##### **replicationInterval**

Um valor de número inteiro (em segundos) que define o tempo entre a gravação de sessões atualizadas para ObjectGrid. O padrão é 10 segundos. Os

valores possíveis são de 0 a 60. 0 significa que as sessões atualizadas são gravadas no ObjectGrid no final da chamada de método de serviço de servlet para cada pedido. Um valor de **replicationInterval** mais alto melhora o desempenho porque menos atualizações são gravadas na grade de dados. No entanto, um valor mais alto torna a configuração menos tolerante a falhas.

Esta configuração se aplica somente quando `objectGridType` é configurado como REMOTE.

#### **sessionTableSize**

Um valor de número inteiro que define o número de referências de sessão mantidas na memória. O padrão é 1000.

Esta configuração pertence somente a uma topologia REMOTE porque a topologia EMBEDDED já possui os dados da sessão na mesma camada que o contêiner da web.

As sessões são despejadas a partir da tabela na memória com base na lógica menos utilizada recentemente (LRU). Quando uma sessão é despejada da tabela em memória, ela é invalidada do contêiner da Web. No entanto, os dados não são removidos da grade; portanto, as solicitações subsequentes para essa sessão ainda podem recuperar os dados. Este valor deve ser configurado acima do valor de conjunto de encadeamentos máximo do contêiner da web, que reduz a contenção no cache de sessão.

#### **fragmentedSession**

Um valor de sequência igual a `true` ou `false`. O valor padrão é `true`. Use esta configuração para controlar se o produto armazena dados da sessão como uma entrada inteira ou armazena cada atributo separadamente.

Configure o parâmetro `fragmentedSession` como `true` se a sessão do aplicativo da web tiver muitos atributos ou atributos com tamanhos grandes. Configure `fragmentedSession` como `false` se uma sessão tiver menos atributos, porque todos os atributos são armazenados na mesma chave na grade de dados.

Na implementação anterior, baseada em filtro, esta propriedade foi mencionada como `persistenceMechanism`, com os valores possíveis de `ObjectGridStore` (`fragmentado`) e `ObjectGridAtomicSessionStore` (`não fragmentado`).

#### **securityEnabled**

Um valor de sequência igual a `true` ou `false`. O valor padrão é `false`. Esta configuração ativa a segurança do cliente do eXtreme Scale. Ele deve corresponder à configuração de **securityEnabled** no arquivo de propriedade de servidor do eXtreme Scale. Se as configurações não corresponderem, ocorrerá uma exceção.

#### **credentialGeneratorClass**

O nome da classe que implementa a interface `com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator`. Essa classe é usada para obter credenciais para os clientes.

#### **credentialGeneratorProps**

As propriedades para a classe de implementação `CredentialGenerator`. As propriedades são configuradas para o objeto com o método `setProperty(String)`. O valor `credentialGeneratorProps` é usado apenas se o valor da propriedade **credentialGeneratorClass** não for nulo.

#### **objectGridXML**

O local de arquivo do arquivo `objectgrid.xml`. O arquivo XML integrado empacotado na biblioteca do eXtreme Scale é carregado automaticamente se `objectGridType=EMBEDDED` e a propriedade **`objectGridXML`** não forem especificados.

#### **objectGridDeploymentXML**

Especifica o local do arquivo XML da política de implementação do ObjectGrid. O arquivo XML integrado empacotado na biblioteca do eXtreme Scale é carregado automaticamente se `objectGridType=EMBEDDED` e a propriedade **`objectGridDeploymentXML`** não forem especificados.

#### **traceSpec**

Especifica a especificação de rastreamento do IBM WebSphere como um valor de sequência. Use esta configuração para servidores de aplicativos diferentes do WebSphere Application Server.

#### **traceFile**

Especifica o local do arquivo de rastreamento como um valor de sequência. Use esta configuração para servidores de aplicativos diferentes do WebSphere Application Server.

#### **cookieDomain**

Especifica se você precisa que as sessões fiquem acessíveis nos hosts. Configure o valor para o nome do domínio comum entre os hosts.

#### **reuseSessionID**

Configure como `true` se o contêiner da web subjacente reutilizar IDs de sessão entre solicitações para diferentes hosts. O valor padrão é `false`. O valor desta propriedade deve ser igual ao valor no contêiner da Web. Se estiver usando o WebSphere Application Server e configurando a persistência de sessão HTTP do eXtreme Scale usando o console administrativo ou o script da ferramenta **`wsadmin`**, a propriedade customizada do contêiner da web `HttpSessionIdReuse=true` será incluída por padrão. O **`reuseSessionID`** também é configurado como `true`. Se você não deseja que IDs de sessão sejam reutilizados, configure a propriedade customizada `HttpSessionIdReuse=false` na propriedade customizada do contêiner da web antes de configurar a persistência de sessão do eXtreme Scale.

#### **shareSessionsAcrossWebApps**

Especifica se as sessões são compartilhadas nos aplicativos da Web, especificados como um valor de sequência de `true` ou `false`. O padrão é `false`. A especificação de servlet indica que as Sessões HTTP não podem ser compartilhadas nos aplicativos da Web. Uma extensão para a especificação de servlet é fornecida para permitir este compartilhamento.

#### **useURLEncoding**

Configurado como `true` se desejar ativar a regravação de URL. O valor padrão é `false`, indicando que os cookies são usados para armazenar dados de sessão. O valor deste parâmetro deve ser a mesma que as configurações do contêiner da web para o gerenciamento de sessão.

### **Arquivo `splicer.properties`**

O arquivo `splicer.properties` contém todas as opções de configuração para configurar um gerenciador de sessões baseado em filtro de servlet.

## Propriedades do splicer de Amostra

Se optar por usar qualquer uma das propriedades adicionais que são descritas neste arquivo, certifique-se de remover o comentário das linhas das propriedades que você deseja ativar.

```
# Properties file that contains all the configuration
#
# This properties file can be made to hold all the default
# individual settings can be overridden using ANT Task
# with the filtersplicer ANT task.

# A string value of either "REMOTE" or "EMBEDDED". The default is REMOTE.
# If it is set to "REMOTE", the session data will be stored outside of
# the server on which the web application is running. If it is set to
# "EMBEDDED", an embedded WebSphere eXtreme Scale container will start
# in the application server process on which the web application is running.

objectGridType = REMOTE

# A string value that defines the name of the ObjectGrid
# instance used for a particular web application. O nome padrão
# is session. This property must reflect the objectGridName in both
# the objectgrid xml and deployment xml files used to start the eXtreme
# Scale containers.

objectGridName = session

# Catalog Server can be contacted to obtain a client side
# The value needs to be of the
# form "host:port<,host:port>", where the host is the listener host
# on which the catalog server is running, and the port is the listener
# port for that catalog server process.
# This list can be arbitrarily long and is used for bootstrapping only.
# The first viable address will be used. It is optional inside WebSphere
# if the catalog.services.cluster property is configured.

# catalogHostPort = host:port<,host:port>

# An integer value (in seconds) that defines the time in seconds between
# writing of updated sessions to ObjectGrid. The default is 10. This property
# is only used when objectGridType is set to REMOTE. Possible values are
# from 0 to 60. 0 means updated sessions are written to the ObjectGrid
# at the end of servlet service method call for each request.

replicationInterval = 10

# An integer value that defines the number of session references
# kept in memory. O padrão é 1000. This property is only used when
# objectGridType is set to REMOTE. When the number of sessions stored
# in memory in the web container exceeds this value, the least recently
# accessed session is invalidated from the web container. If a request
# comes in for that session after it's been invalidated, a new session
# will be created (with a new session ID if reuseSessionId=false),
# populated with the invalidated session's attributes. This value should
# always be set to be higher than the maximum size of the web container
# thread pool to avoid contention on this session cache.

sessionTableSize = 1000

# A string value of either "true" or "false", default is "true".
# It is to control whether we store session data as a whole entry
# or store each attribute separately.
# This property was referred to as persistenceMechanism in the
# previous filter-based implementation, with the possible values
# of ObjectGridStore (fragmented) and ObjectGridAtomicSessionStore
# (not fragmented).
```

```

fragmentedSession = true

# A string value of either "true" or "false", default is "false".
# Enables eXtreme Scale client security. This setting needs to match
# the securityEnabled setting in the eXtreme Scale server properties
# file. If the settings do not match, an exception occurs.

securityEnabled = false

# Specifies the client credential authentication support.
# The possible values are:
# Never - The client does not support credential authentication.
# Supported* - The client supports the credential authentication if and only if the server
# supports too.
# Required - The client requires the credential authentication.
# The default value is Supported.

# credentialAuthentication =

# Specifies the retry count for authentication if the credential
# is expired. If the value is set to 0, there will not be
# any authentication retry.

# authenticationRetryCount =

# Specifies the name of the class that implements the
# com.ibm.websphere.objectgrid.security.plugins.CredentialGenerator
# interface. This class is used to get credentials for clients.

# credentialGeneratorClass =

# Specifies the properties for the CredentialGenerator implementation
# class. The properties are set to the object with the setProperties(String)
# method. The credentialGeneratorProps value is used only if the value of the
# credentialGeneratorClass property is not null.

# credentialGeneratorProps =

# The file location of the objectgrid xml file.
# The built-in xml file packaged in the eXtreme Scale library
# will automatically be loaded if this property
# is not specified and if objectGridType=EMBEDDED

# objectGridXML =

# The file location of the objectGrid deployment policy xml file.
# The built-in xml file packaged in the eXtreme Scale library
# will automatically be loaded if this property
# is not specified and if objectGridType=EMBEDDED

# objectGridDeploymentXML =

# A string of IBM WebSphere trace specification,
# useful for all other application servers besides WebSphere.

# traceSpec =

# A string of trace file location.
# useful for all other application servers besides WebSphere.

# traceFile=

# This property should be set if you require sessions to be
# accessible across hosts. The value will be the name of the
# common domain between the hosts.

```



```

# cookieDomain=

# Set to true if the underlying web container will reuse
# session ID's across requests to different hosts. Default is
# false. The value of this should be the same as what is set
# in the web container.

# reuseSessionId=

# A string value of either "true" or "false". The default is
# "false". Per the servlet specification, HTTP Sessions cannot
# be shared across web applications. An extension to the servlet
# specification is provided to allow this sharing.

# shareSessionsAcrossWebApps = false

# Set to true if you want to enable urlRewriting. Default is
# false, which means cookies will be used to store data. The
# value of this should reflect what is set in the web container
# settings for session management.

# useURLEncoding = false

```

## Configurando o Provedor de Cache Dinâmico para o WebSphere eXtreme Scale

A instalação e configuração do provedor de cache dinâmico para o eXtreme Scale depende de quais são seus requisitos e do ambiente configurado.

### Antes de Iniciar

- Para usar o provedor de cache dinâmico, o WebSphere eXtreme Scale deve ser instalado além das implementações de nó do WebSphere Application Server, incluindo o nó do gerenciador de implementação. Consulte “Instalando o WebSphere eXtreme Scale ou o WebSphere eXtreme Scale Client com WebSphere Application Server” na página 161 para obter mais informações.
- A segurança global deve ser ativada no console administrativo do WebSphere Application Server, se os servidores de catálogo em seu domínio de serviço de catálogo tiver o Secure Sockets Layer (SSL) ativado e desejar usar o SSL para um domínio de serviço de catálogo com SSL suportado. Você requer SSL para um servidor de catálogos configurando o atributo `transportType` como `SSL-Required` no Arquivo de Propriedades do Servidor. Para obter informações adicionais sobre como configurar a segurança global, consulte Configurações de Segurança Global.

### Sobre Esta Tarefa

Para obter informações sobre como usar o provedor de cache dinâmico do eXtreme Scale com o IBM WebSphere Commerce, consulte os tópicos a seguir na documentação do IBM WebSphere Commerce:

- Ativando o Serviço de Cache Dinâmico e o Armazenamento em Cache do Servlet
- Ativando o Cache de Dados do WebSphere Commerce

Se você não estiver direcionando especificamente seu armazenamento em cache para uma instância Cache do Objeto ou Cache do Servlet definida, é provável que as chamadas da API do Cache Dinâmico estejam sendo atendidas por `baseCache`. Se desejar usar o provedor de cache dinâmico do eXtreme Scale para o JPS, serviços da Web ou armazenamento em cache de comando, será necessário configurar a instância `baseCache` para usar o provedor de cache dinâmico do

eXtreme Scale. As mesmas propriedades de configuração são usadas para configurar a instância baseCache. Lembre-se também de que essas propriedades de configuração precisam ser definidas como Propriedades customizadas da Java Virtual Machine (JVM). Essa condição se aplica a qualquer propriedade de configuração de cache abordada nessa seção, exceto para armazenamento em cache de servlet. Para usar o eXtreme Scale com o provedor de cache dinâmico para armazenamento em cache de servlet, certifique-se de configurar a ativação nas propriedades do sistema em vez de nas propriedades customizadas.

## Procedimento

1. Ative o provedor de cache dinâmico do eXtreme Scale.

- **WebSphere Application Server Versão 7.0 e superior:**

É possível configurar o serviço de cache dinâmico para usar o provedor de cache dinâmico do eXtreme Scale com o console administrativo. Depois de instalar o eXtreme Scale, o provedor de cache dinâmico do eXtreme Scale é imediatamente disponibilizado como uma opção **Provedor de Cache** no console administrativo. Para obter mais informações, consulte Centro de Informações do WebSphere Application Server Versão 7.0: Selecionando um Provedor de Serviço de Cache.

- **WebSphere Application Server Versão 6.1:**

Use uma propriedade customizada para configurar o serviço de cache dinâmico para usar o provedor de cache dinâmico do eXtreme Scale. Essas propriedades customizadas também podem ser usadas no WebSphere Application Server Versão 7.0 e superior. Para criar uma propriedade personalizada em uma instância de cache, clique em **Recursos > Instâncias de Cache > cache\_instance\_type > cache\_instance\_name > Propriedades Customizadas > Novo**. Se estiver usando a instância de cache base, crie as propriedades customizadas na JVM.

**com.ibm.ws.cache.CacheConfig.cacheProviderName**

Para usar o provedor de cache dinâmico do eXtreme Scale, configure o valor para `com.ibm.ws.objectgrid.dynacache.CacheProviderImpl`. Essa propriedade customizada pode ser criada em uma instância de cache dinâmico ou em uma instância de cache base. Se você optar por configurar a propriedade customizada na instância de cache de base, todas as outras instâncias de cache no servidor usarão o provedor do eXtreme Scale por padrão. Quaisquer propriedades de configuração do provedor de cache dinâmico do eXtreme Scale configuradas para baseCache são as propriedades de configuração padrão para todas as instâncias de cache suportadas pelo eXtreme Scale. Para substituir a instância de cache base e fazer com que uma instância de cache dinâmico particular use o provedor de cache dinâmico padrão, crie a propriedade customizada `com.ibm.ws.cache.CacheConfig.cacheProviderName` na instância de cache dinâmico e configure o valor como `default`.

2. Opcional: Se estiver usando as instâncias de cache replicadas, defina a configuração de replicação para o cache.

Com o provedor de cache dinâmico do eXtreme Scale, você pode ter instâncias de cache locais ou instâncias de cache replicadas. Se você estiver usando apenas instâncias de cache locais, esta etapa poderá ser ignorada.

Use um dos seguintes métodos para configurar o cache replicado:

- Ative a replicação de cache com o console administrativo. É possível ativar a replicação de cache a qualquer momento no WebSphere Application Server Versão Versão 7.0. No WebSphere Application Server Versão 6.1, você deve criar um domínio de replicação DRS.
  - Ative a replicação de cache com a propriedade customizada `com.ibm.ws.cache.CacheConfig.enableCacheReplication` para forçar o cache a informar que isso é um cache replicado, mesmo se o domínio de replicação DRS não tiver sido designado para ele. Configure o valor dessa propriedade customizada para `true`. Configure essa propriedade customizada na instância de cache, se você estiver usando um cache de objeto de cache de servlet, ou na JVM se você estiver utilizando a instância `baseCache`.
3. Opcional: Se você estiver usando o eXtreme Scale como um cache de fragmento de JSP, configure a propriedade customizada `com.ibm.ws.cache.CacheConfig.disableTemplateInvalidation` para `true` para desativar as invalidações baseadas em modelo durante os recarregamentos do JSP.
  4. Configure a topologia para o serviço de cache dinâmico.  
O único parâmetro de configuração necessário para o provedor de cache dinâmico do eXtreme Scale é a topologia do cache. Configure a propriedade customizada na instância de cache ou para o serviço de cache dinâmico se estiver usando a instância `baseCache`. Insira o nome da propriedade customizada como: `com.ibm.websphere.xs.dynacache.topology`.  
Os três valores possíveis para essa propriedade são os seguintes. Você deve usar um dos valores permitidos:
    - `integrado`
    - `integrado_particionado`
    - `remoto`
 Se você estiver usando topologias integradas ou particionadas integradas, considere configurar a propriedade customizada `com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent` para `true` para economizar alguns custos de serialização. Configure essa propriedade customizada na instância de cache ou a JVM se estiver usando a instância `baseCache`.
  5. Opcional: Se você estiver usando uma topologia particionada integrada, configure o número de contêineres iniciais para o serviço de cache dinâmico.  
É possível aumentar o desempenho dos caches que estão usando a topologia particionada integrada ao configurar o número de contêineres iniciais. Configure a variável como uma propriedade de sistema na WebSphere Application Server Java virtual machine.  
Insira o nome da propriedade como:  
`com.ibm.websphere.xs.dynacache.num_initial_containers`.  
O valor recomendado dessa propriedade de configuração é um número inteiro que seja igual ou um pouco menor que o número total de instâncias do WebSphere Application Server que acessam essa instância de cache distribuído. Por exemplo, se um serviço de cache dinâmico for compartilhado entre os membros da grade de dados, o valor deverá ser configurado para o número de membros da grade de dados.  
Para topologias integradas ou `embedded_partitioned`, você deve estar usando a Versão 7.0 do WebSphere Application Server. Configure a seguinte propriedade customizada no processo JVM para garantir que os contêineres iniciais estejam disponíveis imediatamente.  
`com.ibm.ws.cache.CacheConfig.createCacheAtServerStartup=true`

6. Configurar a grade de serviço de catálogo do eXtreme Scale.

Quando estiver usando o eXtreme Scale como o provedor de cache dinâmico para uma instância de cache distribuído, você deverá configurar um domínio do serviço de catálogo do eXtreme Scale.

Um único domínio do serviço de catálogo pode atender vários provedores do serviço de cache dinâmico retornados pelo eXtreme Scale.

Um serviço de catálogo pode ser executado dentro ou fora de processos WebSphere Application Server. Iniciando com o eXtreme Scale Versão 7.1, quando o console administrativo é usado para configurar domínios de serviço de catálogo, o cache dinâmico usa essas configurações. Não é necessário executar etapas adicionais para configurar um serviço de catálogo. Para obter informações adicionais, consulte “Criando Domínios do Serviço de Catálogo no WebSphere Application Server” na página 252.
7. Configurar os objetos de chave customizados.

Quando estiver usando os objetos customizados como chaves, os objetos deverão implementar a interface Serializável ou Externalizável. Quando estiver usando as topologias integradas ou particionadas integradas, é necessário colocar os objetos no caminho de biblioteca compartilhado do WebSphere, como se eles estivessem sendo usados com o provedor de cache dinâmico padrão. Consulte Usando as interfaces DistributedMap e DistributedObjectCache para o cache dinâmico no centro de informações do WebSphere Application Server Network Deployment para obter mais detalhes.

Se você estiver usando a topologia remota, você deverá colocar os objetos de chaves customizados no CLASSPATH para os contêineres do eXtreme Scale independentes. Consulte o “Iniciando Servidores de Contêiner” na página 390 para obter informações adicionais.
8. Opcional: Se você estiver usando uma topologia remota, configure os servidores de contêiner do eXtreme Scale.
  - **Topologia integrada ou particionada integrada:**

Os dados em cache são armazenados nos servidores de contêiner do WebSphere eXtreme Scale. Os servidores de contêiner podem ser executados dentro ou fora dos processos do WebSphere Application Server. O provedor do eXtreme Scale cria automaticamente contêineres dentro do processo do WebSphere quando estiver usando topologias integradas ou particionadas integradas para uma instância de cache. Nenhuma configuração adicional é necessária para essas topologias.
  - **Topologia Remota:**

Quando estiver usando uma topologia remota, você deve inicializar os servidores de contêiner do eXtreme Scale antes que as instâncias do WebSphere Application Server que acessam a instância de cache sejam iniciadas. Consulte as etapas para iniciar os servidores de contêiner independentes no *Guia de Administração* para obter mais informações. Verifique se todos os servidores de contêiner de um serviço de cache dinâmico específico apontam para os mesmos terminais de serviço de catálogo.

Os arquivos de configuração XML para o provedor de cache dinâmico do eXtreme Scale independente estão no diretório `wxs_install_root/customLibraries/ObjectGrid/dynacache/etc` para instalações sobre o WebSphere Application Server ou no diretório `wxs_install_root/ObjectGrid/dynacache/etc` para instalações independentes. Os arquivos são chamados de `dynacache-remote-objectgrid.xml` e `dynacache-remote-definition.xml`. Faça cópias desses arquivos para editar e usar quando estiver iniciando contêineres independentes para o provedor de cache dinâmico do eXtreme

Scale. O parâmetro **numInitialContainers** no arquivo **dynacache-remote-deployment.xml** deve corresponder ao número de processos do contêiner que estão em execução. Observe que o atributo **numberOfPartitions** no arquivo **dynacache-remote-objectgrid.xml** tem um valor padrão de 47.

**Nota:** O conjunto de processos do servidor de contêiner deve ter memória livre suficiente para atender a todas as instâncias de cache dinâmico que estiverem configuradas para usar a topologia remota. Qualquer processo do WebSphere Application Server que compartilham valores iguais ou equivalentes para a propriedade customizada **catalog.services.cluster** deve usar o mesmo conjunto de contêineres independente. O número de contêineres e o número de servidores nos quais eles residem devem ser redimensionados adequadamente. Consulte “Planejamento de Capacidade de Cache Dinâmico” na página 56 para obter detalhes adicionais.

Uma de entrada da linha de comandos que inicia um contêiner independente para o provedor de cache dinâmico do eXtreme Scale é a seguinte:

#### UNIX

```
startOgServer.sh container1 -objectGridFile
../dynacache/etc/dynacache-remote-objectgrid.xml -deploymentPolicyFile
../dynacache/etc/dynacache-remote-deployment.xml -catalogServiceEndPoints
MyServer1.company.com:2809
```

9. Para topologias distribuídas ou integradas, ative o agente de dimensionamento para melhorar as estimativas de consumo de memória.

O agente de dimensionamento faz a estimativa do consumo de memória (estatística **usedBytes**). O agente requer uma JVM Java 5 ou superior.

Carregue o agente incluindo o seguinte argumento na linha de comandos da JVM:

```
-javaagent:WXS lib directory/wxssizeagent.jar
```

Para uma topologia integrada, inclua o argumento na linha de comandos do processo do WebSphere Application Server.

Para uma topologia distribuída, inclua o argumento na linha de comandos dos processos (contêineres) do eXtreme Scale e do processo do WebSphere Application Server.

## Plug-in do Cache JPA Nível 2 (L2)

O WebSphere eXtreme Scale inclui plug-ins de cache de nível 2 (L2) para ambos os provedores OpenJPA e Hibernate Java Persistence API (JPA). Quando você usar um desses plug-ins, seu aplicativo usará a API do JPA. Uma grade de dados é introduzida entre o aplicativo e o banco de dados, melhorando os tempos de resposta.

Usar o eXtreme Scale como um provedor de cache L2 aumenta o desempenho na leitura e consulta de dados e reduz a carga para o banco de dados. O WebSphere eXtreme Scale tem vantagens sobre as implementações de cache integradas porque o cache é automaticamente replicado entre todos os processos. Quando um cliente armazena em cache um valor, todos os outros clientes conseguem usar o valor armazenado em cache que está localmente na memória.

É possível configurar a topologia e as propriedades para o provedor de cache L2 no arquivo **persistence.xml**. Para obter mais informações sobre como configurar essas propriedades, consulte “Propriedades de Configuração do Cache JPA” na página 332.

**Dica:** O plug-in do cache JPA L2 requer um aplicativo que usa as APIs do JPA. Se desejar usar as APIs do WebSphere eXtreme Scale para acessar uma origem de dados JPA, use o carregador JPA. Para obter informações adicionais, consulte Carregadores JPA.

## Considerações sobre Topologia de Cache do JPA L2

Os fatores a seguir afetam qual tipo de topologia configurar:

### 1. Quantos dados você espera que sejam armazenados em cache?

- Se os dados puderem ser ajustados em um único heap de JVM, use o “Topologia Integrada” na página 328 ou o “Topologia Intradomínio” na página 327.
- Se os dados não puderem ser ajustados em um único heap de JVM, use o “Topologia Integrada e Particionada” na página 329 ou o “Topologia Remota” na página 330

### 2. Qual é a proporção de "leitura para gravação" esperada?

A proporção de "leitura para gravação" afeta o desempenho do cache L2. Cada topologia manipula as operações de leitura e gravação de forma diferente.

- “Topologia Integrada” na página 328: leitura local, gravação remota
- “Topologia Intradomínio” na página 327: leitura local, gravação local
- “Topologia Integrada e Particionada” na página 329: Particionado: leitura remota, gravação remota
- “Topologia Remota” na página 330: leitura remota, gravação remota.

Aplicativos que são principalmente somente leitura devem usar topologias integradas e intradomínio quando possível. Aplicativos que executam mais gravação devem usar topologias intradomínio.

### 3. Qual é a porcentagem de dados consultados versus localizados por uma chave?

Quando ativado, as operações de consulta usam o cache de consulta de JPA. Ative o cache de consulta de JPA somente para aplicativos com altas proporções de leitura para gravação, por exemplo quando você está se aproximando de 99% de operações de leitura. Se você usar o cache de consulta de JPA, deverá usar o “Topologia Integrada” na página 328 ou “Topologia Intradomínio” na página 327.

A operação de localização por chave busca uma entidade de destino se a entidade de destino não tem nenhum relacionamento. Se a entidade de destino tiver relacionamentos com o tipo de busca EAGER, esses relacionamentos serão buscados juntamente com a entidade de destino. No cache de dados de JPA, a busca desses relacionamentos faz com que alguns acertos do cache obtenham todos os dados de relacionamento.

### 4. Qual é o nível de deterioração tolerado dos dados?

Em um sistema com algumas JVMs, existe latência de replicação de dados para operações de gravação. O objetivo do cache é manter uma visualização de dados sincronizados final entre todas as JVMs. Quando você estiver usando a topologia intradomínio, existirá um atraso de replicação de dados para operações de gravação. Aplicativos que usam essa topologia devem ser capazes de tolerar leituras antigas e gravações simultâneas que podem sobrescrever os dados.

#### 7.1.1+



## Topologia Intradomínio

Com uma topologia intradomínio, os shards primários são colocados em cada servidor de contêiner na topologia. Esses shards primários contêm o conjunto completo de dados para a partição. Qualquer um desses shards primários também podem concluir as operações de gravação em cache. Essa configuração elimina o gargalo na topologia integrada na qual todas as operações de gravação de cache deve passar por um shard primário único.

Em uma topologia intradomínio, nenhum shard de réplica é criado, mesmo se as réplicas tiverem sido definidas em seus arquivos de configuração. Cada shard primário redundante contém uma cópia completa dos dados, de modo que cada shard primário também pode ser considerado como um shard de réplica. Essa configuração usa uma única partição, semelhante à topologia integrada.

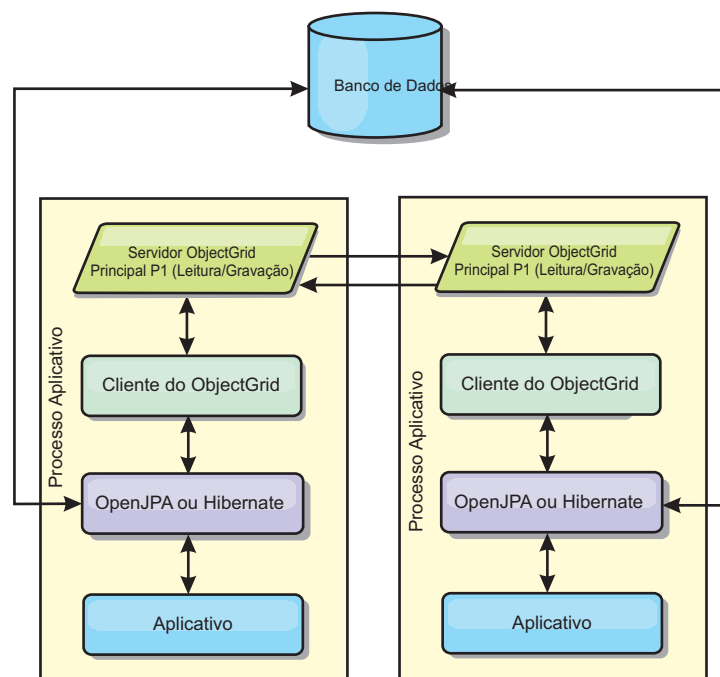


Figura 39. Topologia de Intradomínio do JPA

As propriedades de configuração de cache JPA relacionadas para a topologia intradomínio são:

```
ObjectGridName=objectgrid_name,ObjectGridType=EMBEDDED,PlacementScope=CONTAINER_SCOPE,PlacementScopeTopology=HUB | RING
```

Vantagens:

- As leituras e atualizações do cache serão locais.
- Simples para configurar.

Limitações:

- Essa topologia é mais adequada para quando os servidores de contêiner puderem conter o conjunto inteiro de dados da partição.
- Os shards de réplica, mesmo se estiverem configurados, nunca são colocados porque cada servidor de contêiner hospeda um shard primário. No entanto, todos os shards primários são replicados com os outros shards primários, portanto, esses shards primários se tornam réplicas entre si.



## Topologia Integrada

**Dica:** Considere usar uma topologia intradomínio para obter melhor desempenho.

Uma topologia integrada cria um servidor de contêiner dentro do espaço do processo de cada aplicativo. O OpenJPA e o Hibernate lêem a cópia na memória do cache diretamente e gravam para todas as outras cópias. É possível melhorar o desempenho de gravação usando replicação assíncrona. Esta topologia padrão executa melhor quando a quantidade de dados armazenados em cache é pequena o suficiente para ajustar-se a um único processo. Com uma topologia integrada, crie uma única partição para os dados.

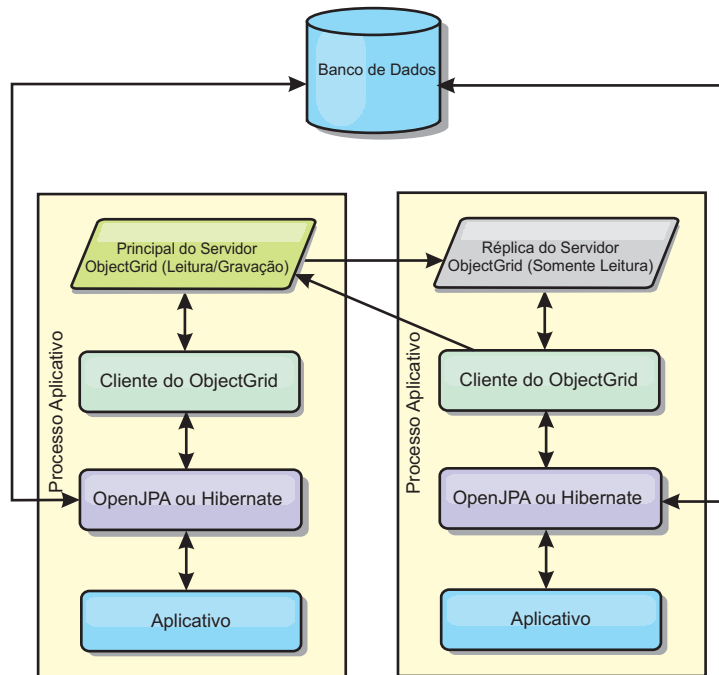


Figura 40. Topologia Integrado do JPA

As propriedades de configuração de cache JPA relacionadas para a topologia integrada são:

```
ObjectGridName=objectgrid_name,ObjectGridType=EMBEDDED,MaxNumberOfReplicas=num_replicas,ReplicaMode=SYNC | ASYNC | NONE
```

Vantagens:

- Todas as leituras de cache são acessos locais rápidos.
- Simples para configurar.

Limitações:

- A quantidade de dados é limitada ao tamanho do processo.
- Todas as atualizações de cache são enviadas por meio de um shard primário, que cria um gargalo.

## Topologia Integrada e Particionada

**Dica:** Considere usar uma topologia intradomínio para obter melhor desempenho.

### CUIDADO:

Não use o cache de consulta de JPA com uma topologia particionada integrada. O cache de consulta armazena os resultados da consulta que são uma coleção de chaves de entidade. O cache de consulta vai para o cache de dados para buscar todos os dados da entidade. Como o cache de dados é dividido entre diversos processos, essas chamadas adicionais podem negar os benefícios do cache L2.

Quando os dados em cache são muito grandes para se ajustarem em um único processo, é possível usar a topologia particionada integrada. Esta topologia divide os dados sobre diversos processos. Os dados são divididos entre os shards primários, portanto, cada shard primário contém um subconjunto dos dados. Ainda é possível usar esta opção quando a latência do banco de dados é alta.

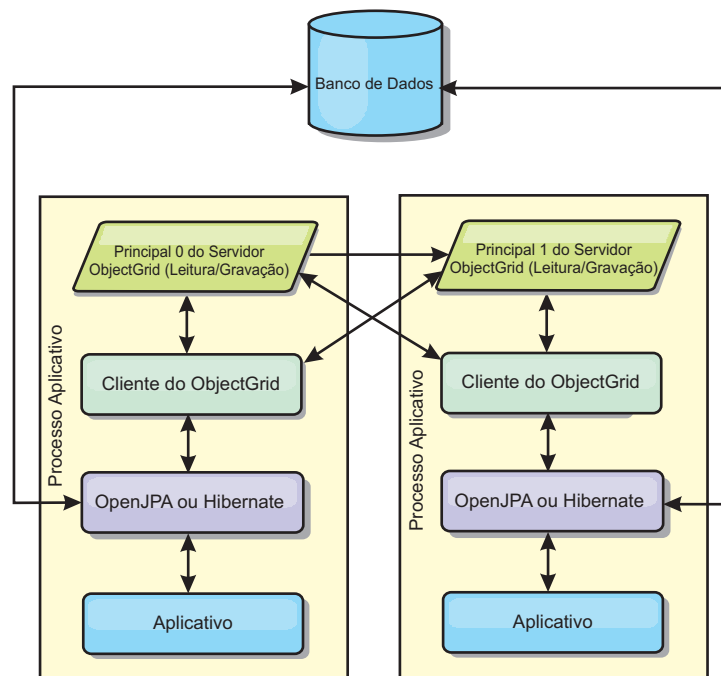


Figura 41. Topologia Particionada Integrada do JPA

As propriedades de configuração de cache do JPA relacionadas à topologia particionada integrada são:

```
ObjectGridName=objectgrid_name,ObjectGridType=EMBEDDED_PARTITION,ReplicaMode=SYNC | ASYNC | NONE,  
NumberOfPartitions=num_partitions,ReplicaReadEnabled=TRUE | FALSE
```

### Vantagens:

- Armazena grandes quantidades de dados.
- Simples para configurar
- As atualizações de cache são propagadas para vários processos.

### Limitação:

- A maioria das leituras e atualizações de cache é remota.

Por exemplo, para armazenar em cache 10 GB de dados com um máximo de 1 GB por JVM, 10 Java Virtual Machines são necessários. Portanto, o número de partições deve ser configurado para 10 ou mais. De maneira ideal, o número de partições deve ser configurado com um número primo no qual cada shard armazena uma quantidade razoável de memória. Geralmente, a configuração de `numberOfPartitions` é igual ao número de Java Virtual Machines. Com esta configuração, cada JVM armazena uma partição. Se você ativar replicação, será necessário aumentar o número de Java Virtual Machines no sistema. Caso contrário, cada JVM também armazenará uma partição de réplica, que consome tanta memória quanto uma partição primária.

Leia sobre o dimensionamento de memória e o cálculo da contagem de partições no *Guia de Administração* para maximizar o desempenho da sua configuração escolhida.

Por exemplo, em um sistema com quatro Java Virtual Machines e com o valor da configuração de `numberOfPartitions` de 4, cada JVM hospeda uma partição primária. Uma operação de leitura tem 25% mais chance de buscar dados de uma partição disponível localmente, o que é muito mais rápido comparado ao obter dados de uma JVM remota. Se uma operação de leitura, como a execução de uma consulta, precisar buscar uma coleta de dados que envolva 4 partições igualmente, 75% das chamadas são remotas e 25% das chamadas são locais. Se a configuração de `ReplicaMode` for definida para `SYNC` ou `ASYNC` e a configuração de `ReplicaReadEnabled` for definida para `true`, as quatro partições de réplica são criadas e espalhadas pelos quatro Java Virtual Machines. Cada JVM hospeda uma partição primária e uma partição de réplica. A chance de que a operação de leitura execute localmente aumenta em 50%. A operação de leitura que busca uma coleta de dados que envolve quatro partições igualmente tem somente 50% de chamadas remotas e 50% de chamadas locais. Chamadas locais são muito mais rápidas do que chamadas remotas. Sempre que ocorrem chamada remotas, o desempenho cai.

## Topologia Remota

### **CUIDADO:**

**Não use o cache de consulta de JPA com uma topologia remota. O cache de consulta armazena os resultados da consulta que são uma coleção de chaves de entidade. O cache de consulta vai para o cache de dados para buscar todos os dados da entidade. Como o cache de dados é remoto, estas chamadas adicionais podem negar os benefícios do cache L2.**

**Dica:** Considere usar uma topologia intradomínio para obter melhor desempenho.

Uma topologia remota armazena todos os dados armazenados em cache em um ou mais processos separados, reduzindo o uso da memória dos processos do aplicativo. É possível obter vantagem da distribuição de dados em processos separados ao implementar uma grade de dados replicada e particionada do eXtreme Scale. Ao contrário das configurações integradas e particionadas integradas descritas nas seções anteriores, se desejar gerenciar a grade de dados remota, você deverá fazer isso independente do aplicativo e do provedor JPA.

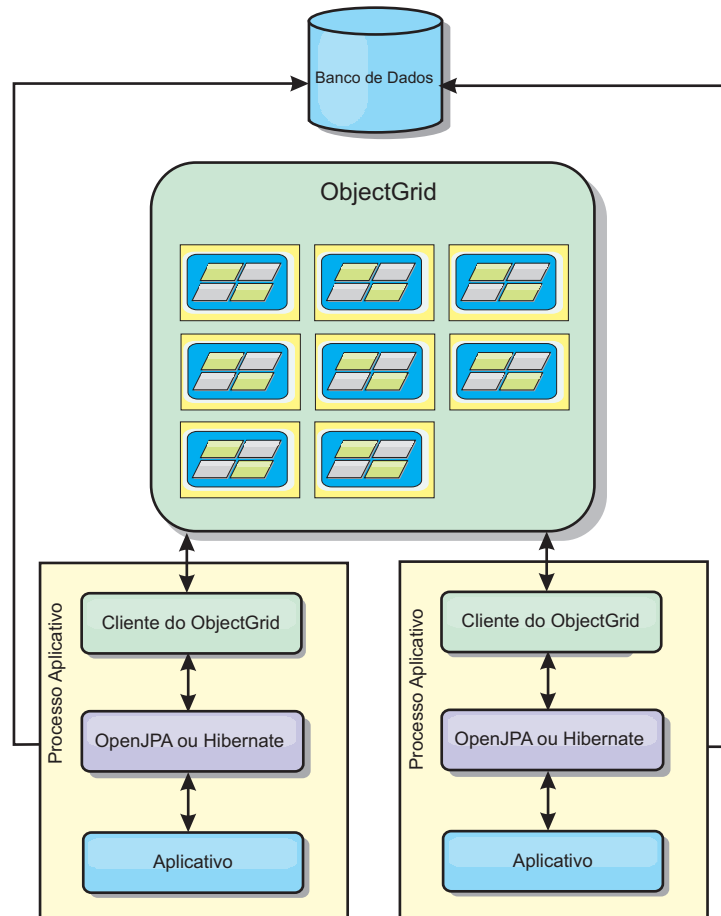


Figura 42. Topologia Remota do JPA

As propriedades de configuração de cache JPA relacionadas à topologia remota são:  
`ObjectGridName=objectgrid_name, ObjectGridType=REMOTE`

O tipo de ObjectGrid REMOTE não requer nenhuma configuração de propriedade porque a política de ObjectGrid e de implementação é definida separadamente a partir do aplicativo JPA. O plug-in do cache JPA conecta-se remotamente a um ObjectGrid remoto existente.

Como toda a interação com o ObjectGrid é remota, essa topologia possui o menor desempenho entre todos os tipos de ObjectGrid.

Vantagens:

- Armazena grandes quantidades de dados.
- O processo aplicativo é livre de dados em cache.
- As atualizações de cache são propagadas para vários processos.
- Opções de configuração flexíveis.

Limitação:

- Todas as leituras e atualizações de cache são remotas.

## Propriedades de Configuração do Cache JPA

O WebSphere eXtreme Scale inclui plug-ins de cache de nível 2 para ambos os provedores, OpenJPA e Hibernate Java Persistence API (JPA). Para configurar o plug-in de cache L2, você deve atualizar as propriedades no arquivo `persistence.xml`.

**Dica:** O plug-in do cache JPA L2 requer um aplicativo que usa as APIs do JPA. Se desejar usar as APIs do WebSphere eXtreme Scale para acessar uma origem de dados JPA, use o carregador JPA. Para obter informações adicionais, consulte “Configurando Utilitários de Carga do JPA” na página 347.

## Propriedades Locais

É possível configurar essas propriedades no arquivo `persistence.xml`. A sintaxe para especificar as propriedades nesse arquivo varia dependendo se você estiver usando OpenJPA ou Hibernate:

- **OpenJPA :** É possível configurar as propriedades no `DataCache` ou no `QueryCache`:

```
<property name="openjpa.DataCache"
  value="object_grid_datacache_class(<property>=<value>,...)" />
```

ou

```
<property name="openjpa.QueryCache"
  value="object_grid_querycache_class(<property>=<value>,...)" />
```

- **Hibernate:**

```
<property name="objectgrid.configuration" value="<property>=<value>,..." />
```

## Topologia e Propriedades Padrão

Os seguintes valores de propriedade padrão serão usados se não nenhum valor for especificado na configuração:

- **ObjectGridName:** nome da unidade de persistência
- **ObjectGridType:** EMBEDDED
- **NumberOfPartitions:** 1 (não é possível alterar quando o tipo de ObjectGrid for EMBEDDED)
- **ReplicaMode:** SYNC
- **ReplicaReadEnabled:** TRUE (não é possível alterar quando o tipo de ObjectGrid for EMBEDDED)
- **MaxUsedMemory:** TRUE
- **MaxNumberOfReplicas:** 47 (deve ser menor ou igual ao número de Java Virtual Machines em um sistema distribuído)

## Propriedades

É possível configurar plug-ins de cache do JPA com as propriedades a seguir.

### ObjectGridName

Especifica o nome exclusivo do ObjectGrid. O valor padrão é o nome da unidade de persistência definido. Se o nome da unidade de persistência não estiver disponível a partir do provedor JPA, um nome gerado será usado.

### ObjectGridType

Especifica o tipo de ObjectGrid.

## Valores válidos:

### EMBEDDED

O tipo de configuração padrão e recomendado. As configurações padrão incluem: `NumberOfPartitions=1`, `ReplicaMode=SYNC`, `ReplicaReadEnabled=true` e `MaxNumberOfReplicas=47`. Use o parâmetro **ReplicaMode** para configurar o modo de replicação e o parâmetro **MaxNumberOfReplicas** para configurar o número máximo de réplicas. Se um sistema tiver mais de 47 Java Virtual Machines, configure o valor **MaxNumberOfReplicas** para que seja igual ao número de Java Virtual Machines.

### EMBEDDED\_PARTITION

O tipo a ser usado quando o sistema precisar armazenar em cache uma grande quantidade de dados em um sistema distribuído. O número padrão de partições é 47 com um modo de réplica de NONE. Em um sistema pequeno com apenas alguns Java Virtual Machines, configure o valor **NumberOfPartitions** para que seja igual ou menor que o número de Java Virtual Machines. Você pode especificar os valores **ReplicaMode**, **NumberOfPartitions** e **ReplicaReadEnabled** para ajustar o sistema.

**REMOTE** O cache tenta se conectar com um ObjectGrid remoto e distribuído a partir do serviço de catálogo.

### MaxNumberOfReplicas

Especifica o número máximo de réplicas a serem usadas para o cache. Esse valor é aplicado apenas para o tipo EMBEDDED. Esse número deve ser igual ou maior que o número de Java Virtual Machines em um sistema. O valor padrão é 47.

**Valores válidos:** maior ou igual a 1

### MaxUsedMemory

**Valores válidos:** TRUE ou FALSE Ativa a liberação das entradas de cache quando a memória se tornar limitada. O valor padrão é TRUE e despeja os dados quando o limite de uso do heap da JVM exceder 70%. É possível modificar a porcentagem do limite de uso do heap JVM padrão ao configurar a propriedade `memoryThresholdPercentage` no arquivo `objectGridServer.properties` e colocar esse arquivo no caminho de classe. Para obter mais informações sobre os evictors, consulte o Plug-ins para Despejar Objetos de Cache informações sobre os evictors no *Visão Geral do Produto*. Para obter mais informações sobre o arquivo de propriedades do servidor, consulte Arquivo de Propriedades do Servidor.

### NumberOfPartitions

**Valores válidos:** maior ou igual a 1 Especifica o número de partições a serem usadas para o cache. Essa propriedade é aplicada quando o valor `ObjectGridType` for configurado para EMBEDDED\_PARTITION. O valor padrão é 47. Para o tipo EMBEDDED, o valor **NumberOfPartitions** é sempre 1.

### 7.1.1+ PlacementScope

Indica a granularidade de uma única instância de um conjunto de mapas.

**Valores válidos:**

#### DOMAIN\_SCOPE

(Padrão) Coloca um shard primário para cada partição em um servidor de contêiner dentro do domínio de serviço de catálogo. Os

shards de réplica para cada partição são colocados nos outros servidores de contêiner dentro do domínio de serviço de catálogo.

#### **CONTAINER\_SCOPE**

Coloca um shard primário em cada servidor de contêiner no domínio de serviço de catálogo.

#### **7.1.1+ PlacementScopeTopology**

Define a topologia de vinculação dos servidores de contêiner dentro do domínio de serviço de catálogo. Este valor é usado apenas quando o valor PlacementScope estiver definido como algo diferente de DOMAIN\_SCOPE.

##### **Valores válidos:**

**HUB** (Padrão) Se a topologia hub for selecionada, uma grade de dados única é selecionada para ser o hub. Todas as outras grades de dados se conectam ao hub. Essa topologia é muito escalável porque o spokes possuem uma única conexão. O hub pode se tornar um gargalo e um ponto de falha único temporário. O hub é realocado para outro servidor de contêiner quando ele falha. A vantagem desta configuração é que um código de arbitragem mais complexo pode ser gravado, permitindo que um único ponto, o hub, manipule todas as colisões.

**RING** Se a topologia em anel for selecionada, cada grade de dados será vinculada a duas outras grades de dados. A ordenação dos links não é garantida. No entanto, cada contêiner que é iniciado provavelmente é vinculado ao primeiro e ao último contêiner a ser incluído ao anel. Essa topologia é a mais escalável, porém apenas dois links podem falhar antes de ser temporariamente cortado. Se os servidores de contêiner falharem, os links serão estabelecidos entre os sobreviventes após a falha ser descoberta.

#### **ReplicaMode**

**Valores válidos:** SYNC/ASYNC/NONE Especifica o método que é usado para copiar o cache para as réplicas. Essa propriedade é aplicada ao configurar o valor ObjectGridType para EMBEDDED ou EMBEDDED\_PARTITION. O valor padrão é NONE para o tipo EMBEDDED\_PARTITION e SYNC para o tipo EMBEDDED. Se o valor **ReplicaMode** for configurado para NONE para o ObjectGridType EMBEDDED, o tipo EMBEDDED ainda usará um **ReplicaMode** de SYNC.

#### **ReplicaReadEnabled**

**Valores válidos:** TRUE ou FALSE Quando ativado, os clientes leem a partir das réplicas. Essa propriedade é aplicada para o tipo EMBEDDED\_PARTITION. O valor padrão é FALSE para o tipo EMBEDDED\_PARTITION. O tipo EMBEDDED sempre é configurado para o valor **ReplicaReadEnabled** para TRUE.

#### **writeBehind**

**Apenas para provedores Hibernate:** Quando writeBehind é ativado, as atualizações são temporariamente armazenadas em um armazenamento de dados de escopo JVM até a condição writeBehindInterval ou writeBehindMaxBatchSize for atendidas.

**Atenção:** A menos que writeBehind seja ativada, as outras definições de configuração write behind são desconsideradas.

**Importante:** Tome cuidado ao usar a função write behind. As configurações write behind introduzem uma latência mais longa de



sincronização de entre todas as JVMs e aumenta a chance de atualizações serem perdidas. Em um sistema que tiver a configuração write behind ativada com quatro ou mais JVMs, a atualização executada em uma JVM possui um atraso de aproximadamente 15 segundos antes que a atualização fique disponível para outras JVMs. Se qualquer uma das duas JVMs atualizar a mesma entrada, aquela que limpar a atualização perderá primeiro sua atualização.

**Valores válidos:** TRUE ou FALSE

**Valor padrão:** FALSE

#### **writeBehindInterval**

**Apenas para provedores Hibernate:** Especifica o intervalo de tempo em milissegundos para limpar as atualizações no cache.

**Valores válidos:** maior ou igual a 1

**Valor padrão:** 5000 (5 segundos)

#### **writeBehindPoolSize**

**Apenas para provedores Hibernate:** Especifica o tamanho máximo do conjunto de encadeamentos usado para limpar as atualizações no cache.

**Valores válidos:** maior ou igual a 1

**Valor padrão:** 5

#### **writeBehindMaxBatchSize**

**Apenas para provedores Hibernate:** Especifica o tamanho de lote máximo por cache de região ao limpar as atualizações no cache. Por exemplo, se o tamanho for configurado para 1.000 e as atualizações armazenadas no armazenamento write behind de um cache de região exceder 1.000 entradas, as atualizações serão limpas no cache, mesmo se a condição writeBehindInterval especificada não for atendida. As atualizações são limpas no cache em aproximadamente durante o número de segundos especificado pelo valor writeBehindInterval ou sempre que o tamanho do armazenamento write behind de cada cache de região exceder 1000 entradas. Observe, no caso em que a condição writeBehindMaxBatchSize é atendida, que apenas o cache da região que atende a esta condição limpa suas atualizações no armazenamento write behind no cache. Um cache de região geralmente corresponde a uma entidade ou a uma consulta.

**Valores válidos:** maior ou igual a 1

**Valor padrão:** 1000

## **Configurando o Plug-in de Cache OpenJPA**

É possível configurar ambas as implementações DataCache e QueryCache para OpenJPA.

### **Antes de Iniciar**

- Você deve determinar a topologia do plug-in do cache JPA que deseja usar. Consulte “Plug-in do Cache JPA Nível 2 (L2)” na página 325 para obter mais informações sobre as diferentes configurações e sobre as propriedades para configurar cada topologia.
- Você deve ter um aplicativo que usa as APIs do JPA. Se desejar usar APIs do WebSphere eXtreme Scale para acessar os dados com o JPA, use o carregador JPA. Para obter informações adicionais, consulte “Configurando Utilitários de Carga do JPA” na página 347.

## Procedimento

1. Configure as propriedades no arquivo `persistence.xml` para configurar o plug-in do cache do OpenJPA: Essas propriedades podem ser configuradas em uma implementação de cache `DataQuery` ou `Query`.

As configurações `DataCache` e `QueryCache` são independentes uma da outra. É possível ativar qualquer uma das configurações. Entretanto, se as duas estiverem ativadas, a configuração do `QueryCache` utilizará a mesma configuração que a do `DataCache` e suas propriedades de configuração serão descartadas.

```
<property name="openjpa.DataCache"
  value="<object_grid_datacache_class(<property>=<value>,...)" />
```

ou

```
<property name="openjpa.QueryCache"
  value="<object_grid_querycache_class(<property>=<value>,...)" />
```

**Nota:** É possível ativar a configuração de `QueryCache` somente para topologias integradas e intradomínio integradas.

Você pode executar a propriedade `ObjectGridName`, a propriedade `ObjectGridType` e outras propriedades relacionadas à política de implementação simples na lista de propriedades da classe do cache `ObjectGrid` para customizar a configuração do cache. Este é um exemplo:

```
<property name="openjpa.DataCache"
  value="com.ibm.websphere.objectgrid.openjpa.ObjectGridDataCache(
  ObjectGridName=BasicTestObjectGrid,ObjectGridType=EMBEDDED,
  maxNumberOfReplicas=4)" />
<property name="openjpa.QueryCache"
  value="com.ibm.websphere.objectgrid.openjpa.ObjectGridQueryCache()" />
<property name="openjpa.RemoteCommitProvider" value="sjvm" />
```

Consulte “Propriedades de Configuração do Cache JPA” na página 332 para obter uma lista de propriedades que podem ser configuradas.

2. No arquivo `persistence.xml`, você também deve configurar a propriedade `openjpa.RemoteCommitProvider` como `sjvm`.

```
<property name="openjpa.RemoteCommitProvider" value="sjvm" />
```

3. Opcional: Para customizar ainda mais a grade de dados usada pelo cache, é possível fornecer configurações adicionais com arquivos XML.

Para a maioria dos cenários, a configuração das propriedades de cache devem ser suficientes. Para customizar ainda mais o `ObjectGrid` usado pelo cache, é possível fornecer os arquivos XML de configuração OpenJPA `ObjectGrid` no diretório `META-INF` semelhante ao arquivo `persistence.xml`. Durante a inicialização, o cache tenta localizar esses arquivos XML e processa-os se localizados.

Há três tipos de arquivos XML de configuração do OpenJPA `ObjectGrid`:

- `openjpa-objectGrid.xml` (Configuração `ObjectGrid`)

**Caminho do arquivo:** `META-INF/openjpa-objectGrid.xml`

Este arquivo é utilizado para customizar a configuração do `ObjectGrid` para os tipos `EMBEDDED` e `EMBEDDED_PARTITION`. Com o tipo `REMOTE`, este arquivo é ignorado. Por padrão, cada classe de entidade é mapeada para sua própria configuração do `BackingMap` denominada como um nome de classe de entidade na configuração do `ObjectGrid`. Por exemplo, a classe de entidade `com.mycompany.Employee` é mapeada para o `BackingMap` `com.mycompany.Employee`. A configuração padrão do `BackingMap` é `readOnly="false"`, `copyKey="false"`, `lockStrategy="NONE"` e

copyMode="NO\_COPY". É possível customizar alguns BackingMaps com uma configuração escolhida. A palavra-chave reservada ALL\_ENTITY\_MAPS pode ser utilizada para representar todos os mapas, exceto outros mapas customizados listados no arquivo openjpa-objectGrid.xml. Os BackingMaps que não estiverem listados neste arquivo openjpa-objectGrid.xml utilizarão a configuração padrão. Se os BackingMaps customizados não especificarem o atributo ou as propriedades de BackingMaps e estes atributos forem especificados na configuração padrão, os valores de atributo da configuração padrão serão aplicados. Por exemplo, se uma classe de entidade for anotada com timeToLive=30, a configuração padrão do BackingMap para essa entidade terá timeToLive=30. Se o arquivo openjpa-objectGrid.xml customizado também incluir esse BackingMap mas não especificar o valor timeToLive, o BackingMap customizado terá timeToLive=30 por padrão. O arquivo openjpa-objectGrid.xml pretende substituir ou estender a configuração padrão.

- openjpa-objectGridDeployment.xml (política de implementação)

**Caminho do arquivo:** META-INF/openjpa-objectGridDeployment.xml

Este arquivo é utilizado para customizar a política de implementação. Ao customizar a política de implementação, se o arquivo openjpa-objectGridDeployment.xml for fornecido, a política de implementação padrão será descartada. Todos os valores de atributo da política de implementação são fornecidos pelo arquivo openjpa-objectGridDeployment.xml.

- openjpa-objectGrid-client-override.xml (configuração de substituição do cliente ObjectGrid)

**Caminho do arquivo:** META-INF/openjpa-objectGrid-client-override.xml

Esse arquivo é usado para customizar um ObjectGrid do lado do cliente. Por padrão, o cache do ObjectGrid aplica uma configuração do ObjectGrid de substituição do cliente que desativa um cache local. Se um aplicativo requerer um cache perto, ele pode fornecer este arquivo e especificar numberOfBuckets="xxx". A substituição do cliente padrão desativa o cache perto ao configurar numberOfBuckets="0". O cache perto pode estar ativo ao reconfigurar numberOfBuckets com um valor maior que 0 com o arquivo openjpa-objectGrid-client-override.xml. O arquivo openjpa-objectGrid-client-override.xml funciona da mesma forma que o arquivo openjpa-objectGrid.xml: Ele substitui ou estende a configuração padrão de substituição do ObjectGrid cliente.

Dependendo da topologia do eXtreme Scale configurada, é possível fornecer qualquer um destes três arquivos XML para customizar essa topologia.

Para ambos os tipos EMBEDDED e EMBEDDED\_PARTITION, é possível fornecer qualquer um dos três arquivos XML para customizar o ObjectGrid, a política de implementação e a configuração de substituição do ObjectGrid cliente.

Para um REMOTE ObjectGrid, o cache ObjectGrid não cria um ObjectGrid dinâmico. Em vez disso, ele obtém um ObjectGrid do lado do cliente a partir do serviço do catálogo. É possível fornecer apenas o arquivo openjpa-objectGrid-client-override.xml para customizar a configuração de substituição do ObjectGrid cliente.

4. Opcional: (Apenas configurações remotas) Configure o sistema eXtreme Scale externo se desejar configurar um cache com um tipo de ObjectGrid REMOTE.

É necessário configurar um sistema eXtreme Scale externo se desejar configurar um cache com um tipo de ObjectGrid REMOTE. É necessário ambos os arquivos XML de configuração ObjectGrid e ObjectGridDeployment, que se baseiam no arquivo persistence.xml, para configurar um sistema externo. Para

obter exemplos desses arquivos de configuração, consulte “Exemplo: Arquivos XML do ObjectGrid OpenJPA”.

## Resultados

### Configuração EMBEDDED, EMBEDDED\_PARTITION ou intradomínio:

Quando um aplicativo é iniciado, o plug-in detecta ou inicia automaticamente um serviço de catálogo, inicia um servidor de contêiner e conecta os servidores de contêiner no serviço de catálogo. Em seguida, o plug-in se comunica com o contêiner ObjectGrid e seus equivalentes que estão em execução em outros processos do servidor de aplicativos que usam a conexão do cliente.

### Configuração REMOTE:

A política de implementação é especificada separadamente do aplicativo JPA. Um sistema do ObjectGrid externo possui ambos os processos de serviço de catálogo e servidor de contêiner. É necessário iniciar um serviço de catálogos antes de iniciar servidores de contêiner. Consulte “Iniciando Servidores Independentes” na página 387 e “Iniciando Servidores de Contêiner” na página 390 para obter mais informações.

## O que Fazer Depois

- Desenvolva um aplicativo OpenJPA que use a configuração. Para obter informações adicionais, consulte Exemplo: Usando o Plug-in Hibernate para Pré-Carregar Dados no Cache do ObjectGrid.
- Em um ambiente de produção, crie domínios do serviço de catálogo para seus processos criados automaticamente para sua configuração EMBEDDED ou EMBEDDED\_PARTITION.
  - Ambiente independente:

Se você não estiver executando seus servidores dentro de um processo do WebSphere Application Server, os hosts e as portas do domínio do serviço de catálogo serão especificados usando o arquivo de propriedades denominado `objectGridServer.properties`. Este arquivo deve ser armazenado no caminho de classe do aplicativo e ter a propriedade `catalogServiceEndpoints` definida. O domínio de serviço de catálogo é iniciado independentemente dos processos do aplicativo e deve ser iniciado antes que os processos do aplicativo sejam iniciados.

O formato do arquivo `objectGridServer.properties` é:

```
catalogServiceEndpoints=<hostname1>:<port1>,<hostname2>:<port2>
```
  - Ambiente do WebSphere Application Server:

Se você estiver executando dentro de um processo do WebSphere Application Server, o plug-in do cache JPA conectará automaticamente ao serviço de catálogo ou ao domínio do serviço de catálogo definido para a célula do WebSphere Application Server.
- Quando estiver usando o valor de `ObjectGridType` EMBEDDED ou EMBEDDED\_PARTITION em um ambiente do Java SE, use o método `System.exit(0)` no final do programa para parar o servidor eXtreme Scale integrado. Caso contrário, o programa poderá parecer não estar respondendo.

### Exemplo: Arquivos XML do ObjectGrid OpenJPA:

Os arquivos XML OpenJPA do ObjectGrid devem ser criados com base na configuração da unidade de persistência.

## Arquivo persistence.xml

A seguir há um arquivo persistence.xml de exemplo que representa a configuração de uma unidade de persistência:

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="1.0">
  <persistence-unit name="AnnuityGrid">
    <provider>org.apache.openjpa.persistence.PersistenceProviderImpl</provider>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistibleObject</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.FixedAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.EquityAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Person</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityHolder</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Address</class>
    <exclude-unlisted-classes>true</exclude-unlisted-classes>

    <properties>
    <!-- Database setting -->

    <!-- enable cache -->
    <property name="openjpa.DataCache"
      value="com.ibm.websphere.objectgrid.openjpa.ObjectGridDataCache(objectGridName=Annuity,
        objectGridType=EMBEDDED, maxNumberOfReplicas=4)" />
    <property name="openjpa.RemoteCommitProvider" value="sjvm" />
    <property name="openjpa.QueryCache"
      value="com.ibm.websphere.objectgrid.openjpa.ObjectGridQueryCache()" />
    </properties>
  </persistence-unit>
</persistence>
```

## Arquivo openjpa-objectGrid.xml

O arquivo openjpa-objectGrid.xml é usado para customizar a configuração do ObjectGrid para ambos os tipos EMBEDDED e EMBEDDED\_PARTITION. A seguir há o arquivo openjpa-objectGrid.xml que corresponde ao arquivo persistence.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="Annuity">
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistibleObject"
        readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistibleObject" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <backingMap name="ObjectGridQueryCache" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" pluginCollectionRef="ObjectGridQueryCache"
        evictionTriggers="MEMORY_USAGE_THRESHOLD" />
    </objectGrid>
  </objectGrids>
  <backingMapPluginCollections>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity">
      <bean id="ObjectTransformer"
```

```

        className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
</backingMapPluginCollection>
<backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address">
    <bean id="ObjectTransformer"
        className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
</backingMapPluginCollection>
<backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor">
    <bean id="ObjectTransformer"
        className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
</backingMapPluginCollection>
<backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person">
    <bean id="ObjectTransformer"
        className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
</backingMapPluginCollection>
<backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact">
    <bean id="ObjectTransformer"
        className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
</backingMapPluginCollection>
<backingMapPluginCollection
    id="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistibleObject">
    <bean id="ObjectTransformer"
        className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
</backingMapPluginCollection>
<backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider">
    <bean id="ObjectTransformer"
        className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
</backingMapPluginCollection>
<backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout">
    <bean id="ObjectTransformer"
        className="com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer" />
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
</backingMapPluginCollection>
<backingMapPluginCollection id="ObjectGridQueryCache">
    <bean id="MapIndexPlugin" className="com.ibm.websphere.objectgrid.plugins.index.HashIndex" >
        <property name="Name" type="java.lang.String"
            value="QueryCacheKeyIndex" description="name of index"/>
        <property name="POJOKeyIndex" type="boolean" value="true" description="POJO Key Index" />
    </bean>
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
</backingMapPluginCollection>
</backingMapPluginCollections>
</objectGridConfig>

```

### Importante:

1. Cada entidade é mapeada para um BackingMap denominado como o nome completo da classe de entidade.

Por padrão, as entidades fazem parte do cache de segundo nível. Nas classes de Entidade que precisam ser excluídas do cache, é possível incluir a anotação `@DataCache(enabled=false)` na classe de entidade que deseja excluir do cache L2 :

```

import org.apache.openjpa.persistence.DataCache;
@Entity
@DataCache(enabled=false)
public class OpenJPACacheTest { ... }

```

2. Se as classes de entidade estão em uma hierarquia de herança, as classes-filha serão mapeadas para o BackingMap pai. A hierarquia de herança compartilha um único BackingMap.
3. O mapa `ObjectGridQueryCache` é necessário para suportar `QueryCache`.
4. O `backingMapPluginCollection` para cada mapa de entidade deve ter o `ObjectTransformer` utilizando a classe `com.ibm.ws.objectgrid.openjpa.ObjectGridPCDataObjectTransformer`.



- O `backingMapPluginCollection` para o mapa `ObjectGridQueryCache` deve ter o índice de chaves denominado como `QueryCacheKeyIndex`, conforme mostrado no modelo.
- O evictor é opcional para cada mapa.

### Arquivo `openjpa-objectGridDeployment.xml`

O arquivo `openjpa-objectGridDeployment.xml` é usado para customizar a política de implementação. A seguir há o arquivo `openjpa-objectGridDeployment.xml` que corresponde ao arquivo `persistence.xml`:

`openjpa-objectGridDeployment.xml`

```
<?xml version="1.0" encoding="UTF-8" ?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="Annuity">
    <mapSet name="MAPSET_Annuity" numberOfPartitions="1" numInitialContainers="1"
minSyncReplicas="0" maxSyncReplicas="4" maxAsyncReplicas="0"
replicaReadEnabled="true">
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Address" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistibleObject" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <map ref="ObjectGridQueryCache" />
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

**Nota:** O mapa `ObjectGridQueryCache` é necessário para suportar `QueryCache`.

## Configurando o Plug-in de Cache do Hibernate

É possível ativar o cache para usar o plug-in de cache do Hibernate ao especificar arquivos de propriedades.

### Antes de Iniciar

- Você deve determinar a topologia do plug-in do cache JPA que deseja usar. Consulte “Plug-in do Cache JPA Nível 2 (L2)” na página 325 para obter mais informações sobre as diferentes configurações.
- Você deve ter um aplicativo que usa as APIs do JPA. Se desejar usar APIs do WebSphere eXtreme Scale para acessar os dados com o JPA, use o carregador JPA. Para obter informações adicionais, consulte “Configurando Utilitários de Carga do JPA” na página 347.

### Procedimento

- Se estiver usando o WebSphere Application Server, coloque os arquivos Java Archive (JAR) nos locais apropriados para sua configuração.

O plug-in do cache do Hibernate é empacotado no arquivo `oghibernate-cache.jar` e é instalado no diretório `was_root/optionalLibraries/ObjectGrid`. Para usar o plug-in de cache do Hibernate, é necessário incluir o arquivo `oghibernate-cache.jar` na biblioteca Hibernate. Por exemplo, se você incluir a biblioteca Hibernate em seu aplicativo, também deverá incluir o arquivo `oghibernate-cache.jar`. Se você definir uma biblioteca compartilhada para incluir a biblioteca Hibernate, o arquivo `oghibernate-cache.jar` deverá ser incluído no diretório da biblioteca compartilhada.

O eXtreme Scale não instala o arquivo `cglib.jar` no ambiente do WebSphere Application Server. Se você tiver aplicativos existentes ou bibliotecas compartilhadas, como Hibernate, que depende do `cglib.jar`, localize o arquivo `cglib.jar` e inclua-o no caminho de classe. Por exemplo, se seu aplicativo



incluir todos os arquivos JAR da biblioteca Hibernate, mas excluir o arquivo `cglib.jar` disponível com Hibernate, você deverá incluir o arquivo `cglib.jar` proveniente do Hibernate em seu aplicativo.

2. Configure as propriedades no arquivo `persistence.xml` para configurar o plug-in do cache do Hibernate

A sintaxe para configurar as propriedades no arquivo `persistence.xml` é a seguinte:

```
<property name="hibernate.cache.provider_class"
  value="com.ibm.websphere.objectgrid.hibernate.cache.ObjectGridHibernateCacheProvider" />
<property name="hibernate.cache.use_query_cache" value="true"/>
<property name="objectgrid.configuration" value="<property>=<value>,..." />
<property name="objectgrid.hibernate.regionNames" value="<regionName>,..." />
```

- **hibernate.cache.provider\_class:** O valor da propriedade `provider_class` é a classe `com.ibm.websphere.objectgrid.hibernate.cache.ObjectGridHibernateCacheProvider`.
- **hibernate.cache.use\_query\_cache:** Para ativar o cache de consulta, defina o valor para `true` na propriedade `use_query_cache`.

**Nota:** É possível ativar o cache de consulta somente para topologias integradas e intradomínio integradas.

- **objectgrid.configuration:** Use a propriedade `objectgrid.configuration` para especificar as propriedades de configuração de cache eXtreme Scale, incluindo o atributo `ObjectGridType` que especifica como colocar os shards na grade de dados.

É necessário especificar um único valor da propriedade `ObjectGridName` para evitar potenciais conflitos de nomenclatura. As outras propriedades de configuração de cache do eXtreme Scale são opcionais.

Para ativar o cache `write-behind`, use os seguintes atributos `write-behind` na propriedade `objectgrid.configuration`. Quando em cache `write-behind` é ativado, as atualizações são armazenadas temporariamente em um armazenamento de dados de escopo JVM até a condição `writeBehindInterval` ou `writeBehindMaxBatchSize` ser atendidas, quando os dados são esvaziados para o cache.

```
writeBehind=true, writeBehindInterval=5000, writeBehindPoolSize=10, writeBehindMaxBatchSize=1000
```

**Atenção:** A menos que `writeBehind` seja ativada, as outras definições de configuração `write behind` são desconsideradas.

Para obter mais informações sobre os valores que podem ser configurados na propriedade `objectgrid.configuration`, consulte "Propriedades de Configuração do Cache JPA" na página 332.

- **objectgrid.hibernate.regionNames:** A propriedade `objectgrid.hibernate.regionNames` é opcional e deve ser especificada quando os valores `regionNames` forem definidos após o cache eXtreme Scale ser inicializado. Considere o exemplo de uma classe de entidade mapeada para uma `regionName` com a classe de entidade não-especificada no arquivo `persistence.xml` ou não incluída no arquivo de mapeamento Hibernate. Além disso, suponha que ele possua a anotação `Entity`. Então, o `regionName` para esta classe de entidade será resolvido no momento do carregamento da classe quando o cache do eXtreme Scale for inicializado. Outro exemplo é a execução do método `Query.setCacheRegion(String regionName)` que é executado após a inicialização do cache do eXtreme Scale. Nas situações acima, inclua todos os `regionNames` possíveis determinados como dinâmico na propriedade `objectgrid.hibernate.regionNames` para que o cache do eXtreme Scale possa preparar o `BackingMaps` para todos os `regionNames`.

3. Opcional: Para customizar ainda mais a grade de dados usada pelo cache, é possível fornecer configurações adicionais com arquivos XML.

Para a maioria dos cenários, a configuração das propriedades de cache devem ser suficientes. Para customizar ainda mais o ObjectGrid usado pelo cache, é possível fornecer os arquivos XML de configuração Hibernate ObjectGrid no diretório META-INF da mesma forma que o arquivo `persistence.xml`. Durante a inicialização, o cache tenta localizar esses arquivos XML e processa-os se localizados.

Há três tipos de arquivos XML de configuração do Hibernate ObjectGrid:

- `hibernate-objectGrid.xml` (Configuração ObjectGrid)

**Caminho de arquivo:** META-INF/hibernate-objectGrid.xml

Por padrão, cada classe de entidade possui uma `regionName` associada (padrão para o nome da classe de entidade) que é mapeada para uma configuração de `BackingMap` denominada como `regionName` na configuração do ObjectGrid. Por exemplo, a classe de entidade `com.mycompany.Employee` possui uma `regionName` associada definida por padrão como `com.mycompany.Employee BackingMap`. A configuração padrão do `BackingMap` é `readOnly="false"`, `copyKey="false"`, `lockStrategy="NONE"` e `copyMode="NO_COPY"`. É possível customizar alguns `BackingMaps` com uma configuração escolhida. A palavra-chave reservada `"ALL_ENTITY_MAPS"` pode ser utilizada para representar todos os mapas, exceto outros mapas customizados listados no arquivo `hibernate-objectGrid.xml`. Os `BackingMaps` que não estiverem listados neste arquivo `hibernate-objectGrid.xml` utilizarão a configuração padrão.

- `hibernate-objectGridDeployment.xml` (política de implementação)

**Caminho de arquivo:** META-INF/hibernate-objectGridDeployment.xml

Este arquivo é utilizado para customizar a política de implementação. Ao customizar a política de implementação, se o arquivo `hibernate-objectGridDeployment.xml` for fornecido, a política de implementação padrão será descartada. Todos os valores de atributo da política de implementação serão fornecidos pelo arquivo `hibernate-objectGridDeployment.xml`.

- `hibernate-objectGrid-client-override.xml` (configuração de substituição ObjectGrid do cliente)

**Caminho de arquivo:** META-INF/hibernate-objectGrid-client-override.xml

Este arquivo é utilizado para customizar um ObjectGrid do lado do cliente. Por padrão, o cache do ObjectGrid aplica uma configuração de substituição do cliente padrão que desativa o cache local. Se o aplicativo requerer um cache perto, ele pode fornecer este arquivo e especificar `numberOfBuckets="xxx"`. A substituição do cliente padrão desativa o cache perto ao configurar `numberOfBuckets="0"`. O cache perto pode estar ativo ao reconfigurar `numberOfBuckets` com um valor superior a 0 através de `hibernate-objectGrid-client-override.xml`. A maneira como o arquivo `hibernate-objectGrid-client-override.xml` funciona é similar ao `hibernate-objectGrid.xml`: ele substitui ou estende a configuração de substituição do ObjectGrid do cliente padrão.

Dependendo da topologia do eXtreme Scale configurada, é possível fornecer qualquer um destes três arquivos XML para customizar essa topologia.

Para os tipos `EMBEDDED` e `EMBEDDED_PARTITION`, é possível fornecer qualquer um dos três arquivos XML para customizar o ObjectGrid, a política de implementação e a configuração de substituição do ObjectGrid cliente.

Para um `REMOTE` ObjectGrid, o cache não cria um ObjectGrid dinâmico. O cache obtém apenas um ObjectGrid do lado do cliente a partir do serviço do

catálogo. É possível fornecer apenas um arquivo `hibernate-objectGrid-client-override.xml` para customizar a configuração de substituição do ObjectGrid do cliente.

4. Opcional: (Apenas configurações remotas) Configure o sistema eXtreme Scale externo se desejar configurar um cache com um tipo de ObjectGrid REMOTE. É necessário configurar um sistema eXtreme Scale externo se desejar configurar um cache com um tipo de ObjectGrid REMOTE. É necessário ambos os arquivos XML de configuração ObjectGrid e ObjectGridDeployment, que se baseiam no arquivo `persistence.xml`, para configurar um sistema externo. Para obter exemplos desses arquivos de configuração, consulte “Exemplo: Arquivos XML do ObjectGrid Hibernate” na página 345.

## Resultados

### Configuração EMBEDDED ou EMBEDDED\_PARTITION:

Quando um aplicativo é iniciado, o plug-in detecta ou inicia automaticamente um serviço de catálogo, inicia um servidor de contêiner e conecta os servidores de contêiner no serviço de catálogo. Em seguida, o plug-in se comunica com o contêiner ObjectGrid e seus equivalentes que estão em execução em outros processos do servidor de aplicativos que usam a conexão do cliente.

Cada entidade JPA possui um mapa de apoio independente designado para usar o nome da classe da entidade. Cada BackingMap possui os seguintes atributos.

- `readOnly="false"`
- `copyKey="false"`
- `lockStrategy="NONE"`
- `copyMode="NO_COPY"`

### Configuração REMOTE:

A política de implementação é especificada separadamente do aplicativo JPA. Um sistema do ObjectGrid externo possui ambos os processos de serviço de catálogo e servidor de contêiner. É necessário iniciar um serviço de catálogos antes de iniciar servidores de contêiner. Consulte “Iniciando Servidores Independentes” na página 387 e “Iniciando Servidores de Contêiner” na página 390 para obter mais informações.

## O que Fazer Depois

- Desenvolva um aplicativo Hibernate que usa a configuração. Para obter informações adicionais, consulte Exemplo: Usando o Plug-in Hibernate para Pré-Carregar Dados no Cache do ObjectGrid.
- Em um ambiente de produção, crie domínios do serviço de catálogo para seus processos criados automaticamente para sua configuração EMBEDDED ou EMBEDDED\_PARTITION.
  - Ambiente independente:

Se você não estiver executando seus servidores dentro de um processo do WebSphere Application Server, os hosts e as portas do domínio do serviço de catálogo serão especificados usando o arquivo de propriedades denominado `objectGridServer.properties`. Este arquivo deve ser armazenado no caminho de classe do aplicativo e ter a propriedade `catalogServiceEndpoints` definida.

O domínio de serviço de catálogo é iniciado independentemente dos processos do aplicativo e deve ser iniciado antes que os processos do aplicativo sejam iniciados.

O formato do arquivo `objectGridServer.properties` é:

```
catalogServiceEndPoints=<hostname1>:<port1>,<hostname2>:<port2>
```

– Ambiente do WebSphere Application Server:

Se você estiver executando dentro de um processo do WebSphere Application Server, o plug-in do cache JPA conectará automaticamente ao serviço de catálogo ou ao domínio do serviço de catálogo definido para a célula do WebSphere Application Server.

- Quando estiver usando o valor de `ObjectGridType EMBEDDED` ou `EMBEDDED_PARTITION` em um ambiente do Java SE, use o método `System.exit(0)` no final do programa para parar o servidor eXtreme Scale integrado. Caso contrário, o programa poderá parecer não estar respondendo.

### Exemplo: Arquivos XML do ObjectGrid Hibernate:

Os arquivos XML Hibernate ObjectGrid devem ser criados com base na configuração de uma unidade de persistência.

#### Arquivo `persistence.xml`

Um arquivo `persistence.xml` de exemplos que representa a configuração de uma unidade de persistência está a seguir:

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  version="1.0">
  <persistence-unit name="AnnuityGrid">
    <provider>org.hibernate.ejb.HibernatePersistence</provider>

    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityPersistibleObject</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.FixedAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.EquityAnnuity</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Person</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.AnnuityHolder</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact</class>
    <class>com.ibm.wssvt.acme.annuity.common.bean.jpa.Address</class>

    <exclude-unlisted-classes>true</exclude-unlisted-classes>

    <properties>
      <property name="hibernate.show_sql" value="false" />
      <property name="hibernate.connection.url" value="jdbc:db2:Annuity" />
      <property name="hibernate.connection.driver_class" value="com.ibm.db2.jcc.DB2Driver" />
      <property name="hibernate.default_schema" value="EJB30" />

      <!-- Cache -->
      <property name="hibernate.cache.provider_class"
        value="com.ibm.websphere.objectgrid.hibernate.cache.ObjectGridHibernateCacheProvider" />
      <property name="hibernate.cache.use_query_cache" value="true"/>
      <property name="objectgrid.configuration" value="ObjectGridType=EMBEDDED,
        ObjectGridName=Annuity, MaxNumberOfReplicas=4" />
    </properties>
  </persistence-unit>
</persistence>
```

#### Arquivo `hibernate-objectGridDeployment.xml`

Use o arquivo `hibernate-objectGridDeployment.xml` para, opcionalmente, customizar a política de implementação. Se esse arquivo for fornecido no diretório `META-INF/hibernate-objectGridDeployment.xml`, a política de implementação padrão será substituída pela configuração neste arquivo.

```
<?xml version="1.0" encoding="UTF-8" ?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
```

```

xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
<objectgridDeployment objectgridName="Annuity">
  <mapSet name="MAPSET_Annuity" numberOfPartitions="1" numInitialContainers="1" minSyncReplicas="0"
    maxSyncReplicas="4" maxAsyncReplicas="0" replicaReadEnabled="true">
    <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
    <map ref="defaultCacheMap" />
    <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
    <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
    <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
    <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
    <map ref="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
    <map ref="org.hibernate.cache.UpdateTimestampsCache" />
    <map ref="org.hibernate.cache.StandardQueryCache" />
  </mapSet>
</objectgridDeployment>
</deploymentPolicy>

```

## Arquivo hibernate-objectGrid.xml

Se você não estiver utilizando o Hibernate com o Java Persistence API (JPA), use o seguinte exemplo hibernate-objectGrid.xml para criar sua configuração Hibernate:

```

<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="Annuity">
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity" />
      <backingMap name="defaultCacheMap" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="defaultCacheMap" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider" />
      <backingMap name="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout" />
      <backingMap name="org.hibernate.cache.UpdateTimestampsCache" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="org.hibernate.cache.UpdateTimestampsCache" />
      <backingMap name="org.hibernate.cache.StandardQueryCache" readOnly="false" copyKey="false"
        lockStrategy="NONE" copyMode="NO_COPY" evictionTriggers="MEMORY_USAGE_THRESHOLD"
        pluginCollectionRef="org.hibernate.cache.StandardQueryCache" />
    </objectGrid>
  </objectGrids>
  <backingMapPluginCollections>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Annuity">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="defaultCacheMap">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payor">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Contact">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Person">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Rider">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="com.ibm.wssvt.acme.annuity.common.bean.jpa.Payout">
      <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
      </bean>
    </backingMapPluginCollection>
    <backingMapPluginCollection id="org.hibernate.cache.UpdateTimestampsCache">

```

```

    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>

    </backingMapPluginCollection>
    <backingMapPluginCollection id="org.hibernate.cache.StandardQueryCache">
    <bean id="Evictor" className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor">
    </bean>
    </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>

```

**Nota:** Os mapas `org.hibernate.cache.UpdateTimestampsCache`, `org.hibernate.cache.StandardQueryCache` e `defaultCacheMap` são necessários.

---

## Configurando Integração do Banco de Dados

O WebSphere eXtreme Scale pode ser usado para diminuir a carga nos bancos de dados. É possível usar a Java Persistence API (JPA) entre o WebSphere eXtreme Scale e o banco de dados para integrar as mudanças como um carregador.

### Antes de Iniciar

Para obter um resumo das várias topologias que podem ser criadas com um banco de dados, consulte “Integração com o Banco de Dados: Armazenamento em Cache Write-behind, Sequencial e Lateral” na página 17.

## Configurando Utilitários de Carga do JPA

Um Utilitário de Carga do Java Persistence API (JPA) é uma implementação de plug-in que utiliza JPA para interagir com o banco de dados.

### Antes de Iniciar

- É necessário ter uma implementação do JPA, como Hibernate ou OpenJPA.
- O banco de dados pode ser qualquer back end que seja suportado pelo provedor JPA escolhido.
- Decida se você usará o plug-in do `JPAEntityLoader` ou o plug-in `JPAEntityLoader Use` o plug-in do `JPAEntityLoader` ao armazenar dados usando a API `ObjectMap`. Use o plug-in do `JPAEntityLoader` ao armazenar dados usando a API `EntityManager`.

**Nota:** Se estiver usando as APIs JPA para acessar a origem de dados JPA, use o plug-in de cache JPA L2. O plug-in do cache introduz a grade de dados entre seu aplicativo e a origem de dados JPA, enquanto ainda usa um aplicativo JPA. Para obter informações adicionais, consulte “Plug-in do Cache JPA Nível 2 (L2)” na página 325.

### Sobre Esta Tarefa

Para obter mais informações sobre como o Java Persistence API (JPA) Loader funciona, consulte o Carregadores JPA.

### Procedimento

1. Configure os parâmetros necessários que o JPA requer para interagir com o banco de dados.

Os seguintes parâmetros são necessários. Esses parâmetros são configurados no bean `JPALoader` ou `JPAEntityLoader` e no bean `JPATxCallback`.

- **persistenceUnitName:** Especifica o nome da unidade de persistência. Esse parâmetro é necessário para duas finalidades: criar um `EntityManagerFactory`



do JPA e localizar os metadados da entidade JPA no arquivo `persistence.xml`. Este atributo é configurado no bean `JPATxCallback`.

- **JPAPropertyFactory**: Especifica o factory para criar um mapa de propriedade de persistência para substituir as propriedades de persistência padrão. Este atributo é configurado no bean `JPATxCallback`. Para configurar esse atributo, a configuração de estilo Spring é necessária.
- **entityClassName**: Especifica o nome da classe de entidade que é necessário para usar os métodos JPA, como `EntityManager.persist`, `EntityManager.find` e assim por diante. O plug-in `JPAEntityLoader` requer este parâmetro, mas o parâmetro é opcional para `JPAEntityLoader`. Para o plug-in `JPAEntityLoader`, se um parâmetro **entityClassName** não estiver configurado, a classe de entidade configurada no mapa de entidade `ObjectGrid` será usada. A mesma classe deve ser usada para o `eXtreme Scale EntityManager` e para o provedor JPA. Este atributo é configurado no bean `JPALoader` ou `JPAEntityLoader`.
- **preloadPartition**: Especifica a partição na qual o pré-carregamento do mapa será iniciado. Se a partição do pré-carregamento for menor que zero ou maior que o número total de partições menos 1, o pré-carregamento do mapa não será iniciado. O valor padrão é -1, significando que o pré-carregamento não é iniciado por padrão. Este atributo é configurado no bean `JPALoader` ou `JPAEntityLoader`.

Diferente dos quatro parâmetros JPA a serem configurados no `eXtreme Scale`, os metadados JPA são usados para recuperar a chave das entidades JPA. Os metadados JPA podem ser configurados como anotação ou como um arquivo `orm.xml` especificado no arquivo `persistence.xml`. Eles não fazem parte da configuração do `eXtreme Scale`.

## 2. Configure os arquivos XML para a configuração do JPA.

Para configurar um `JPALoader` ou `JPAEntityLoader`, consulte Plug-ins para a Comunicação com os Bancos de Dados.

Configure um retorno de chamada de transação `JPATxCallback` junto com a configuração do carregador. O exemplo a seguir é um arquivo descritor XML do `ObjectGrid` (`objectgrid.xml`) que possui um `JPAEntityLoader` e `JPATxCallback` configurado:

### Configurando um utilitário de carga incluindo um retorno de chamada - XML de exemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="JPAEM" entityMetadataXMLFile="jpaEMD.xml">
      <bean id="TransactionCallback"
        className="com.ibm.websphere.objectgrid.jpa.JPATxCallback">
        <property
          name="persistenceUnitName"
          type="java.lang.String"
          value="employeeEMPU" />
        </bean>
      <backingMap name="Employee" pluginCollectionRef="Employee" />
    </objectGrid>
  </objectGrids>

  <backingMapPluginCollections>
    <backingMapPluginCollection id="Employee">
      <bean id="Loader"
        className="com.ibm.websphere.objectgrid.jpa.JPAEntityLoader">
        <property
          name="entityClassName"
          type="java.lang.String"
```



```

        value="com.ibm.ws.objectgrid.jpa.test.entity.Employee"/>
    </bean>
</backingMapPluginCollection>
</backingMapPluginCollections>
</objectGridConfig>

```

Se desejar configurar um JPAPropertyFactory, será necessário usar uma configuração de estilo Spring. A seguir há uma amostra de arquivo de configuração XML, JPAEM\_spring.xml, que configura um bean Spring a ser usado para as configurações do eXtreme Scale.

**Configurando um utilitário de carga incluindo um factory de propriedade JPA - XML de exemplo**

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:aop="http://www.springframework.org/schema/aop"
    xmlns:tx="http://www.springframework.org/schema/tx"
    xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        http://www.springframework.org/schema/beans/spring-beans-2.0.xsd">

    <objectgrid:JPAEntityLoader id="jpaLoader" entityClassName=
        "com.ibm.ws.objectgrid.jpa.test.entity.Employee"/>
    <objectgrid:JPATxCallback id="jpaTxCallback" persistenceUnitName="employeeEMPU" />
</beans>

```

O arquivo XML de configuração Objectgrid.xml vem a seguir. Observe que o nome do ObjectGrid é JPAEM, que corresponde ao nome do ObjectGrid no arquivo de configuração Spring JPAEM\_spring.xml.

**Configuração de um utilitário de carga JPAEM - XML de exemplo**

```

<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
    xmlns="http://ibm.com/ws/objectgrid/config">
    <objectGrids>
        <objectGrid name="JPAEM" entityMetadataXMLFile="jpaEMD.xml">
            <bean id="TransactionCallback"
                className="{spring}jpaTxCallback"/>
            <backingMap name="Employee" pluginCollectionRef="Employee"
                writeBehind="T4"/>
        </objectGrid>
    </objectGrids>

    <backingMapPluginCollections>
        <backingMapPluginCollection id="Employee">
            <bean id="Loader" className="{spring}jpaLoader" />
        </backingMapPluginCollection>
    </backingMapPluginCollections>
</objectGridConfig>

```

Uma entidade pode ser anotada com as anotações do JPA e com as anotações do gerenciador de entidade do eXtreme Scale. Cada anotação possui um XML equivalente que pode ser utilizado. Assim, o eXtreme Scale incluiu o espaço de nomes Spring. Também é possível configurá-los utilizando o suporte a espaço de nomes Spring. Para obter informações adicionais, consulte Visão Geral da Estrutura Spring.

## Configurando um Atualizador de Dados Baseado em Tempo do JPA

É possível configurar uma atualização de banco de dados baseado em tempo usando o XML para uma configuração do eXtreme Scale local ou distribuída. Também é possível definir uma configuração local programaticamente.

### Sobre Esta Tarefa

Para obter mais informações sobre como o atualizador de dados baseado em tempo do Java Persistence API (JPA) trabalha, consulte Atualizador de Dados Baseado em Tempo JPA.

## Procedimento

Criar uma configuração timeBasedDBUpdate.

- **Com o arquivo XML:**

O seguinte exemplo mostra um arquivo objectgrid.xml que contém uma configuração timeBasedDBUpdate:

**Atualizador baseado em tempo do JPA - exemplo de XML**

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="changeOG"
      entityMetadataXMLFile="userEMD.xml">
      <backingMap name="user" >
        <timeBasedDBUpdate timestampField="rowChgTs"
          persistenceUnitName="userderby"
          entityClass="com.test.UserClass"
          mode="INVALIDATE_ONLY"
        />
      </backingMap>
    </objectGrid>
  </objectGrids>
  <backingMapPluginCollections>
</objectGridConfig>
```

Neste exemplo, o mapa "user" é configurado com a atualização do banco de dados baseada em tempo. O modo de atualização de banco de dados é INVALIDATE\_ONLY e o campo do registro de data e hora é rowChgTs.

Quando um ObjectGrid "changeOG" distribuído é iniciado no servidor de contêiner, um encadeamento de atualização de banco de dados baseado em tempo é iniciado automaticamente na partição 0.

- **Programaticamente:**

Se você criar um ObjectGrid local, também é possível criar um objeto TimeBasedDBUpdateConfig e configurá-lo na instância BackingMap:

```
public void setTimeBasedDBUpdateConfig(TimeBasedDBUpdateConfig dbUpdateConfig);
```

Para obter mais informações sobre como configurar um objeto na instância BackingMap, consulte as informações sobre a interface BackingMap na Documentação de API.

Por outro lado, não é possível anotar o campo do registro de data e hora na classe de entidade usando a anotação com.ibm.websphere.objectgrid.jpa.dbupdate.annotation.Timestamp. Ao configurar o valor na classe, não é necessário configurar o timestampField na configuração XML.

## O que Fazer Depois

Inicie um atualizador de dados baseado em tempo do JPA. Consulte Iniciando o Atualizador Baseado em Tempo do JPA para obter mais informações.

---

## Configurando Serviços de Dados REST

É possível usar o serviço de dados REST do WebSphere eXtreme Scale com o WebSphere Application Server versão 7.0, o WebSphere Application Server Community Edition e o Apache Tomcat.

### Sobre Esta Tarefa

A amostra incluída tem código de origem e binários compilados para executar uma grade de dados particionada. Essa amostra demonstra como criar uma grade de

dados simples, modelar os dados utilizando entidades e fornece dois aplicativos clientes de linha de comandos que permitem incluir e consultar entidades com o usando Java ou C#.

O cliente Java de amostra utiliza a API EntityManager do eXtreme Scale Java para persistir e consultar dados na grade de dados. Esse cliente pode ser executado no Eclipse ou utilizando um script de linha de comandos. Observe que o cliente Java de amostra não demonstra o serviço de dados REST, mas permite a atualização de dados na grade, portanto, um navegador da Web ou outros clientes podem ler os dados.

O cliente Microsoft WCF Data Services C# se comunica com a grade de dados do eXtreme Scale por meio dos serviços de dados REST usando a estrutura .NET. O cliente WCF Data Services pode ser usado para atualizar e consultar a grade de dados.

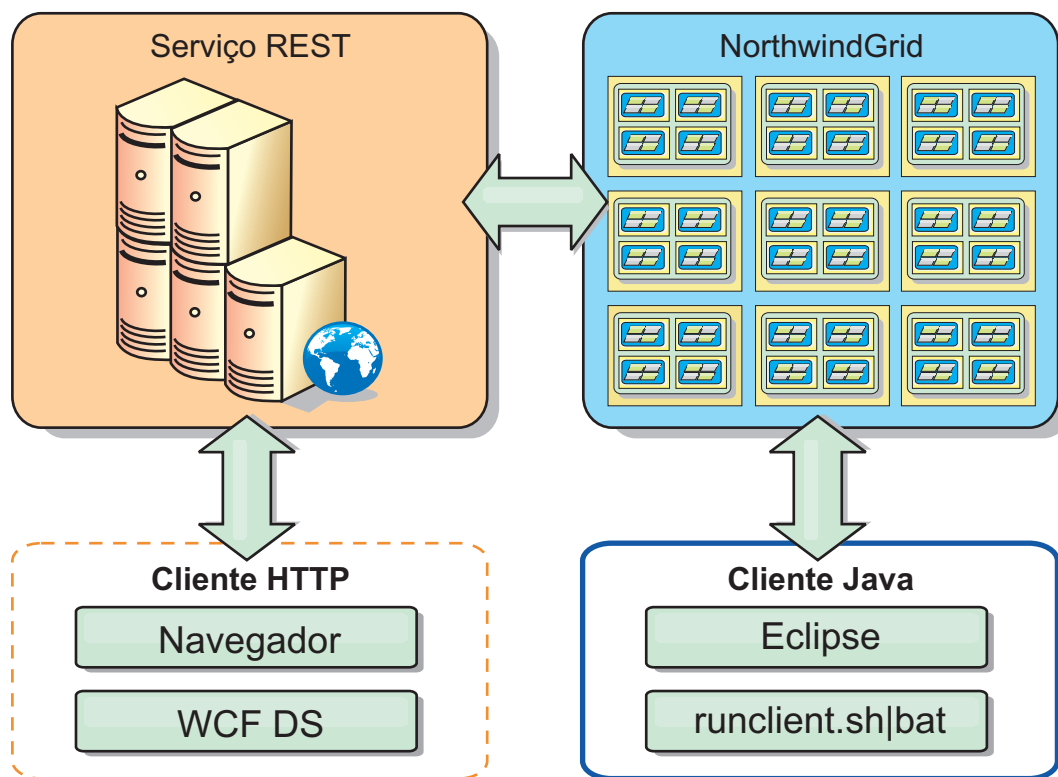


Figura 43. Topologia de Amostra de Introdução. Os clientes HTTP que usam o serviço de dados REST e os clientes Java podem acessar a mesma grade de dados.

### Procedimento

1. Configure e inicie a grade de dados do eXtreme Scale. Consulte “Ativando o Serviço de Dados REST” na página 352.
2. Configure e inicie o serviço de dados REST em um servidor da Web. Consulte “Configurando Servidores de Aplicativos para o Serviço de Dados REST” na página 360.
3. Execute um cliente para interagir com o serviço de dados REST. Duas opções estão disponíveis:
  - a. Execute o cliente Java de amostra para preencher a grade com dados usando a API EntityManager e consulte os dados na grade usando um

navegador da Web e o serviço de dados REST do eXtreme Scale. Consulte “Utilizando um Cliente Java com Serviço de Dados REST” na página 376.

- b. Execute o cliente WCF Data Services C# de amostra. Consulte “Cliente Visual Studio 2008 WCF com Serviço de Dados REST” na página 378.

## Ativando o Serviço de Dados REST

O serviço de dados REST pode representar metadados de entidade do WebSphere eXtreme Scale para representar cada entidade como um EntitySet.

### Iniciando uma Grade de Dados do eXtreme Scale de Amostra

No geral, antes de iniciar o serviço de dados REST, inicie a grade de dados eXtreme Scale. As etapas a seguir iniciarão um único processo do serviço de catálogo do eXtreme Scale e dois processos do servidor de contêiner.

O WebSphere eXtreme Scale pode ser instalado usando três métodos diferentes:

- Instalação por tentativa
- Implementação independente
- Implementação integrada do WebSphere Application Server

### Modelo de Dados Escalável no eXtreme Scale

A amostra Northwind da Microsoft utiliza a tabela Detalhes do Pedido para estabelecer uma associação muitos-para-muitos entre Pedidos e Produtos.

Object to relational mapping specifications (ORMs), como ADO.NET Entity Framework e Java Persistence API (JPA), podem mapear as tabelas e os relacionamentos utilizando entidades. No entanto, esta arquitetura não é escalada. Tudo deve estar localizado na mesma máquina ou em um cluster de máquinas caro para uma boa execução.

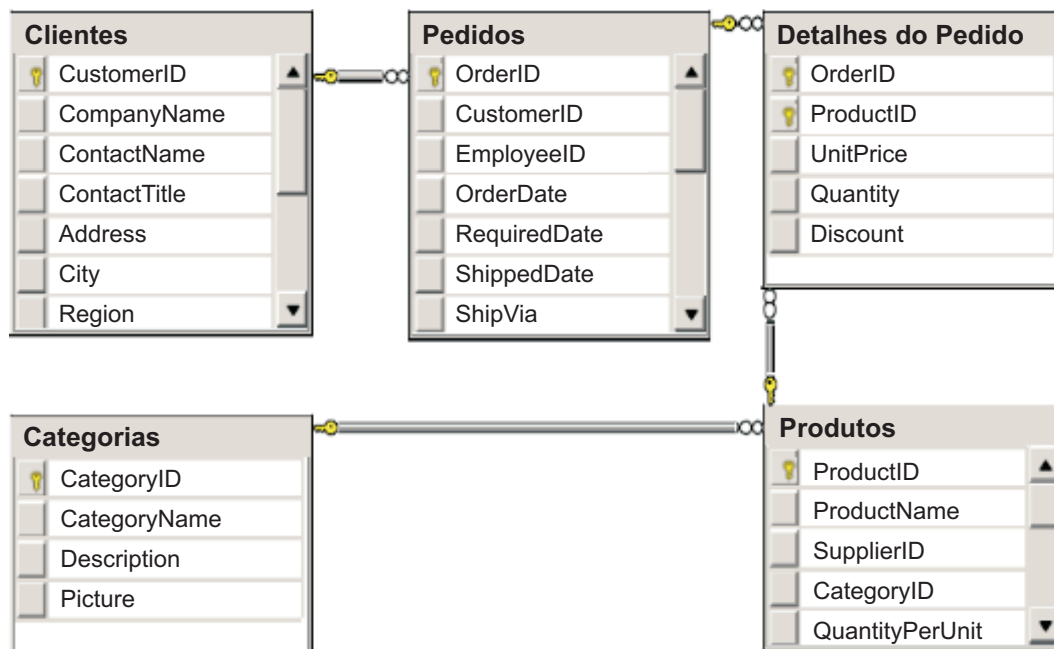


Figura 44. Diagrama do Esquema da Amostra Northwind do Microsoft SQL Server

Para criar uma versão escalável da amostra, as entidades devem ser modeladas de forma que cada entidade ou grupo de entidades relacionadas possa ser particionado com base em uma única chave. Ao criar partições em uma única chave, os pedidos podem ser espalhados entre vários servidores independentes. Para atingir essa configuração, as entidades foram divididas em duas árvores: a árvore Cliente e Ordem e a árvore Produto e Categoria. Neste modelo, cada árvore pode ser particionada de forma independente e, portanto, pode crescer a taxas diferentes, aumentando a escalabilidade.

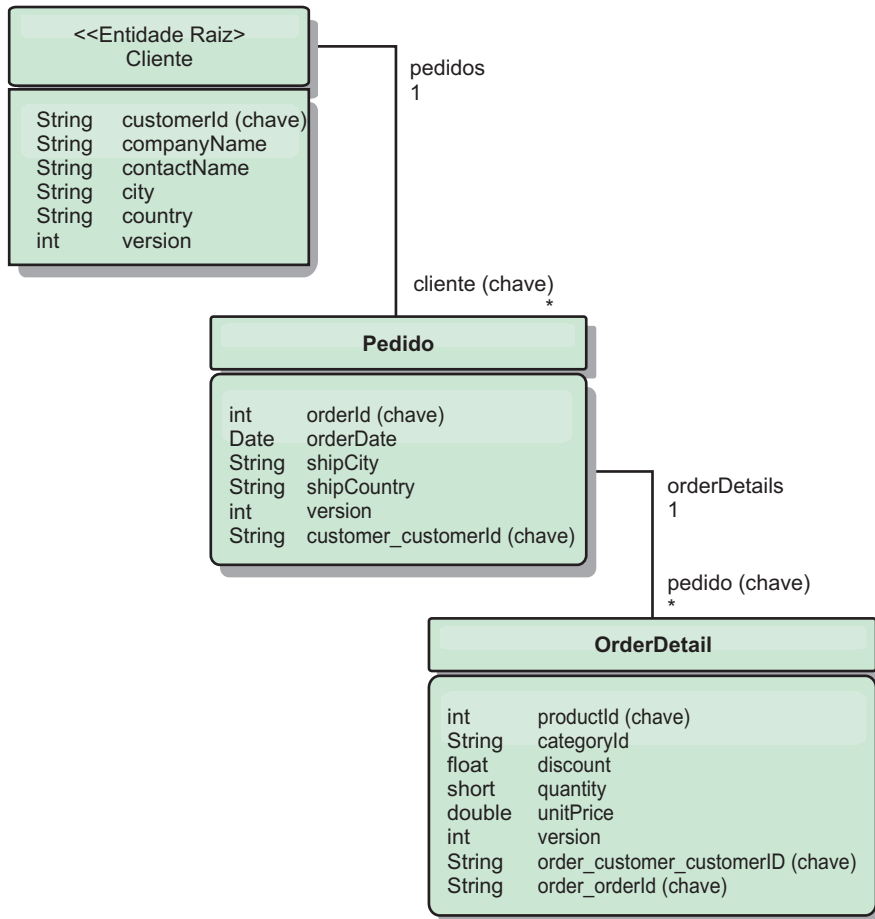


Figura 45. Diagrama do Esquema de Entidade Cliente e Ordem

Por exemplo, Order e Product têm números inteiros separados exclusivos como chaves. De fato, as tabelas Order e Product são realmente independentes uma da outra. Por exemplo, considere o efeito do tamanho de um catálogo, o número de produtos vendidos, com o número total de pedidos. Intuitivamente, pode parecer que ter muitos produtos implica também em ter muitas ordens, mas este não é necessariamente o caso. Se isso fosse verdade, você poderia facilmente aumentar as vendas incluindo mais produtos em seu catálogo. Orders e Products têm suas próprias tabelas independentes. É possível estender mais este conceito para que as ordens e os produtos tenham, cada um, suas próprias grades de dados separadas. Com grades de dados independente, é possível controlar o número de partições e servidores, além do tamanho de cada grade de dados separadamente para que seu aplicativo possa escalar. Se dobrar o tamanho de seu catálogo, a grade dados de produtos deverá ser dobrada, mas a grade de ordens pode permanecer inalterada. O contrário é verdade para um pico de pedidos ou pico de pedidos esperado.

No esquema, um Customer tem zero ou mais Orders, e um Order tem itens de linha (OrderDetail), cada um com um produto específico. Um Produto é identificado pelo ID (a chave do Produto) em cada OrderDetail. Uma única grade de dados armazena Clients, Orders e OrderDetails com o Client como a entidade raiz da grade de dados. É possível recuperar os Clientes pelo ID, mas é necessário obter as Ordens que começam com o ID do Cliente. Então, o ID do cliente é incluído na Ordem como parte de sua chave. Da mesma forma, o ID do cliente e o ID da ordem fazem parte do ID do OrderDetail.

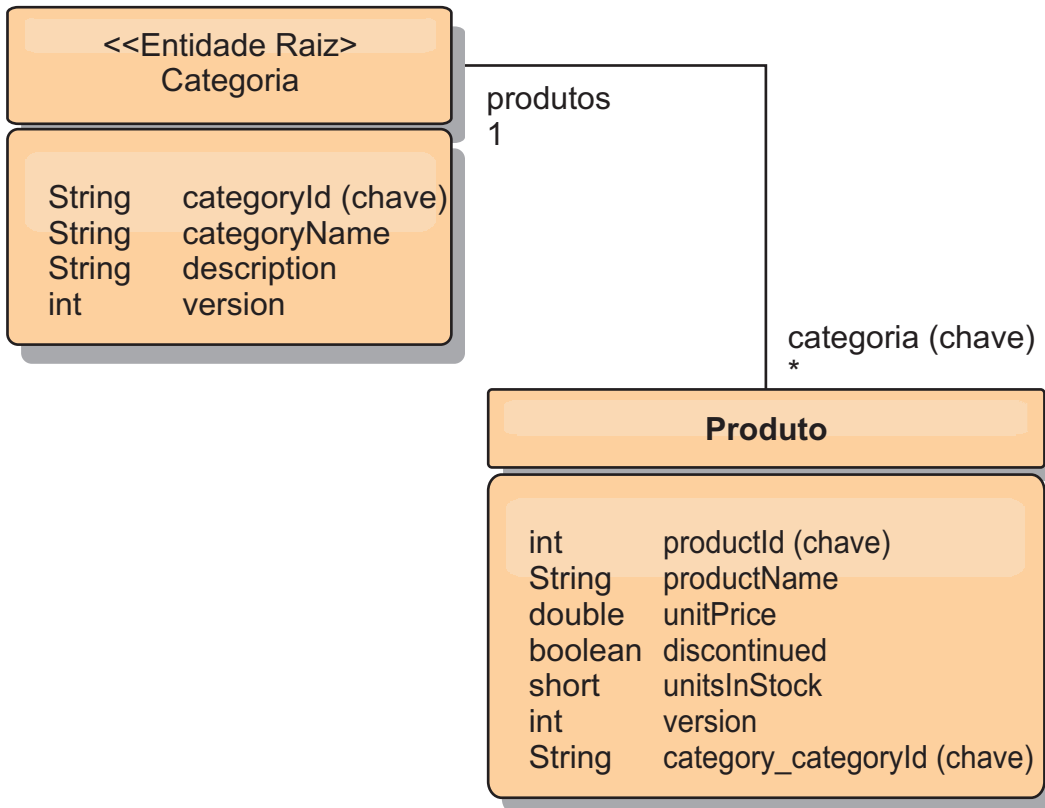


Figura 46. Diagrama do Esquema de Entidade Categoria e Produto

No esquema Category e Product, Category é a raiz do esquema. Com este esquema, os clientes podem consultar os produtos por categoria. Consulte “Recuperando e Atualizando Dados com o REST” para obter detalhes adicionais sobre as principais associações e sua importância.

### Recuperando e Atualizando Dados com o REST

O protocolo OData requer que todas as entidades possam ser endereçadas por sua forma canônica. Isso significa que cada entidade deve incluir a chave da entidade raiz particionada, a raiz do esquema.

A seguir está um exemplo de como utilizar a associação de uma entidade raiz para endereçar um filho em:

```
/Customer('ACME')/order(100)
```

No WCF Data Services, a entidade-filha deve ser diretamente endereçável, o que significa que a chave na raiz do esquema deve fazer parte da chave da filha:

```
/Order(customer_customerId='ACME', orderId=100). Isso é conseguido por meio da criação de uma associação com a entidade raiz, em que a associação
```

um-para-um ou muitos-para-um com a entidade raiz também é identificada como uma chave. Quando entidades são incluídas como parte da chave, os atributos da entidade pai são expostos como propriedades-chave.

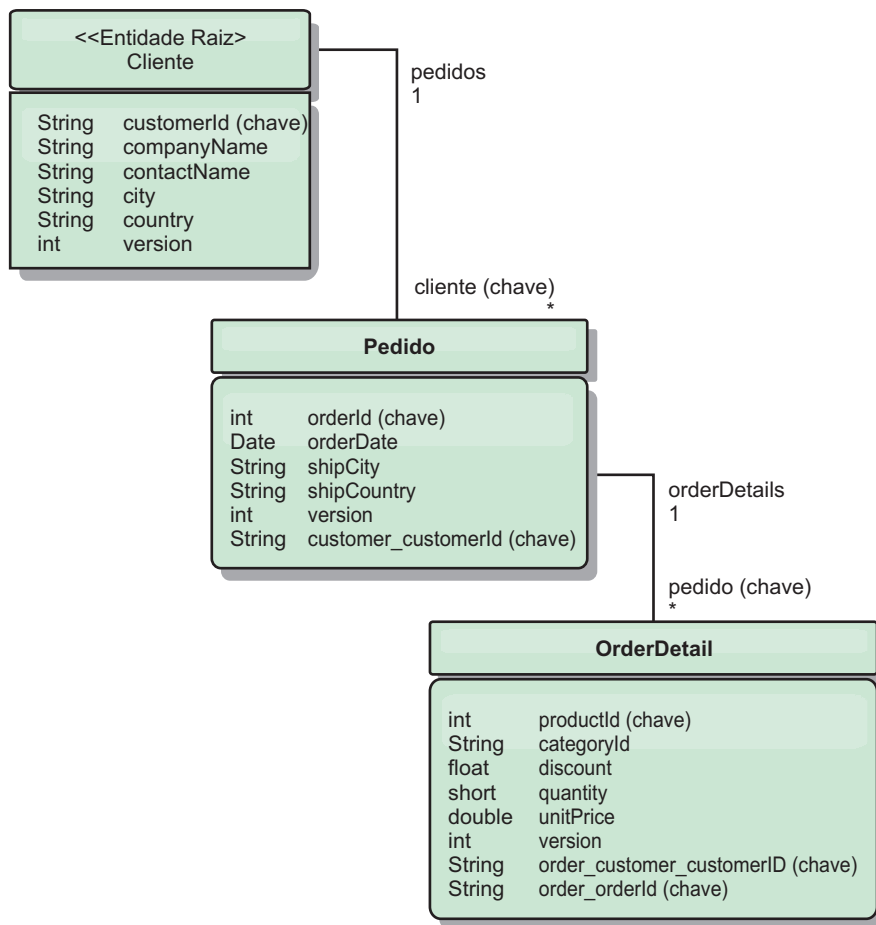


Figura 47. Diagrama do Esquema de Entidade Cliente e Ordem

O diagrama do esquema de entidade Cliente/Ordem ilustra como cada entidade é particionada usando o Cliente. A entidade Order inclui Customer como parte de sua chave e é, portanto, diretamente acessível. O serviço de dados REST expõe todas as associações chave como propriedades individuais: Order tem `customer_customerId` e OrderDetail tem `order_customer_customerId` e `order_orderId`.

Utilizando a API EntityManager, é possível localizar Order utilizando o id de Customer e Order:

```

transaction.begin();
// Look-up the Order using the Customer. We only include the Id
// in the Customer class when building the OrderId key instance.
Order order = (Order) em.find(Order.class,
    new OrderId(100, new Customer('ACME')));
...
transaction.commit();
  
```

Ao utilizar o serviço de dados REST, Order pode ser recuperada com qualquer uma das seguintes URLs:

- `/Order(orderId=100, customer_customerId='ACME')`
- `/Customer('ACME')/orders?$filter=orderId eq 100`



A chave do cliente é endereçada utilizando o nome do atributo da entidade Customer, um caractere sublinhado e o nome do atributo do id de Customer: customer\_customerId.

Uma entidade também pode incluir uma entidade não raiz como parte de sua chave se todos os ancestrais para a entidade não raiz tiverem associações de chave com a raiz. Neste exemplo, OrderDetail tem uma associação de chave com Order, e Order tem uma associação de chave com a entidade Customer raiz. Utilizando a API EntityManager:

```
transaction.begin();
// Construct an OrderDetailId key instance. It includes
// The Order and Customer with only the keys set.
Customer customerACME = new Customer("ACME");
Order order100 = new Order(100, customerACME);
OrderDetailId orderDetailKey =
    new OrderDetailId(order100, "COMP");
OrderDetail orderDetail = (OrderDetail)
    em.find(OrderDetail.class, orderDetailKey);
...
```

O serviço de dados REST permite endereçar OrderDetail diretamente:

```
/OrderDetail(productId=500,
order_customer_customerId='ACME', order_orderId =100)
```

A associação da entidade OrderDetail com a entidade Product foi quebrada para permitir o particionamento de Orders e Product Inventory independentemente. A entidade OrderDetail armazena o ID do produto e categoria de um relacionamento sólido. Ao desacoplar os dois esquemas de entidade, apenas uma partição é acessada por vez.

O esquema Categoria e Produto ilustrado no diagrama mostra que a entidade-raiz é Categoria e cada Produto tem uma associação com uma entidade Categoria. A entidade Category está incluída na identidade de Product. O serviço de dados REST expõe uma propriedade-chave: category\_categoryId, que permite o endereçamento direto de Product.

Como Categoria é a entidade-raiz, em um ambiente particionado, a Categoria deve ser conhecida para localizar o Produto. Utilizando a API EntityManager, a transação deve ser retida na entidade Category antes de localizar Product.

Utilizando a API EntityManager:

```
transaction.begin();
// Create the Category root entity with only the key. This
// allows us to construct a ProductId without needing to find
// The Category first. The transaction is now pinned to the
// partition where Category "COMP" is stored.
Category cat = new Category("COMP");
Product product = (Product) em.find(Product.class,
    new ProductId(500, cat));
...
```

O serviço de dados REST permite endereçar Product diretamente:

```
/Product(productId=500,
category_categoryId='COMP')
```

## Iniciando uma Grade de Dados Independente para os Serviços de Dados REST

Siga estas etapas para iniciar a grade de dados de amostra do serviço REST do WebSphere eXtreme Scale para uma implementação do eXtreme Scale independente.

### Antes de Iniciar

Instale o Teste do WebSphere eXtreme Scale ou o produto integral:

- Instale a versão independente do produto e aplique quaisquer correções subsequentes.
- Faça download e extraia a avaliação do WebSphere eXtreme Scale Versão 7.1 ou posterior, que inclui o serviço de dados REST do WebSphere eXtreme Scale.

### Sobre Esta Tarefa

Inicie a grade de dados de amostra do WebSphere eXtreme Scale.

### Procedimento

1. Inicie o processo do serviço de catálogo. Abra uma janela do terminal ou da linha de comandos e configure a variável de ambiente JAVA\_HOME:

- **Linux** **UNIX** `export JAVA_HOME=java_home`
- **Windows** `set JAVA_HOME=java_home`

2. `cd restservice_home/gettingstarted`

3. Inicie o processo do serviço de catálogo. Para iniciar o serviço *sem* a segurança do eXtreme Scale, use os comandos a seguir.

- **Linux** **UNIX** `./runcat.sh`
- **Windows** `runcat.bat`

Para iniciar o serviço com a segurança do eXtreme Scale, use os comandos a seguir.

- **Linux** **UNIX** `./runcat_secure.sh`
- **Windows** `runcat_secure.bat`

4. Inicie dois processos do servidor de contêiner. Abra uma janela do terminal ou de linha de comandos e configure a variável de ambiente JAVA\_HOME:

- **Linux** **UNIX** `export JAVA_HOME=java_home`
- **Windows** `set JAVA_HOME=java_home`

5. `cd restservice_home/gettingstarted`

6. Inicie um processo do servidor de contêiner:

Para iniciar o servidor sem a segurança do eXtreme Scale, use os comandos a seguir:

- **Linux** **UNIX** `./runcontainer.sh container0`
- **Windows** `runcontainer.bat container0`

Para iniciar o servidor com a segurança do eXtreme Scale, use os comandos a seguir.

- **Linux** **UNIX** `./runcontainer_secure.sh container0`
- **Windows** `runcontainer_secure.bat container0`

7. Abra uma janela do terminal ou de linha de comandos e configure a variável de ambiente `JAVA_HOME`:

- `Linux` `UNIX` `export JAVA_HOME=java_home`
- `Windows` `set JAVA_HOME=java_home`

8. `cd restservice_home/gettingstarted`

9. Inicie um segundo processo do servidor de contêiner.

Para iniciar o servidor sem a segurança do eXtreme Scale, use os comandos a seguir.

- `Linux` `UNIX` `./runcontainer.sh container1`
- `Windows` `runcontainer.bat container1`

Para iniciar o servidor com a segurança do eXtreme Scale, use os comandos a seguir.

- `Linux` `UNIX` `./runcontainer_secure.sh container1`
- `Windows` `runcontainer_secure.bat container1`

## Resultados

Aguarde até que os contêineres do eXtreme Scale estejam prontos antes de continuar com as próximas etapas. Os servidores de contêiner estão prontos quando a seguinte mensagem é exibida na janela do terminal:

```
CWOBJ1001I: O Servidor de ObjectGrid container_name está pronto para processar pedidos.
```

Em que *container\_name* é o nome do contêiner que foi iniciado.

## Iniciando uma Grade de Dados para os Serviços de Dados REST no WebSphere Application Server

Siga estas etapas para iniciar uma grade de dados de amostra do serviço REST do WebSphere eXtreme Scale independente para uma implementação do WebSphere eXtreme Scale que está integrada ao WebSphere Application Server. Embora o WebSphere eXtreme Scale esteja integrado ao WebSphere Application Server, estas etapas iniciam um processo e um contêiner do serviço de catálogo do WebSphere eXtreme Scale independente.

### Antes de Iniciar

Instale o produto em um diretório de instalação do WebSphere Application Server Versão 7.0.0.5 ou posterior com a segurança desativada. Aumente pelo menos um perfil do servidor de aplicativos.

### Sobre Esta Tarefa

Inicie a grade de dados de amostra do WebSphere eXtreme Scale.

### Procedimento

1. Inicie o processo do serviço de catálogo. Abra uma janela do terminal ou da linha de comandos e configure a variável de ambiente `JAVA_HOME`:

- `Linux` `UNIX` `export JAVA_HOME=java_home`
- `Windows` `set JAVA_HOME=java_home`

- ```
cd restservice_home/gettingstarted
```
- Inicie o processo do serviço de catálogo.  
Para iniciar o servidor sem a segurança do eXtreme Scale, use os comandos a seguir.
    - `Linux` `UNIX` `./runcat.sh`
    - `Windows` `runcat.bat`Para iniciar o servidor com a segurança do eXtreme Scale, use os comandos a seguir.
    - `Linux` `UNIX` `./runcat_secure.sh`
    - `Windows` `runcat_secure.bat`
  - Inicie dois processos do servidor de contêiner. Abra uma janela do terminal ou de linha de comandos e configure a variável de ambiente `JAVA_HOME`:
    - `Linux` `UNIX` `export JAVA_HOME=java_home`
    - `Windows` `set JAVA_HOME=java_home`
  - Inicie um processo do servidor de contêiner.  
Para iniciar o servidor sem a segurança do eXtreme Scale, use os comandos a seguir.
    - Abra uma janela de linha de comandos.
    - `cd restservice_home/gettingstarted`
    - Para iniciar o servidor *sem* a segurança do eXtreme Scale, use os comandos a seguir.
      - `Linux` `UNIX` `./runcontainer.sh container0`
      - `Windows` `runcontainer.bat container0`
    - Para iniciar o servidor com a segurança do eXtreme Scale, use os comandos a seguir.
      - `Linux` `UNIX` `./runcontainer_secure.sh container0`
      - `Windows` `runcontainer_secure.bat container0`
  - Inicie um segundo processo do servidor de contêiner.
    - Abra uma janela de linha de comandos.
    - `cd restservice_home/gettingstarted`
    - Para iniciar o servidor *sem* a segurança do eXtreme Scale, use os comandos a seguir.
      - `Linux` `UNIX` `./runcontainer.sh container1`
      - `Windows` `runcontainer.bat container1`
    - Para iniciar o servidor *com* a segurança do eXtreme Scale, use os comandos a seguir.
      - `Linux` `UNIX` `./runcontainer_secure.sh container1`
      - `Windows` `runcontainer_secure.bat container1`

## Resultados

Aguarde até que os servidores de contêiner estejam prontos antes de continuar com as próximas etapas. Os servidores de contêiner estarão prontos quando a seguinte mensagem for exibida:

CWOBJ1001I: O Servidor de ObjectGrid *container\_name* está pronto para processar pedidos.

Em que *container\_name* é o nome do contêiner que foi iniciado na etapa anterior.

## Configurando Servidores de Aplicativos para o Serviço de Dados REST

É possível configurar vários servidores de aplicativos para usar o serviço de dados REST.

### Implementando o Serviço de Dados REST no WebSphere Application Server

Este tópico descreve como configurar o serviço de dados REST do WebSphere eXtreme Scale no WebSphere Application Server ou no WebSphere Application Server Network Deployment Versão 6.1.0.25 ou posterior. Estas instruções também se aplicam às implementações nas quais o WebSphere eXtreme Scale é integrado com a implementação do WebSphere Application Server.

#### Antes de Iniciar

Você deve ter um dos seguintes ambientes em seu sistema para configurar e implementar o serviço de dados REST para o WebSphere eXtreme Scale.

- O WebSphere Application Server com o cliente WebSphere eXtreme Scale independente:
  - O WebSphere eXtreme Scale Trial Versão 7.1 com o serviço de dados REST é transferido por download e extraído ou o WebSphere eXtreme Scale Versão 7.1.0.0 com a correção 2 acumulativa do produto é instalado em um diretório independente.
  - O WebSphere Application Server Versão 6.1.0.25 ou 7.0.0.5 ou posterior está instalado e em execução.
- O WebSphere Application Server integrado com o WebSphere eXtreme Scale: O WebSphere eXtreme Scale Versão 7.1.0.0 com a correção acumulativa 2 ou posterior é instalado sobre o WebSphere Application Server Versão 6.1.0.25 ou 7.0 ou posterior.

**Dica:** O serviço de dados REST do WebSphere eXtreme Scale requer apenas que a opção do cliente do WebSphere eXtreme Scale seja instalada. O perfil não precisa ser aumentado.

Leia sobre como ativar a segurança Java 2 no centro de informações do WebSphere Application Server.

#### Procedimento

1. Configure e inicie uma grade de dados.
  - a. Para obter detalhes sobre como configurar uma grade de dados para usar com o serviço de dados REST, consulte “Iniciando uma Grade de Dados para os Serviços de Dados REST no WebSphere Application Server” na página 358.
  - b. Verifique se um cliente pode se conectar com e acessar entidades na grade de dados. Para obter um exemplo, consulte “Tutorial: Introdução ao WebSphere eXtreme Scale” na página 1.
2. Construa a JAR da configuração do serviço eXtreme Scale REST ou o diretório. Consulte as informações sobre como empacotar e implementar o serviço REST no “Instalando o Serviço de Dados REST” na página 198.

3. Inclua o diretório ou JAR da configuração do serviço de dados REST no caminho de classe do servidor de aplicativos:
  - a. Abra o console administrativo do WebSphere Application Server
  - b. Navegue para **Ambiente > Bibliotecas Compartilhadas**
  - c. Clique em **Novo**
  - d. Inclua as seguintes entradas nos campos apropriados:
    - Nome: `extremescale_rest_configuration`
    - Caminho de classe: <diretório ou jar da configuração do serviço REST>
  - e. Clique em **OK**
  - f. Salve as mudanças na configuração principal
4. Inclua o JAR de tempo de execução do cliente do WebSphere eXtreme Scale, `wsogclient.jar`, e o JAR ou o diretório de configuração do serviço de dados REST no caminho de classe do servidor de aplicativos: Essa etapa não será necessária se o WebSphere eXtreme Scale for integrado com o WebSphere Application Server.
  - a. Abra o console administrativo do WebSphere Application Server.
  - b. Navegue para **Ambiente > Bibliotecas Compartilhadas**.
  - c. Clique em **Novo**.
  - d. Inclua as seguintes entradas nos campos:
    - Nome: `extremescale_client_v71`
    - Caminho da classe: `wxs_home/lib/wsogclient.jar`

**Lembre-se:** Inclua cada caminho em uma linha separada.

- e. Clique em **OK**.
- f. Salve as alterações na configuração principal.
5. Instale o arquivo EAR do serviço de dados REST, `wxsrestservice.ear`, no WebSphere Application Server usando o console administrativo:
  - a. Abra o console administrativo do WebSphere Application Server.
  - b. Clique em **Aplicativos > Novo aplicativo**.
  - c. Navegue para o arquivo `/lib/wxsrestservice.ear` no sistema de arquivos, selecione-o e clique em **Avançar**.
    - Se estiver usando o WebSphere Application Server Versão 7.0, clique em **Avançar**.
    - Se estiver usando o WebSphere Application Server Versão 6.1, insira um valor de Raiz de Contexto com o nome: `/wxsrestservice` e continue na próxima etapa.
  - d. Escolha uma opção de instalação detalhada e clique em **Avançar**.
  - e. Na tela de avisos de segurança do aplicativo, clique em **Continuar**.
  - f. Escolha as opções de instalação padrão e clique em **Avançar**.
  - g. Escolha um servidor para o qual o aplicativo será mapeado e clique em **Avançar**.
  - h. Na página de recarregamento de JSP, utilize os padrões e clique em **Avançar**.
  - i. Na página de bibliotecas compartilhadas, mapeie o módulo `wxsrestservice.war` para as bibliotecas compartilhadas definidas:
    - `extremescale_rest_configuration`
    - `extremescale_client_v71`

- Dica:** Esta biblioteca compartilhada será necessária apenas se o WebSphere eXtreme Scale não estiver integrado ao WebSphere Application Server.
- j. Na página de relacionamentos da biblioteca compartilhada do mapa, utilize os padrões e clique em **Avançar**.
  - k. Na página hosts virtuais do mapa, clique nos padrões e em **Avançar**.
  - l. Na página de raízes de contexto do mapa, configure a raiz de contexto como: `wxsrestservice`
  - m. Na tela Resumo, clique em **Concluir** para concluir a instalação.
  - n. Salve as alterações na configuração principal.
6. Inicie o aplicativo de serviço de dados REST `wxsrestservice`:
- a. Acesse o aplicativo no console administrativo.
    - WebSphere Application Server Versão 7.0: No console administrativo, clique em **Aplicativos > Tipos de Aplicativos > Aplicativos WebSphere**.
    - WebSphere Application Server Versão 6.0: No console administrativo, clique em **Aplicativos > Aplicativos Corporativos**.
  - b. Marque a caixa de seleção ao lado do aplicativo `wxsrestservice` e clique em **Iniciar**.
  - c. Revise o arquivo `SystemOut.log` para o perfil do servidor de aplicativos. Quando o serviço de dados REST for iniciado com êxito, a seguinte mensagem será exibida no arquivo `SystemOut.log` para o perfil do servidor:  
`CW0BJ4000I: O serviço de dados REST do WebSphere eXtreme Scale foi iniciado.`
7. Verifique se o serviço de dados do REST está funcionando: O número da porta pode ser localizado no arquivo `SystemOut.log` dentro do diretório de logs do perfil do servidor de aplicativos ao consultar a primeira porta exibida para o identificador de mensagem: `SRVE0250I..` A porta padrão é 9080.  
Por exemplo:`http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/` Resultado: O documento de serviço AtomPub é exibido.  
Por exemplo: `http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata`. O documento Entity Model Data Extensions (EDMX) é exibido.
8. Para parar os processos da grade de dados, use CTRL+C na respectiva janela de comando.

### **Iniciando Serviços de Dados REST com o WebSphere eXtreme Scale integrado ao WebSphere Application Server 7.0:**

Este tópico descreve como configurar e iniciar o serviço de dados REST do eXtreme Scale usando o WebSphere Application Server versão 7.0 que foi integrado e aumentado com o WebSphere eXtreme Scale.

#### **Antes de Iniciar**

Verifique se a grade de dados do eXtreme Scale independente de amostra foi iniciada. Consulte o “Ativando o Serviço de Dados REST” na página 352 para obter detalhes sobre como iniciar a grade de dados.

#### **Sobre Esta Tarefa**

Para introduzir o serviço de dados REST do WebSphere eXtreme Scale usando o WebSphere Application Server, siga estas etapas:



## Procedimento

1. Inclua o JAR de configuração de amostra do serviço de dados REST do WebSphere eXtreme Scale no caminho de classe:
  - a. Abra o Console Administrativo do WebSphere
  - b. Navegue para Ambiente -> Bibliotecas Compartilhadas
  - c. Clique em Novo
  - d. Inclua as seguintes entradas nos campos apropriados:
    - 1) Nome: extremescale\_gettingstarted\_config
    - 2) Caminho de Classe
      - restservice\_home/gettingstarted/restclient/bin
      - restservice\_home/gettingstarted/common/bin

**Lembre-se:** Cada caminho deve aparecer em uma linha diferente.

- e. Clique em **OK**
- f. Salve as mudanças na configuração principal
2. Instale o arquivo EAR do serviço de dados REST, wxsrestservice.ear, no WebSphere Application Server utilizando o console administrativo do WebSphere:
  - a. Abra o console administrativo do WebSphere
  - b. Navegue para Aplicativos -> Novo Aplicativo
  - c. Navegue para o arquivo restservice\_home/lib/wxsrestservice.ear no sistema de arquivos. Selecione o arquivo e clique em **Avançar**.
  - d. Escolha as opções de instalação detalhadas e clique em **Avançar**.
  - e. Na tela de avisos de segurança do aplicativo, clique em **Continuar**.
  - f. Escolha as opções de instalação padrão e clique em **Avançar**.
  - g. Escolha um servidor para o qual mapear o módulo wxsrestservice.war e clique em **Avançar**.
  - h. Na página de recarregamento de JSP, utilize os padrões e clique em **Avançar**.
  - i. Na página bibliotecas compartilhadas, mapeie o módulo "wxsrestservice.war" para as seguintes bibliotecas compartilhadas que foram definidas durante a etapa 1: extremescale\_gettingstarted\_config
  - j. Na página de relacionamentos da biblioteca compartilhada do mapa, utilize os padrões e clique em **Avançar**.
  - k. Na página hosts virtuais do mapa, clique nos padrões e em **Avançar**.
  - l. Na página de raízes de contexto do mapa, configure a raiz de contexto como: wxsrestservice.
  - m. Na tela Resumo, clique em **Concluir** para concluir a instalação.
  - n. Salve as alterações na configuração principal.
3. Se a grade de dados do eXtreme Scale foi iniciada com a segurança do eXtreme Scale ativada, configure a seguinte propriedade no arquivo restservice\_home/gettingstarted/restclient/bin/wxsRestService.properties.

ogClientPropertyFile=restservice\_home/gettingstarted/security/security.ogclient.properties

4. Inicie o servidor de aplicativos e o aplicativo do serviço de dados REST do eXtreme Scale "wxsrestservice".

Após o aplicativo ser iniciado, revise o SystemOut.log para o servidor de aplicativos e verifique se a seguinte mensagem aparece: CW0BJ4000I: O serviço de dados REST do WebSphere eXtreme Scale foi iniciado.

5. Verifique se o serviço de dados REST está funcionando:
  - a. Abra um navegador e navegue para:  
`http://localhost:9080/wxsrestservice/restservice/NorthwindGrid`  
 O documento de serviço para NorthwindGrid é exibido.
  - b. Navegue para:  
`http://localhost:9080/wxsrestservice/restservice/NorthwindGrid/$metadata`  
 O documento Entity Model Data Extensions (EDMX) é exibido
6. Para parar os processos da grade de dados, use CTRL+C na respectiva janela de comando para parar o processo.

## Implementando o Serviço de Dados REST no WebSphere Application Server Community Edition

É possível configurar o serviço de dados REST do eXtreme Scale no WebSphere Application Server Community Edition Version 2.1.1.3 ou posterior.

### Antes de Iniciar

- Um IBM (recomendado) ou Sun JRE ou JDK, Versão 5 ou posterior está instalado e a variável de ambiente JAVA\_HOME está configurada.
- Faça download e instale o WebSphere Application Server Community Edition Versão 2.1.1.3 ou posterior no diretório `wasce_root`, por exemplo, o diretório `/opt/IBM/wasce`. Leia as instruções de instalação para obter informações sobre o versão 2.1.1 ou outras versões.

### Procedimento

1. Configure e inicie uma grade de dados.
  - a. Para obter detalhes sobre como configurar uma grade de dados do eXtreme Scale para usar com o serviço de dados REST, consulte o “Iniciando uma Grade de Dados Independente para os Serviços de Dados REST” na página 357.
  - b. Verifique se um cliente do eXtreme Scale pode se conectar com, e acessar, as entidades na grade. Para obter um exemplo, consulte “Tutorial: Introdução ao WebSphere eXtreme Scale” na página 1.
2. Construa a JAR da configuração do serviço eXtreme Scale REST ou o diretório. Consulte as informações de empacotamento e implementação no tópico “Instalando o Serviço de Dados REST” na página 198 para obter detalhes.
3. Inicie o servidor WebSphere Application Server Community Edition:
  - a. Para iniciar o servidor sem a segurança Java SE ativada, execute o seguinte comando:
 

UNIX
Linux
`wasce_root/bin/startup.sh`

Windows
`wasce_root/bin/startup.bat`
  - b. Para iniciar o servidor com a segurança Java SE ativada, siga estas etapas:
 

UNIX
Linux

    - 1) Abra uma janela de linha de comandos ou de terminal e execute o seguinte comando de cópia (ou copie o conteúdo do arquivo de política especificado na política existente): `cp restservice_home/gettingstarted/wasce/geronimo.policy wasce_root/bin`
    - 2) Edite o arquivo `wasce_root/bin/setenv.sh`

- 3) Depois da linha que contém "WASCE\_JAVA\_HOME=", inclua o seguinte:
 

```
export JAVA_OPTS="-Djava.security.manager
-Djava.security.policy=geronimo.policy"
```

**Windows**

- 1) Abra uma janela de linha de comandos e execute o seguinte comando de cópia, ou copie o conteúdo do arquivo de políticas especificado na política existente:
 

```
copy restservice_home\gettingstarted\wasce\geronimo.policy\bin
```
  - 2) Edite o arquivo `wasce_root\bin\setenv.bat`
  - 3) Após a linha contendo "set WASCE\_JAVA\_HOME=", inclua o seguinte:
 

```
set JAVA_OPTS="-Djava.security.manager
-Djava.security.policy=geronimo.policy"
```
4. Inclua o JAR de tempo de execução do cliente ObjectGrid ao repositório WebSphere Application Server Community Edition:
    - a. Abra o console administrativo WebSphere Application Server Community Edition e efetue login. A URL padrão é: `http://localhost:8080/console`, o ID do usuário padrão é `system` e a senha é `manager`.
    - b. Clique no link **Repositório** no lado esquerdo da janela do console, na pasta **Serviços**.
    - c. Na seção **Incluir Archive no Repositório**, preencha o seguinte nas caixas de texto de entrada:

Tabela 24. Incluir Archive no Repositório

| Caixa de texto | Valor                                  |
|----------------|----------------------------------------|
| File           | <code>wxs_home/lib/ogclient.jar</code> |
| Grupo          | <code>com.ibm.websphere.xs</code>      |
| Artefato       | <code>ogclient</code>                  |
| Versão         | <code>7.1</code>                       |
| Tipo           | JAR                                    |

- d. Clique no botão Instalar
- Consulte a seguinte nota técnica para obter detalhes sobre as diferentes maneiras pelas quais as dependências de classe e de biblioteca podem ser configuradas: Especificando dependências externas para aplicativos em execução no WebSphere Application Server Community Edition.
5. Implemente e inicie o módulo do serviço de dados REST, o arquivo `wxsrestservice.war`, para o servidor WebSphere Application Server Community Edition.
    - a. Copie e edite o arquivo XML do plano de implementação de amostra: `restservice_home/gettingstarted/wasce/geronimo-web.xml` para incluir dependências do caminho para o JAR ou o diretório de configuração do seu serviço de dados REST. Consulte a seção para obter um exemplo sobre como configurar o caminho de classe para incluir seu arquivo `wxsRestService.properties` e outros arquivos de configuração e classes de metadados.
    - b. Abra o console administrativo WebSphere Application Server Community Edition e efetue login.

**Dica:** A URL padrão é: `http://localhost:8080/console`. O ID de usuário padrão é `system` e a senha é `manager`.

- c. Clique no link **Implementar Novo** no lado esquerdo da janela do console.
- d. Na página **Instalar Novos Aplicativos**, insira os seguintes valores nas caixas de texto:

Tabela 25. Instalar Novos Aplicativos

| Caixa de texto | Valor                                                  |
|----------------|--------------------------------------------------------|
| Archive        | restservice_home/lib/wxsrestservice.war                |
| Plano          | restservice_home/gettingstarted/wasce/geronimo-web.xml |

**Dica:** Use o caminho para o arquivo `geronimo-web.xml` que você copiou e editou na etapa 3.

- e. Clique no botão **Instalar**. A página do console indica que o aplicativo foi instalado e iniciado com sucesso.
  - f. Examine o log de saída do sistema ou o console do WebSphere Application Server Community Edition para verificar se o serviço de dados REST foi iniciado com êxito. A seguinte mensagem deve aparecer:  
 CWOBJ4000I: O serviço de dados REST do WebSphere eXtreme Scale foi iniciado.
6. Inicie o servidor WebSphere Application Server Community Edition executando o seguinte comando:
- **UNIX** **Linux** `wasce_root/bin/startup.sh`
  - **Windows** `wasce_root/bin/startup.bat`
7. Instale o serviço de dados REST do eXtreme Scale e a amostra fornecida no servidor WebSphere Application Server Community Edition:
- a. Inclua o JAR de tempo de execução do cliente ObjectGrid ao repositório WebSphere Application Server Community Edition:
    - 1) Abra o console administrativo WebSphere Application Server Community Edition e efetue login. A URL padrão é: `http://localhost:8080/console`. O ID de usuário padrão é `system` e a senha é `manager`.
    - 2) Clique no link **Repositório** no lado esquerdo da janela do console, na pasta **Serviços**.
    - 3) Na seção **Incluir Archive no Repositório**, preencha o seguinte nas caixas de texto de entrada:

Tabela 26. Incluir Archive no Repositório

| Caixa de texto | Valor                     |
|----------------|---------------------------|
| File           | wxs_home/lib/ogclient.jar |
| Grupo          | com.ibm.websphere.xs      |
| Artefato       | ogclient                  |
| Versão         | 7.1                       |
| Tipo           | JAR                       |

- 4) Clique no botão **Instalar**.

**Dica:** Consulte a seguinte nota técnica para obter detalhes sobre as diferentes maneiras pelas quais as dependências de classe e de biblioteca

podem ser configuradas: Especificando dependências externas para aplicativos em execução no WebSphere Application Server Community Edition

- b. Implemente o módulo de serviço de dados REST: `wxsrestservice.war` para o servidor WebSphere Application Server Community Edition.
  - 1) Edite o arquivo XML de implementação `restservice_home/gettingstarted/wasce/geronimo-web.xml` de amostra para incluir dependências do caminho nos diretórios do caminho de classe de amostra de introdução:
    - Altere "classesDirs" para os dois GBeans do cliente de introdução:  
O caminho "classesDirs" para o GBean `GettingStarted_Client_SharedLib` deve ser configurado para: `restservice_home/gettingstarted/restclient/bin`  
O caminho "classesDirs" para o GBean `GettingStarted_Common_SharedLib` deve ser configurado para: `restservice_home/gettingstarted/common/bin`
  - 2) Abra o console administrativo WebSphere Application Server Community Edition e efetue login.
  - 3) Clique no link **Implementar Novo** no lado esquerdo da janela do console.
  - 4) Na página **Instalar Novos Aplicativos**, insira os seguintes valores nas caixas de texto:

Tabela 27. Instalar Novos Aplicativos

| Caixa de texto | Valor                                                               |
|----------------|---------------------------------------------------------------------|
| Archive        | <code>restservice_home/lib/wxsrestservice.war</code>                |
| Plano          | <code>restservice_home/gettingstarted/wasce/geronimo-web.xml</code> |

- 5) Clique no botão **Instalar**.  
A página do console indica que o aplicativo foi instalado e iniciado com sucesso.
  - 6) Examine o log de saída do sistema WebSphere Application Server Community Edition para verificar se o serviço de dados REST foi iniciado com êxito verificando se a seguinte mensagem está presente:  
`CW0BJ4000I: O serviço de dados REST do WebSphere eXtreme Scale foi iniciado.`
8. Verifique se o serviço de dados REST está funcionando:  
Abra um navegador da Web e navegue para a seguinte URL:  
`http://<host>:<port>/<context root>/restservice/<Grid Name>`  
A porta padrão para WebSphere Application Server Community Edition é 8080 e é definida usando a propriedade "HTTPPort" no arquivo `/var/config/config-substitutions.properties`.  
Por exemplo: `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/`

## Resultados

O documento de serviço AtomPub é exibido.

## Iniciando o Serviço de Dados REST no WebSphere Application Server Community Edition:

Este tópico descreve como configurar e iniciar o serviço de dados REST do eXtreme Scale usando o WebSphere Application Server Community Edition.

### Antes de Iniciar

Verifique se a grade de dados de amostra foi iniciada. Consulte “Ativando o Serviço de Dados REST” na página 352 para obter detalhes sobre como iniciar a grade.

### Procedimento

1. Faça download e instale o WebSphere Application Server Community Edition Versão 2.1.1.3 ou posterior para `wasce_root`, como: `/opt/IBM/wasce`
2. Inicie o servidor WebSphere Application Server Community Edition executando o seguinte comando:
  - **Linux** **UNIX** `wasce_root/bin/startup.sh`
  - **Windows** `wasce_root/bin/startup.bat`
3. Se a grade do eXtreme Scale tiver sido iniciada com a segurança do eXtreme Scale ativada, configure as seguintes propriedades no arquivo `restservice_home/gettingstarted/restclient/bin/wxsRestService.properties`.

```
ogClientPropertyFile=restservice_home/gettingstarted/security/security.ogclient.properties  
loginType=none
```

4. Instale o serviço de dados REST do eXtreme Scale e a amostra fornecida no servidor WebSphere Application Server Community Edition:
  - a. Inclua o JAR de tempo de execução do cliente ObjectGrid ao repositório WebSphere Application Server Community Edition:
    - 1) Abra o console administrativo WebSphere Application Server Community Edition e efetue login.

**Dica:** A URL padrão é: `http://localhost:8080/console`. O ID de usuário padrão é `system` e a senha é `manager`.
    - 2) Clique no **Repositório**, na pasta **Serviços**.
    - 3) Na seção **Incluir Archive no Repositório**, preencha o seguinte nas caixas de texto de entrada:

Tabela 28. Archive para Repositório

| Caixa de texto | Valor                                  |
|----------------|----------------------------------------|
| File           | <code>wxs_home/lib/ogclient.jar</code> |
| Grupo          | <code>com.ibm.websphere.xs</code>      |
| Artefato       | <code>ogclient</code>                  |
| Versão         | <code>7.0</code>                       |
| Tipo           | <code>jar</code>                       |

- 4) Clique no botão **Instalar**.

**Dica:** Consulte a seguinte nota técnica para obter detalhes sobre os diferentes métodos de dependências de classe e de biblioteca de

configuração: Especificando dependências externas para aplicativos em execução no WebSphere Application Server Community Edition.

- b. Implemente o módulo do serviço de dados REST, que é o arquivo `wxsrestservice.war`, para o servidor WebSphere Application Server Community Edition.

- 1) Edite o arquivo XML de implementação `restservice_home/gettingstarted/wasce/geronimo-web.xml` de amostra para incluir dependências do caminho nos diretórios do caminho de classe de amostra de introdução:

Altere os caminhos `classesDirs` para os dois GBeans do cliente de introdução:

- O caminho "classesDirs" para o GBean `GettingStarted_Client_SharedLib` deve ser configurado para: `restservice_home/gettingstarted/restclient/bin`
- O caminho "classesDirs" para o GBean `GettingStarted_Common_SharedLib` deve ser configurado para: `restservice_home/gettingstarted/common/bin`

- 2) Abra o console administrativo do WebSphere Application Server Community Edition e efetue login.

**Dica:** A URL padrão é: `http://localhost:8080/console`. O ID de usuário padrão é `system` e a senha é `manager`.

- 3) Clique em **Implementar Novo**.
- 4) Na página **Instalar Novos Aplicativos**, insira os seguintes valores nas caixas de texto:

Tabela 29. Valores de Instalação

| Caixa de texto | Valor                                                               |
|----------------|---------------------------------------------------------------------|
| Archive        | <code>restservice_home/lib/wxsrestservice.war</code>                |
| Plano          | <code>restservice_home/gettingstarted/wasce/geronimo-web.xml</code> |

- 5) Clique no botão Instalar.

A página do console deve indicar que o aplicativo foi instalado e iniciado com sucesso.

- 6) Examine o log de saída do sistema ou o console do WebSphere Application Server Community Edition para verificar se o serviço de dados REST foi iniciado com êxito verificando se a seguinte mensagem está presente:

`CWOBJ4000I: O serviço de dados REST do WebSphere eXtreme Scale foi iniciado.`

5. Verifique se o serviço de dados REST está funcionando:

- a. Abra o seguinte link em uma janela do navegador: `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid`. O documento de serviço para a grade `NorthwindGrid` é exibido.
- b. Abra o seguinte link em uma janela do navegador: `http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/$metadata`. O documento `Entity Model Data Extensions (EDMX)` é exibido.

6. Para parar os processos da grade, use `CTRL+C` na respectiva janela de comando para parar o processo.



7. Para parar o WebSphere Application Server Community Edition, use o seguinte comando:

- `UNIX` `Linux` `wasce_root/bin/shutdown.sh`
- `Windows` `wasce_root\bin\shutdown.bat`

**Dica:** O ID de usuário padrão é `system` e a senha é `manager`. Se você estiver usando uma porta customizada, use a opção `-port`.

## Implementando o Serviço de Dados REST no Apache Tomcat

Este tópico descreve como configurar o serviço de dados REST do WebSphere eXtreme Scale no Apache Tomcat Versão 5.5 ou posterior.

### Sobre Esta Tarefa

- Um IBM ou Sun JRE ou JDK, Versão 5 ou posterior está instalado e uma variável de ambiente `JAVA_HOME` especificada.
- Apache Tomcat Versão 5.5 ou posterior está instalado. Consulte Apache Tomcat para obter detalhes sobre como instalar o Tomcat.
- Uma instalação independente do WebSphere eXtreme Scale.

### Procedimento

1. Se estiver utilizando Sun JRE ou JDK, instale IBM ORB no Tomcat:

a. Tomcat versão 5.5:

Copie todos os arquivos JAR de:

o diretório `wxs_home/lib/endorsed`

para:

o diretório `tomcat_root/common/endorsed`

b. Tomcat versão 6.0:

Crie um diretório "endorsed":

```
UNIX Linux mkdir tomcat_root/endorsed
```

```
Windows md tomcat_root/endorsed
```

Copie todos os arquivos JAR de:

`wxs_home/lib/endorsed`

para:

`tomcat_root/common/endorsed`

2. Configure e inicie uma grade de dados.

a. Para obter detalhes sobre como configurar uma grade de dados para usar com o serviço de dados REST, consulte Capítulo 6, "Configurando", na página 219.

b. Verifique se um cliente do eXtreme Scale pode conectar e acessar entidades na grade. Para obter um exemplo, consulte "Configurando Serviços de Dados REST" na página 350.

3. Construa a JAR da configuração do serviço eXtreme Scale REST ou o diretório. Consulte as informações de empacotamento e implementação no "Instalando o Serviço de Dados REST" na página 198 para obter detalhes.

4. Implemente o módulo de serviço de dados REST: `wxsrestservice.war` para o servidor Tomcat.

Copie o arquivo `wxsrestservice.war` a partir de:

`restservice_home/lib`

para:

`tomcat_root/webapps`

5. Inclua o JAR de tempo de execução do cliente ObjectGrid e o JAR do aplicativo no caminho de classe compartilhado no Tomcat:
  - a. Edite o arquivo `tomcat_root/conf/catalina.properties`
  - b. Anexe os seguintes nomes de caminhos no final da propriedade `shared.loader`, separando cada nome de caminho por uma vírgula:
    - `wxs_home/lib/ogclient.jar`
    - `restservice_home/gettingstarted/restclient/bin`
    - `restservice_home/gettingstarted/common/bin`
6. Se você estiver usando a segurança Java 2, inclua permissões de segurança para o arquivo de políticas tomcat:
  - Se estiver utilizando Tomcat versão 5.5:

Mesclre o conteúdo do arquivo 5.5 `catalina.policy` de amostra localizado em `restservice_home/gettingstarted/tomcat/catalina-5_5.policy` com o arquivo `tomcat_root/conf/catalina.policy`.
  - Se estiver utilizando Tomcat versão 6.0:

Mesclre o conteúdo do arquivo 6.0 `catalina.policy` de amostra localizado em `restservice_home/gettingstarted/tomcat/catalina-6_0.policy` com o arquivo `tomcat_root/conf/catalina.policy`.
7. Inicie o servidor Tomcat:
  - **Se estiver usando Tomcat 5.5 no UNIX ou Windows ou a distribuição ZIP de Tomcat 6.0:**
    - a. `cd tomcat_root/bin`
    - b. Inicie o servidor:
      - Sem a segurança Java 2 ativada:

```
UNIX Linux ./catalina.sh run
Windows catalina.bat run
```
      - Com a segurança Java 2 ativada:

```
UNIX Linux ./catalina.sh run -security
Windows catalina.bat run -security
```
    - c. Os logs do Apache Tomcat são exibidos no console. Quando o serviço de dados REST for iniciado com êxito, a seguinte mensagem será exibida no console administrativo:  
CWOBJ4000I: O serviço de dados REST do WebSphere eXtreme Scale foi iniciado.
  - **Se estiver usando Tomcat 6.0 no Windows usando a distribuição do Windows Installer:**
    - a. `cd /bin`
    - b. Inicie a ferramenta de configuração do Apache Tomcat 6:  
`tomcat6w.exe`
    - c. Para ativar a segurança Java 2 (opcional):

Inclua as seguintes entradas em Opções Java na guia Java na janela de propriedades do Apache Tomcat 6:

```
-Djava.security.manager
-Djava.security.policy=\conf\catalina.policy
```

- d. Clique no botão Iniciar na janela de propriedades do Apache Tomcat 6 para iniciar o servidor Tomcat.
  - e. Revise os seguintes logs para verificar se o servidor Tomcat foi iniciado com sucesso:
    - `tomcat_root/bin/catalina.log`  
Exibe o status do mecanismo do servidor Tomcat
    - `tomcat_root/bin/stdout.log`  
Exibe o log de saída do sistema
  - f. Quando o serviço de dados REST for iniciado com sucesso, a seguinte mensagem será exibida no log de saída do sistema:  
CW0BJ4000I: O serviço de dados REST do WebSphere eXtreme Scale foi iniciado.
8. Verifique se o serviço de dados REST está funcionando.  
Abra um navegador da Web e navegue para a seguinte URL:  
`http://host:port/context_root/restservice/grid_name`  
A porta padrão para o Tomcat é 8080 e está configurada no arquivo `tomcat_root/conf/server.xml` no elemento `<Connector>`.  
Por exemplo:  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/`

## Resultados

O documento de serviço AtomPub é exibido.



### Iniciando Serviços de Dados REST no Apache Tomcat:

Este tópico descreve como configurar e iniciar o serviço de dados REST do eXtreme Scale usando o Apache Tomcat, versão 5.5 ou posterior.

#### Antes de Iniciar

Verifique se a grade de dados do eXtreme Scale de amostra foi iniciada. Consulte o “Ativando o Serviço de Dados REST” na página 352 para obter detalhes sobre como iniciar a grade de dados.

#### Procedimento

1. Faça download e instale o Apache Tomcat Versão 5.5 ou posterior em `tomcat_root`. Por exemplo: `/opt/tomcat`
2. Instale o serviço de dados REST do eXtreme Scale e a amostra fornecida no servidor Tomcat da seguinte maneira:
  - a. Se estiver usando um Sun JRE ou JDK, você deve instalar o IBM ORB no Tomcat:
    - Para Tomcat versão 5.5  
Copie todos os arquivos JAR de:  
`wxs_home/lib/endorsed`  
para  
`tomcat_root/common/endorsed`
    - Para Tomcat versão 6.0
      - 1) Crie um diretório "endorsed"
        -   `mkdir tomcat_root/endorsed`

- **Windows** md tomcat\_root/endorsed
- 2) Copie todos os arquivos JAR de:
  - wxs\_home/lib/endorsed
  - para
  - tomcat\_root/endorsed
- b. Implemente o módulo de serviço de dados REST: wxsrestservice.war para o servidor Tomcat.
  - Copie o arquivo wxsrestservice.war de:
    - restservice\_home/lib
    - para:
    - tomcat\_root/webapps
- c. Inclua o JAR de tempo de execução do cliente ObjectGrid e o JAR do aplicativo no caminho de classe compartilhado no Tomcat:
  - 1) Edite o arquivo tomcat\_root/conf/catalina.properties
  - 2) Anexe os seguintes nomes de caminho no final da propriedade shared.loader no formato de lista separada por vírgula:
    - wxs\_home/lib/ogclient.jar
    - restservice\_home/gettingstarted/restclient/bin
    - restservice\_home/gettingstarted/common/bin

**Importante:** O separador de caminho deve ser uma **barra**.

- 3. Se a grade de dados do eXtreme Scale foi iniciada com a segurança do eXtreme Scale ativada, configure as seguintes propriedades no arquivo restservice\_home/gettingstarted/restclient/bin/wxsRestService.properties.

```
ogClientPropertyFile=restservice_home/gettingstarted/security/security.ogclient.properties
loginType=none
```

- 4. Inicie o servidor Tomcat com o serviço de dados REST:
  - Se estiver utilizando Tomcat 5.5 no UNIX ou Windows, ou Tomcat 6.0 no UNIX:
    - a. cd tomcat\_root/bin
    - b. Inicie o servidor:
      - **UNIX** **Linux** ./catalina.sh run
      - **Windows** catalina.bat run
    - c. O console exibirá logs do Apache Tomcat. Quando o serviço de dados REST for iniciado com sucesso, a seguinte mensagem será exibida no console administrativo:
 

```
CWOBJ4000I: O serviço de dados REST do WebSphere eXtreme Scale foi iniciado.
```
  - Se estiver utilizando Tomcat 6.0 no Windows:
    - a. cd tomcat\_root/bin
    - b. Inicie a ferramenta de configuração do Apache Tomcat 6 com o seguinte comando: tomcat6w.exe
    - c. Clique no botão Iniciar na janela de propriedades do Apache Tomcat 6 para iniciar o servidor Tomcat.
    - d. Revise os seguintes logs para verificar se o servidor Tomcat foi iniciado com sucesso:
      - tomcat\_root/bin/catalina.log

- Exibe o status do mecanismo do servidor Tomcat
- tomcat\_root/bin/stdout.log
- Exibe o log de saída do sistema.
- e. Quando o serviço de dados REST for iniciado com sucesso, a seguinte mensagem será exibida no log de saída do sistema: CW0BJ4000I: O serviço de dados REST do WebSphere eXtreme Scale foi iniciado.
5. Verifique se o serviço de dados REST está funcionando:
    - a. Abra um navegador e navegue para:  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid`  
O documento de serviço para NorthwindGrid é exibido.
    - b. Navegue para:  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/$metadata`  
O documento Entity Model Data Extensions (EDMX) é exibido.
  6. Para parar os processos da grade de dados, use CTRL+C na respectiva janela de comando.
  7. Para parar o Tomcat, utilize CTRL +C na janela onde o iniciou.

## Configurando Navegadores da Web para Acessar os feeds ATOM do Serviço de Dados REST

O serviço de dados REST do eXtreme Scale cria feeds ATOM por padrão ao usar um navegador da Web. O formato feed ATOM pode não ser compatível com navegadores antigos ou pode ser interpretado de modo que os dados não possam ser visualizados como XML. É possível configurar o Internet Explorer Versão 8 e o Firefox Versão 3 para exibir os feeds ATOM e o XML dentro do navegador.

### Sobre Esta Tarefa

O serviço de dados REST do eXtreme Scale cria feeds ATOM por padrão ao usar um navegador da Web. O formato feed ATOM pode não ser compatível com navegadores antigos ou pode ser interpretado de modo que os dados não possam ser visualizados como XML. Para os navegadores antigos, será solicitado que você salve os arquivos no disco. Após os arquivos serem transferidos por download, utilize seu leitor XML favorito para examinar os arquivos. O XML gerado não é formatado para ser exibido, portanto, tudo será impresso em uma linha. A maioria dos programas de leitura XML, como Eclipse, suporta reformatação de XML em um formato legível.

Para navegadores modernos, como Microsoft Internet Explorer Versão 8 e Firefox Versão 3, os arquivos XML ATOM podem ser exibidos nativamente no navegador. Os tópicos a seguir fornecem detalhes sobre como configurar o Internet Explorer Versão 8 e o Firefox Versão 3 para exibir os feeds ATOM e o XML dentro do navegador.

### Procedimento

#### Configure o Internet Explorer Versão 8

- Para permitir que o Internet Explorer leia os feeds ATOM que o serviço de dados REST gera, use as seguintes etapas:
  1. Clique em **Ferramentas > Opções da Internet**
  2. Selecione a guia **Conteúdo**

3. Clique no botão **Configurações** na seção **Feeds e Web Slices**
4. Desmarque a caixa: "Ativar visualização de leitura de feed"
5. Clique em **OK** para retornar ao navegador.
6. Reinicie o Internet Explorer.

### Configure o Firefox Versão 3

- O Firefox não exibe automaticamente páginas com tipo de conteúdo: application/atom+xml. Na primeira vez que uma página é exibida, o Firefox solicita o salvamento do arquivo. Para exibir a página, abra o próprio arquivo com o Firefox da seguinte forma:
  1. Na caixa de diálogo do seletor de aplicativo, selecione o botão de opções "Abrir com" e clique no botão **Navegar**.
  2. Navegue para o diretório de instalação do Firefox. Por exemplo: C:\Program Files\Mozilla Firefox
  3. Selecione firefox.exe e pressione o botão **OK**.
  4. Marque "Fazer isto automaticamente para arquivos como este..." .
  5. Clique no botão **OK**.
  6. Em seguida, o Firefox exibe a página XML ATOM em uma nova janela do navegador ou guia
- O Firefox renderiza automaticamente feeds ATOM em formato legível. Entretanto, os feeds que o serviço de dados REST cria incluem XML. O Firefox não pode exibir XML, a menos que você desative o renderizador de feed. Ao contrário do Internet Explorer, no Firefox, o plug-in de renderização de feed ATOM deve ser editado explicitamente. Para configurar o Firefox para ler ATOM feeds como arquivos XML, siga estas etapas:
  1. Abra o seguinte arquivo em um editor de texto: <firefoxInstallRoot>\components\FeeDConverter.js. No caminho, <firefoxInstallRoot> é o diretório-raiz onde o Firefox está instalado.  
Para sistemas operacionais Windows, o diretório padrão é: C:\Program Files\Mozilla Firefox.
  2. Procure o fragmento semelhante ao seguinte:
 

```
// show the feed page if it wasn't sniffed and we have a document,
// or we have a document, title, and link or id
if (result.doc && (!this._sniffed ||
    (result.doc.title && (result.doc.link ||
    result.doc.id)))) {
```
  3. Comente as duas linhas que começam com if e result colocando // (duas barras) na frente delas.
  4. Anexe a seguinte instrução ao fragmento: if(0) {.
  5. O texto resultante deve ser como o seguinte:
 

```
// show the feed page if it wasn't sniffed and we have a document,
// or we have a document, title, and link or id
//if (result.doc && (!this._sniffed ||
//    (result.doc.title && (result.doc.link ||
    result.doc.id)))) {
if(0) {
```
  6. Salve o arquivo.
  7. Reinicie o Firefox
  8. Agora o Firefox pode exibir automaticamente todos os feeds no navegador.
- Teste sua configuração tentando algumas URLs.

## Exemplo

Esta seção descreve algumas URLs de exemplo que podem ser usadas para visualizar os dados que foram incluídos pela amostra de introdução fornecida com o serviço de dados REST. Antes de usar as seguintes URLs, inclua o conjunto de dados padrão na grade de dados de amostra do eXtreme Scale utilizando o cliente Java de amostra ou o cliente Visual Studio WCF Data Services de amostra.

Os exemplos a seguir assumem que a porta seja 8080, o que pode variar. Consulte a seção para obter detalhes sobre como configurar o serviço de dados REST em servidores de aplicativos diferentes.

- Visualizar um único cliente com o ID de "ACME":  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')`
- Visualizar todas as ordens para o cliente "ACME":  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')/orders`
- Visualizar o cliente "ACME" e as ordens:  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer('ACME')?$expand=orders`
- Visualizar a ordem 1000 para o cliente "ACME":  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Order(orderId=1000,customer_customerId='ACME')`
- Visualizar a ordem 1000 para o cliente "ACME" e seu Cliente associado:  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Order(orderId=1000,customer_customerId='ACME')?$expand=customer`
- Visualizar a ordem 1000 para o cliente "ACME" e seu Cliente e OrderDetails associados:  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Order(orderId=1000,customer_customerId='ACME')?$expand=customer,orderDetails`
- Visualizar todas as ordens para o cliente "ACME" para o mês de outubro de 2009 (GMT):  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer(customerId='ACME')/orders?$filter=orderDate ge datetime'2009-10-01T00:00:00' and orderDate lt datetime'2009-11-01T00:00:00'`
- Visualizar todas as 3 primeiras ordens e orderDetails para o cliente "ACME" para o mês de outubro de 2009 (GMT):  
`http://localhost:8080/wxsrestservice/restservice/NorthwindGrid/Customer(customerId='ACME')/orders?$filter=orderDate ge datetime'2009-10-01T00:00:00' and orderDate lt datetime'2009-11-01T00:00:00' &&$orderby=orderDate&&$top=3&&$expand=orderDetails`

## Utilizando um Cliente Java com Serviço de Dados REST

O aplicativo cliente Java usa a API EntityManager do eXtreme Scale para inserir dados na grade.

### Sobre Esta Tarefa

As seções anteriores descreveram como criar uma grade de dados do eXtreme Scale e configurar e iniciar o serviço de dados REST do eXtreme Scale. O aplicativo cliente Java usa a API EntityManager do eXtreme Scale para inserir dados na grade. Ele não demonstra como utilizar interfaces REST. O objetivo desse cliente é demonstrar como a API EntityManager é usada para interagir com a grade de dados do eXtreme Scale e permitir a modificação de dados na grade. Para visualizar dados na grade usando o serviço de dados REST, use um navegador da Web ou use o aplicativo cliente Visual Studio 2008.



## Procedimento

Para incluir rapidamente conteúdo na grade de dados do eXtreme Scale, execute o seguinte comando:

1. Abra uma janela do terminal ou da linha de comandos e configure a variável de ambiente JAVA\_HOME:

- `Linux` `UNIX` `export JAVA_HOME=java_home`
- `Windows` `set JAVA_HOME=java_home`

2. `cd restservice_home/gettingstarted`

3. Insira alguns dados na grade. Os dados que são inseridos serão recuperados depois com o uso de um navegador da Web e do serviço de dados REST.

Se a grade de dados tiver sido iniciada *sem* a segurança do eXtreme Scale, use os comandos a seguir.

- `UNIX` `Linux` `./runclient.sh load default`
- `Windows` `runclient.bat load default`

Se a grade de dados tiver sido iniciada *com* a segurança do eXtreme Scale, use os comandos a seguir.

- `UNIX` `Linux` `./runclient_secure.sh load default`
- `Windows` `runclient_secure.bat load default`

Para um cliente Java, use a seguinte sintaxe de comando:

- `UNIX` `Linux` `runclient.sh comando`
- `Windows` `runclient.bat comando`

Os comandos a seguir estão disponíveis:

- `load default`

Carrega um conjunto predefinido de entidades Customer, Category e Product na grade de dados e cria um conjunto aleatório de Orders para cada cliente.

- `load category categoryId categoryName firstProductId num_products`

Cria uma Category de produto e um número fixo de entidades Product na grade de dados. O parâmetro `firstProductId` identifica o número do ID do primeiro produto, e a cada produto subsequente é designado o próximo id até o número especificado de produtos ser criado.

- `load customer companyCode contactNamecompanyName numOrders firstOrderIdshipCity maxItems discountPct`

Carrega um novo Customer na grade de dados e cria um conjunto fixo de entidades Order para qualquer produto aleatório atualmente carregado na grade. O número de Orders é determinado pela configuração do parâmetro `<numOrders>`. Cada Order terá um número aleatório de entidades OrderDetail até `<maxItems>`

- `display customer companyCode`

Exibe uma entidade Customer e as entidades Order e OrderDetail associadas.

- `display category categoryId`

Exibe uma entidade Category do produto e as entidades Product associadas.

## Resultados

- `runclient.bat load default`
- `runclient.bat load customer IBM "John Doe" "IBM Corporation" 5 5000 Rochester 5 0.05`

- runclient.bat load category 5 "Household Items" 100 5
- runclient.bat display customer IBM
- runclient.bat display category 5

## Executando e Construindo a Grade de Dados de Amostra e o Cliente Java com Eclipse

A amostra de introdução do serviço de dados REST pode ser atualizada e aprimorada com uso do Eclipse. Para obter detalhes sobre como configurar seu ambiente Eclipse, consulte o documento de texto: `restservice_home/gettingstarted/ECLIPSE_README.txt`.

Depois que o projeto `WXSRestGettingStarted` for importado para o Eclipse e estiver sendo construído com êxito, a amostra automaticamente recompilará e os arquivos de script usados para iniciar o servidor e o cliente de contêiner automaticamente selecionarão os arquivos de classe e os arquivos XML. O serviço de dados REST detectará automaticamente quaisquer mudanças, já que o servidor da Web está configurado para ler os diretórios de compilação do Eclipse automaticamente.

**Importante:** Ao alterar arquivos de origem ou de configuração, o servidor de contêiner do eXtreme Scale e o aplicativo do serviço de dados REST devem ser reiniciados. O servidor de contêiner do eXtreme Scale deve ser iniciado antes do aplicativo da Web do serviço de dados REST.

## Cliente Visual Studio 2008 WCF com Serviço de Dados REST

A amostra de introdução ao serviço de dados REST do eXtreme Scale inclui um cliente de Serviços de Dados WCF que pode interagir com o serviço de dados REST do eXtreme Scale. A amostra é gravada como um aplicativo de linha de comandos em C#.

### Requisitos de Software

O cliente da amostra C# do WCF Data Services requer o seguinte:

- Sistema Operacional
  - Microsoft Windows XP
  - Microsoft Windows Server 2003
  - Microsoft Windows Server 2008
  - Microsoft Windows Vista
- Microsoft Visual Studio 2008 com Service Pack 1

**Dica:** Consulte o link anterior para conhecer os requisitos adicionais de hardware e software.

- Microsoft .NET Framework 3.5 Service Pack 1
- Microsoft Support: Uma atualização para o .NET Framework 3.5 Service Pack 1 está disponível

### Construindo e Executando o Cliente de Introdução

O cliente de amostra do WCF Data Services inclui um projeto e uma solução do Visual Studio 2008 e o código de origem para a execução da amostra. A amostra deve ser carregada no Visual Studio 2008 e compilada em um programa executável Windows antes de poder ser executada. Para construir e executar a amostra,

consulte o documento de texto: `restservice_home/gettingstarted/VS2008_README.txt`.

## Sintaxe de Comando C# do Cliente WCF Data Services

**Windows** `WXSRESTGettingStarted.exe <service URL> <command>`

`<service URL>` é a URL do serviço de dados REST do eXtreme Scale configurada na seção.

### Os comandos a seguir estão disponíveis:

- `load default`  
Carrega um conjunto predefinido de entidades `Customer`, `Category` e `Product` na grade de dados e cria um conjunto aleatório de `Orders` para cada cliente.
- `load category <categoryId> <categoryName> <firstProductId> <numProducts>`  
Cria uma `Category` de produto e um número fixo de entidades `Product` na grade de dados. O parâmetro `firstProductId` identifica o número do ID do primeiro produto, e a cada produto subsequente é designado o próximo id até o número especificado de produtos ser criado.
- `load customer <companyId> <contactName> <companyName> <numOrders> <firstOrderId> <shipCity> <maxItems> <discountPct>`  
Carrega um novo `Customer` na grade de dados e cria um conjunto fixo de entidades `Order` para qualquer produto aleatório atualmente carregado na grade de dados. O número de `Orders` é determinado pela configuração do parâmetro `<numOrders>`. Cada `Order` terá um número aleatório de entidades `OrderDetail` até `<maxItems>`
- `display customer <companyId>`  
Exibe uma entidade `Customer` e as entidades `Order` e `OrderDetail` associadas.
- `display category <categoryId>`  
Exibe uma entidade `Category` do produto e as entidades `Product` associadas.
- `unload`  
Remove todas as entidades que foram carregadas utilizando o comando "default load".

Os exemplos a seguir ilustram vários comandos.

- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid load default`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid load customer`
- `IBM "John Doe" "IBM Corporation" 5 5000 Rochester 5 0.05`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid load category 5 "Household Items" 100 5`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid display customer IBM`
- `WXSRestGettingStarted.exe http://localhost:8080/wxsrestservice/restservice/NorthwindGrid display category 5`

## Configurando Servidores para OSGi

O WebSphere eXtreme Scale inclui um pacote configurável OSGi do servidor, permitindo iniciar e configurar servidores e contêineres dentro de uma estrutura OSGi. Os tópicos de configuração descrevem como usar o pacote configurável do servidor do eXtreme Scale, o serviço OSGi Blueprint e a configuração do eXtreme Scale para executar os servidores do eXtreme Scale em uma estrutura do Eclipse Equinox OSGi.

### Sobre Esta Tarefa

As tarefas a seguir são necessárias para iniciar um servidor do eXtreme Scale dentro do Eclipse Equinox:

### Procedimento

1. Crie um pacote configurável OSGi que irá armazenar os plug-ins do eXtreme Scale, expondo-os como serviços e atualize o arquivo XML do descritor do ObjectGrid para referenciar os serviços.
2. Configure o OSGi para iniciar um servidor de contêiner do eXtreme Scale.
3. Instale e inicie o pacote configurável do servidor do eXtreme Scale na estrutura do OSGi.
4. Instale e inicie o pacote configurável do OSGi que contém os plug-ins do eXtreme Scale.

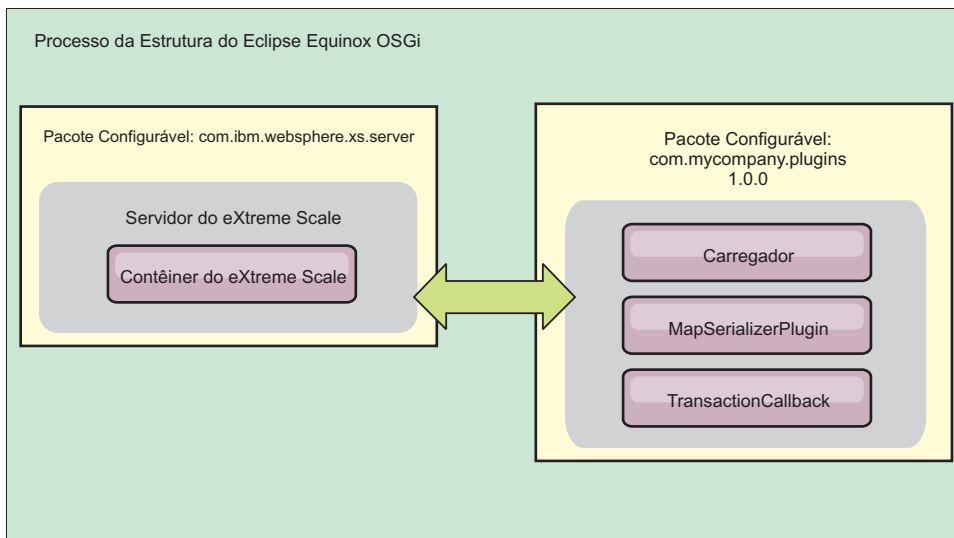


Figura 48. Processo do Eclipse Equinox para Instalar e Iniciar Pacotes Configuráveis OSGi com Plug-ins do eXtreme Scale

## Configurando os Plug-ins do eXtreme Scale com o OSGi Blueprint

Todos os plug-ins do ObjectGrid e do BackingMap do eXtreme Scale podem ser definidos como beans e serviços OSGi usando o Serviço OSGi Blueprint disponível com o Eclipse Gemini ou o Aries Apache.

## Antes de Iniciar

Antes de poder configurar seus plug-ins como serviços OSGi, você deve primeiro empacotar seus plug-ins em um pacote configurável OSGi e entender os princípios fundamentais dos plug-ins necessários. O pacote configurável deve importar os pacotes do cliente ou servidor do WebSphere eXtreme Scale e outros pacotes dependentes requeridos pelos plug-ins ou criar uma dependência do pacote configurável nos pacotes configuráveis do servidor ou cliente do eXtreme Scale. Este tópico descreve como configurar o XML Blueprint para criar beans de plug-in e expô-los como serviços OSGi para o eXtreme Scale usar.

## Sobre Esta Tarefa

Beans e serviços são definidos em um arquivo XML do Blueprint e o contêiner do Blueprint descobre, cria e liga os beans juntos e os expõe como serviços. O processo torna os beans disponíveis para outros pacotes configuráveis OSGi, incluindo os pacotes configuráveis de servidor e cliente do eXtreme Scale.

Ao criar serviços de plug-in customizados para uso com o eXtreme Scale, o pacote configurável que deve hospedar os plug-ins deve ser configurado para usar Blueprint. Além disso, um arquivo XML do Blueprint deve ser criado e armazenado dentro do pacote configurável. Leia sobre construção de aplicativos OSGi com a especificação do Contêiner do Blueprint para obter um entendimento geral da especificação.

## Procedimento

1. Crie um arquivo XML do Blueprint. É possível nomear o arquivo de qualquer jeito. No entanto, você deve incluir o namespace do projeto:

```
<?xml version="1.0" encoding="UTF-8"?>
<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0">
...
</blueprint>
```

2. Crie definições de bean no arquivo XML do Blueprint para cada plug-in do eXtreme Scale.

Beans são definidos usando o elemento `<bean>` e podem ser ligados a outras referências de bean e podem incluir parâmetros de inicialização.

**Importante:** Ao definir um bean, você deve usar o escopo correto. O Blueprint suporta os escopos singleton e de protótipo. O eXtreme Scale também suporta um escopo de shard customizado.

Defina a maioria dos plug-ins do eXtreme Scale como protótipo ou beans com escopo definido em shard, uma vez que todos os beans devem ser exclusivos para cada shard do ObjectGrid ou instância do BackingMap à qual eles estão associados. Beans com escopo definido em shard podem ser úteis ao usar osbeans em outros contextos para permitir a recuperação da instância correta.

Para definir um bean com escopo definido em protótipo, use o atributo `scope="prototype"` no bean:

```
<bean id="myPluginBean" class="com.mycompany.MyBean" scope="prototype">
...
</bean>
```

Para definir um bean com escopo definido em shard, você deve incluir o namespace `objectgrid` no esquema XML e usar o atributo `scope="objectgrid:shard"` no bean:

```

<?xml version="1.0" encoding="UTF-8"?>

<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
           xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"

           xsi:schemaLocation="http://www.ibm.com/schema/objectgrid
                               http://www.ibm.com/schema/objectgrid/objectgrid.xsd">

  <bean id="myPluginBean" class="com.mycompany.MyBean"
        scope="objectgrid:shard">
    ...
  </bean>
  ...

```

3. Crie definições de bean `PluginServiceFactory` para cada bean de plug-in. Todos os beans do eXtreme Scale devem ter um bean do `PluginServiceFactory` definido para que o escopo do bean correto possa ser aplicado. O eXtreme Scale inclui um `BlueprintServiceFactory` que você pode usar. Ele inclui duas propriedades que devem ser configuradas. Você deve configurar a propriedade `blueprintContainer` com a referência `blueprintContainer` e a propriedade `beanId` deve ser configurada com o nome do identificador de bean. Quando o eXtreme Scale consulta o serviço para instanciar os beans apropriados, o servidor procura a instância do componente do bean usando o contêiner `Blueprint`.

```

bean id="myPluginBeanFactory"
    class="com.ibm.websphere.objectgrid.plugins.osgi.BluePrintServiceFactory">
  <property name="blueprintContainer" ref="blueprintContainer"/>
  <property name="beanId" value="myPluginBean" />
</bean>

```

4. Crie um gerenciador de serviços para cada bean `PluginServiceFactory`. Cada gerenciador de serviços expõe o bean `PluginServiceFactory`, usando o elemento `<service>`. O elemento de serviço identifica o nome a ser exposto para o OSGi, a referência para o bean `PluginServiceFactory`, a interface a ser exposta e a classificação do serviço. O eXtreme Scale usa a classificação do gerenciador de serviços para executar upgrades de serviço quando a grade do eXtreme Scale está ativa. Se a classificação não for especificada, a estrutura do OSGi assumirá uma classificação igual a 0. Leia sobre como atualizar as classificações de serviço para obter mais informações.

`Blueprint` inclui várias opções para configurar gerenciadores de serviços. Para definir um gerenciador de serviços simples para um bean `PluginServiceFactory`, crie um elemento `<service>` para cada bean `PluginServiceFactory`:

```

<service ref="myPluginBeanFactory"
         interface="com.ibm.websphere.objectgrid.plugins.osgi.PluginServiceFactory"
         ranking="1">
</service>

```

5. Armazene o arquivo XML do `Blueprint` no pacote configurável de plug-ins. O arquivo XML do `Blueprint` deve ser armazenado no diretório `OSGI-INF/blueprint` para o contêiner do `Blueprint` a ser descoberto.

Para armazenar o arquivo XML do `Blueprint` em um diretório diferente, você deve especificar o seguinte cabeçalho de manifesto `Bundle-Blueprint`:

```
Bundle-Blueprint: OSGI-INF/blueprint.xml
```

## Resultados

Os plug-ins do eXtreme Scale agora são configurados para serem expostos em um contêiner OSGi `Blueprint`, além disso, o arquivo XML do descritor do `ObjectGrid` é configurado para referenciar os plug-ins usando o serviço OSGi `Blueprint`.

## Configurando Servidores com o OSGi Blueprint

É possível configurar os servidores de contêiner do WebSphere eXtreme Scale usando um arquivo XML do OSGi Blueprint, permitindo o empacotamento e o desenvolvimento simplificados de pacotes configuráveis do servidor autocontidos.

### Antes de Iniciar

Este tópico assume que as seguintes tarefas foram concluídas:

- A estrutura do Eclipse Equinox OSGi foi instalada e iniciada com o contêiner de projeto do Eclipse Gemini ou do Apache Aries.
- O pacote configurável do servidor eXtreme Scale foi instalado e iniciado.
- O pacote configurável de plug-ins dinâmicos do eXtreme Scale foi criado.
- O arquivo XML do descritor do ObjectGrid do eXtreme Scale e o arquivo XML da política de implementação foram criados.

### Sobre Esta Tarefa

Esta tarefa descreve como configurar um servidor do eXtreme Scale com um contêiner usando um arquivo XML do projeto. O resultado do procedimento é um pacote configurável do contêiner. Quando o pacote configurável do contêiner for iniciado, o pacote configurável do servidor eXtreme Scale controlará o pacote configurável, analisará o XML do servidor e iniciará um servidor e um contêiner.

Um pacote configurável do contêiner pode ser, opcionalmente, combinado com o aplicativo e os plug-ins do eXtreme Scale quando atualizações do plug-in dinâmico não são necessárias ou os plug-ins não suportam a atualização dinâmica.

### Procedimento

1. Crie um arquivo XML do Blueprint com o namespace `objectgrid` incluído. É possível nomear o arquivo de qualquer jeito. No entanto, ele deve incluir o namespace do projeto:

```
<?xml version="1.0" encoding="UTF-8"?>

<blueprint xmlns="http://www.osgi.org/xmlns/blueprint/v1.0.0"
           xmlns:objectgrid="http://www.ibm.com/schema/objectgrid"
           xsi:schemaLocation="http://www.ibm.com/schema/objectgrid
                               http://www.ibm.com/schema/objectgrid/objectgrid.xsd">
  ...
</blueprint>
```

2. Inclua a definição de XML para o servidor eXtreme Scale com as propriedades de servidor apropriadas. Consulte o arquivo XML do descritor do Spring para obter detalhes sobre todas as propriedades de configuração disponíveis. Consulte o exemplo a seguir da definição de XML:

```
objectgrid:server
  id="xsServer"
  tracespec="ObjectGridOSGi=all=enabled"
  tracefile="logs/osgi/wxserver/trace.log"
  jmxport="1199"
  listenerPort="2909">
  <objectgrid:catalog host="catserver1.mycompany.com" port="2809" />
  <objectgrid:catalog host="catserver2.mycompany.com" port="2809" />
</objectgrid:server>
```

3. Inclua a definição de XML para o contêiner do eXtreme Scale com a referência para a definição de servidor e os arquivos XML do descritor do ObjectGrid e de implementação do ObjectGrid integrados no pacote configurável; por exemplo:



```
<objectgrid:container id="container"
  objectgridxml="/META-INF/objectGrid.xml"
  deploymentxml="/META-INF/objectGridDeployment.xml"
  server="xsServer" />
```

4. Armazene o arquivo XML do Blueprint no pacote configurável do contêiner. O XML do Blueprint deve ser armazenado no diretório OSGI-INF/blueprint para que o contêiner do Blueprint seja localizado.

Para armazenar o XML do Blueprint em um diretório diferente, você deve especificar o cabeçalho de manifesto Bundle-Blueprint; por exemplo:

```
Bundle-Blueprint: OSGI-INF/blueprint.xml
```

5. Empacote os arquivos em um arquivo JAR do pacote configurável único. Consulte o exemplo a seguir de uma hierarquia do diretório do pacote configurável:

```
MyBundle.jar
  /META-INF/manifest.mf
  /META-INF/objectGrid.xml
  /META-INF/objectGridDeployment.xml
  /OSGI-INF/blueprint/blueprint.xml
```

## Resultados

Um pacote configurável do contêiner do eXtreme Scale agora está criado e pode ser instalado no Eclipse Equinox. Quando o pacote configurável do contêiner é iniciado, o ambiente de tempo de execução do servidor eXtreme Scale no pacote configurável do servidor eXtreme Scale irá iniciar automaticamente servidor do eXtreme Scale de singleton usando os parâmetros definidos no pacote configurável e um servidor de contêiner é iniciado. O pacote configurável pode ser interrompido e iniciado, o que resulta no contêiner parando e iniciando. O servidor é um singleton e não para quando o pacote configurável é iniciado pela primeira vez.

## Configurando Servidores com a Administração de Configuração do OSGI

É possível usar o serviço de administração de configuração do OSGi (admin config) para configurar os servidores de contêiner do WebSphere eXtreme Scale.

### Sobre Esta Tarefa

Para configurar um servidor, o identificador persistente (PID) do ManagedService, com.ibm.websphere.xs.server, é configurado para fazer referência ao arquivo de propriedades do servidor do ObjectGrid no sistema de arquivos. Para configurar um contêiner, o PID do ManagedServiceFactory, com.ibm.websphere.xs.container, é configurado para fazer referência ao arquivo XML de implementação do ObjectGrid e ao arquivo XML da política de implementação do ObjectGrid no sistema de arquivos.

Quando os dois PIDs estão configurados no serviço config admin, o serviço do servidor do eXtreme Scale inicializa automaticamente o servidor e inicia o contêiner com os arquivos de configuração especificados. Os PIDs de administração de configuração são persistidos para o diretório de configuração do OSGi. Se a configuração não for limpa, as configurações serão retidas entre reinicializações da estrutura.

Vários utilitários de terceiros existem para definir propriedades de administração de configuração. Os utilitários a seguir são exemplos de ferramentas que o produto suporta:

- O cliente da linha de comando Luminis OSGi Configuration Admin permite a configuração da linha de comandos.
- A Instalação do Arquivo Apache Felix permite especificar configurações do PID de administração de configuração nos arquivos de propriedades padrão.

Para configurar servidores de contêiner do eXtreme Scale com o cliente de linha de comandos de Administração de Configuração do OSGi para Luminis, conclua as etapas a seguir:

## Procedimento

1. Crie um PID de serviço gerenciado para o arquivo de propriedades do servidor ObjectGrid no console OSGi, executando os seguintes comandos:

```
osgi> cm create com.ibm.websphere.xs.server
osgi> cm put com.ibm.websphere.xs.server objectgrid.server.props /mypath/server.properties
```

2. Crie um PID do identificador de persistência de factory do serviço gerenciado para o contêiner do ObjectGrid no console do OSGi executando os comandos a seguir.

**Atenção:** Use o PID que é criado com o comando `config admin createf`. O PID que é usado no fragmento de código a seguir é apenas um exemplo.

```
osgi> cm createf com.ibm.websphere.xs.container
PID: com.ibm.websphere.xs.container-123456789-0
osgi> cm put com.ibm.websphere.xs.container-123456789-0 objectgridFile /mypath/objectGrid.xml
osgi> cm put com.ibm.websphere.xs.container-123456789-0 deploymentPolicyFile /mypath/deployment.xml
```

## Resultados

Servidores de contêiner do eXtreme Scale agora estão configurados para iniciar em uma estrutura do Eclipse Equinox OSGi.

## O que Fazer Depois

Os servidores de contêiner também podem ser criados programaticamente usando a API `ServerFactory` e ativadores de pacote configurável OSGi. Para obter detalhes sobre como usar a API `ServerFactory`, consulte a documentação da API.



---

## Capítulo 7. Administrando



Administrar e operar o ambiente do produto inclui iniciar e parar servidores, gerenciar a disponibilidade da grade de dados e recuperar cenários de falha do datacenter. Após configurar seus servidores de catálogos e servidores de contêiner, será possível iniciar e parar os servidores usando vários métodos. O método usado para iniciar e parar os servidores depende de se estiver usando uma topologia integrada, uma topologia independente ou uma topologia que está em execução no WebSphere Application Server.

---

### Iniciando e Parando Servidores Independentes

É possível iniciar e parar servidores de catálogos e de contêiner independentes com os scripts **start0gServer** e **stop0gServer** ou com a API do servidor integrado.

#### Antes de Iniciar

Se estiver iniciando ou parando servidores em um ambiente independente que usa um provedor de segurança do cliente externo, você deve configurar a variável de ambiente *CLIENT\_AUTH\_LIB* antes de executar os scripts **start0gServer** ou **stop0gServer**. Para obter mais informações sobre como configurar essa variável de ambiente, consulte “Iniciando Servidores Seguros em um Ambiente Independente” na página 520.

### Iniciando Servidores Independentes

Quando estiver executando uma configuração independente, o ambiente é composto por servidores de catálogos, servidores de contêineres e processos do cliente. Os servidores do WebSphere eXtreme Scale também podem ser integrados em aplicativos Java existentes usando a API do servidor integrado. É necessário configurar e iniciar estes processos manualmente.

#### Antes de Iniciar

É possível iniciar os servidores do WebSphere eXtreme Scale em um ambiente que não possua o WebSphere Application Server instalado. Se você estiver utilizando o WebSphere Application Server, consulte “Configurando o WebSphere eXtreme Scale com o WebSphere Application Server” na página 252.

### Iniciando um Serviço de Catálogo Independente

Você deve iniciar o serviço de catálogo manualmente quando estiver utilizando um ambiente do WebSphere eXtreme Scale distribuído que não esteja em execução no WebSphere Application Server.

#### Antes de Iniciar

- Se estiver usando o WebSphere Application Server, o serviço de catálogo iniciará automaticamente dentro dos processos existentes. Consulte Iniciando o Serviço de Catálogo no WebSphere Application Server para obter mais informações.

## Sobre Esta Tarefa

Inicie o serviço de catálogo com o script **startOgServer**. Ao chamar o comando iniciar, use o script **startOgServer.sh** em plataformas Unix ou **startOgServer.bat** no Windows.

O serviço de catálogo pode ser executado em um único processo ou pode incluir diversos servidores de catálogos para formar um domínio de serviço de catálogo. Um domínio do serviço de catálogo é necessário em um ambiente de produção para obter alta disponibilidade. Para obter mais informações sobre os domínios de serviço de catálogo, consulte as informações sobre os domínios de serviço de catálogo no *Visão Geral do Produto*. Também é possível especificar parâmetros adicionais para o script vincular o Object Request Broker (ORB) a um host e porta específicos, especificar o domínio ou ativar a segurança.

## Procedimento

- **Iniciar um processo de servidor de catálogo único**

Para iniciar um servidor de catálogo único, digite os seguintes comandos a partir da linha de comandos:

1. Navegue até o diretório bin.  

```
cd objectgridRoot/bin
```
2. Executar o comando **startOgServer**.  

```
startOgServer.bat|sh catalogServer
```

Para obter uma lista de todos os parâmetros de linha de comandos disponíveis, consulte “Script**startOgServer**” na página 393. Não utilize uma única Java Virtual Machine (JVM) para executar o serviço de catálogo em um ambiente de produção. Se o serviço de catálogo falhar, nenhum novo cliente poderá rotear para o eXtreme Scale implementado, e nenhuma nova instância ObjectGrid poderá ser incluída no domínio. Por esses motivos, você deve iniciar um conjunto de Java Virtual Machines para executar um domínio do serviço de catálogo.

- **Inicie um domínio de serviço de catálogo que consiste de diversos terminais.**

Para iniciar um conjunto de servidores para executar um serviço de catálogo, é necessário utilizar a opção **-catalogServiceEndpoints** no script **startOgServer**. Este argumento aceita uma lista de terminais de serviço de catálogo no formato de *serverName:hostname:clientPort:peerPort*. O exemplo a seguir mostra como iniciar a primeira das três Java Virtual Machines para hospedar um serviço de catálogo:

1. Navegue até o diretório bin.  

```
cd wxs_install_root/bin
```
2. Executar o comando **startOgServer**.  

```
startOgServer.bat|sh cs1 -catalogServiceEndpoints  
cs1:MyServer1.company.com:6601:6602,  
cs2:MyServer2.company.com:6601:6602,  
cs3:MyServer3.company.com:6601:6602
```

Neste exemplo, o servidor cs1 no host MyServer1.company.com é iniciado. Este nome do servidor é o primeiro argumento que é passado para o script. Durante a inicialização do servidor cs1, os parâmetros **catalogServiceEndpoints** são examinados para determinar quais portas estão alocadas para este processo. A lista também é utilizada para permitir que o servidor cs1 aceite conexões de outros servidores: cs2 e cs3.

3. Para iniciar os servidores de catálogos restantes na lista, transmita os argumentos a seguir para o script startOgServer. Iniciando o servidor cs2 no host MyServer2.company.com.

```
startOgServer.bat|sh cs2 -catalogServiceEndpoints  
cs1:MyServer1.company.com:6601:6602,  
cs2:MyServer2.company.com:6601:6602,  
cs3:MyServer3.company.com:6601:6602
```

Iniciando cs3 no MyServer3.company.com:

```
startOgServer.bat|sh cs3 -catalogServiceEndpoints  
cs1:MyServer1.company.com:6601:6602,  
cs2:MyServer2.company.com:6601:6602,  
cs3:MyServer3.company.com:6601:6602
```

**Importante: Inicie pelo menos dois servidores de catálogo simultaneamente.**

Você deve iniciar os servidores de catálogos que estão em uma grade de dados simultaneamente, porque cada servidor pausa para aguardar os outros servidores de catálogos se unirem ao grupo principal. Um servidor de catálogos que está configurado para uma grade de dados não inicia até identificar os demais membros no grupo. O servidor de catálogos eventualmente expira se outros servidores tornam-se disponíveis.

- **Ligue o ORB a um host e porta específicos.**

Não considerando as portas definidas no argumento **catalogServiceEndpoints**, cada serviço de catálogo também utiliza um Object Request Broker (ORB) para aceitar conexões de clientes e contêineres. Por padrão, o ORB atende na porta 2809 do host local. Se desejar vincular o ORB a um host específico em um JVM de serviço de catálogo, use os argumentos **-listenerHost** e **-listenerPort**. O exemplo a seguir mostra como iniciar um único servidor de catálogos JVM com seu ORB conectado à porta 7000 no MyServer1.company.com:

```
startOgServer.sh catalogServer -listenerHost MyServer1.company.com  
-listenerPort 7000
```

Cada contêiner e cliente do eXtreme Scale deve ser fornecidos com dados do terminal ORB do serviço de catálogo. Os clientes precisam apenas de um subconjunto destes dados, mas você deve utilizar pelo menos dois terminais para alta disponibilidade.

- **Opcional: Nome do domínio de serviço de catálogo**

Um nome de domínio do serviço de catálogo não é necessário quando iniciar um serviço de catálogo. No entanto, se você estiver usando replicação multimestre ou estiver usando diversos domínios de serviço de catálogo dentro do mesmo conjunto de processos, será necessário definir um nome exclusivo para o domínio de serviço de catálogo. O nome do domínio padrão é `DefaultDomain`. Para dar um nome ao seu domínio, use a opção **-domain**. O exemplo a seguir demonstra como iniciar uma única JVM de serviço de catálogo com o nome de domínio `myDomain`.

```
startOgServer.sh catalogServer -domain myDomain
```

Para obter mais informações sobre como configurar a replicação multimestre, consulte o “Configurando Diversas Topologias do Datacenter” na página 276.

- **Iniciar um serviço de catálogo seguro** Consulte o “Iniciando Servidores Seguros em um Ambiente Independente” na página 520 para obter informações adicionais.
- **Iniciar o serviço de catálogo programaticamente**

Qualquer configuração JVM que estiver sinalizada pelo método `CatalogServerProperties.setCatalogServer` pode hospedar o serviço de catálogo no eXtreme Scale. Este método indica ao tempo de execução do servidor eXtreme Scale para instanciar o serviço de catálogo quando o servidor é iniciado. O código a seguir mostra como instanciar o servidor de catálogos do eXtreme Scale:

```
CatalogServerProperties catalogServerProperties =
    ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

//The getInstance() method will start the catalog service.
Server server = ServerFactory.getInstance();
```

Para obter mais informações sobre como iniciar servidores programaticamente, consulte o “Usando a API do Servidor Integrado para Iniciar e Parar Servidores” na página 402.

## Iniciando Servidores de Contêiner

É possível iniciar servidores de contêiner a partir da linha de comandos utilizando uma topologia de implementação ou utilizando um arquivo `server.properties`.

### Sobre Esta Tarefa

Para iniciar um processo de contêiner, é necessário um arquivo XML ObjectGrid. O arquivo XML ObjectGrid especifica quais servidores eXtreme Scale o contêiner hospeda. Assegure-se de que o contêiner esteja equipado para hospedar cada ObjectGrid no XML transmitido para ele. Todas as classes que estes ObjectGrids requerem devem estar no caminho de classe para o contêiner. Para obter mais informações sobre o arquivo XML do ObjectGrid, consulte Arquivo `objectGrid.xsd`.

### Procedimento

- **Inicie o processo do contêiner a partir da linha de comandos.**

1. A partir da linha de comandos, navegue até o diretório `bin`:

```
cd wxs_install_root/bin
```

2. Execute o seguinte comando:

```
startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml
-catalogServiceEndpoints MyServer1.company.com:2809
```

**Importante:** No contêiner, a opção `-catalogServiceEndpoints` é utilizada para fazer referência ao host do Object Request Broker (ORB) e porta no serviço de catálogo. O serviço de catálogo utiliza as opções `-listenerHost` e `-listenerPort` para especificar o host e a porta para ligação ORB ou aceita a ligação padrão. Quando estiver iniciando um contêiner, use a opção `-catalogServiceEndpoints` para referenciar os valores que são transmitidos para as opções `-listenerHost` e `-listenerPort` no serviço de catálogo. Se as opções `-listenerHost` e `-listenerPort` não forem utilizadas quando o serviço de catálogo é iniciado, o ORB se conecta à porta 2809 no host local para o serviço de catálogo. Não use a opção `-catalogServiceEndpoints` para referenciar os hosts e portas que foram transmitidos para a opção `-catalogServiceEndpoints` no serviço de catálogo. No serviço de catálogo, a opção `-catalogServiceEndpoints` é usada para especificar portas necessárias para a configuração de servidor estático. Esse processo é identificado por `c0`, o primeiro argumento transmitido para o script. Utilize `companyGrid.xml` para iniciar o contêiner. Se o seu ORB do servidor de catálogos estiver em execução em um host diferente do seu contêiner ou estiver usando uma porta não-padrão, deverá usar o argumento



**-catalogServiceEndpoints** para se conectar com o ORB. Para este exemplo, assumo que um único serviço de catálogo está em execução na porta 2809 no MyServer1.company.com

- **Inicie o contêiner usando uma política de implementação.**

Embora não seja necessário, uma política de implementação é recomendada durante a inicialização do contêiner. A política de implementação é utilizada para configurar o particionamento e a replicação para o eXtreme Scale. A política de implementação também pode ser utilizada para influenciar o comportamento de disposição. Como o exemplo anterior não forneceu um arquivo de política de implementação, o exemplo recebe todos os valores-padrão com relação a replicação, particionamento e disposição. Portanto, os mapas no CompanyGrid estão em um mapSet. O mapSet não é particionado ou replicado. Para obter mais informações sobre os arquivos de política de implementação, consulte Arquivo Descritor XML de Política de Implementação. O exemplo a seguir utiliza o arquivo companyGridDpReplication.xml para iniciar uma JVM do contêiner, a JVM c0:

1. A partir da linha de comandos, navegue até o diretório bin:

```
cd wxs_install_root/bin
```

2. Execute o seguinte comando:

```
startOgServer.sh c0 -objectGridFile ../xml/companyGrid.xml  
-deploymentPolicyFile ../xml/companyGridDpReplication.xml  
-catalogServiceEndpoints MyServer1.company.com:2809
```

**Nota:** Se você tiver classes Java armazenadas em um diretório específico, em vez de alterar o script StartOgServer, é possível ativar o servidor com argumentos semelhantes aos seguintes: -jvmArgs -cp C:\ . . . \DirectoryPOJOs\POJOs.jar . No arquivo companyGridDpReplication.xml, um único conjunto de mapas contém todos os mapas. Esse mapSet é dividido em 10 partições. Cada partição possui uma réplica síncrona e nenhuma réplica assíncrona. Qualquer contêiner que utiliza a política de implementação companyGridDpReplication.xml em par com o arquivo XML do ObjectGrid companyGrid.xml também está apto a hospedar shards do CompanyGrid. Inicie outra JVM do contêiner, a JVM c1:

1. A partir da linha de comandos, navegue até o diretório bin:

```
cd wxs_install_root/bin
```

2. Execute o seguinte comando:

```
startOgServer.sh c1 -objectGridFile ../xml/companyGrid.xml  
-deploymentPolicyFile ../xml/companyGridDpReplication.xml  
-catalogServiceEndpoints MyServer1.company.com:2809
```

Cada política de implementação contém um ou mais elementos objectgridDeployment. Quando um contêiner é iniciado, ele publica sua política de implementação no serviço de catálogo. O serviço de catálogo examina cada elemento objectgridDeployment. Se o atributo objectgridName corresponde ao atributo objectgridName de um elemento objectgridDeployment anteriormente recebido, o elemento objectgridDeployment mais recente é ignorado. O primeiro elemento objectgridDeployment recebido para um atributo objectgridName específico é usado como o elemento principal. Por exemplo, assumo que a JVM c2 utiliza uma política de implementação que divide o mapSet em um número diferente de partições:

**companyGridDpReplicationModified.xml**

```
<?xml version="1.0" encoding="UTF-8"?>  
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy  
  ../deploymentPolicy.xsd"  
  xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
```

```

    <objectgridDeployment objectgridName="CompanyGrid">
      <mapSet name="mapSet1" numberOfPartitions="5"
        minSyncReplicas="1" maxSyncReplicas="1"
        maxAsyncReplicas="0">
        <map ref="Customer" />
        <map ref="Item" />
        <map ref="OrderLine" />
        <map ref="Order" />
      </mapSet>
    </objectgridDeployment>
  </deploymentPolicy>

```

Agora, é possível iniciar uma terceira JVM, a JVM c2:

1. A partir da linha de comandos, navegue até o diretório bin:

```
cd wxs_install_root/bin
```

2. Execute o seguinte comando:

```
startOgServer.sh c2 -objectGridFile ../xml/companyGrid.xml
-deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
-catalogServiceEndpoints MyServer1.company.com:2809
```

O contêiner na JVM c2 é iniciado com uma política de implementação que especifica 5 partições para mapSet1. No entanto, o serviço de catálogo já mantém a cópia principal do objectgridDeployment para o CompanyGrid. Quando a JVM c0 foi iniciada, ela especificou que 10 partições existem para esse mapSet. Como ela foi o primeiro contêiner a iniciar e publicar a política de implementação, sua política de implementação se tornou o principal. Portanto, qualquer valor de atributo objectgridDeployment que seja igual ao CompanyGrid em uma política de implementação subsequente será ignorado.

- **Inicie um contêiner utilizando um arquivo de propriedades do servidor.**

É possível usar um arquivo de propriedades de servidor para configurar o rastreamento e configurar a segurança em um contêiner. Execute os seguintes comandos para iniciar o contêiner c3 com um arquivo de propriedades do servidor:

1. A partir da linha de comandos, navegue até o diretório bin:

```
cd wxs_install_root/bin
```

2. Execute o seguinte comando:

```
startOgServer.sh c3 -objectGridFile ../xml/companyGrid.xml
-deploymentPolicyFile ../xml/companyGridDpReplicationModified.xml
-catalogServiceEndpoints MyServer1.company.com:2809
-serverProps ../serverProps/server.properties
```

A seguir há um exemplo do arquivo server.properties:

```

server.properties
workingDirectory=
traceSpec==all=disabled
systemStreamToFileEnabled=true
enableMBeans=true
memoryThresholdPercentage=50

```

Este é um arquivo de propriedades de servidor básico que não possui a segurança ativada. Para obter mais informações sobre o arquivo server.properties, consulte Arquivo de Propriedades do Servidor.

- **Inicie um servidor de contêiner programaticamente**

Para obter mais informações sobre como iniciar os servidores de contêiner programaticamente, consulte o “Usando a API do Servidor Integrado para Iniciar e Parar Servidores” na página 402.

## Script `startOgServer`

O script `startOgServer` inicia servidores de catálogos e de contêiner. É possível utilizar uma variedade de parâmetros quando você inicia seu servidores para ativar o rastreamento, especificar números de porta e assim por diante.

### Propósito

É possível usar o script `startOgServer` para iniciar servidores.

### Local

O script `startOgServer` está no diretório `bin` do diretório-raiz, por exemplo:

```
cd wxs_install_root/bin
```

**Nota:** Se você tiver classes Java armazenadas em um diretório específico, em vez de alterar o script `startOgServer`, é possível ativar o servidor com argumentos como os seguintes: `-jvmArgs -cp C:\ . . . \DirectoryPOJ0s\POJ0s.jar`

## Uso para Servidores de Catálogos

Para iniciar um servidor de catálogos:

Windows

```
startOgServer.bat <server> [options]
```

UNIX

```
startOgServer.sh <server>[options]
```

Para iniciar um servidor de catálogos configurado padrão, utilize os seguintes comandos:

Windows

```
startOgServer.bat catalogServer
```

UNIX

```
startOgServer.sh catalogServer
```

## Opções para Iniciar Servidores de Catálogos

Os seguintes parâmetros são opcionais.

**Parâmetros para iniciar um servidor de catálogos:**

**-catalogServiceEndpoints <serverName:hostName:clientPort:peerPort>**

No contêiner, referencia o host e a porta do Object Request Broker (ORB) no serviço de catálogo. Cada atributo é definido conforme a seguir:

**serverName**

Especifica o nome para identificar o processo que você está ativando.

**hostName**

Especifica o nome do host para o computador onde o servidor é ativado.

**clientPort**

Especifica a porta que é usada para comunicação de serviço de catálogo peer.

**peerPort**

Este valor é o mesmo que haManagerPort. Especifica a porta que é usada para comunicação de serviço de catálogo peer.

O exemplo a seguir inicia o servidor de catálogos cs1, que está no mesmo domínio de serviço de catálogo que os servidores cs2 e cs3:

```
startOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
```

**-clusterSecurityFile <arquivo xml de segurança do cluster>**

Especifica o arquivo objectGridSecurity.xml no disco rígido, que descreve as propriedades de segurança que são comuns a todos os servidores (incluindo servidores de catálogos e servidores de contêiner). Um dos exemplos de propriedade é a configuração do autenticador que representa o registro do usuário e o mecanismo de autenticação.

**Exemplo:**/opt/xs/ogsecurity.xml

**-clusterSecurityUrl <URL do xml de segurança do cluster>**

Especifica o arquivo objectGridSecurity.xml como uma URL no arquivo no disco rígido ou na rede, que descreve as propriedades de segurança que são comuns a todos os servidores (incluindo servidores de catálogos e servidores de contêiner). Um dos exemplos de propriedade é a configuração do autenticador que representa o registro do usuário e o mecanismo de autenticação.

**Exemplo:**file:///opt/xs/ogsecurity.xml

**-domain <nome de domínio>**

Especifica o nome do domínio de serviço de catálogo para este servidor de catálogos. O domínio de serviço de catálogo cria um grupo de servidores de catálogo altamente disponíveis. Cada servidor de catálogos para um único domínio deve especificar o mesmo valor para o parâmetro **-domain**.

**-JMXConnectorPort <port>**

Define a porta Secure Sockets Layer (SSL) à qual o serviço Java Management Extensions (JMX) se liga.

**-haManagerPort <porta>**

Sinônimo de porta de peer. Especifica o número da porta usada pelo gerenciador de alta disponibilidade. Se essa propriedade não for configurada, o serviço de catálogo gera uma porta disponível automaticamente. Essa propriedade se aplica ao servidor de contêiner e ao serviço de catálogo. (Necessário somente para ambientes do WebSphere Application Server.)

**-JMXServicePort <porta>**

Especifica o número da porta na qual o servidor MBean recebe a comunicação com Java Management Extensions (JMX). Você deve usar um número da porta diferente para cada JVM na configuração. Se desejar usar JMX/RMI, especifique explicitamente **JMXServicePort** e o número da porta, mesmo se desejar usar o valor de porta padrão. Esta propriedade se aplica ao servidor de contêiner e ao serviço de catálogo.

**Padrão:** 1099

**-jvmArgs <argumentos JVM>**

Especifica um conjunto de argumentos JVM. Cada opção após a opção **-jvmArgs** é usada para iniciar o servidor do Java virtual machine (JVM). Quando o parâmetro **-jvmArgs** é utilizado, certifique-se de que este seja o último argumento de script opcional especificado.

**Exemplo:** **-jvmArgs -Xms256M -Xmx1G**

**-listenerHost <nome do host>**

Especifica o nome do host ao qual o Object Request Broker (ORB) se conecta para comunicação com o Internet Inter-ORB Protocol (IIOP). O valor deve ser um nome de domínio ou endereço IP completo. Se sua configuração envolver diversas placas de rede, configure o host e a porta do listener para permitir que o Object Request Broker na JVM conheça o endereço IP para o qual se ligar. Se você não especificar qual endereço IP usar, sintomas tais como tempos limites de conexão, falhas de API incomuns e clientes que parecem ser interrompidos podem ocorrer. **Padrão:** localhost

**-listenerPort <porta>**

Especifica o número da porta ao qual o Object Request Broker (ORB) se conecta. Esta configuração define que contêineres e clientes se comunicam com o serviço de catálogo por meio do ORB. No WebSphere Application Server, a listenerPort é herdada pela configuração de porta BOOTSTRAP\_ADDRESS. Esta propriedade se aplica ao servidor de contêiner e ao serviço de catálogo. **Padrão:** 2809

**-quorum true|false**

Ativa o quorum para o servidor de catálogo. Consulte o Quorums de Servidores de Catálogo para obter informações adicionais.

**-script <arquivo de script>**

Especifica o local de um script customizado para os comandos que você especifica para iniciar servidores de catálogo ou contêineres e depois parametriza ou edita conforme necessário.

**-serverProps <arquivo de propriedades do servidor>**

Especifica o arquivo de propriedades do servidor que contém as propriedades de segurança específicas do servidor. O nome do arquivo especificado para esta propriedade está simplesmente no formato de caminho de arquivo simples, tal como c:/tmp/og/catalogserver.props.

**-traceSpec <especificação de rastreamento>**

Especifica uma cadeia que especifica o escopo do rastreamento que é ativado quando o servidor inicia.

**Exemplo:**

- ObjectGrid=all=enabled
- ObjectGrid\*=all=enabled

**-traceFile <arquivo de rastreamento>**

Especifica o caminho de um arquivo no qual salvar informações de rastreamento.

**Exemplo:** ../logs/c4Trace.log

**-timeout <segundos>**

Especifica um número de segundos antes do servidor expirar.

**Uso dos Servidores de Contêiner** Windows

```
startOgServer.bat <server> -objectgridFile <xml file>  
-deploymentPolicyFile <arquivo xml> [options]
```

#### Windows

```
startOgServer.bat <servidor> -objectgridUrl  
<URL xml> -deploymentPolicyUrl <URL xml> [options]
```

#### UNIX

```
startOgServer.sh <server> -objectgridFile <xml file>  
-deploymentPolicyFile <arquivo xml> [options]
```

#### UNIX

```
startOgServer.sh <servidor> -objectgridUrl <URL xml> -deploymentPolicyUrl  
<URL xml> [options]
```

## Opções para Servidores de Contêiner

### **-catalogServiceEndpoints<hostName:port,hostName:port>**

Especifica o host e a porta do Object Request Broker (ORB) no serviço de catálogo.

**Padrão:** localhost:2809

### **-deploymentPolicyFile <arquivo xml da política de implementação>**

Especifica o caminho para o arquivo de política de implementação no disco rígido. A política de implementação é usada para configurar o particionamento e a replicação. A política de implementação também pode ser utilizada para influenciar o comportamento de disposição.

**Exemplo:** ../xml/SimpleDP.xml

### **-deploymentPolicyUrl <deployment policy url>**

Especifica a URL para o arquivo de política de implementação no disco rígido ou na rede. A política de implementação é usada para configurar o particionamento e a replicação. A política de implementação também pode ser utilizada para influenciar o comportamento de disposição.

**Exemplo:** file://xml/SimpleDP.xml

### **-JMXConnectorPort <port>**

Define a porta Secure Sockets Layer (SSL) à qual o serviço Java Management Extensions (JMX) se conecta.

### **-JMXServicePort <porta>**

Especifica o número da porta na qual o servidor MBean recebe a comunicação com Java Management Extensions (JMX). Você deve usar um número da porta diferente para cada JVM na configuração. Se desejar usar JMX/RMI, especifique explicitamente **JMXServicePort** e o número da porta, mesmo se desejar usar o valor de porta padrão. Esta propriedade se aplica ao servidor de contêiner e ao serviço de catálogo.**Padrão:** 1099

### **-jvmArgs <argumentos JVM>**

Especifica um conjunto de argumentos JVM. Cada opção após a opção **-jvmArgs** é usada para iniciar o servidor do Java virtual machine (JVM). Quando o parâmetro **-jvmArgs** é utilizado, certifique-se de que este seja o último argumento de script opcional especificado.

**Exemplo:** **-jvmArgs** -Xms256M -Xmx1G

### **-listenerHost <nome do host>**

Especifica o nome do host ao qual o Object Request Broker (ORB) se conecta para comunicação com o Internet Inter-ORB Protocol (IIOP). O valor deve ser um nome de domínio ou endereço IP completo. Se sua configuração envolver

diversas placas de rede, configure o host e a porta do listener para permitir que o Object Request Broker na JVM conheça o endereço IP para o qual se ligar. Se você não especificar qual endereço IP usar, sintomas tais como tempos limites de conexão, falhas de API incomuns e clientes que parecem ser interrompidos podem ocorrer.**Padrão:** localhost

**-listenerPort <porta>**

Especifica o número da porta ao qual o Object Request Broker (ORB) se conecta. Esta configuração define que contêineres e clientes se comunicam com o serviço de catálogo por meio do ORB. No WebSphere Application Server, a listenerPort é herdada pela configuração de porta BOOTSTRAP\_ADDRESS. Esta propriedade se aplica ao servidor de contêiner e ao serviço de catálogo.**Padrão:** 2809

**-objectgridFile <ObjectGrid descriptor xml file>**

Especifica o caminho para o arquivo descritor ObjectGrid. O arquivo XML ObjectGrid especifica quais servidores eXtreme Scale o contêiner hospeda.

**-objectgridUrl <ObjectGrid descriptor url>**

Especifica uma URL para o arquivo descritor ObjectGrid. O arquivo XML ObjectGrid especifica quais servidores eXtreme Scale o contêiner hospeda.

**-script <arquivo de script>**

Especifica o local de um script customizado para os comandos que você especifica para iniciar servidores de catálogo ou contêineres e depois parametriza ou edita conforme necessário.

**-serverProps <arquivo de propriedades do servidor>**

Especifica o caminho para o arquivo de propriedades do servidor.

**Exemplo:** ../security/server.props

**-timeout <segundos>**

Especifica um número de segundos antes do servidor expirar.

**-traceFile <arquivo de rastreamento>**

Especifica o caminho de um arquivo no qual salvar informações de rastreamento.

**Exemplo:** ../logs/c4Trace.log

**-traceSpec <especificação de rastreamento>**

Especifica uma cadeia que especifica o escopo do rastreamento que é ativado quando o servidor inicia.

**Exemplo:**

- ObjectGrid=all=enabled
- ObjectGrid\*=all=enabled

**-zone <nome da zona>**

Especifica a zona a utilizar para todos os contêineres no servidor. Consulte o "Roteamento para Zonas Preferenciais" na página 239 informações sobre zonas no *Visão Geral do Produto* para obter mais informações sobre como configurar zonas.

## Parando Servidores Independentes

É possível usar o script stop0gServer para parar processos do servidor eXtreme Scale.



## Sobre Esta Tarefa

Execute o script **stop0gServer** ao navegar para o diretório bin:

```
cd wxs_install_root/bin
```

### Procedimento

- **Para um servidor de contêiner único.**

Execute o script **stop0gServer** para parar o servidor de contêiner. Use esse comando apenas quando parar um servidor de contêiner único. Se executar o comando para parar um servidor de catálogos único ou vários servidores de contêiner sucessivamente, poderão ocorrer problemas de desempenho ao desordenar o posicionamento dosshards.

```
stop0gServer containerServer -catalogServiceEndpoints MyServer1.company.com:2809
```

**Atenção:** A opção **-catalogServiceEndpoints** deve corresponder à opção **-catalogServiceEndpoints** usada para iniciar o contêiner. Se um **-catalogServiceEndpoints** não tiver sido usado para iniciar o contêiner, os valores padrão serão, provavelmente, localhost ou hostname e 2809 para a porta ORB para conectar ao serviço de catálogo. Caso contrário, use os valores transmitidos para **-listenerHost** e **-listenerPort** no serviço de catálogo. Se as opções **-listenerHost** e **-listenerPort** não são utilizadas ao iniciar o serviço de catálogo, o ORB conecta-se à porta 2809 no host local para o serviço de catálogo.

- **Parar diversos servidores de contêiner.**

Para evitar problemas de desordem e de desempenho de posicionamento de shard quando desejar parar diversos servidores de contêiner ao mesmo tempo, use o seguinte formato de comando. Separe uma lista de servidores de contêiner com vírgulas:

```
stop0gServer containerServer0,containerServer1,containerServer2  
-catalogServiceEndpoints MyServer1.company.com:2809
```

Se desejar parar todos os contêineres em uma zona ou host específico, será possível usar o parâmetro **-teardown**. Consulte o “Parando Servidores Corretamente com o Utilitário **xscmd**” na página 400 para obter informações adicionais.

- **Parar servidores de catálogos.**

Execute o script **stop0gServer** para parar o servidor de catálogos.

```
stop0gServer.sh catalogServer -catalogServiceEndpoints MyServer1.company.com:2809
```

**Atenção:** Quando estiver parando um serviço de catálogo, use a opção **-catalogServiceEndpoints** para referenciar o host e a porta do Object Request Broker (ORB) no serviço de catálogo. O serviço de catálogo utiliza as opções **-listenerHost** e **-listenerPort** para especificar o host e a porta para a ligação ORB ou aceita a ligação padrão. Se as opções **-listenerHost** e **-listenerPort** não são utilizadas ao iniciar o serviço de catálogo, o ORB conecta-se à porta 2809 no host local para o serviço de catálogo. A opção **-catalogServiceEndpoints** é diferente quando parar um serviço de catálogo de quando iniciar o serviço de catálogo.

Iniciar um serviço de catálogo requer portas de acesso peer e portas de acesso do cliente, se as portas padrão não tiverem sido usadas. Parar um serviço de catálogo requer apenas a porta ORB.

- **Parar o servidor do console da web.** Para parar o servidor do console da web, execute o script **stopConsoleServer.bat|sh**. Esse script está no diretório

`wxs_install_root/ObjectGrid/bin` de sua instalação. Para obter informações adicionais, consulte “Iniciando e Efetuando Login no Console da Web” na página 435.

- **Ativar o rastreo para o processo de interrupção do servidor.**

Se um contêiner falhar ao parar, é possível ativar o rastreo para ajudar com a depuração do problema. Para ativar o rastreo durante a interrupção de um servidor, inclua os parâmetros **-traceSpec** e **-traceFile** nos comandos de interrupção. O parâmetro **-traceSpec** especifica o tipo de rastreo a ser ativado e o parâmetro **-traceFile** especifica o caminho e o nome do arquivo a ser criado e utilizado para os dados de rastreo.

1. A partir da linha de comandos, navegue até o diretório `bin`.

```
cd wxs_install_root/bin
```

2. Execute o script **stopOgServer** com o rastreo ativado.

```
stopOgServer.sh c4 -catalogServiceEndpoints MyServer1.company.com:2809  
-traceFile ../logs/c4Trace.log -traceSpec ObjectGrid=all=enabled
```

Após o rastreo ser obtido, consulte os erros relacionados aos conflitos de porta, às classes ausente, aos arquivos XML ausentes ou incorretos ou a quaisquer rastreios de pilha. As especificações de rastreo de inicialização sugeridas são:

- ObjectGrid=all=enabled
- ObjectGrid\*=all=enabled

Para todas as opções de especificação de rastreo, consulte “Opções de Rastreo” na página 527.

- **Parar servidores de catálogos integrados programaticamente.**

Para obter mais informações sobre como parar servidores integrados programaticamente, consulte o “Usando a API do Servidor Integrado para Iniciar e Parar Servidores” na página 402.

## Script stopOgServer

O script **stopOgServer** para servidores de catálogos e de contêiner.

### Propósito

Execute o script **stopOgServer** para parar um servidor. Você deve fornecer o nome do servidor e seus terminais de serviço de catálogo.

### Local

O script **stopOgServer** está no diretório `bin` do diretório `root`, por exemplo:

```
cd wxs_install_root/bin
```

### Uso

**Para parar um servidor de catálogos ou de contêiner:** Windows

```
stopOgServer.bat <server_name> -catalogServiceEndpoints  
<csHost:csListenerPort,csHost:csListenerPort> [options]
```

UNIX

```
stopOgServer.sh <server_name> -catalogServiceEndpoints  
<csHost:csListenerPort,csHost:csListenerPort> [options]
```

## Opções

### **-catalogServiceEndpoints <csHost:csListenerPort, csHost:csListenerPort...>**

Especifica o hoste o número da porta do Object Request Broker (ORB).

**Para servidores de contêiner:** A lista de terminais de serviço de catálogo deve ser a mesma lista que foi usada para iniciar o servidor de contêiner. Se esta opção não foi especificada ao iniciar o servidor de contêiner, use o valor padrão de localhost:2809.

**Para servidores de catálogos:** Se você estiver parando o serviço de catálogo, use os valores indicados para as opções **-listenerHost** e **-listenerPort** quando você iniciou o serviço de catálogo. Se essas opções não foram especificadas ao iniciar o servidor de catálogos, use o valor padrão de localhost:2809. O valor **-catalogServiceEndpoints** usado ao parar o serviço de catálogo é diferente do valor usado ao iniciar o serviço de catálogo.

### **-clientSecurityFile <arquivo de propriedades de segurança>**

Especifica o caminho para o arquivo de propriedades do cliente que define as propriedades de segurança para o cliente. Consulte Arquivo de Propriedades do Cliente para obter mais informações sobre as configurações de segurança neste arquivo.

### **-traceSpec <especificação de rastreo>**

Especifica uma cadeia que especifica o escopo do rastreo que é ativado quando o servidor inicia.

#### **Exemplo:**

- ObjectGrid=all=enabled
- ObjectGrid\*=all=enabled

### **-traceFile <arquivo de rastreo>**

Especifica o caminho de um arquivo no qual salvar informações de rastreo.

**Exemplo:** ../logs/c4Trace.log

### **-jvmArgs <argumentos JVM>**

Especifica um conjunto de argumentos JVM. Cada opção após a opção **-jvmArgs** é usada para iniciar o servidor do Java virtual machine (JVM). Quando o parâmetro **-jvmArgs** é utilizado, certifique-se de que este seja o último argumento de script opcional especificado.

**Exemplo:**-jvmArgs -Xms256M -Xmx1G

## **Parando Servidores Corretamente com o Utilitário xscmd**

É possível usar o utilitário **xscmd** com o comando **-c teardown** para parar uma lista ou grupo de servidores de contêiner e de catálogos. Este comando simplifica o encerramento de toda a, ou parte de uma, grade de dados, evitando ações de posicionamento e de recuperação desnecessárias pelo serviço de catálogo que normalmente ocorrem quando os processos estão parados ou interrompidos.

## **Procedimento**

- Pare uma lista de servidores específica.  
Forneça uma lista de servidores após o parâmetro **-teardown:**  
xscmd -c teardown
- Pare todos os servidores em uma zona específica.

Use o parâmetro **-z** e forneça o nome da zona. O servidor de catálogos determina os servidores que estão em execução na zona e o utilitário **xscmd** solicita uma lista dos servidores na zona selecionada antes de encerrar os servidores:

```
xscmd -c teardown -z zone_name
```

- Pare todos os servidores em um host específico.

Use o parâmetro **-hf** e forneça o nome do host. Por exemplo, para encerrar todos os servidores em `myhost.mycompany.com`, digite `-hf myhost.mycompany.com`. O servidor de catálogos determina os servidores que estão em execução no host e o utilitário **xscmd** solicita uma lista dos servidores no host selecionado antes de encerrar os servidores:

```
xscmd -teardown -hf <host_name>
```

---

## Iniciando e Parando Servidores em um Ambiente do WebSphere Application Server

Os servidores de catálogos e de contêiner podem ser iniciados automaticamente em um ambiente do WebSphere Application Server ou do WebSphere Application Server Network Deployment.

### Antes de Iniciar

Configure os servidores de catálogos e os servidores de contêiner para serem executados no WebSphere Application Server:

- “Configurando o Serviço de Catálogo no WebSphere Application Server” na página 252
- “Configurando Servidores de Contêiner no WebSphere Application Server” na página 270

### Sobre Esta Tarefa

O ciclo de vida dos servidores de catálogo e de contêiner do WebSphere Application Server é vinculado aos processos nos quais esses servidores são executados.

### Procedimento

- **Iniciando serviços de catálogos no WebSphere Application Server:**

O ciclo de vida de um servidor de catálogos é ligado ao processo do WebSphere Application Server. Após configurar o domínio de serviço de catálogo no WebSphere Application Server, reinicie cada servidor que for definido como parte do domínio de serviço de catálogo. O serviço de catálogo é iniciado automaticamente nos servidores que estiverem associados ao domínio de serviço de catálogo. O serviço de catálogo também pode iniciar automaticamente nos seguintes cenários, dependendo da edição do WebSphere Application Server:

- **WebSphere Application Server de Base:** Seu aplicativo pode ser configurado para iniciar automaticamente um servidor de contêiner e um serviço de catálogo. Este recurso simplifica o teste de unidade em ambientes de desenvolvimento como o Rational Application Developer, pois você não precisa iniciar explicitamente um serviço de catálogo. Consulte o “Configurando os Aplicativos do WebSphere Application Server para Iniciar Automaticamente os Servidores de Contêiner” na página 270 para obter informações adicionais.

- **WebSphere Application Server Network Deployment:** O serviço de catálogo iniciará automaticamente no processo do gerenciador de implementação se o nó do gerenciador de implementação tiver o WebSphere eXtreme Scale instalado e se o perfil do gerenciador de implementação estiver aumentado. Consulte o “Configurando o Serviço de Catálogo no WebSphere Application Server” na página 252 para obter informações adicionais.
- **Iniciando servidores de contêiner no WebSphere Application Server:**

O ciclo de vida de um servidor de contêiner é ligado ao aplicativo WebSphere Application Server. Quando você inicia o aplicativo configurado, os servidores de contêiner também são iniciados.
- **Parando uma grade de dados inteira de servidores:**

É possível parar os servidores de catálogos e de contêiner ao parar os aplicativos e os servidores de aplicativos associados. No entanto, também é possível parar uma grade de dados inteira com o utilitário **xscmd** ou com o MBeans:

  - **No utilitário xscmd:**

Consulte o “Parando Servidores Corretamente com o Utilitário **xscmd**” na página 400 para obter mais informações sobre como parar uma grade de dados inteira.
  - **Com Mbeans:**

Use a operação `tearDownServers` no MBean `PlacementServiceMBean`.

---

## Usando a API do Servidor Integrado para Iniciar e Parar Servidores

Com o WebSphere eXtreme Scale, é possível usar uma API programática para gerenciar o ciclo de vida de servidores e contêineres integrados. É possível configurar programaticamente o servidor com qualquer uma das opções que também podem ser configuradas com as opções de linha de comandos ou com as propriedades de servidor baseadas em arquivo. Você pode configurar o servidor integrado para que seja um servidor de contêiner, um serviço de catálogo ou ambos.

### Antes de Iniciar

É necessário ter um método para executar o código a partir de um Java Virtual Machine já existente. As classes do eXtreme Scale deverão estar disponíveis na árvore do carregador de classes.

### Sobre Esta Tarefa

É possível executar muitas tarefas de administração com a API de Administração. Um uso comum da API é como um servidor interno para armazenar o estado do aplicativo da Web. O servidor da Web pode iniciar um servidor WebSphere eXtreme Scale integrado, relatar o servidor de contêiner para o serviço de catálogo e o servidor será, em seguida, incluído como um membro de uma grade distribuída maior. Esse uso pode fornecer escalabilidade e alta disponibilidade para um armazém de dados voláteis do contrário.

É possível controlar programaticamente o ciclo de vida completo de um servidor eXtreme Scale integrado. Os exemplos são genéricos o máximo possível e mostram apenas exemplos de código diretos para etapas realçadas.

### Procedimento

1. Obtenha o objeto `ServerProperties` da classe `ServerFactory` e configure quaisquer opções necessárias.

Cada servidor eXtreme Scale possui um conjunto de propriedades configuráveis. Quando um servidor é iniciado na linha de comandos, essas propriedades são configuradas para os padrões, mas é possível substituir várias propriedades ao fornecer uma origem ou arquivo externo. No escopo integrado, é possível configurar diretamente as propriedades com um objeto `ServerProperties`. Essas propriedades devem ser configuradas antes de obter uma instância do servidor a partir da classe `ServerFactory`. O seguinte exemplo de fragmento de código obtém um objeto `ServerProperties`, configura o campo `CatalogServiceBootstrap` e inicializa várias configurações do servidor opcionais. Consulte a documentação da API para obter uma lista de definições configuráveis.

```
ServerProperties props = ServerFactory.getServerProperties();
props.setCatalogServiceBootstrap("host:port"); // necessário para conectar a um serviço
de catálogo específico
props.setServerName("ServerOne"); // nome do servidor
props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled"); // Configura a espec. de rastreamento
```

2. Se desejar que o servidor seja um serviço de catálogo, obtenha o objeto `CatalogServerProperties`.

Cada servidor integrado pode ser um serviço de catálogo, um servidor de contêiner ou ambos. O seguinte exemplo obtém o objeto `CatalogServerProperties`, ativa a opção de serviço de catálogo e define várias configurações de serviço de catálogo.

```
CatalogServerProperties catalogProps = ServerFactory.getCatalogProperties();
catalogProps.setCatalogServer(true); // false por padrão, necessário para configurar
como um serviço de catálogo catalogProps.setQuorum(true); // ativa / desativa quorum
```

3. Obtenha a instância `Servidor` a partir da classe `ServerFactory`. A instância `doServidor` é um singleton de escopo definido de processo responsável por gerenciar a associação na grade. Depois que essa instância for inicializada, esse processo será conectado e altamente disponibilizado com outros servidores na grade. A seguir há um exemplo de como criar a instância `Servidor`:

```
Server server = ServerFactory.getInstance();
```

Revisando o exemplo anterior, a classe `ServerFactory` fornece um método estático que retorna uma instância `Servidor`. A classe `ServerFactory` deve ser a única interface para obter uma instância `Servidor`. Portanto, a classe garante que a instância seja um singleton ou uma instância para cada JVM ou carregador de classes isolado. O método `getInstance` inicializa a instância `Servidor`. É necessário configurar todas as propriedades do servidor antes de inicializar a instância. A classe `Servidor` é responsável por criar novas instâncias do Contêiner. É possível usar ambas as classes `ServerFactory` e `Servidor` para gerenciar o ciclo de vida da instância do `Servidor` integrada.

4. Inicie uma instância Contêiner usando a instância do `Servidor`.

Antes que os shards possam ser colocados em um servidor integrado, é necessário criar um contêiner no servidor. A interface `Servidor` possui um método `createContainer` que utiliza um argumento `DeploymentPolicy`. O seguinte exemplo utiliza a instância do servidor obtida para criar um contêiner usando um arquivo `DeploymentPolicy` criado. Note que os contêineres requerem um carregador de classes que possui os binários do aplicativo disponíveis para serialização. É possível tornar esses binários disponíveis ao chamar o método `createContainer` com o carregador de classes de contexto Encadeamento para configurar o carregador de classes que deseja usar.

```
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy
(new URL("file://urltodeployment.xml"),
 new URL("file://urltoobjectgrid.xml"));
Container container = server.createContainer(policy);
```

5. Remover e limpar um contêiner.



É possível remover e limpar um servidor de contêiner usando o método `teardown` em execução na instância do Contêiner obtida. Executar o método `teardown` adequadamente em um contêiner limpa o contêiner e remove o contêiner do servidor integrado.

O processo de limpeza do contêiner inclui o movimento e a remoção de todos os shards que são colocados dentro desse contêiner. Cada servidor pode conter muitos contêineres e shards. Limpar um contêiner não afeta o ciclo de vida da instância do Servidor pai. O seguinte exemplo demonstra como executar o método `teardown` em um servidor. O método `teardown` é disponibilizado por meio da interface `ContainerMBean`. Ao usar a interface `ContainerMBean`, se você não tiver mais acesso programático nesse contêiner, ainda poderá remover e limpar o contêiner com o `MBean`. Um método `terminate` também existe na interface `Contêiner`, mas não use-o a menos que realmente seja necessário. Esse método é mais forte e não coordena o movimento e a limpeza apropriados dos shards.

```
container.teardown();
```

## 6. Parar o servidor integrado.

Ao parar um servidor integrado, todos os contêineres e shards que estiverem em execução também são parados no servidor. Ao parar um servidor integrado, é necessário limpar todas as conexões abertas e mover ou remover todos os shards. O seguinte exemplo mostra como parar um servidor e o uso do método `waitFor` na interface do Servidor para garantir que a instância do Servidor seja parada completamente. Da mesma forma que o exemplo de contêiner, o método `stopServer` é disponibilizado por meio da interface `ServerMBean`. Com essa interface, é possível parar um servidor com o bean gerenciado (`MBean`) correspondente.

```
ServerFactory.stopServer(); // Usa o factory para interromper o singleton de Servidor
// ou
server.stopServer(); // Usa a instância do Servidor diretamente
server.waitFor(); // Retorna quando o servidor tiver concluído corretamente
os procedimentos de encerramento
```

### Código de exemplo completo:

```
import java.net.MalformedURLException;
import java.net.URL;

import com.ibm.websphere.objectgrid.ObjectGridException;
import com.ibm.websphere.objectgrid.deployment.DeploymentPolicy;
import com.ibm.websphere.objectgrid.deployment.DeploymentPolicyFactory;
import com.ibm.websphere.objectgrid.server.Container;
import com.ibm.websphere.objectgrid.server.Server;
import com.ibm.websphere.objectgrid.server.ServerFactory;
import com.ibm.websphere.objectgrid.server.ServerProperties;

public class ServerFactoryTest {

    public static void main(String[] args) {

        try {

            ServerProperties props = ServerFactory.getServerProperties();
            props.setCatalogServiceBootstrap("catalogservice-hostname:catalogservice-port");
            props.setServerName("ServerOne"); // name server
            props.setTraceSpecification("com.ibm.ws.objectgrid=all=enabled"); // TraceSpec

            /*
             * Na maioria dos casos, o servidor atuará apenas como um servidor de contêiner
             * e será conectado a um serviço de catálogo externo. Essa é a forma mais
             * disponibilizada de fazer as coisas. O trecho de código comentado a seguir
             * ativará esse Servidor como um serviço de catálogo.
             */
            /*
             * CatalogServerProperties catalogProps =
             * ServerFactory.getCatalogProperties();
             * catalogProps.setCatalogServer(true); // ativar serviço de catálogo
             * catalogProps.setQuorum(true); // ativar quorum
             */
        }
    }
}
```





```

// Set the name of the zone this process is contained in.
serverProperties.setZoneName("EmbeddedZone1");

// Set the end point information required to bootstrap to the catalog service.
serverProperties.setCatalogServiceBootstrap("localhost:2809");

// Set the ORB listener host name to use to bind to.
serverProperties.setListenerHost("host.local.domain");

// Set the ORB listener port to use to bind to.
serverProperties.setListenerPort(9010);

// Turn off all MBeans for this process.
serverProperties.setMBeansEnabled(false);

Server server = ServerFactory.getInstance();

```

## Integrando o Serviço de Catálogo

Qualquer configuração JVM que é sinalizada pelo método `CatalogServerProperties.setCatalogServer` pode hospedar o serviço de catálogo para o eXtreme Scale. Este método indica ao tempo de execução do servidor do eXtreme Scale para instanciar o serviço de catálogo quando o servidor for iniciado. O código a seguir mostra como instanciar o servidor de catálogos do eXtreme Scale:

```

CatalogServerProperties catalogServerProperties =
    ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

Server server = ServerFactory.getInstance();

```

## Integrando o Contêiner do eXtreme Scale

Emita o método `Server.createContainer` para qualquer JVM para hospedar vários contêineres do eXtreme Scale. O código a seguir mostra como instanciar um contêiner do eXtreme Scale:

```

Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new
    File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL());
Container container = server.createContainer(policy);

```

## Processo do Servidor Autocontido

Você pode iniciar todos os serviços juntos, o que é útil para o desenvolvimento e também prático na produção. Ao iniciar os serviços juntos, um único processo faz todo o seguinte: inicia o serviço de catálogo, inicia um conjunto de contêineres e executa a lógica de conexão do cliente. A inicialização dos serviços desta forma classifica os problemas de programação antes da implementação em um ambiente distribuído. O código a seguir mostra como instanciar um servidor eXtreme Scale autocontido:

```

CatalogServerProperties catalogServerProperties =
    ServerFactory.getCatalogProperties();
catalogServerProperties.setCatalogServer(true);

Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new
    File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL());
Container container = server.createContainer(policy);

```

## Incorporando o eXtreme Scale no WebSphere Application Server

A configuração para o eXtreme Scale é definida automaticamente quando o eXtreme Scale é instalado em um ambiente do WebSphere Application Server. Você não precisa configurar quaisquer propriedades antes de acessar o servidor para criar um contêiner. O código a seguir mostra como instanciar um servidor do eXtreme Scale no WebSphere Application Server:

```
Server server = ServerFactory.getInstance();
DeploymentPolicy policy = DeploymentPolicyFactory.createDeploymentPolicy(
    new
    File("META-INF/embeddedDeploymentPolicy.xml").toURI().toURL(),
    new File("META-INF/embeddedObjectGrid.xml").toURI().toURL());
Container container = server.createContainer(policy);
```

Para obter um exemplo passo a passo de como iniciar um serviço de catálogo integrado e contêiner programaticamente, consulte “Usando a API do Servidor Integrado para Iniciar e Parar Servidores” na página 402.

---

## Administrando com o Utilitário xscmd

Com o **xscmd**, é possível concluir tarefas administrativas no ambiente como: estabelecer links de replicação multimestre, substituir o quorum e parar grupos de servidores com o comando `teardown`.

### Antes de Iniciar

- Seus servidores de catálogos e servidores de contêiner devem ser iniciados. Se seus servidores de catálogos estiverem em um domínio de serviço de catálogo, pelo menos dois servidores de catálogo deverão ser iniciados.
- Verifique se a variável de ambiente `JAVA_HOME` está configurada para usar o ambiente de tempo de execução que foi instalado com o produto. Se estiver usando a versão de avaliação do produto, você deverá configurar a variável de ambiente `JAVA_HOME`.

### Sobre Esta Tarefa

O utilitário **xscmd** substitui o utilitário **xsadmin** de amostra como uma ferramenta de monitoramento e de administração totalmente suportadas. Seria possível concluir operações semelhantes com a ferramenta **xsadmin**, mas esta ferramenta não é suportada. A amostra **xsadmin** fornece um método para analisar e descobrir dados de implementação atuais e pode ser usada como uma base para gravar utilitários customizados. Se a ferramenta **xsadmin** era usada anteriormente para monitoramento e administração, considere atualizar seus scripts para usar o utilitário **xscmd**. Para obter informações sobre os comandos **xsadmin** para os novos comandos **xscmd**, consulte “Ferramenta **xsadmin** para a Migração de Ferramenta **xscmd**” na página 213.

### Procedimento

1. Abra uma janela de linha de comandos. Na linha de comandos, configure as variáveis de ambiente apropriadas.
  - a. Configure a variável de ambiente `CLIENT_AUTH_LIB`:
    - **Windows** `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`
    - **UNIX** `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`  
`export CLIENT_AUTH_LIB`
2. Acesse o diretório `wxs_home/bin`.

```
cd wxs_home/bin
```

3. Exiba a ajuda para as várias opções **xscmd**.

- Para exibir a ajuda geral, execute o seguinte comando:

```
– UNIX ./xscmd.sh -h
```

```
– Windows xscmd.bat -h
```

- Para exibir uma lista de todos os comandos, execute o comando a seguir:

```
– UNIX ./xscmd.sh -lc
```

```
– Windows xscmd.bat -lc
```

- Para exibir a ajuda para um comando específico, execute o seguinte comando:

```
– UNIX ./xscmd.sh -h command_name
```

```
– Windows xscmd.bat -h command_name
```

- Para exibir uma lista dos grupos de comandos, execute o seguinte comando:

```
– UNIX ./xscmd.sh -lcg
```

```
– Windows xscmd.bat -lcg
```

- Para exibir uma lista dos comandos dentro de um grupo de comandos, execute o seguinte comando:

```
– UNIX ./xscmd.sh -lc command_group_name
```

```
– Windows xscmd.bat -lc command_group_name
```

4. Execute comandos que se conectam a servidores de catálogos específicos. Por padrão, **xscmd** se conecta ao servidor de catálogos no host local, usando o nome do host e a porta do localhost:2809. Também é possível fornecer uma lista de nomes de host e portas para o comando para que você possa se conectar aos servidores de catálogo em outros hosts. Na lista, um dos utilitários **xscmd** se conecta a um host aleatório. A lista de hosts que você fornece deve estar dentro do mesmo domínio de serviço de catálogo.

- Forneça uma lista de servidores de catálogos independentes a serem conectados:

```
– UNIX ./xscmd.sh -c <command_name> -cep  
hostname:port(,hostname:port)
```

```
– Windows xscmd.bat -c <command_name> -cep  
hostname:port(,hostname:port)
```

Nos comandos, o *command\_name* é o nome do comando que você está executando. O valor *hostname:port* é o nome do host e a porta do listener do servidor de catálogos. O valor da porta do listener em um servidor de catálogos independente é especificado quando você executa o comando **startOgServer**.

- Forneça uma lista de servidores de catálogos do WebSphere Application Server a serem conectados. Não é possível conectar-se aos servidores de catálogos que estão em execução no WebSphere Application Server com o valor do host local padrão:

```
– UNIX ./xscmd.sh -c <command_name> -cep  
was_hostname:port(,hostname:port)
```

```
– Windows xscmd.bat -c <command_name> -cep  
was_hostname:port(,hostname:port)
```

Nos comandos, o *command\_name* é o nome do comando que você está executando. O valor *was\_hostname* é o nome do host do servidor de catálogos na célula do WebSphere Application Server. O valor de *port* é a porta do listener. O valor da porta do listener no WebSphere Application Server é herdado pela configuração da porta BOOTSTRAP\_ADDRESS. O valor padrão é 9809 se o servidor de catálogos está em execução no gerenciador de implementação. Se você estiver executando o servidor de catálogos em um servidor de aplicativos, verifique a configuração da porta BOOTSTRAP\_ADDRESS do servidor de aplicativos para determinar o número da porta.

---

## Iniciando Servidores do eXtreme Scale Usando a Estrutura do Eclipse Equinox OSGi

Os servidores de contêiner do WebSphere eXtreme Scale podem ser iniciados em uma estrutura do Eclipse Equinox OSGi usando vários métodos.

### Antes de Iniciar

Antes de poder iniciar um contêiner do eXtreme Scale, você deve ter concluído as tarefas a seguir:

1. O pacote configurável do servidor do WebSphere eXtreme Scale deve estar instalado no Eclipse Equinox.
2. Seu aplicativo deve ser empacotado como um pacote configurável OSGi.
3. Seus plug-ins do WebSphere eXtreme Scale (se houver) devem ser empacotados como um pacote configurável OSGi. Eles podem ser empacotados no mesmo pacote configurável que seu aplicativo ou como pacotes configuráveis separados.

### Sobre Esta Tarefa

Esta tarefa descreve como iniciar um servidor de contêiner do eXtreme Scale em uma estrutura do Eclipse Equinox OSGi. É possível usar qualquer um dos seguintes métodos para iniciar os servidores de contêiner usando a implementação do Eclipse Equinox:

- Serviço do OSGi Blueprint

É possível incluir toda a configuração e os metadados em um pacote configurável OSGi. Consulte a imagem a seguir para compreender o processo do Eclipse Equinox para este método:

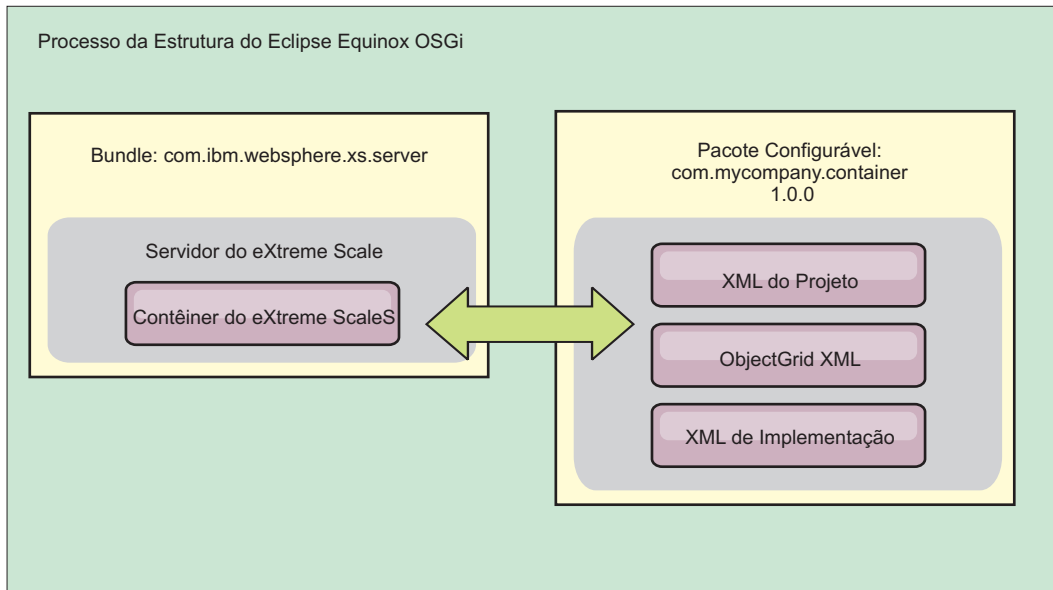


Figura 49. Processo do Eclipse Equinox para Incluir Toda a Configuração e Todos os Metadados em um Pacote Configurável OSGi

- Serviço de Administração de Configuração do OSGi  
É possível especificar a configuração e os metadados fora de um pacote configurável OSGi. Consulte a imagem a seguir para compreender o processo do Eclipse Equinox para este método:

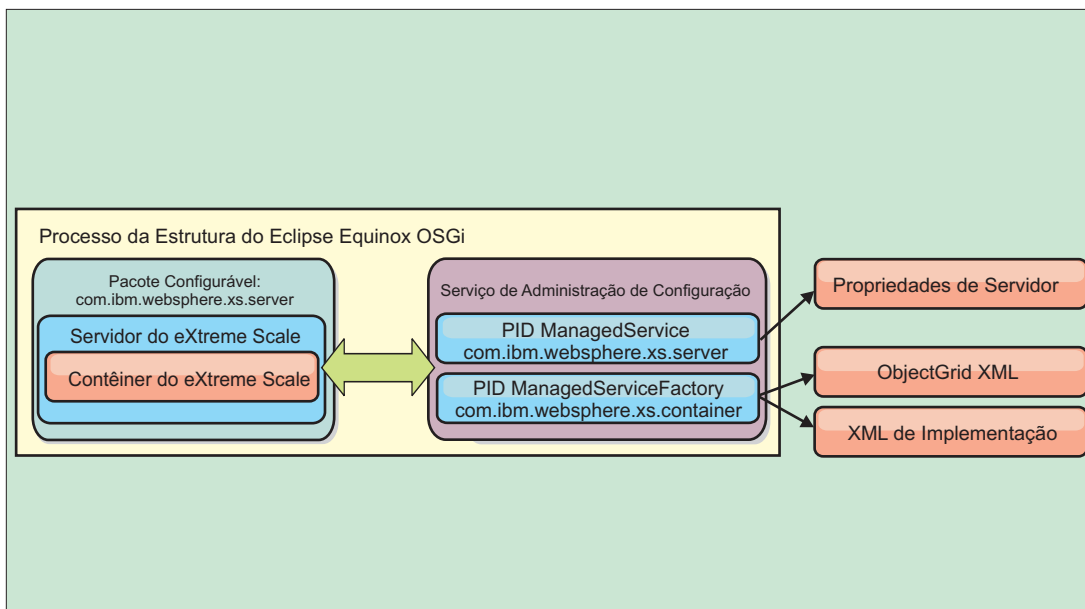


Figura 50. Processo do Eclipse Equinox para Especificar a Configuração e os Metadados Fora de um Pacote Configurável OSGi

- Programaticamente  
Suporta soluções de configuração customizadas.

Em cada caso, um singleton do servidor do eXtreme Scale é configurado e um ou mais contêineres são configurados.

O pacote configurável do servidor do eXtreme Scale, `objectgrid.jar`, inclui todas as bibliotecas necessárias para iniciar e executar um contêiner de grade do eXtreme Scale em uma estrutura OSGi. O ambiente de tempo de execução do servidor se comunica com os plug-ins e objetos de dados fornecidos pelo usuário usando o gerenciador de serviço OSGi.

**Importante:** Depois de um pacote configurável do servidor do eXtreme Scale ser iniciado e o servidor do eXtreme Scale ser inicializado, ele não poderá ser reiniciado. O processo do Eclipse Equinox deve ser reiniciado para reiniciar um servidor do eXtreme Scale.

É possível usar o suporte do eXtreme Scale para o namespace do Spring para configurar os servidores de contêiner do eXtreme Scale em um arquivo XML do Blueprint. Quando os elemento XML do servidor e do contêiner são incluídos no arquivo XML do Blueprint, o manipulador de namespace do eXtreme Scale inicia automaticamente um servidor de contêiner usando os parâmetros que são definidos no arquivo XML do Blueprint quando o pacote configurável é iniciado. A manipulação para o contêiner quando o pacote configurável é interrompido.

Para configurar servidores de contêiner do eXtreme Scale com o XML do Blueprint, conclua as etapas a seguir:

### Procedimento

- Inicie um servidor de contêiner do eXtreme Scale usando o OSGi Blueprint.
  1. Crie um pacote configurável de contêiner.
  2. Instale o pacote configurável de contêiner na estrutura do Eclipse Equinox OSGi. Consulte “Instalando e Iniciando Plug-ins Ativados pelo OSGi”.
  3. Inicie o pacote configurável do contêiner.
- Inicie uma servidor de contêiner do eXtreme Scale usando a administração de configuração do OSGi.
  1. Configure o servidor e o contêiner usando a administração de configuração.
  2. Quando o pacote configurável do servidor do eXtreme Scale é iniciado ou os identificadores persistentes são criados com `config admin`, o servidor e o contêiner iniciam automaticamente.
- Inicie um servidor de contêiner do eXtreme Scale usando a API do `ServerFactory`. Consulte a documentação da API do servidor.
  1. Crie uma classe de ativador do pacote configurável OSGi e use a API do `ServerFactory` do eXtreme Scale para iniciar um servidor.

---

## Instalando e Iniciando Plug-ins Ativados pelo OSGi

Nesta tarefa, você instala o pacote configurável de plug-in dinâmico na estrutura do OSGi. Em seguida, inicia o plug-in.

### Antes de Iniciar

Este tópico assume que as seguintes tarefas foram concluídas:

- O pacote configurável do servidor ou cliente do eXtreme Scale foi instalado na estrutura do Eclipse Equinox OSGi. Consulte “Instalando Pacotes Configuráveis do eXtreme Scale” na página 202.
- Um ou mais plug-ins `BackingMap` ou `ObjectGrid` dinâmicos foram implementados. Consulte `Construindo Plug-ins Dinâmicos do eXtreme Scale`.



- Os plug-ins dinâmicos foram empacotados como serviços OSGi nos pacotes configuráveis OSGi.

## Sobre Esta Tarefa

Esta tarefa descreve como instalar o pacote configurável usando o console do Eclipse Equinox. O pacote configurável pode ser instalado usando vários métodos diferentes, incluindo a modificação do arquivo de configuração `config.ini`. Os produtos que incorporam o Eclipse Equinox incluem métodos alternativos para gerenciar pacotes configuráveis. Para obter mais informações sobre como incluir pacotes configuráveis no arquivo `config.ini` no Eclipse Equinox, consulte as opções de tempo de execução do Eclipse.

O OSGi permite que pacotes configuráveis que possuem serviços duplicados sejam iniciados. O WebSphere eXtreme Scale usa a classificação de serviço mais recente. Ao iniciar diversas estruturas OSGi em uma grade de dados do eXtreme Scale, você deve se certificar de que as classificações de serviço corretas sejam iniciados em cada servidor. Não fazer isso faz com que a grade seja iniciada com uma mistura de diferentes versões.

Para ver quais versões estão em uso pela grade de dados, use o utilitário `xscmd` para verificar as classificações atuais e disponíveis. Para obter mais informações sobre classificações de serviço disponíveis, consulte “Atualizando Serviços OSGi para Plug-ins do eXtreme Scale com `xscmd`” na página 416.

## Procedimento

Instale o pacote configurável de plug-in na estrutura do Eclipse Equinox OSGi usando o console do OSGi.

1. Inicie a estrutura do Eclipse Equinox com o console ativado; por exemplo:  

```
<java_home>/bin/java -jar <equinox_root>/plugins/org.eclipse.osgi_3.6.1.R36x_v20100806.jar -console
```
2. Instale o pacote configurável de plug-in no console do Equinox.  

```
osgi> install file:///<path to bundle>
```

O Equinox exibe o ID do pacote configurável para o pacote configurável recém-instalado:

```
Bundle id is 17
```

3. Insira a linha a seguir para iniciar o pacote configurável no console do Equinox, em que `<id>` é o ID do pacote configurável designado quando o pacote configurável foi instalado:  

```
osgi> install <id>
```
4. Recupere o status do serviço no console do Equinox para verificar se o pacote configurável foi iniciado:  

```
osgi> ss
```

Quando o pacote configurável foi iniciado com êxito, o pacote configurável exibe o estado ACTIVE; por exemplo:

```
17      ACTIVE      com.mycompany.plugin.bundle_VRM
```

Instale o pacote configurável de plug-in na estrutura do Eclipse Equinox OSGi usando o arquivo `config.ini`.

5. Copie o pacote configurável de plug-in no diretório de plug-ins do Eclipse Equinox; por exemplo:  

```
<equinox_root>/plugins
```

6. Edite o arquivo de configuração `config.ini` do Eclipse Equinox e inclua o pacote configurável na propriedade `osgi.bundles`; por exemplo:

```
osgi.bundles=\
org.eclipse.osgi.services_3.2.100.v20100503.jar@1:start, \
org.eclipse.osgi.util_3.2.100.v20100503.jar@1:start, \
org.eclipse.equinox.cm_1.0.200.v20100520.jar@1:start, \
com.mycompany.plugin.bundle_VRM.jar@1:start
```

**Importante:** Verifique se existe uma linha em branco após o nome do último pacote configurável. Cada pacote configurável é separado por uma vírgula.

7. Inicie a estrutura do Eclipse Equinox com o console ativado; por exemplo:

```
<java_home>/bin/java -jar <equinox_root>/plugins/org.eclipse.osgi_3.6.1.R36x.v20100806.jar -console
```

8. Recupere o status de serviço no console do Equinox para verificar se o pacote configurável foi iniciado; por exemplo:

```
osgi> ss
```

Quando o pacote configurável foi iniciado com êxito, o pacote configurável exibe o estado `ACTIVE`; por exemplo:

```
17      ACTIVE      com.mycompany.plugin.bundle_VRM
```

## Resultados

O pacote configurável de plug-in agora está instalado e iniciado. O contêiner ou cliente do eXtreme Scale agora pode ser iniciado. Para obter mais informações sobre como desenvolver plug-ins do eXtreme Scale, consulte o tópico APIs e Plug-ins do Sistema APIs e Plug-ins do Sistema.

---

## Administrando Serviços Ativados pelo OSGi Usando o Utilitário `xscmd`

É possível usar o utilitário `xscmd` para concluir as tarefas de administrador, como visualizar serviços e suas classificações que estão sendo usados por cada contêiner e atualizar o ambiente de tempo de execução para utilizar novas versões dos pacotes configuráveis.

### Sobre Esta Tarefa

Com a estrutura do Eclipse Equinox OSGi, é possível instalar diversas versões do mesmo pacote configurável e você pode atualizar esses pacotes configuráveis durante o tempo de execução. O WebSphere eXtreme Scale é um ambiente distribuído que executa os servidores de contêiner em muitas instâncias da estrutura do OSGi.

Os administradores são responsáveis por copiar, instalar e iniciar manualmente pacotes configuráveis na estrutura do OSGi. O eXtreme Scale inclui um `ServiceTrackerCustomizer` OSGi para controlar quaisquer serviços que foram identificados como plug-ins do eXtreme Scale no arquivo XML do descritor do ObjectGrid. Use o utilitário `xscmd` para validar qual versão do plug-in é usado, quais versões estão disponíveis para serem usadas e para executar upgrades do pacote configurável.

O eXtreme Scale usa o número de classificação do serviço para identificar a versão de cada serviço. Quando dois ou mais serviços são carregados com a mesma referência, o eXtreme Scale usa automaticamente o serviço com a classificação mais alta.

## Procedimento

- Execute o comando **osgiCurrent** e verifique se cada servidor eXtreme Scale está usando a classificação do serviço de plug-in correta.

Como o eXtreme Scale escolhe automaticamente a referência de serviço com a classificação mais alta, é possível que a grade de dados possa iniciar com diversas classificações de um serviço de plug-in.

Se o comando detecta uma incompatibilidade de classificações ou se ele é incapaz de localizar um serviço, um nível de erro diferente de zero é configurado. Se o comando foi concluído com êxito, o nível de erro é configurado como 0.

O exemplo a seguir mostra a saída do comando **osgiCurrent** quando dois plug-ins estão instalados na mesma grade em quatro servidores. O plug-in loaderPlugin está usando classificação de 1 e txCallbackPlugin está usando classificação 2.

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
-----
loaderPlugin      1           MyGrid      MapSetA     server1
loaderPlugin      1           MyGrid      MapSetA     server2
loaderPlugin      1           MyGrid      MapSetA     server3
loaderPlugin      1           MyGrid      MapSetA     server4
txCallbackPlugin  2           MyGrid      MapSetA     server1
txCallbackPlugin  2           MyGrid      MapSetA     server2
txCallbackPlugin  2           MyGrid      MapSetA     server3
txCallbackPlugin  2           MyGrid      MapSetA     server4
```

O exemplo a seguir mostra a saída do comando **osgiCurrent** quando server2 foi iniciado com uma classificação mais nova do loaderPlugin:

```
OSGi Service Name Current Ranking ObjectGrid Name MapSet Name Server Name
-----
loaderPlugin      1           MyGrid      MapSetA     server1
loaderPlugin      2           MyGrid      MapSetA     server2
loaderPlugin      1           MyGrid      MapSetA     server3
loaderPlugin      1           MyGrid      MapSetA     server4
txCallbackPlugin  2           MyGrid      MapSetA     server1
txCallbackPlugin  2           MyGrid      MapSetA     server2
txCallbackPlugin  2           MyGrid      MapSetA     server3
txCallbackPlugin  2           MyGrid      MapSetA     server4
```

- Execute o comando **osgiAll** para verificar se os serviços de plug-in foram iniciados corretamente em cada servidor de contêiner do eXtreme Scale.

Quando pacotes configuráveis que contêm serviços que uma configuração do ObjectGrid está referenciando são iniciados, o ambiente de tempo de execução do eXtreme Scale controla automaticamente o plug-in, mas não o usa imediatamente. O comando **osgiAll** mostra quais plug-ins estão disponíveis para cada servidor.

Quando executados sem quaisquer parâmetros, todos os serviços são mostrados para todas as grades e todos os servidores. Filtros adicionais, incluindo o filtro **-serviceName <service\_name>**, podem ser especificados para limitar a saída para um único serviço ou um subconjunto da grade de dados.

O exemplo a seguir mostra a saída do comando **osgiAll** quando dois plug-ins são iniciados em dois servidores. O loaderPlugin possui ambas as classificações, 1 e 2, iniciadas e o txCallbackPlugin tem a classificação 1 iniciada. A mensagem de resumo no final da saída confirma que ambos os servidores consultam as mesmas classificações de serviço:

```
Server: server1
  OSGi Service Name  Available Rankings
  -----
  loaderPlugin       1, 2
  txCallbackPlugin   1
```

```

Server: server2
  OSGi Service Name  Available Rankings
  -----
  loaderPlugin      1, 2
  txCallbackPlugin  1

```

Summary - All servers have the same service rankings.

O exemplo a seguir mostra a saída do comando **osgiAll** quando o pacote configurável que inclui o loaderPlugin com classificação 1 é interrompido no server1. A mensagem de resumo na parte inferior da saída confirma que o server1 agora está ausente no loaderPlugin com classificação 1:

```

Server: server1
  OSGi Service Name  Available Rankings
  -----
  loaderPlugin      2
  txCallbackPlugin  1

```

```

Server: server2
  OSGi Service Name  Available Rankings
  -----
  loaderPlugin      1, 2
  txCallbackPlugin  1

```

Summary - The following servers are missing service rankings:

```

Server  OSGi Service Name Missing Rankings
-----
server1 loaderPlugin      1

```

O exemplo a seguir mostra a saída se o nome do serviço é especificado com o argumento **-sn**, mas o serviço não existe:

```

Server: server2
  OSGi Service Name  Available Rankings
  -----
  invalidPlugin      No service found

```

```

Server: server1
  OSGi Service Name  Available Rankings
  -----
  invalidPlugin      No service found

```

Summary - All servers have the same service rankings.

- Execute o comando **osgiCheck** para verificar conjuntos de serviços de plug-in e classificações para ver se eles estão disponíveis.

O comando **osgiCheck** aceita um ou mais conjuntos de classificações de serviço no formato: `-serviceRankings <serviceName>;<ranking>[,<serviceName>;<ranking>]`

Quando as classificações estão todas disponíveis, o método retorna com um nível de erro igual a 0. Se uma ou mais classificações estão indisponíveis, um erro de nível diferente de zero é configurado e uma tabela de todos os servidores que não incluem as classificações de serviço especificadas. Filtros adicionais podem ser usados para limitar a verificação de serviço para um subconjunto dos servidores disponíveis no domínio do eXtreme Scale.

Por exemplo, se a classificação ou o serviço especificado estiver ausente, a seguinte mensagem será exibida:

```

Server  OSGi Service Unavailable Rankings
-----
server1 loaderPlugin 3
server2 loaderPlugin 3

```

- Execute o comando **osgiUpdate** para atualizar a classificação de um ou mais plug-ins para todos os servidores em um único ObjectGrid e MapSet em uma única operação.

O comando aceita um ou mais conjuntos de classificações de serviço no formato:  
`-serviceRankings <service name>;<ranking>[,<serviceName>;<ranking>] -g <grid name> -ms <mapset name>`

Com este comando, é possível concluir as operações a seguir:

- Verifique se os serviços especificados estão disponíveis para atualização em cada um dos servidores.
- Altere o estado da grade para offline usando a interface StateManager. Consulte “Gerenciando a Disponibilidade do ObjectGrid” na página 421 para obter mais informações. Este processo coloca a grade em modo quiesce e aguarda até que qualquer transação em execução tenha concluído e impede o início de qualquer nova transação. Este processo também sinaliza quaisquer plug-ins ObjectGridLifecycleListener e BackingMapLifecycleListener para descontinuar qualquer atividade transacional. Consulte Plug-ins para Fornecer Listeners de Eventos para obter informações sobre plug-ins do listener de eventos.
- Atualize cada contêiner do eXtreme Scale em execução em uma estrutura OSGi para usar as novas versões de serviço.
- Altere o estado da grade para online, permitindo que as transações continuem.

O processo de atualização é idempotente, de forma que, se um cliente falhar ao concluir qualquer tarefa, isto resultará na operação sendo recuperada. Se um cliente for incapaz de executar a recuperação ou for interrompido durante o processo de atualização, o mesmo comando poderá ser emitido novamente e ele continuará na etapa apropriada.

Se o cliente for incapaz de continuar, e o processo for reiniciado a partir de um outro cliente, use a opção `-force` para permitir que o cliente execute a atualização. O comando `xscmd.bat/xscmd.sh` impede que diversos clientes atualizem o mesmo conjunto de mapas simultaneamente. Para obter mais detalhes sobre o comando **osgiUpdate**, consulte “Atualizando Serviços OSGi para Plug-ins do eXtreme Scale com `xscmd`”.

## Atualizando Serviços OSGi para Plug-ins do eXtreme Scale com `xscmd`

O WebSphere eXtreme Scale suporta o upgrade de pacotes configuráveis de plug-in do servidor de contêiner enquanto a grade está ativa. Este suporte permite que administradores concluam atualizações e adições do aplicativo sem precisar reiniciar processos da grade.

### Antes de Iniciar

Conclua as etapas a seguir antes de atualizar pacotes configuráveis OSGi do eXtreme Scale para uma nova versão:

1. Inicie servidores do eXtreme Scale em uma estrutura OSGi suportada.
2. Separe todos os plug-ins do eXtreme Scale em pacotes configuráveis e eles devem usar classificações de serviço para identificar cada versão de the plug-ins.
3. Especifique objetos de cache como tipos primitivos Java como `byte[]`, `Integer` ou `String`, ou eles devem ser armazenados usando um plug-in `MapSerializerPlugin`. Os objetos de dados são armazenados no pacote

configurável do eXtreme Scale e não são atualizados. Apenas os plug-ins que interagem com os dados são atualizados.

4. Projete os dados de objeto de cache para serem compatíveis na versão. Novos plug-ins devem ser capazes de interagir com os dados criados por plug-ins antigos.
5. Projete plug-ins para atender aos eventos de ObjectGridLifecycle e BackingMapLifecycle para atualizar quaisquer referências a outros plug-ins ou os metadados que os plug-ins possam ter para que eles possam ser atualizados quando ele for atualizado.
6. O processo de atualização do OSGi do eXtreme Scale afeta somente servidores. Você deve atualizar de modo independente quaisquer clientes que estão usando plug-ins.

## Sobre Esta Tarefa

Sem de ativação do OSGi, se um administrador precisar atualizar os plug-ins do aplicativo ou objetos de cache, cada nó da grade deverá ser atualizado um por um, causando uma sobrecarga na rede, memória e utilização da CPU. Isso é necessário pois os plug-ins e objetos Java do cache são armazenados diretamente na grade. Quando classes são atualizadas sem reiniciar os processos, os plug-ins da grade têm conflitos porque cada classe possui um ClassLoader diferente.

O produto eXtreme Scale inclui o utilitário xscmd e MBeans que permitem aos administradores visualizar todos os pacotes configuráveis do plug-in instalados em cada estrutura OSGi de hosting do contêiner da grade e escolher qual revisão usar. Quando o xscmd é usado para atualizar os plug-ins para uma nova classificação, a grade é colocada no modo quiesce e todas as transações são drenadas, os plug-ins são atualizados e a grade é ativada novamente. Se ocorrer um erro durante o processo de atualização, o processo é retrocedido e a classificação antiga é restaurada.

## Procedimento

1. Crie uma versão do pacote configurável, aumentando o número da versão no manifesto do pacote configurável e aumentando a classificação para cada serviço de plug-in do eXtreme Scale. Se a versão do pacote configurável original for `Bundle-Version: 1.0.0`, a próxima versão poderá ser definida como `Bundle-Version: 1.1.0`.

Se a classificação de serviço original for `ranking="1"`, a próxima classificação poderá ser definida como `ranking="2"`.

**Importante:** As classificações do serviço OSGi devem ser números inteiros.

2. Copie o novo pacote configurável para cada nó de estrutura do OSGi que está hospedando um servidor de contêiner do eXtreme Scale.
3. Instale o novo pacote configurável na estrutura do OSGi. O pacote configurável é designado a um identificador de pacote configurável; por exemplo:  

```
osgi> install <URL to bundle>
```
4. Inicie o novo pacote configurável usando o identificador de pacote configurável designado; por exemplo:  

```
osgi> start <id>
```

Depois que o novo pacote configurável é iniciado, o rastreador de serviço OSGi do eXtreme Scale detecta o pacote configurável e o torna disponível para atualização.

- Use o comando **xscmd -c osgiAll** para verificar se cada servidor de contêiner vê o novo pacote configurável. O comando **osgiAll** consulta todos os contêineres na grade para todos os serviços que são referenciados no arquivo XML do descritor do ObjectGrid e exibe todas as classificações que estão disponíveis; por exemplo:

```
xscmd -c osgiAll
```

```
Server: server1
  OSGi Service Name      Available Rankings
  -----
  myLoaderServiceFactory 1, 2
  mySerializerServiceFactory 1, 2
```

```
Server: server2
  OSGi Service Name      Available Rankings
  -----
  myLoaderServiceFactory 1, 2
  mySerializerServiceFactory 1, 2
```

Summary - All servers have the same service rankings.

- Use o comando **xscmd -c osgiCheck** para verificar se uma ou mais classificações de serviço são destinos de atualização válidos; por exemplo:

```
xscmd -c osgiCheck -sr
mySerializerServiceFactory;2,myLoaderServiceFactory;2
```

CWXSIO040I: The command osgiCheck has completed successfully.

- Se o comando **osgiCheck** não localizou nenhum erro resultante, suspenda o balanceador do serviço de posicionamento para evitar movimentos de shard, no caso de uma falha durante o processo de atualização. Para suspender o posicionamento, use o comando **xscmd -c suspendBalancing** para cada grade de objeto e conjunto de mapas que são afetados pela atualização; por exemplo:

```
xscmd -c suspendBalancing -g MyGrid -ms MyMapSet
```

- Após o balanceamento ter sido suspenso para cada grade de objeto e conjunto de mapas, use o comando **xscmd -c osgiCheck** novamente para verificar se uma ou mais classificações de serviço são destinos de atualização válidos; por exemplo:

```
xscmd -c osgiCheck -sr
mySerializerServiceFactory;2,myLoaderServiceFactory;2
```

CWXSIO040I: The command osgiCheck has completed successfully.

- Após o balanceamento ter sido suspenso para a grade de objeto e o conjunto de mapas, use o comando **osgiUpdate** para atualizar o serviço em todos os servidores para uma grade de objeto e um conjunto de mapas; por exemplo:

```
xscmd -c osgiUpdate -sr
mySerializerServiceFactory;2,myLoaderServiceFactory;2 -g MyGrid -ms MyMapSet
```

- Verifique se o upgrade foi bem-sucedido; por exemplo:

```
Update succeeded for the following service rankings:
Service      Ranking
-----
mySerializerServiceFactory 2
myLoaderServiceFactory 2
```

- Depois de verificar que a classificação foi atualizada com êxito, ative o balanceamento novamente, usando o comando **xscmd -c resumeBalancing**; por exemplo:

```
xscmd -c resumeBalancing -g MyGrid -ms MyMapSet
```



12. Pare e desinstale o pacote configurável antigo em cada estrutura do OSGi que está hospedando o contêiner do eXtreme Scale. Por exemplo, insira o seguinte código no console do Equinox Eclipse:

```
osgi> stop <id>  
osgi> uninstall <id>
```

## Resultados

O pacote configurável do eXtreme Scale foi atualizado para uma nova versão.

---

## Controlando o Posicionamento

É possível usar várias opções diferentes para controlar quando os shards são colocados em vários servidores de contêiner na configuração. Durante a inicialização, você pode optar por atrasar o posicionamento de shards. Quando você estiver executando todos os seus servidores de contêiner, poderá precisar suspender, retomar ou alterar o posicionamento enquanto mantém os servidores.

### Procedimento

Controlando o posicionamento durante a inicialização

É possível controlar quando os shards começam a ser colocados enquanto seu ambiente está iniciando. Algum controle está ativo por padrão. Se você não executar nenhuma ação para controlar o posicionamento, os shards começarão a ser colocados imediatamente. Quando os shards são colocados imediatamente, eles podem não ser colocados uniformemente conforme os servidores de contêiner subsequentes iniciam e operações de posicionamento adicionais são executadas para equilibrar a distribuição.

- Suspenda temporariamente o balanceamento de shards para evitar o posicionamento de shard imediato quando seus servidores de contêiner estão iniciando.

Antes de iniciar seus servidores de contêiner, use o comando **xscmd -c suspendBalancing** para parar o balanceamento de shards para uma grade de dados e um conjunto de mapas específicos. Após os servidores de contêiner serem iniciados, é possível usar o comando **xscmd -c resumeBalancing** para iniciar o posicionamento de shards nos servidores de contêiner.

- **7.1.1+** Configure a propriedade **placementDeferralInterval**.

A propriedade **placementDeferralInterval** minimiza o número de ciclos de posicionamento de shard nos servidores de contêiner. O posicionamento de shard é acionado no intervalo de tempo definido.

Configure a propriedade **placementDeferralInterval** no arquivo de propriedades do servidor para o servidor de catálogos. Se estiver usando a API do servidor integrado, use o método `setPlacementDeferralInterval` na interface `CatalogServerProperties`. Esta propriedade configura vários milissegundos antes de os shards serem colocados nos servidores de contêiner. O valor padrão para esta propriedade é 15 segundos. Com o valor padrão, quando um servidor de contêiner é iniciado, o posicionamento não é iniciado até após o tempo especificado na propriedade ter passado. Se diversos servidores de contêiner estiverem iniciando sucessivamente, o cronômetro do intervalo de adiamento será reconfigurado se um novo servidor de contêiner for iniciado dentro do intervalo determinado. Por exemplo, se um segundo servidor de contêiner iniciar 10 segundos depois que o primeiro servidor de contêiner, o posicionamento não será iniciado até 15 segundos após o segundo servidor de

contêiner ser iniciado. No entanto, se um terceiro servidor de contêiner iniciar 20 segundos após o segundo servidor de contêiner, o posicionamento já iniciou nos primeiros dois servidores de contêiner.

Quando servidores de contêiner se tornam indisponíveis, o posicionamento é acionado assim que o servidor de catálogos tem conhecimento do evento de forma que a recuperação possa ocorrer o mais rapidamente possível.

É possível usar as seguintes dicas para ajudar a determinar se seu valor de adiamento de posicionamento está configurado com a quantidade correta de tempo:

- Já que você inicia simultaneamente os servidores de contêiner, consulte as mensagens CWOBJ1001 no arquivo SystemOut.log para cada servidor de contêiner. O registro de data e hora destas mensagens em cada arquivo de log do servidor de contêiner indica o horário de início do servidor de contêiner real. Você pode considerar ajustar a propriedade **placementDeferralInterval** para incluir mais inícios do servidor de contêiner. Por exemplo, se o primeiro servidor de contêiner inicia 90 segundos antes do último servidor de contêiner, você pode configurar a propriedade como 90 segundos.
- Observe por quanto tempo as mensagens CWOBJ1511 ocorrem depois das mensagens CWOBJ1001. Este período de tempo pode indicar se o adiamento ocorreu com êxito.
- Se você estiver usando um ambiente de desenvolvimento, considere o comprimento do intervalo quando estiver testando seu aplicativo.

- Configure o atributo **numInitialContainers**.

Se você usou anteriormente o atributo **numInitialContainers**, poderá continuar usando o atributo. No entanto, o uso dos comandos **xscmd -c suspendBalancing** e **xscmd -c resumeBalancing** seguidos pelo **placementDeferralInterval** é sugerido sobre o atributo **numInitialContainers** para controlar o posicionamento. O atributo **numInitialContainers** especifica o número de servidores de contêiner que são necessários antes do posicionamento inicial ocorrer para os shards neste elemento mapSet. O atributo **numInitialContainers** está no arquivo XML do descritor da política de implementação. Se você tiver **numInitialContainers** e **placementDeferralInterval** configurados, observe que até que o valor de **numInitialContainers** ser atendido, nenhum posicionamento ocorre, independentemente do valor da propriedade **placementDeferralInterval**.

Controlando o posicionamento após a inicialização inicial

- Force a ocorrência do posicionamento.

É possível usar o comando **xscmd -c triggerPlacement -g my\_OG -ms my\_Map\_Set**, em que *my\_OG* e *my\_Map\_Set* são configurados com valores para sua grade de dados e seu conjunto de mapas, para forçar que o posicionamento ocorra durante um ponto no tempo no qual o posicionamento pode não ocorrer de outra forma. Por exemplo, você pode executar esse comando quando o período de tempo especificado pela propriedade **placementDeferralInterval** ainda não passou ou quando o balanceamento é suspenso.

- Redesigne um shard primário.

Use o comando **xscmd -c swapShardWithPrimary** para designar um shard de réplica para ser o novo shard primário. O shard primário anterior se torna uma réplica.

- Rebalanceie os shards primários e de réplica.

Use o comando **xscmd -c balanceShardTypes** para ajustar a proporção de shards primários e de réplica para ser equitativa entre os servidores de contêiner em execução na configuração. A proporção é consistente dentro de um shard em cada servidor de contêiner.

- Suspenda ou retome o posicionamento.  
Use o comando `xscmd -c suspendBalancing` ou o comando `xscmd -c resumeBalancing` para parar e iniciar o balanceamento de shards para uma grade de dados e um conjunto de mapas específicos. Quando o balanceamento tiver sido suspenso, as ações de posicionamento a seguir ainda poderão ser executadas:
  - A promoção de shard pode ocorrer quando os servidores de contêiner falham.
  - A troca de função de shard com o comando `xscmd -c swapShardWithPrimary`.
  - O balanceamento acionado pelo posicionamento de shard com o comando `xscmd -c triggerPlacement -g myOG -ms myMapSet`.

## O que Fazer Depois

É possível monitorar o posicionamento no ambiente com o comando `xscmd -c placementServiceStatus`.

---

## Gerenciando a Disponibilidade do ObjectGrid

O estado de disponibilidade de uma instância do ObjectGrid determina quais pedidos podem ser processados em qualquer momento específico. É possível usar a interface StateManager para configurar e recuperar o estado de uma instância do ObjectGrid.

### Sobre Esta Tarefa

Quatro estados de disponibilidade existem para uma determinada instância do ObjectGrid.

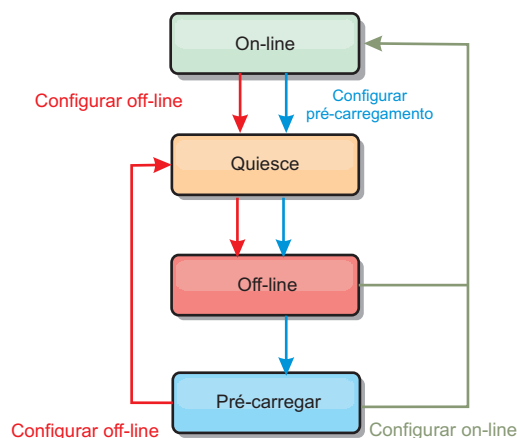


Figura 51. Estados de Disponibilidade de uma Instância do ObjectGrid

### ONLINE

O estado ONLINE é o estado padrão de disponibilidade padrão de um ObjectGrid. Um ObjectGrid ONLINE é capaz de processar qualquer pedido de um cliente típico do eXtreme Scale. Entretanto, os pedidos de um cliente de pré-carregamento são rejeitados enquanto o ObjectGrid está ONLINE.

### QUIESCE

O estado QUIESCE é temporário. Um ObjectGrid que está em QUIESCE logo é movido para o estado OFFLINE. Enquanto estiver no estado QUIESCE, um ObjectGrid tem permissão para processar transações pendentes. Entretanto, qualquer nova transação será rejeitada. Um

ObjectGrid pode permanecer em QUIESCE por até 30 segundos. Após este tempo, o estado de disponibilidade é movido para OFFLINE.

#### **OFFLINE**

O estado OFFLINE resulta na rejeição de todas as transações que são enviadas para o ObjectGrid.

#### **PRELOAD**

O estado PRELOAD pode ser utilizado para carregar dados em um ObjectGrid a partir de um cliente de pré-carregamento. Enquanto o ObjectGrid está em um estado PRELOAD, apenas um cliente de pré-carregamento poderá executar o commit em transações junto ao ObjectGrid. Todas as outras transações são rejeitadas.

Uma solicitação é rejeitada se um ObjectGrid não estiver no estado de disponibilidade apropriado para suportar essa solicitação. Uma exceção AvailabilityException resulta sempre na rejeição de um pedido.

### **Procedimento**

1. Configure o estado inicial de um ObjectGrid com o arquivo XML de configuração do ObjectGrid.

O atributo **initialState** pode ser usado em um ObjectGrid para indicar seu estado de inicialização. Normalmente, quando um ObjectGrid conclui sua inicialização, ele está disponível para roteamento. O estado pode ser posteriormente alterado para evitar o tráfego de roteamento para um ObjectGrid. Se o ObjectGrid precisar ser inicializado, mas não ficar imediatamente disponível, será possível usar o atributo **initialState**.

O atributo initialState é configurado no arquivo XML de configuração do ObjectGrid. O estado padrão é ONLINE. Os valores válidos incluem:

- ONLINE (padrão)
- PRELOAD
- OFFLINE

Consulte Arquivo XML descritor do ObjectGrid para obter informações adicionais sobre o atributo **initialState**.

Se o atributo initialState estiver configurado em um ObjectGrid, o estado deverá ser explicitamente configurado de volta como on-line ou o ObjectGrid permanecerá indisponível. Uma exceção AvailabilityException resultará se o ObjectGrid não estiver no estado ON-LINE.

Consulte AvailabilityState Documentação da API para obter mais informações.

#### **Utilizando o atributo initialState para pré-carregamento**

Se o ObjectGrid for pré-carregado com dados, pode haver um período de tempo entre quando o ObjectGrid está disponível e a mudança para um estado pré-carregado para bloquear o tráfego do cliente. Para evitar este período de tempo, o estado inicial em um ObjectGrid pode ser configurado como PRELOAD. O ObjectGrid ainda conclui todas as inicializações necessárias, mas ele bloqueia o tráfego até que o estado mude e permite que o pré-carregamento ocorra.

Ambos os estados PRELOAD e OFFLINE bloqueiam o tráfego, mas você deve utilizar o estado PRELOAD se deseja iniciar um pré-carregamento.

#### **Comportamento de failover e equilíbrio**

Se uma grade de dados de réplica for promovida para ser uma grade de dados primária, a réplica não usará a configuração **initialState**. Se a grade de dados primária for movida para um re-equilíbrio, a configuração **initialState** não

será usada porque os dados são copiados para o novo local do shard primário antes de a movimentação ser concluída. Se a replicação não for configurada, o primário irá para a configuração **initialState** se ocorrer failover e um novo primário deverá ser colocado.

2. Altere o estado da disponibilidade com a interface `StateManager`.

Use a interface `StateManager` para configurar o estado de disponibilidade de um `ObjectGrid`. Para configurar o estado de disponibilidade de um `ObjectGrid` em execução nos servidores, transmita um cliente `ObjectGrid` correspondente para a interface `StateManager`. O código a seguir demonstra como alterar o estado de disponibilidade de um `ObjectGrid`.

```
ClientClusterContext client = ogManager.connect("localhost:2809", null, null);
ObjectGrid myObjectGrid = ogManager.getObjectGrid(client, "myObjectGrid");
StateManager stateManager = StateManagerFactory.getStateManager();
stateManager.setObjectGridState(AvailabilityState.OFFLINE, myObjectGrid);
```

Cada shard de `ObjectGrid` muda para o estado desejado quando o método `setObjectGridState` é chamado na interface `StateManager`. Quando o método retornar, todos os shards no `ObjectGrid` deverão estar no estado adequado.

Use um plug-in `ObjectGridEventListener` para alterar o estado de disponibilidade de um `ObjectGrid` do lado do servidor. Altere o estado de disponibilidade de um `ObjectGrid` do lado do servidor apenas quando o `ObjectGrid` tiver uma única partição. Se o `ObjectGrid` tiver múltiplas partições, o método `shardActivated` será chamado em cada primário, o que resultará em chamadas desnecessárias para alterar o estado do `ObjectGrid`

```
public class OGListener implements ObjectGridEventListener,
ObjectGridEventGroup.ShardEvents
{
    public void shardActivated(ObjectGrid grid) {
        StateManager stateManager = StateManagerFactory.getStateManager();
        stateManager.setObjectGridState(AvailabilityState.PRELOAD, grid);
    }
}
```

Como **QUIESCE** é o estado transicional, você não pode usar a interface `StateManager` para colocar um `ObjectGrid` em um estado **QUIESCE**. Um `ObjectGrid` transmite passa por este estado em seu caminho para o estado **OFFLINE**.

3. Recupere o estado da disponibilidade.

Use o método `getObjectGridState` da interface `StateManager` para recuperar o estado de disponibilidade de um `ObjectGrid` específico.

```
StateManager stateManager = StateManagerFactory.getStateManager();
AvailabilityState state = stateManager.getObjectGridState(inventoryGrid);
```

O método `getObjectGridState` escolhe um shard primário aleatório no `ObjectGrid` e retorna seu `AvailabilityState`. Como todos os shards de um `ObjectGrid` devem estar no mesmo estado de disponibilidade ou em transição para o mesmo estado de disponibilidade, este método fornece um resultado aceitável para o estado de disponibilidade atual do `ObjectGrid`.

---

## Gerenciando Falhas do Datacenter

Quando o datacenter entra em um cenário de falha, considere substituir o quorum para que os eventos do servidor de contêiner não sejam ignorados. É possível usar o utilitário **xscmd** para consultar informações sobre as tarefas quorum e como executá-las, como o status do quorum e a substituição do quorum.

## Antes de Iniciar

- Defina o mecanismo quorum para que todos os seus servidores de catálogos tenham a mesma configuração. Consulte o Configurando o Mecanismo Quorum para obter informações adicionais.
- O quorum é o número mínimo de servidores de catálogos que são necessários para conduzir operações de posicionamento para a grade de dados e é o conjunto completo de servidores de catálogos, a menos que um número inferior tenha sido configurado. O WebSphere eXtreme Scale espera perder o quorum pelos seguintes motivos:

- Falha do membro da JVM de serviço de catálogo
- Queda de energia da rede
- perda do datacenter

A seguinte mensagem indica que o quorum foi perdido. Procure por essa mensagem nos logs de serviço de catálogo.

CW0BJ1254W: O serviço de catálogo está aguardando pelo quorum.

## Sobre Esta Tarefa

Substitua o quorum apenas em um cenário de falha do datacenter. Quando substituir o quorum, qualquer instância do servidor de catálogos sobrevivente pode ser usada. Todas as instâncias sobreviventes são notificadas quando uma for instruída para substituir o quorum.

## Procedimento

- Consulte o status do quorum com o utilitário **xscmd**.

```
xscmd -c showQuorumStatus -cep cathost:2809
```

Use esta opção para exibir o status do quorum de uma instância de serviço de catálogo. Um dos seguintes resultados é exibido:

- Quorum está desativado: Os servidores de catálogos estão em execução em um modo de quorum desativado. O modo de quorum desativado é um modo de desenvolvimento ou de datacenter único. Não use o modo de quorum desativado para diversas configurações do datacenter.
  - O Quorum está ativado e o servidor de catálogos possui quorum: O quorum está ativado e o sistema trabalha normalmente.
  - O Quorum está ativado, mas o servidor de catálogos está aguardando quorum): O quorum está ativado, porém o quorum foi perdido.
  - O Quorum está ativado, mas o quorum foi substituído: O quorum está ativado, porém ele foi substituído.
  - O status do quorum é ilegal: Quando uma indisponibilidade de energia ocorre, divide o serviço de catálogos em duas partições, A e B. O servidor de catálogos A possui quorum substituído. A partição de rede é resolvida e o servidor na partição B é declarado ilegal, requerendo um reinício da JVM. Isso também ocorre se a JVM de catálogo na partição B for reiniciada durante a queda de energia e, em seguida, a energia voltar.
- Substitua o quorum com o utilitário **xscmd**.
- ```
xscmd -c overrideQuorum -cep cathost:2809
```

Executar esse comando força a servidores de catálogos sobreviventes a restabelecerem um quorum.

- Faça diagnóstico do quorum com o utilitário **xscmd**.
  - **Exiba uma lista dos grupos principais:**



Use a opção **-c listCoreGroups** para exibir uma lista de todos os grupos principais para o servidor de catálogos.

```
xscmd -c listCoreGroups -cep cathost:2809
```

– **Servidores Teardown:**

Use a opção **-c teardown** para remover um servidor manualmente da grade de dados. Geralmente não é necessário remover um servidor da grade. Os servidores serão removidos automaticamente caso uma falha deles seja detectada, porém o comando é fornecido para uso sob orientação do suporte IBM. Consulte o “Parando Servidores Corretamente com o Utilitário **xscmd**” na página 400 para obter mais informações sobre como usar esse comando.

```
xscmd -c teardown server1,server2,server3 -cep cathost:2809 -g Grid
```

– **Exiba a tabela de rota:**

Use a opção **-c routetable** para exibir a tabela de rota atual ao simular uma nova conexão do cliente para a grade de dados. Ele também valida a tabela de rota ao confirmar que todos os servidores de contêineres reconhecem sua função na tabela de rota, como o tipo de shard para uma determinada partição.

```
xscmd -c routetable -cep cathost:2809 -g myGrid
```

– **Verifique os tamanhos de mapa:**

Use a opção **-c showMapSizes** para verificar se a distribuição de chaves é uniforme sobre os shards na chave. Se algum dos servidores de contêiner tiver mais chaves do que outros, é provável que a função hash nos objetos chave tenha sido mal distribuída.

```
xscmd -c showMapSizes -cep cathost:2809 -g myGrid -ms myMapSet
```

– **Configure sequências de rastreamento:**

Use a opção **-c setTraceSpec** para definir as configurações de rastreamento para todas as JVMs que correspondem ao filtro especificado para o comando **xscmd**. Essa configuração altera apenas as configurações de rastreamento, até que outro comando seja usado ou até as JVMs modificadas falharem ou pararem.

```
xscmd -c setTraceSpec -spec ObjectGrid*=event=enabled -cep cathost:1099  
-g myGrid -hf host1
```

Essa sequência ativa o rastreamento para todas as JVMs no servidor com o nome do host especificado que, neste caso, é host1.

– **Exibir shards não designados:**

Use a opção **-c showPlacement -sf U** para exibir a lista de shards que não podem ser colocados na grade de dados. Os shards não podem ser colocados quando o serviço de posicionamento tiver uma restrição que esteja evitando o posicionamento. Por exemplo, se você iniciar as JVMs em um único servidor físico enquanto estiver no modo de produção, apenas os shards primários poderão ser colocados. As réplicas não são designadas até as JVMs serem iniciadas em um segundo servidor físico. O serviço de posicionamento coloca as réplicas apenas nas JVMs com endereços IP diferentes das JVMs que hospedam os shards primários. Não ter nenhuma JVM em uma zona também faz com que os shards não sejam designados.

```
xscmd -c showPlacement -sf U -cep cathost:2809 -g myGrid
```

---

## Administrando com Beans Gerenciados (MBeans)

É possível usar vários tipos diferentes de Java Management Extensions (JMX) MBeans para administrar e monitorar as implementações. Cada MBean refere-se a uma entidade específica, como um mapa, grade de dados, servidor ou serviço.



## Interfaces MBean JMX e WebSphere eXtreme Scale

Cada MBean possui métodos get que representam valores de atributo. Estes métodos get não podem ser chamados diretamente a partir do seu programa. A especificação JMX trata os atributos de maneira diferente das operações. É possível visualizar atributos com um console JMX do fornecedor e executar operações em seu programa ou com um console JMX do fornecedor.

### Pacote com.ibm.websphere.objectgrid.management

Consulte a documentação da API para obter uma visão geral e especificações de programação detalhadas para todos os MBeans disponíveis: Pacote com.ibm.websphere.objectgrid.management .

## Acessando Beans Gerenciados (MBeans) Usando a Ferramenta wsadmin

É possível usar o utilitário wsadmin fornecido no WebSphere Application Server para acessar as informações do bean gerenciado (MBean).

### Procedimento

Execute a ferramenta wsadmin a partir do diretório bin em sua instalação do WebSphere Application Server. O exemplo a seguir recupera uma visualização da disposição do shard atual em um eXtreme Scale dinâmico. A ferramenta wsadmin pode ser executada a partir de qualquer instalação na qual o eXtreme Scale está em execução. Não é necessário executar a ferramenta wsadmin no serviço de catálogo.

```
$ wsadmin.sh -lang jython
wsadmin>placementService = AdminControl.queryNames
("com.ibm.websphere.objectgrid:*,type=PlacementService")
wsadmin>print AdminControl.invoke(placementService,
"listObjectGridPlacement","library ms1")

<objectGrid name="library" mapSetName="ms1">
  <container name="container-0" zoneName="DefaultDomain"
    hostName="host1.company.org" serverName="server1">
    <shard type="Primary" partitionName="0"/>
    <shard type="SynchronousReplica" partitionName="1"/>
  </container>
  <container name="container-1" zoneName="DefaultDomain"
    hostName="host2.company.org" serverName="server2">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="Primary" partitionName="1"/>
  </container>
  <container name="UNASSIGNED" zoneName="_ibm_SYSTEM"
    hostName="UNASSIGNED" serverName="UNNAMED">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="AsynchronousReplica" partitionName="0"/>
  </container>
</objectGrid>
```

## Acessando Beans Gerenciados (MBeans) Programaticamente

É possível se conectar aos MBeans com os aplicativos Java. Esses aplicativos podem usar as interfaces no pacote com.ibm.websphere.objectgrid.management.

### Sobre Esta Tarefa

Os métodos programáticos para acessar os MBeans variam, dependendo do tipo de servidor ao qual você está se conectando.

- Conectar-se a um servidor MBean do serviço de catálogo independente
- Conectar-se a um servidor MBean do contêiner
- Conectar-se a um servidor MBean do serviço de catálogo que é hospedado no WebSphere Application Server
- Conectar-se a um servidor MBean do serviço de catálogo com a segurança ativada

### **Procedimento**

- **Conectar-se a um servidor MBean do serviço de catálogo independente:**  
O programa de exemplo a seguir conecta a um servidor MBean do serviço de catálogo independente e retorna uma sequência XML formatada que lista cada servidor de contêiner junto com seus shards alocado para um determinado ObjectGrid e MapSet.

```

package com.ibm.websphere.sample.xs.admin;

import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Collects the placement information from the Catalog Server for a given ObjectGrid.
 */
public final class CollectPlacementPlan {
    private static String hostName = "localhost";

    private static int port = 1099;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Connects to the ObjectGrid Catalog Service to retrieve placement information and
     * prints it out.
     *
     * @param args
     * @throws Exception
     *
     * If there is a problem connecting to the catalog service MBean server.
     */
    public static void main(String[] args) throws Exception {
        String serviceURL = "service:jmx:rmi:///jndi/rmi://" + hostName + ":" + port +
            "/objectgrid/MBeanServer";
        JMXServiceURL jmxUrl = new JMXServiceURL(serviceURL);
        JMXConnector jmxCon = JMXConnectorFactory.connect(jmxUrl);

        try {
            MBeanServerConnection catalogServerConnection = jmxCon.getMBeanServerConnection();

            Set placementSet = catalogServerConnection.queryNames(new ObjectName
("com.ibm.websphere.objectgrid"
+ ":", type=PlacementService"), null);
            ObjectName placementService = (ObjectName) placementSet.iterator().next();
            Object placementXML = catalogServerConnection.invoke(placementService,
                "listObjectGridPlacement", new Object[] {
                    objectGridName, mapSetName }, new String[] { String.class.getName(),
String.class.getName() });
            System.out.println(placementXML);
        } catch (Exception e) {
            if (jmxCon != null) {
                jmxCon.close();
            }
        }
    }
}

```

Figura 52. *CollectPlacementPlan.java*

Algumas notas sobre o programa de amostra:

- O valor **JMXServiceURL** para o serviço de catálogo está sempre no seguinte formato: `service:jmx:rmi:///jndi/rmi://<host>:<port>/objectgrid/MBeanServer`, em que `<host>` é o host no qual o serviço de catálogo está em execução e `<port>` é a porta do serviço JMX que é fornecida com a opção **-JMXServicePort** quando iniciar o serviço de catálogo. Se nenhuma porta for especificada, o padrão será 1099.

- Para que o ObjectGrid ou as estatísticas do mapa sejam ativadas, você deve especificar a seguinte propriedade no arquivo de propriedades do servidor quando estiver iniciando um contêiner ObjectGrid: statsSpec=all=enabled.
- Para desativar o MBeans que estão em execução nos servidores de contêiner, especifique a seguinte propriedade no arquivo de propriedades do servidor: enableMBeans=false.

Um exemplo da saída é a seguinte. Essa saída indica que dois servidores de contêiner estão ativos. O servidor de contêiner Container-0 hospeda quatro shards primários. O servidor de contêiner Container-1 hospeda uma réplica síncrona para cada um dos shards primários no servidor de contêiner Container-0. Nessa configuração, duas réplicas síncronas e uma réplica assíncrona são configuradas. Como resultado, o servidor de contêiner Não designado é deixado com o restante dos shards. Se dois ou mais servidores de contêiner forem iniciados, o servidor de contêiner Não designado não é exibido.

```
<objectGrid name="library" mapSetName="ms1">
  <container name="Container-1" zoneName="DefaultZone"
    hostname="myhost.mycompany.com" serverName="ogserver">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="SynchronousReplica" partitionName="1"/>
    <shard type="SynchronousReplica" partitionName="2"/>
    <shard type="SynchronousReplica" partitionName="3"/>
  </container>
  <container name="Container-0" zoneName="DefaultZone"
    hostname="myhost.mycompany.com" serverName="ogserver">
    <shard type="Primary" partitionName="0"/>
    <shard type="Primary" partitionName="1"/>
    <shard type="Primary" partitionName="2"/>
    <shard type="Primary" partitionName="3"/>
  </container>
  <container name="library:ms1:_UnassignedContainer_" zoneName="_ibm_SYSTEM"
    hostname="UNASSIGNED" serverName="UNNAMED">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="SynchronousReplica" partitionName="1"/>
    <shard type="SynchronousReplica" partitionName="2"/>
    <shard type="SynchronousReplica" partitionName="3"/>
    <shard type="AsynchronousReplica" partitionName="0"/>
    <shard type="AsynchronousReplica" partitionName="1"/>
    <shard type="AsynchronousReplica" partitionName="2"/>
    <shard type="AsynchronousReplica" partitionName="3"/>
  </container>
</objectGrid>
```

- **Conectar-se a um servidor MBean de contêiner:**

Servidores de contêiner hospedam MBeans para consultar informações sobre os mapas e instâncias do ObjectGrid individuais que estão em execução dentro do servidor de contêiner. O programa de exemplo a seguir imprime o status de cada servidor de contêiner que é hospedado pelo servidor de catálogos com o endereço JMX de localhost:1099:

```

package com.ibm.websphere.sample.xs.admin;

import java.util.List;
import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectInstance;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Collects placement status from each of the available containers directly.
 */
public final class CollectContainerStatus {
    private static String hostName = "localhost";

    private static int port = 1099;

    /**
     * @param args
     */
    public static void main(String[] args) throws Exception {
        String serviceURL = "service:jmx:rmi:///jndi/rmi://" + hostName + ":" + port + "/objectgrid/MBeanServer";
        JMXServiceURL jmxUrl = new JMXServiceURL(serviceURL);
        JMXConnector jmxCon = JMXConnectorFactory.connect(jmxUrl);

        try {
            MBeanServerConnection catalogServerConnection = jmxCon.getMBeanServerConnection();

            Set placementSet = catalogServerConnection.queryNames(new ObjectName("com.ibm.websphere.objectgrid"
                + ".*:*,type=PlacementService"), null);

            ObjectName placementService = (ObjectName) placementSet.iterator().next();
            List<String> containerJMXAddresses = (List<String>) catalogServerConnection.invoke(placementService,
                "retrieveAllServersJMXAddresses", new Object[0], new String[0]);
            for (String address : containerJMXAddresses) {
                JMXServiceURL containerJMXURL = new JMXServiceURL(address);
                JMXConnector containerConnector = JMXConnectorFactory.connect(containerJMXURL);
                MBeanServerConnection containerConnection = containerConnector.getMBeanServerConnection();
                Set<ObjectInstance> containers = containerConnection.queryMBeans(
                    new ObjectName("*:*,type=ObjectGridContainer"), null);
                for (ObjectInstance container : containers) {
                    System.out.println(containerConnection.getAttribute(container.getObjectName(), "Status"));
                }
            }
        } finally {
            if(jmxCon != null) {
                jmxCon.close();
            }
        }
    }
}

```

Figura 53. *CollectContainerStatus.java*

O programa de exemplo imprime o status do servidor de contêiner para cada contêiner. A seguir está um exemplo de saída:

```

<container name="Container-0" zoneName="DefaultZone" hostName="descartes.rchland.ibm.com"
serverName="ogserver">
  <shard type="Primary" partitionName="1"/>
  <shard type="Primary" partitionName="0"/>
  <shard type="Primary" partitionName="3"/>
  <shard type="Primary" partitionName="2"/>
</container>

```

- **Conectar-se a um servidor MBean do serviço de catálogo que é hospedado no WebSphere Application Server:**

O método para acessar programaticamente os MBeans no WebSphere Application Server é um pouco diferente de acessar os MBeans em uma configuração independente.

1. Crie e compile um programa Java para se conectar ao servidor MBean. A seguir há um exemplo do programa:

```

package com.ibm.websphere.sample.xs.admin;

import java.util.Set;

import javax.management.MBeanServerConnection;
import javax.management.ObjectName;
import javax.management.remote.JMXConnector;
import javax.management.remote.JMXConnectorFactory;
import javax.management.remote.JMXServiceURL;

/**
 * Collects the placement information from the catalog server running in a deployment manager for a given ObjectGrid.
 */
public final class CollectPlacementPlanWAS {
    private static String hostName = "localhost";

    private static int port = 9809;

    private static String objectGridName = "library";

    private static String mapSetName = "ms1";

    /**
     * Connects to the catalog service to retrieve placement information and prints it out.
     *
     * @param args
     * @throws Exception
     * If there is a problem connecting to the catalog service MBean server.
     */
    public static void main(String[] args) throws Exception {

        // connect to bootstrap port of the deployment manager
        String serviceURL = "service:jmx:iiop://" + hostName + ":" + port + "/jndi/JMXConnector";
        JMXServiceURL jmxUrl = new JMXServiceURL(serviceURL);
        JMXConnector jmxCon = JMXConnectorFactory.connect(jmxUrl);

        try {
            MBeanServerConnection catalogServerConnection = jmxCon.getMBeanServerConnection();

            Set placementSet = catalogServerConnection.queryNames(new ObjectName("com.ibm.websphere.objectgrid"
                + ".*:type=PlacementService"), null);

            ObjectName placementService = (ObjectName) placementSet.iterator().next();
            Object placementXML = catalogServerConnection.invoke(placementService,
                "listObjectGridPlacement", new Object[] {
                    objectGridName, mapSetName }, new String[] { String.class.getName(), String.class.getName() });
            System.out.println(placementXML);
        } finally {
            if(jmxCon != null) {
                jmxCon.close();
            }
        }
    }
}

```

Figura 54. *CollectPlacementPlan.java*

2. Execute o seguinte comando.

```

"$JAVA_HOME/bin/java" "$WAS_LOGGING" -Djava.security.auth.login.config="$app_server_root/properties/wsjaas_client.conf" \
-Djava.ext.dirs="$JAVA_HOME/jre/lib/ext:$WAS_EXT_DIRS:$WAS_HOME/plugins:$WAS_HOME/lib/wmq/java/lib" \
-Djava.naming.provider.url=<an_IIOP_URL_or_a_corbaloc_URL_to_your_application_server_machine_name> \
-Djava.naming.factory.initial=com.ibm.websphere.naming.WsnInitialContextFactory \
-Dserver.root="$WAS_HOME" "$CLIENTSAS" "$CLIENTSSL" $USER_INSTALL_PROP \
-classpath "$WAS_CLASSPATH":<list_of_your_application_jars_and_classes> \
<fully_qualified_class_name_to_run> <your_application_parameters>

```

Este comando supõe que o script *was\_root/bin/setupCmdLine.sh* foi executado para configurar as variáveis corretamente. Um exemplo do formato do valor da propriedade *java.naming.provider.url* é *corbaloc:iiop:1,0<host>:<port>/NameService*.

- **Conectar-se a um servidor MBean do serviço de catálogo com a segurança ativada:**

Para obter mais informações sobre como se conectar com o MBean do serviço de catálogo com a segurança ativada, consulte “Segurança do Java Management Extensions (JMX)” na página 507.

## **O que Fazer Depois**

Para obter mais exemplos sobre como exibir estatísticas e executar operações administrativas com MBeans, consulte a amostra do aplicativo **xsadmin**. É possível examinar o código de origem do aplicativo de amostra **xsadmin** no arquivo *wxs\_home/samples/xsadmin.jar* em uma instalação independente ou no arquivo *wxs\_home/xsadmin.jar* em uma instalação do WebSphere Application Server. Consulte o Amostra: Utilitário **xsadmin** para obter mais informações sobre as operações que podem ser concluídas com o aplicativo de amostra **xsAdmin**.

Também é possível localizar mais informações sobre os MBeans no pacote `com.ibm.websphere.objectgrid.management`.



---

## Capítulo 8. Monitoramento



É possível usar o console de monitoramento, APIs, MBeans, logs e utilitários incluídos para monitorar o desempenho do seu ambiente de aplicativo.

---

### Visão Geral de Estatísticas

As estatísticas no WebSphere eXtreme Scale são construídas a partir de uma árvore de estatísticas internas. A API StatsAccessor, módulos da Performance Monitoring Infrastructure (PMI) e a API MBean são construídos a partir da árvore interna.

A figura a seguir mostra a configuração geral das estatísticas para o WebSphere eXtreme Scale.

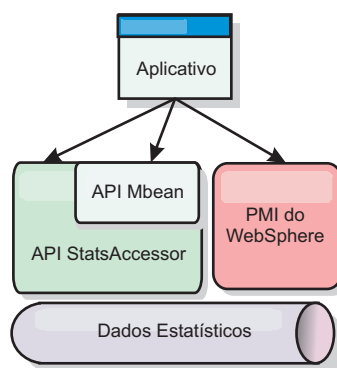


Figura 55. Visão Geral de Estatísticas

Cada uma das APIs oferece uma visualização da árvore de estatísticas, mas são utilizadas por diferentes motivos:

- **API de Estatísticas:** A API de Estatísticas permite que os desenvolvedores tenham acesso direto às estatísticas, o que permite soluções de integração de estatísticas flexíveis e customizáveis, como MBeans customizados ou a criação de log.
- **API do MBean:** A API do MBean é um mecanismo baseado em especificação para monitoramento. A API MBean usa a API Statistics e executa localmente para o servidor do JVM (Java Virtual Machine). As estruturas de API e MBean são projetadas para integração imediata com utilitários de outros fornecedores. Use a API MBean quando estiver executando uma grade de objeto distribuído.
- **WebSphere Application Server Módulos Performance Monitoring Infrastructure (PMI) :** Utilize a PMI se estiver executando o WebSphere eXtreme Scale no WebSphere Application Server. Estes módulos fornecem uma visualização da árvore de estatísticas internas.

#### API de Estatísticas

Muito semelhante a um mapa de árvore, há um caminho e uma chave correspondentes para recuperar um módulo específico, ou neste caso, nível de granularidade ou agregação. Por exemplo, assume que sempre há um nó-raiz arbitrário na árvore e que as estatísticas estão sendo reunidas para um mapa

nomeado como "payroll", pertencendo a um ObjectGrid nomeado como "accounting". Por exemplo, para acessar o módulo para um nível de agregação ou granularidade do mapa, você poderia passar uma String[] dos caminhos. Neste caso, isto seria igual a String[] {root, "accounting", "payroll"}, já que cada String representaria o caminho do nó. A vantagem desta estrutura é que um usuário pode especificar a matriz para qualquer nó no caminho e obter o nível de agregação para tal nó. Portanto, passar String[] {root, "accounting"} forneceria a você as estatísticas do mapa, mas para a grade inteira de "accounting." Isto deixa o usuário tanto com a habilidade de especificar tipos de estatísticas a serem monitorados quanto em que nível de agregação é necessário para o aplicativo.

## **Módulos PMI do WebSphere Application Server**

O WebSphere eXtreme Scale inclui módulos de estatísticas para uso com a PMI do WebSphere Application Server. Quando um perfil do WebSphere Application Server é aumentado com o WebSphere eXtreme Scale, os scripts de aumento integram-se automaticamente aos módulos do WebSphere eXtreme Scale nos arquivos de configuração do WebSphere Application Server. Com a PMI, é possível ativar e desativar módulos de estatísticas, automaticamente agregar estatísticas em várias granularidades e, até mesmo, criar gráficos dos dados utilizando o Tivoli Performance Viewer integrado. Consulte "Monitorando com a PMI do WebSphere Application Server" na página 455 para obter mais informações.

## **Integração de Produtos de Fornecedores com Beans Gerenciados (MBean)**

As APIs do eXtreme Scale e os Beans Gerenciados são projetados para permitir a fácil integração com aplicativos de monitoramento de terceiros. JConsole ou MC4J são alguns exemplos de consoles Java Management Extensions (JMX) leves que podem ser utilizados para analisar informações sobre uma topologia do eXtreme Scale. Também é possível utilizar as APIs programáticas para criar implementações do adaptador para realizar uma captura instantânea ou controlar o desempenho do eXtreme Scale. O WebSphere eXtreme Scale inclui um aplicativo de monitoramento de amostra que permite recursos de monitoramento prontos para utilização e que pode ser utilizado como um modelo para criação de utilitário de monitoramento customizados mais avançados.

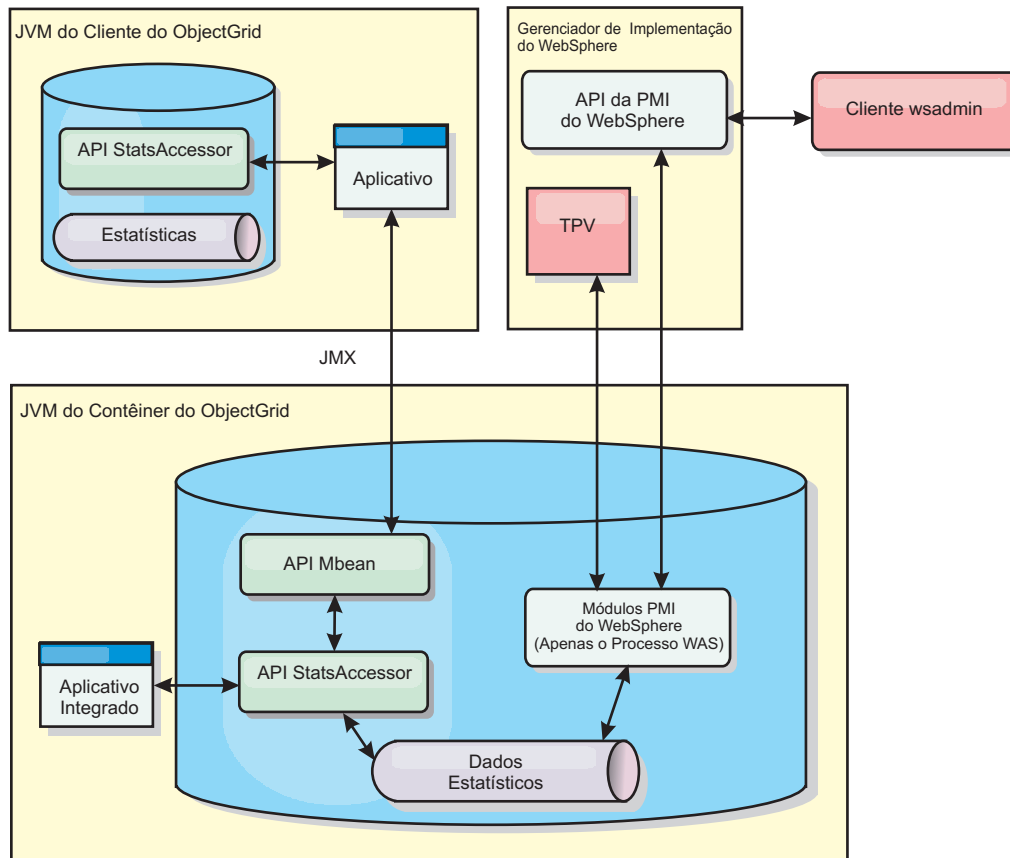


Figura 56. Visão Geral do MBean

Consulte Amostra: Utilitário **xsadmin** para obter mais informações. Para obter informações adicionais sobre a integração com aplicativos de um fornecedor específico, consulte os tópicos a seguir:

- Monitoramento do eXtreme Scale com o IBM Tivoli Monitoring Agent
- “Monitorando o eXtreme Scale com o Hyperic HQ” na página 477
- “Monitorando Aplicativos eXtreme Scale com o CA Wily Introscope” na página 474

## Monitorando com o Console da Web

Com o console da Web, é possível registrar no gráfico as estatísticas atuais e de histórico. Este console fornece alguns gráficos pré-configurados para visões gerais de alto nível e tem uma página de relatórios customizada que pode ser usada para construir gráficos a partir das estatísticas disponíveis. É possível usar os recursos gráficos no console de monitoramento do WebSphere eXtreme Scale para visualizar o desempenho geral das grades de dados em seu ambiente.

### Iniciando e Efetuando Login no Console da Web

Inicie o servidor de console executando o comando **startConsoleServer** e efetuando login no servidor com o ID de usuário e a senha padrão.

#### Antes de Iniciar

- **Requisitos do Sistema**
  - Use um sistema AIX, Linux ou Windows para executar o console da web.

- Instale um servidor WebSphere eXtreme Scale independente no sistema que hospeda o servidor do console. Consulte “Instalando WebSphere eXtreme Scale ou WebSphere eXtreme Scale Client Independente” na página 191 para obter mais informações.
- O sistema do servidor do console deve ser capaz de se conectar a seu serviço de catálogo. O serviço de catálogo também deve ser capaz de se conectar de volta ao servidor do console da web.

- **Requisitos do Navegador da Web**

Use um dos navegadores a seguir com o console da Web:

- Mozilla Firefox, versão 3.5.x e posterior
- Mozilla Firefox, versão 3.6.x e posterior
- Microsoft Internet Explorer, versão 7 ou 8

## Procedimento

1. Opcional: Se desejar executar servidor de console em uma porta diferente da porta padrão, edite o arquivo `wxs_install_root/ObjectGrid/console/config/zero.config`. A porta padrão para o servidor do console é 7080 para HTTP e 7443 para HTTPS. As propriedades a seguir podem ser editadas para alterar os valores padrão:

```
/config/http/port = 7080
/config/https/port = 7443
```

Se esses valores forem editados após o servidor do console já estar iniciado, reinicie o servidor para usar os novos números de porta.

2. Inicie o servidor do console. O script `startConsoleServer.bat|sh` para iniciar o servidor do console está no diretório `wxs_install_root/ObjectGrid/bin` de sua instalação.
3. Efetue logon no console.
  - a. No seu navegador da web, acesse `https://your.console.host:7443`, substituindo `your.console.host` pelo nome do host do servidor na qual o console foi instalado.
  - b. Efetue logon no console.
    - **ID do usuário:** admin
    - **Senha:** admin
 A página de boas-vindas do console é exibida.
4. Edite a configuração do console. Clique em **Definições > Configuração** para revisar a configuração do console. A configuração do console inclui informações como:
  - Cadeia de rastreamento para o cliente do WebSphere eXtreme Scale, como `*=all=disabled`
  - O nome e a senha do Administrador
  - O endereço de e-mail do Administrador

## O que Fazer Depois

- Conecte seus servidores de catálogos no console da web para iniciar o rastreamento de estatísticas. Consulte o “Conectando o Console da Web nos Servidores de Catálogos” na página 437 para obter informações adicionais.
- Se precisar parar o servidor de console da Web, execute o script `stopConsoleServer.bat|sh`. Esse script está no diretório `wxs_install_root/ObjectGrid/bin` de sua instalação.

## Conectando o Console da Web nos Servidores de Catálogos

Para visualizar as estatísticas no console da Web, primeiro você deve se conectar com os servidores de catálogo que deseja monitorar. Etapas adicionais serão necessárias se a segurança estiver ativada nos seus servidores de catálogos.

### Antes de Iniciar

- O servidor do console da web deve estar em execução. Consulte o “Iniciando e Efetuando Login no Console da Web” na página 435 para obter informações adicionais.
- Pelo menos um servidor de catálogos deve estar em execução ao qual deseja se conectar. Consulte “Iniciando um Serviço de Catálogo Independente” na página 387 para obter mais informações.


### Procedimento


1. Se o Secure Sockets Layer (SSL) estiver ativado nos seus servidores de catálogos, você deverá configurar um keystore, um armazenamento confiável e o arquivo de propriedades do cliente. Ative o SSL para um servidor de catálogos ao configurar o atributo `transportType` como SSL Necessário no Arquivo de Propriedades do Servidor.
  - a. Configure um keystore e um armazenamento confiável e, em seguida, troque ou efetue uma importação cruzada dos certificados públicos. Por exemplo, você pode copiar o armazenamento confiável e o keystore para um local no servidor que está executando o console da Web.
  - b. Edite o arquivo de propriedades do cliente no servidor do console da web para incluir as propriedades na configuração SSL. Por exemplo, você pode editar o arquivo `wxs_install_root/ObjectGridProperties/sampleclient.properties`. As seguintes propriedades são necessárias para conexões SSL de saída a partir do console da web:

```
#-----  
# SSL Configuration  
#  
# - contextProvider      (IBMJSSE2, IBMJSSE, IBMJSSEFIPS, etc.)  
# - protocol             (SSL, SSLv2, SSLv3, TLS, TLSv1, etc.)  
# - keyStoreType         (JKS, JCEK, PKCS12, etc.)  
# - trustStoreType       (JKS, JCEK, PKCS12, etc.)  
# - keyStore             (fully qualified path to key store file)  
# - trustStore           (fully qualified path to trust store file)  
# - alias                (string specifying ssl certificate alias to use from keyStore)  
# - keyStorePassword     (string specifying password to the key store - encoded or not)  
# - trustStorePassword  (string specifying password to the trust store - encoded or not)  
#  
# Uncomment these properties to set the SSL configuration.  
#-----  
#alias=clientprivate  
#contextProvider=IBMJSSE  
#protocol=SSL  
#keyStoreType=JKS  
#keyStore=etc/test/security/client.private  
#keyStorePassword={xor}PDM20jErLyg\  
#trustStoreType=JKS  
#trustStore=etc/test/security/server.public  
#trustStorePassword={xor}Ly09MzY8
```

**Importante:** Windows Se estiver usando Windows, você deve escapar qualquer caractere de barra invertida ( \ ) no caminho. Por exemplo, se desejar usar o caminho `C:\opt\ibm`, digite `C:\\opt\\ibm` no arquivo de propriedades.

2. Estabeleça e mantenha conexões com servidores de catálogos que você deseja monitorar. Repita as seguintes etapas para incluir cada servidor de catálogos na configuração.
  - a. Clique em **Configurações > Servidores de Catálogos do eXtreme Scale** .
  - b. Inclua um novo servidor de catálogos.

- 1) Clique no ícone Incluir (  ) para registrar um servidor de catálogos existente.
- 2) Forneça informações, como o nome do host e a porta do listener. Consulte “Planejamento para Portas de Rede” na página 62 para obter mais informações sobre a configuração da porta e os padrões.
- 3) Clique em **OK**.
- 4) Verifique se o servidor de catálogos foi incluído na árvore de navegação.
3. Agrupe os servidores de catálogos que foram criados em um domínio de serviço de catálogo. Você deve criar um domínio de serviço de catálogo quando a segurança estiver ativada nos seus servidores de catálogos porque as configurações de segurança são definidas no domínio de serviço de catálogo.
- Clique na página **Configurações > Domínios do eXtreme Scale** .
  - Inclua um novo domínio do serviço de catálogo.

- 1) Clique no ícone Incluir (  ) para registrar um domínio de serviço de catálogo. Insira um nome para o domínio de serviço de catálogo.
- 2) Depois de criar o domínio de serviço de catálogo, as propriedades poderão ser editadas. As propriedades do domínio de serviço de catálogo são:

**Nome** Indica o nome do host do domínio, conforme designado pelo administrador.

#### **Servidores de catálogos**

Lista um ou mais servidores de catálogos que pertencem ao domínio selecionado. É possível incluir os servidores de catálogos criados na etapa anterior.

#### **Classe de gerador**

Exibe o nome da classe que implementa a interface `CredentialGenerator`. This class is used to get credentials for clients. Se você especificar um valor neste campo, o valor substituirá a propriedade **`credentialGeneratorClass`** no arquivo `client.properties`.

#### **Propriedades do gerador**

Especifica as propriedades para a classe de implementação `CredentialGenerator`. As propriedades são configuradas para o objeto com o método `setProperty(String)`. O valor `credentialGeneratorProps` é usado apenas se o valor da propriedade `credentialGeneratorClass` não for nulo. Se você especificar um valor neste campo, o valor substituirá a propriedade **`credentialGeneratorProps`** no arquivo `client.properties`.

#### **Caminho de Propriedades do Cliente eXtreme Scale**

Especifica o caminho para o arquivo de propriedades do cliente que foi editado para incluir propriedades SSL em uma etapa anterior. Por exemplo, você pode indicar o arquivo `c:\ObjectGridProperties\samplclient.properties`. Se desejar que o console pare de tentar usar conexões SSL, o valor poderá ser excluído neste campo. Depois de configurar o caminho, o console usará uma conexão não segura.

- 3) Clique em **OK**.
- 4) Verifique se o domínio foi incluído na árvore de navegação.

Para visualizar informações sobre um domínio de serviço de catálogo existente, clique no nome do domínio de serviço de catálogo na árvore de navegação na página **Configurações > Domínios do eXtreme Scale**.

4. Visualize o status da conexão. O campo **Domínio Atual** indica o nome do domínio do serviço de catálogo que está atualmente sendo usado para exibir informações no console da web. O status da conexão é exibido ao lado do nome do domínio de serviço de catálogo.

## Visualizando Estatísticas com o Console da Web

É possível monitorar estatísticas e outras informações de desempenho com o console da web.

### Antes de Iniciar

Antes de poder visualizar as estatísticas com o console da web, as seguintes tarefas deverão ser concluídas:

1. Inicie o servidor do console da web. Consulte o “Iniciando e Efetuando Login no Console da Web” na página 435 para obter informações adicionais.
2. Conecte os servidores de catálogo ao servidor do console da web. Consulte o “Conectando o Console da Web nos Servidores de Catálogos” na página 437 para obter informações adicionais.
3. Execute as grades de dados e aplicativos ativos dentro dos servidores que são gerenciados pelo domínio de serviço de catálogo.

### Sobre Esta Tarefa

Depois de criar suas grades de dados e configurar seus aplicativos para usarem as grades de dados, permita que passe algum tempo para que as estatísticas sejam disponibilizadas. Por exemplo, com uma grade de dados de cache dinâmico, as estatísticas não estarão disponíveis até que um WebSphere Application Server, que executa um cache dinâmico, se conecte ao cache dinâmico. No geral, aguarde um minuto após uma mudança de configuração maior para ver as mudanças em suas estatísticas.

**Dica:** Para visualizar informações mais específicas sobre qualquer ponto de dados em um gráfico, é possível mover o ponteiro do mouse sobre o ponto de dados.

### Procedimento

- Para visualizar as estatísticas atuais do servidor, clique em **Monitorar > Visão Geral do Servidor**.
- Para visualizar o desempenho de todas as suas grades de dados, clique em **Monitorar > Visão geral do domínio da grade de dados**.
- Para visualizar as grades de dados individuais, clique em **Monitorar > Visão geral da grade de dados > data\_grid\_name**. Esta página mostra um resumo que inclui o número de entradas de cache, o tempo médio de transação e o rendimento médio.
- Para visualizar detalhes adicionais sobre uma grade de dados específica, clique em **Monitorar > Detalhes da grade de dados**. Uma árvore é exibida com todas as grades de dados em sua configuração. É possível fazer uma pesquisa detalhada em uma grade de dados específica para visualizar os mapas que



fazem parte dessa grade de dados. É possível clicar no nome de uma grade de dados ou em um mapa para obter informações adicionais.

- Para escolher quais estatísticas você deseja que seu relatório customizado contivesse, clique em **Monitor > Relatórios Customizados**.

Use esta visualização para construir gráficos de dados detalhados das várias estatísticas. Use a árvore para explorar os servidores e as grades de dados disponíveis e suas estatísticas associadas. Um menu é aberto quando você clica ou pressiona Enter em um nó que faz referência a dados que podem ser representados graficamente. Crie um novo gráfico contendo as estatísticas ou inclua as estatísticas em um gráfico existente com estatísticas compatíveis. Consulte o “Monitorando com Relatórios Customizados” na página 445 para obter informações adicionais.

## Estatísticas do Console da Web

Dependendo da visualização que você está utilizando no console da Web, é possível visualizar estatísticas diferentes sobre sua configuração. Essas estatísticas incluem a memória usada, as principais grades de dados usadas e o número de entradas de cache.

- “Visão Geral do Domínio da Grade de Dados”
- “Visão Geral da Grade de Dados” na página 441
- “Detalhes da grade de dados” na página 441
- “Visão Geral do Servidor” na página 442
- “Relatórios Customizados: Estatísticas do Domínio de Serviço de Catálogo” na página 442
  - “Relatórios Customizados: Estatísticas do Servidor de Contêiner” na página 442
  - “Relatórios customizados : Grade de dados estatísticos” na página 444
  - “Relatórios Customizados: Estatísticas do Mapa” na página 444

## Visão Geral do Domínio da Grade de Dados

As estatísticas da visão geral de domínio da grade de dados são exibidas na página **Monitor > Visão Geral de Domínio da Grade de Dados**. Clique em uma das seguintes guias para obter mais informações sobre o domínio da grade de dados:

### Guia Capacidade Usada

No gráfico **Distribuição da Capacidade Usada da Grade de Dados Atual**, uma imagem do **Conjunto Total** e da **Maior Capacidade Usada pelo Consumidores** são exibidos. Apenas as 25 principais grades de dados são exibidas. Na **Capacidade Usada ao Longo do Tempo**, o número de bytes que são consumidos pela grade de dados é exibida.

### Guia Rendimento Médio

O gráfico **5 Grades de Dados Mais Ativas por Tempo Médio de Transação em Milissegundos** contém uma lista dos cinco principais cache de dados, organizados por tempo médio de transação. O gráfico **Rendimento Médio Ao Longo do Tempo** exibe os rendimentos médio, máximo e mínimo durante a última hora, dia e semana.

### Guia Tempo Médio da Transação

O gráfico **5 Grades de Dados Mais Lentas** exibe dados sobre as grades de dados mais lentas. O gráfico **Tempo Médio da Transação ao Longo do Tempo** exibe os tempos de transação médio, máximo e mínimo durante a última hora, dia e semana.

## Visão Geral da Grade de Dados

Para visualizar as estatísticas para uma grade de dados individual, clique em **Monitor > Visão Geral da Grade de Dados > *data\_grid\_name***.

### Resumo atual dos últimos 30 segundos

Exibe o número atual de entradas de cache, o tempo médio de transação, o rendimento médio e a taxa de acertos do cache para a grade de dados selecionada.

### Guia Capacidade Usada

O gráfico **Resumo Atual Sobre os Últimos 30 Segundos** exibe o número de entradas de cache e a capacidade em bytes usada durante um intervalo de tempo especificado.

### Guia Uso do Cache

O gráfico **Uso de Cache** ajuda a visualizar o número de consultas bem-sucedidas para o cache e exibe as tentativas de cache, acertos de cache e a taxa de acertos do cache durante um intervalo de tempo especificado.

### Guia Rendimento Médio

O gráfico **Rendimento Médio vs. Tempo de Transação Médio** exibe o tempo e o rendimento da transação sobre um intervalo de tempo especificado.

## Detalhes da grade de dados

As estatísticas da grade de dados são exibidas na página **Monitor > Detalhes da Grade de Dados**. É possível consultar os dados para uma grade selecionada e os mapas que estiverem dentro de uma grade.

### Resumo atual dos últimos 30 segundos

Exibe a capacidade usada atual, o número de entradas de cache, o rendimento médio e o tempo médio de transação para a grade de dados selecionada.

### Distribuição da Capacidade Usada do Mapa da Grade de Objeto do eXtreme Scale Atual

Visualize um conjunto total, que inclui a capacidade por zona e a capacidade total em cada zona. Apenas os 25 principais mapas de ObjectGrid são exibidos. Também é possível visualizar os principais clientes da capacidade usada em cada mapa.

### Distribuição da Capacidade Usada da Zona Atual

Visualize um conjunto total, que inclui o conjunto total e a maior capacidade usada pelos consumidores na zona da grade de dados selecionada. Também é possível visualizar a maior capacidade usada pelos consumidores em cada zona.

### Estatísticas do Mapa:

#### Resumo atual dos últimos 30 segundos

Exibe a capacidade usada atual, o número de entradas de cache, o rendimento médio e o tempo médio da transação para o mapa selecionado.

#### Distribuição da Capacidade Usada da Partição Atual

Visualize uma partição, que inclui o conjunto total e clientes de maior capacidade de uso. Apenas as 25 principais partições são exibidas. Também é possível visualizar a maior capacidade usada pelos consumidores em cada partição.

## Visão Geral do Servidor

As estatísticas do servidor são exibidas na página **Monitor > Visão Geral do Servidor**.

### Distribuição da Memória Atual Usada pelo Servidor

Este gráfico é composto por duas visualizações. **Conjunto Total** exibe a quantidade atual de memória usada (real) no tempo de execução do servidor. **Maior Uso de Memória pelos Consumidores** divide a memória usada por servidor, no entanto, apenas os 25 principais servidores que usam a maior parte da memória são exibidos.

### Total de Memória com o Passar do Tempo

Exibe o uso de memória real no tempo de execução do servidor.

### Memória Usada com o Passar do Tempo

Exibe a quantidade de memória usada no tempo de execução do servidor.

## Relatórios Customizados: Estatísticas do Domínio de Serviço de Catálogo

É possível visualizar as estatísticas de domínio do serviço de catálogo ao criar um relatório customizado. Clique em **Monitorar > Relatórios Customizados**.

### Tempo Médio da Transação (ms)

Exibe o tempo médio necessário para concluir uma transação neste domínio.

### Rendimento Médio da Transação (trans/seg)

Exibe o número médio de transações por segundo neste domínio.

### Tempo Máximo da Transação (ms)

Exibe o tempo gasto pela transação que *mais* consome tempo neste domínio.

### Tempo Mínimo da Transação (ms)

Exibe o tempo gasto pela transação que *menos* consome tempo neste domínio.

### Tempo Total da Transação (ms)

Exibe o tempo total gasto em transações neste domínio, desde o momento em que o domínio foi inicializado.

## Relatórios Customizados: Estatísticas do Servidor de Contêiner

É possível visualizar as estatísticas do servidor de contêiner por meio da criação de um relatório customizado. Clique em **Monitorar > Relatórios Customizados**.

### Tempo Médio da Transação (ms)

Exibe o tempo médio necessário para concluir uma transação para este servidor de catálogos.

### Rendimento Médio da Transação (trans/seg)

Exibe o número médio de transações por segundo para este servidor de catálogos.

### Tempo Máximo da Transação (ms)

Exibe o tempo gasto pela transação que *mais* consome tempo para este servidor de catálogos.

**Tempo Mínimo da Transação (ms)**

Exibe o tempo gasto pela transação que *menos* consome tempo para este servidor de catálogos.

**Tempo Total da Transação (ms)**

Exibe o tempo total gasto em transações para este servidor de catálogos, desde o momento em que este servidor de catálogos foi inicializado.

**Total de Entradas em Cache**

Exibe o número atual de objetos em cache nas grades examinadas por este servidor de catálogos.

**Taxa de ocorrência (porcentagem)**

Exibe a taxa de ocorrência (taxa de acertos) para a grade de dados selecionada. Uma taxa alta de ocorrência é desejável. A taxa de ocorrência indica como a grade está ajudando a evitar o acesso ao armazenamento persistente.

**Bytes Usados**

Exibe o consumo de memória por este mapa. As estatísticas de bytes usadas são precisas apenas quando você estiver usando objetos simples ou o modo de cópia COPY\_TO\_BYTES.

**Mínimo de Bytes Usados**

Exibe o ponto baixo no consumo de memória por este serviço de catálogo e seus mapas. As estatísticas de bytes usadas são precisas apenas quando você estiver usando objetos simples ou o modo de cópia COPY\_TO\_BYTES.

**Máximo de Bytes Usados**

Exibe o ponto alto no consumo de memória por este serviço de catálogo e seus mapas. As estatísticas de bytes usadas são precisas apenas quando você estiver usando objetos simples ou o modo de cópia COPY\_TO\_BYTES.

**Número Total de Ocorrências**

Exibe o número total de vezes que os dados solicitados foram localizados no mapa, evitando a necessidade de acessar o armazenamento persistente.

**Número Total de Gets**

Exibe o número total de vezes que o mapa teve que acessar o armazenamento persistente para obter dados.

**Heap Grátis (MB)**

Exibe a quantia real de heap disponível para a JVM que está sendo usada pelo servidor de catálogos.

**Total de Heap**

Exibe a quantia de heap disponível para a JVM que está sendo usada por este servidor de catálogos.

**Número de Processadores Disponíveis**

Exibe o número de processadores que está disponível para este serviço de catálogo e os seus mapas. Para obter maior estabilidade, execute os seus servidores a um carregamento de processador de 60% e o JVM heaps a um carregamento de heap de 60%. Os picos podem, então, conduzir o uso do processador para 80–90%, mas não execute seus servidores regularmente em níveis superiores a estes

**Tamanho Máximo de Heap (MB)**

Exibe a quantia máxima de heap disponível para a JVM que está sendo usada por este servidor de catálogos.

### **Memória Usada**

Exibe a memória usada na JVM que está sendo usada por este servidor de catálogos.

### **Relatórios customizados : Grade de dados estatísticos**

É possível visualizar as estatísticas da grade de dados criando um relatório customizado. Clique em **Monitorar > Relatórios Customizados**.

#### **Tempo Médio da Transação (ms)**

Exibe o tempo médio necessário para concluir transações envolvendo esta grade.

#### **Rendimento Médio da Transação (trans/seg)**

Exibe o número médio de transações por segundo concluídas por esta grade.

#### **Tempo Máximo da Transação (ms)**

Exibe o tempo gasto pela transação que *mais* consome tempo concluída por esta grade.

#### **Tempo Mínimo da Transação (ms)**

Exibe o tempo gasto pela transação que *menos* consome tempo concluída por esta grade.

#### **Tempo Total da Transação (ms)**

Exibe o período total de tempo de processamento de transações para esta grade.

### **Relatórios Customizados: Estatísticas do Mapa**

É possível visualizar as estatísticas do mapa ao criar um relatório customizado. Clique em **Monitorar > Relatórios Customizados**.

#### **Total de Entradas em Cache**

Exibe o número atual de objetos em cache neste mapa.

#### **Taxa de Ocorrência (porcentagem)**

Exibe a taxa de ocorrência (taxa de acertos) para o mapa selecionado. Uma taxa alta de ocorrência é desejável. A taxa de ocorrência indica como o mapa está ajudando a evitar o acesso ao armazenamento persistente.

#### **Bytes Usados**

Exibe o consumo de memória por este mapa. As estatísticas de bytes usadas são precisas apenas quando você estiver usando objetos simples ou o modo de cópia COPY\_TO\_BYTES.

#### **Mínimo de Bytes Usados**

Exibe o consumo mínimo (em Bytes) para este mapa. As estatísticas de bytes usadas são precisas apenas quando você estiver usando objetos simples ou o modo de cópia COPY\_TO\_BYTES.

#### **Máximo de Bytes Usados**

Exibe o consumo máximo (em Bytes) para este mapa. As estatísticas de bytes usadas são precisas apenas quando você estiver usando objetos simples ou o modo de cópia COPY\_TO\_BYTES.

#### **Número Total de Ocorrências**

Exibe o número total de vezes que os dados solicitados foram localizados no mapa, evitando a necessidade de acessar o armazenamento persistente.

**Número Total de Gets**

Exibe o número total de vezes que o mapa teve que acessar o armazenamento persistente para obter dados.

**Heap Grátis (MB)**

Exibe a quantia atual de heap disponível para este mapa, na JVM que está sendo usada pelo servidor de catálogos.

**Total de Heap (MB)**

Exibe a quantia total de heap disponível para este mapa, na JVM que está sendo usada pelo servidor de catálogos. Para obter maior estabilidade, execute os seus servidores a um carregamento de processador de 60% e o JVM heaps a um carregamento de heap de 60%. Os picos podem, então, conduzir o uso do processador para 80–90%, mas não execute seus servidores regularmente em níveis superiores a estes

**Número de Processadores Disponíveis**

Exibe o número de processadores disponíveis para este mapa. Para obter maior estabilidade, execute os seus servidores a um carregamento de processador de 60% e o JVM heaps a um carregamento de heap de 60%. Os picos podem, então, conduzir o uso do processador para 80–90%, mas não execute seus servidores regularmente em níveis superiores a estes

**Tamanho Máximo de Heap (MB)**

Exibe a quantia máxima de heap disponível para este mapa, na JVM que está sendo usada pelo servidor de catálogos.

**Memória Usada (MB)**

Exibe a quantia usada de memória neste mapa.

## Monitorando com Relatórios Customizados

É possível construir relatórios customizados para salvar vários gráficos que contêm estatísticas sobre os domínios de serviço de catálogo, grades de dados e servidores de contêiner em seu ambiente. É possível salvar os relatórios customizados e carregá-los para visualizar novamente mais tarde.

### Antes de Iniciar


Antes de poder visualizar as estatísticas com o console da web, as seguintes tarefas deverão ser concluídas:

1. Inicie o servidor do console da web. Consulte o “Iniciando e Efetuando Login no Console da Web” na página 435 para obter informações adicionais.
2. Conecte os servidores de catálogo ao servidor do console da web. Consulte o “Conectando o Console da Web nos Servidores de Catálogos” na página 437 para obter informações adicionais.
3. Execute as grades de dados e aplicativos ativos dentro dos servidores que são gerenciados pelo domínio de serviço de catálogo.

### Procedimento

- Crie um relatório customizado.
  1. Clique em **Monitorar > Relatórios Customizados**. Uma lista de domínios do eXtreme Scale que foram definidos é mostrada em um formato em árvore. É possível expandir cada um desses domínios para exibir as estatísticas disponíveis que podem ser incluídas no relatório customizado.

2. Inclua gráficos com as estatísticas que deseja rastrear. As estatísticas

disponíveis são indicadas pelo ícone de gráfico (  ). Clique em uma das estatísticas que deseja rastrear. Escolha **Incluir no novo gráfico** ou **Incluir em um gráfico existente**. Dependendo de sua seleção, a estatística selecionada é exibida em uma nova guia de gráfico ou no gráfico existente selecionado. Uma métrica poderá ser incluída em um gráfico existente apenas se as métricas já estiverem no gráfico e a nova métrica usar as mesmas unidades.

- Salve um relatório customizado. Salvar o relatório customizado salva as estatísticas em todas as guias que foram criadas. Para salvar o relatório, clique em **Salvar**.
- Carregue um relatório customizado. Clique em **Carregar** e escolha o relatório customizado salvo que deseja visualizar.

---

## Monitorando com Arquivos CSV

É possível ativar os dados de monitoramento para serem gravados em arquivos CSV. Esses arquivos CSV podem conter informações sobre a instância da JVM, mapa ou do ObjectGrid.

### Sobre Esta Tarefa

Ao permitir que os dados de monitoramento sejam gravados em arquivos CSV, é possível fazer download e analisar dados históricos para um servidor de contêiner de catálogos individual. Os dados são coletados quando o servidor é iniciado com suas propriedades que ativam os arquivos CSV. Em seguida, é possível fazer download dos arquivos CSV a qualquer momento e usar esses arquivos como desejar.

### Procedimento

1. Atualize o arquivo de propriedades do servidor com as seguintes propriedades que estão relacionadas à ativação dos arquivos CSV.

```
parameter=default value
jvmStatsLoggingEnabled=true
maxJVMStatsFiles=5
maxJVMStatsFileSize=100
jvmStatsFileName=jvmstats
jvmStatsWriteRate=10
```

```
mapStatsLoggingEnabled=true
maxMapStatsFiles=5
maxMapStatsFileSize=100
mapStatsFileName=mapstats
mapStatsWriteRate=10
```

```
ogStatsLoggingEnabled=true
maxOGStatsFiles=5
maxOGStatsFileSize=100
ogStatsFileName=ogstats
ogStatsWriteRate=10
```

Para obter mais informações sobre essas propriedades, consulte Arquivo de Propriedades do Servidor.

2. Reinicie o servidor para selecionar as mudanças para o arquivo de propriedades do servidor.
3. Faça download do arquivo CSV. O arquivo CSV é gravado no diretório `server_name/logs`.



4. Importe o arquivo CSV no programa que está usando para processar os dados, como uma planilha.

## O que Fazer Depois

Para obter mais informações sobre os dados que estão contidos nos arquivos CSV, consulte “Definições de Estatísticas do Arquivo CSV”.

## Definições de Estatísticas do Arquivo CSV

Os arquivos CSV que podem ser transferidos por download para um servidor incluem estatísticas que podem ser usadas para construir gráficos históricos ou outras informações.

### Log de Estatísticas da Java virtual machine (JVM)

**TimeStamp**

Especifica a data e a hora da captura instantânea das estatísticas que foi feita para a JVM.

**ServerName**

Especifica o nome do servidor da JVM.

**Hostname**

Especifica o nome do host da JVM.

**DomainName**

Especifica o domínio do serviço de catálogo ao qual a JVM pertence.

**FreeMemory**

Especifica o número de bytes disponíveis para a JVM.

**MaxMemory**

Especifica o número máximo de bytes que podem ser alocados para a JVM.

**TotalMemory**

Exibe o uso de memória real no tempo de execução do servidor.

**AvailProcs**

Exibe o número de processadores que está disponível para este serviço de catálogo e os seus mapas. Para obter maior estabilidade, execute os seus servidores a um carregamento de processador de 60% e o JVM heaps a um carregamento de heap de 60%. Os picos podem, então, conduzir o uso do processador para 80–90%, mas não execute seus servidores regularmente em níveis superiores a estes

### Log de Estatísticas do Mapa

**TimeStamp**

Especifica a data e a hora da captura instantânea de estatísticas que foi feita para o mapa.

**MapName**

Especifica o nome do mapa.

**OgName**

Especifica o nome da grade de dados à qual esse mapa pertence.

**PartitionId**

Especifica o ID da partição.

**MapSetName**

Especifica o conjunto de mapas ao qual esse mapa pertence.

**HitRate**

Exibe a taxa de ocorrência (taxa de acertos) para o mapa selecionado. Uma taxa alta de ocorrência é desejável. A taxa de ocorrência indica modo com que a grade de dados está ajudando a evitar o acesso ao armazenamento persistente.

**Count** Indica uma contagem de amostras de dados que foram reunidas desde que o servidor foi iniciado. Por exemplo, um valor de 100 indica que a entrada é a 100ª entrada de amostra que foi reunida desde que o servidor foi iniciado.

**TotalGetCount**

Exibe o número total de vezes que o mapa teve que acessar o armazenamento persistente para obter dados.

**TotalHitCount**

Exibe o número total de vezes que os dados solicitados foram localizados no mapa, evitando a necessidade de acessar o armazenamento persistente.

**StartTime**

Especifica o horário em que os contadores iniciaram desde a última chamada de reconfiguração. As reconfigurações ocorrem quando o servidor é iniciado ou reiniciado.

**LastCount**

Especifica a quantidade de tempo desde que a última amostra de dados foi obtida.

**LastTotalGetCount**

Indica o número total atual de operações get a partir do cache menos o número de operações get no período de tempo anterior.

**LastTotalHitCount**

Indica o número total atual de ocorrências do cache menos o número de ocorrências no período de tempo anterior.

**UsedBytes**

Exibe o consumo de memória por este mapa. As estatísticas de bytes usadas são precisas apenas quando você estiver usando objetos simples ou o modo de cópia COPY\_TO\_BYTES.

**MinUsedBytes**

Exibe o ponto baixo no consumo de memória por este serviço de catálogo e seus mapas. As estatísticas de bytes usadas são precisas apenas quando você estiver usando objetos simples ou o modo de cópia COPY\_TO\_BYTES.

**MaxUsedBytes**

Exibe o ponto alto no consumo de memória por este serviço de catálogo e seus mapas. As estatísticas de bytes usadas são precisas apenas quando você estiver usando objetos simples ou o modo de cópia COPY\_TO\_BYTES.

**LastUsedBytes**

Indica o valor UsedBytes atual menos o valor UsedBytes do período de coleção de estatísticas anterior.

**SampleLen**

Indica a duração, em milissegundos, do tempo em que os dados foram amostrados.

## Los de Estatísticas ObjectGrid

**Count** Indica uma contagem de amostras de dados que foram reunidas desde que o servidor foi iniciado. Por exemplo, um valor de 100 indica que a entrada é a 100ª entrada de amostra que foi reunida desde que o servidor foi iniciado.

**TimeStamp**

Especifica a data e a hora em que a captura instantânea de estatísticas foi feita para a grade de dados.

**OgName**

Especifica o nome da grade de dados.

**PartitionId**

Especifica o ID partição.

**Hostname**

Especifica o nome do host.

**DomainName**

Especifica o domínio do serviço de catálogo ao qual essa grade de dados pertence.

**MaxTime**

Exibe o tempo gasto pela transação que *mais* consome tempo para este servidor de catálogos.

**MinTime**

Exibe o tempo gasto pela transação que *menos* consome tempo para este servidor de catálogos.

**MeanTime**

Especifica o tempo médio gasto em uma transação.

**TotalTime**

Exibe o tempo total gasto em transações para este servidor de catálogos, desde o momento em que este servidor de catálogos foi inicializado.

**AvgTransTime**

Exibe o tempo médio necessário para concluir uma transação para este servidor de catálogos.

**AvgThroughPut**

Exibe o número médio de transações por segundo para este servidor de catálogos.

**SumOfSquares**

Especifica a soma dos valores quadrados para o tempo de transação. Esse valor mede o desvio da média em um determinado ponto no tempo.

**SampleLen**

Indica a duração, em milissegundos, do tempo em que os dados foram amostrados.

**LastCount**

Especifica o tempo desde a última amostra obtida.

**LastTotalTime**

Especifica o tempo total atual menos o tempo total anterior para a amostra de dados.

### StartTime

Indica o tempo que as estatísticas começaram a ser coletadas desde a última reconfiguração dos dados. Os dados são reconfigurados quando o servidor é reiniciado.

---

## Monitorando com a API de Estatísticas

A API de Estatísticas é a interface direta para a árvore de estatísticas interna. As estatísticas são desativadas por padrão, mas podem ser ativadas configurando uma interface StatsSpec. Uma interface StatsSpec define como o WebSphere eXtreme Scale deve monitorar estatísticas.

### Sobre Esta Tarefa

É possível usar a API StatsAccessor local para consultar dados e acessar estatísticas em qualquer instância do ObjectGrid que esteja no mesmo Java Virtual Machine (JVM) que o código de execução. Para obter mais informações sobre as interfaces específicas, consulte a documentação da API. Conclua as seguintes etapas para ativar o monitoramento da árvore de estatísticas internas.

### Procedimento

1. Recupere o objeto StatsAccessor. A interface StatsAccessor segue o padrão do singleton. Assim, além dos problemas relacionados ao carregador de classes, uma instância do StatsAccessor deverá existir para cada JVM. Esta classe atua como a interface principal para todas as operações locais de estatísticas. O seguinte código é um exemplo de como recuperar a classe do acessador. Chame essa operação antes de qualquer outra chamada ObjectGrid.

```
public class LocalClient {  
  
    public static void main(String[] args) {  
  
        // retrieve a handle to the StatsAccessor  
        StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();  
  
    }  
  
}
```

2. Configure a interface StatsSpec da grade de dados. Configure esse JVM para coletar todas as estatísticas apenas no nível do ObjectGrid. É necessário garantir que um aplicativo ative todas as estatísticas que podem ser necessárias antes de iniciar quaisquer transações. O exemplo a seguir configura a interface StatsSpec utilizando um campo constante estático e utilizando uma spec String. Usar um campo de constante estático é mais simples porque o campo já definiu a especificação. Entretanto, ao utilizar uma spec String, é possível ativar qualquer combinação de estatísticas que são necessárias.

```
public static void main(String[] args) {  
  
    // retrieve a handle to the StatsAccessor  
    StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();  
  
    // Set the spec via the static field  
    StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);  
    accessor.setStatsSpec(spec);  
  
    // Configurar o spec através do spec String  
    StatsSpec spec = new StatsSpec("og.all=enabled");  
    accessor.setStatsSpec(spec);  
  
}
```

3. Envie as transações para a grade para forçar a coleta dos dados para monitoramento. Para coletar dados úteis para as estatísticas, é necessário enviar as transações para a grade de dados. O seguinte extrato de código insere um registro no MapA, que é um ObjectGridA. Como as estatísticas estão no nível do ObjectGrid, qualquer mapa dentro do ObjectGrid produz os mesmos resultados.

```
public static void main(String[] args) {  
  
    // retrieve a handle to the StatsAccessor  
    StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();  
  
    // Set the spec via the static field  
    StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);  
    accessor.setStatsSpec(spec);  
  
    ObjectGridManager manager =  
    ObjectGridmanagerFactory.getObjectGridManager();  
    ObjectGrid grid = manager.getObjectGrid("ObjectGridA");  
    Session session = grid.getSession();  
    Map map = session.getMap("MapA");  
  
    // Drive insert  
    session.begin();  
    map.insert("SomeKey", "SomeValue");  
    session.commit();  
}
```

4. Consulte um StatsFact utilizando a API do StatsAccessor. Cada caminho de estatísticas é associado com uma interface StatsFact. A interface StatsFact é um marcador genérico que é utilizado para organizar e conter um objeto StatsModule. Antes de poder acessar o módulo de estatísticas real, o objeto StatsFact deve ser recuperado.

```
public static void main(String[] args)  
{  
  
    // retrieve a handle to the StatsAccessor  
    StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();  
  
    // Set the spec via the static field  
    StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);  
    accessor.setStatsSpec(spec);  
  
    ObjectGridManager manager =  
    ObjectGridManagerFactory.getObjectGridManager();  
    ObjectGrid grid = manager.getObjectGrid("ObjectGridA");  
    Session session = grid.getSession();  
    Map map = session.getMap("MapA");  
  
    // Drive insert  
    session.begin();  
    map.insert("SomeKey", "SomeValue");  
    session.commit();  
  
    // Retrieve StatsFact  
  
    StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},  
    StatsModule.MODULE_TYPE_OBJECT_GRID);  
}
```

5. Interaja com o objeto StatsModule. O objeto StatsModule está contido na interface StatsFact. É possível obter uma referência para o módulo utilizando a interface StatsFact. Como a interface StatsFact é uma interface genérica, é necessário converter o módulo retornado para o tipo StatsModule esperado. Como esta tarefa coleta estatísticas do eXtreme Scale, o objeto StatsModule

retornado é convertido para um tipo `OGStatsModule`. Após o módulo ser convertido, é possível acessar todas as estatísticas disponíveis.

```
public static void main(String[] args) {

    // retrieve a handle to the StatsAccessor
    StatsAccessor accessor = StatsAccessorFactory.getStatsAccessor();

    // Set the spec via the static field
    StatsSpec spec = new StatsSpec(StatsSpec.OG_ALL);
    accessor.setStatsSpec(spec);

    ObjectGridManager manager =
    ObjectGridmanagerFactory.getObjectGridManager();
    ObjectGrid grid = manager.getObjectGrid("ObjectGridA");
    Session session = grid.getSession();
    Map map = session.getMap("MapA");

    // Drive insert
    session.begin();
    map.insert("SomeKey", "SomeValue");
    session.commit();

    // Retrieve StatsFact
    StatsFact fact = accessor.getStatsFact(new String[] {"EmployeeGrid"},
    StatsModule.MODULE_TYPE_OBJECT_GRID);

    // Retrieve module and time
    OGStatsModule module = (OGStatsModule)fact.getStatsModule();
    ActiveTimeStatistic timeStat =
    module.getTransactionTime("Default", true);
    double time = timeStat.getMeanTime();

}
```

## Módulos Estatísticos

O WebSphere eXtreme Scale usa um modelo de estatísticas interno para controlar e filtrar dados, que é a estrutura subjacente usada por todas as visualizações de dados para reunir capturas instantâneas das estatísticas. É possível utilizar diversos métodos para recuperar as informações dos módulos estatísticos.

### Visão Geral

As estatísticas no WebSphere eXtreme Scale são controladas e contidas nos componentes `StatsModules`. No modelo estatístico, existem diversos tipos de módulos estatísticos:

#### **OGStatsModule**

Fornece estatísticas para uma instância do `ObjectGrid`, incluindo tempos de resposta da transação.

#### **MapStatsModule**

Fornece estatísticas para um único mapa, incluindo o número de entradas e a taxa de ocorrências.

#### **QueryStatsModule**

Fornece estatísticas para consultas, incluindo criação de planos e tempo de execução.

#### **AgentStatsModule**

Fornece estatísticas para agentes de API do `DataGrid`, incluindo tempos de serialização e tempos de execução.

### HashIndexStatsModule

Fornecer estatísticas para consulta HashIndex e tempos de execução de manutenção.

### SessionStatsModule

Fornecer estatísticas ao plug-in do gerenciador de sessões HTTP.

Para obter detalhes sobre os módulos de estatísticas, consulte a Pacote do com.ibm.websphere.objectgrid.stats na Documentação da API.

## Estatísticas em um Ambiente Local

O modelo é organizado como uma árvore n-ária (uma estrutura em árvore com o mesmo grau para todos os nós) que é composta de todos os tipos de StatsModule mencionados na lista anterior. Devido a esta estrutura de organização, cada nó na árvore é representado pela interface StatsFact. A interface StatsFact pode representar um módulo individual ou um grupo de módulos para propósitos de agregação. Por exemplo, se vários nós-folhas na árvore representarem objetos MapStatsModule específicos, o nó-pai StatsFact para estes nós conterá estatísticas agregadas para todos os módulos-filhos. Após buscar um objeto StatsFact, é possível utilizar a interface para recuperar o StatsModule correspondente.

Muito semelhante a um mapa de árvore, é utilizado um caminho ou chave correspondente para recuperar um StatsFact específico. O caminho é um valor String[] que consiste em cada nó que está ao longo do caminho para o fato solicitado. Por exemplo, você criou um ObjectGrid denominado ObjectGridA, que contém dois Mapas: MapA e MapB. O caminho para o StatsModule para MapA seria semelhante a [ObjectGridA, MapA]. O caminho para as estatísticas agregadas para ambos os mapas seria: [ObjectGridA].

## Estatísticas em um Ambiente Distribuído

Em um ambiente distribuído, os módulos de estatísticas são recuperados utilizando um caminho diferente. Como um servidor pode conter várias partições, a árvore de estatísticas precisa controlar a partição a qual cada módulo pertence. Como resultado, o caminho para consultar um objeto StatsFact específico é diferente. Utilizando o exemplo anterior, mas incluindo nele os mapas existentes na partição 1, o caminho é [1, ObjectGridA, MapA] para recuperação de tal objeto StatsFact para MapA.

---

## Monitorando com o Utilitário xscmd

O utilitário **xscmd** substitui o utilitário **xsadmin** de amostra como uma ferramenta de monitoramento e de administração totalmente suportadas. Com o utilitário **xscmd**, é possível exibir informações sobre sua topologia do WebSphere eXtreme Scale.

### Antes de Iniciar

- Para o utilitário **xscmd** exibir os resultados, você deve criar uma topologia de grade de dados. Os servidores de catálogos e servidores de contêiner devem ser iniciados. Consulte o “Iniciando e Parando Servidores Independentes” na página 387 para obter informações adicionais.
- Consulte “Administrando com o Utilitário **xscmd**” na página 407 para obter mais informações sobre como iniciar o utilitário **xscmd**.



## Sobre Esta Tarefa

É possível usar o utilitário **xscmd** para visualizar o layout atual e o estado específico da grade de dados, como o conteúdo de mapa. Neste exemplo, o layout da grade de dados nesta tarefa consiste de uma única grade de dados *ObjectGridA* com um mapa *MapA* que pertence ao conjunto de mapas *MapSetA*. Este exemplo demonstra como é possível exibir todos os contêineres ativos em uma grade de dados e imprimir métricas filtradas referentes ao tamanho do mapa do mapa *MapA*. Para visualizar todas as opções de comando possíveis, execute o utilitário **xscmd** sem nenhum argumento ou com a opção **-help**.

## Procedimento

Monitore o ambiente com o utilitário **xscmd**.

- Para ativar as estatísticas de todos os servidores, execute o seguinte comando:
  - **UNIX** `./xscmd.sh -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
  - **Windows** `xscmd.bat -c setStatsSpec -spec ALL=enabled -g ObjectGridA`
- Para exibir todos os servidores de contêiner on-line para uma grade de dados, execute o seguinte comando:

- **UNIX** `./xscmd.sh -c showPlacement -g ObjectGridA -ms MapSetA`
- **Windows** `xscmd.bat -c showPlacement -g ObjectGridA -ms MapSetA`

Todas as informações do contêiner são exibidas.

**Atenção:** Para obter essas informações quando o protocolo Segurança da Camada de Transporte/Secure Sockets Layer (TLS/SSL) estiver ativado, você deve iniciar os servidores de catálogos e de contêiner com o conjunto de portas do serviço JMX. Para configurar a porta de serviço JMX, é possível usar a opção **-JMXServicePort** no script **startOgServer** ou chamar o método `setJMXServicePort` na interface `ServerProperties`.

- Para exibir informações sobre os mapas para a grade de dados *ObjectGridA*, execute o seguinte comando:
  - **UNIX** `./xscmd.sh -c showMapSizes -g ObjectGridA -ms MapSetA`
  - **Windows** `xscmd.bat -c showMapSizes -g ObjectGridA -ms MapSetA`
- Para conectar-se ao serviço de catálogo e exibir informações sobre o mapa *MapA* para o domínio de serviço de catálogo inteiro, execute o comando a seguir:
  - **UNIX** `./xscmd.sh -c showMapSizes -g ObjectGridA -ms MapSetA -m MapA -cep CatalogMachine:6645`
  - **Windows** `xscmd.bat -c showMapSizes -g ObjectGridA -ms MapSetA -m MapA -cep CatalogMachine:6645`

O utilitário **xscmd** se conecta ao servidor MBean que está em execução em um servidor de catálogos. Conectando-se a um servidor de catálogos único, é possível recuperar informações sobre o domínio do serviço de catálogo inteiro. Um servidor de catálogos pode ser executado como um processo independente, como um processo do WebSphere Application Server ou integrado em um processo de aplicativo customizado. Use a opção **-cep** para especificar o nome do host e a porta do serviço de catálogo. Se você incluir uma lista de servidores de catálogos para a opção **-cep**, os servidores de catálogos deverão estar dentro do mesmo domínio de serviço de catálogo. É possível recuperar as estatísticas para um domínio de serviço de catálogo por vez.

- Para exibir o posicionamento configurado e de tempo de execução de sua configuração, execute um dos seguintes comandos:

- xscmd -c placementServiceStatus
- xscmd -c placementServiceStatus -g ObjectGridA -ms MapSetA
- xscmd -c placementServiceStatus -ms MapSetA
- xscmd -c placementServiceStatus -g ObjectGridA

É possível definir o escopo do comando para exibir informações de posicionamento para a configuração inteira, para uma grade de dados única, para um único conjunto de mapas ou para uma combinação de uma grade de dados e conjunto de mapas.

---

## Monitorando com a PMI do WebSphere Application Server

O WebSphere eXtreme Scale suporta Performance Monitoring Infrastructure (PMI) ao executar em um WebSphere Application Server ou servidor de aplicativos WebSphere Extended Deployment. A PMI coleta dados de desempenho em aplicativos de tempo de execução e fornece interfaces que suportam aplicativos externos para monitorar dados de desempenho. É possível utilizar o console administrativo ou a ferramenta wsadmin para acessar dados de monitoramento.

### Antes de Iniciar

É possível utilizar a PMI para monitorar seu ambiente quando você está utilizando o WebSphere eXtreme Scale combinado com o WebSphere Application Server.

### Sobre Esta Tarefa

O WebSphere eXtreme Scale utiliza o recurso PMI customizado do WebSphere Application Server para incluir sua própria instrumentação PMI. Com esta abordagem, é possível ativar e desativar a PMI do WebSphere eXtreme Scale com o console administrativo ou com as interfaces Java Management Extensions (JMX) na ferramenta wsadmin. Além disso, é possível acessar as estatísticas do WebSphere eXtreme Scale com a PMI padrão e as interfaces JMX que são utilizadas pela ferramentas de monitoramento, incluindo o Tivoli Performance Viewer.

### Procedimento

1. Ative a PMI do eXtreme Scale. É necessário ativar a PMI para visualizar as estatísticas de PMI. Consulte "Ativando a PMI" para obter mais informações.
2. Recupere as estatísticas de PMI do eXtreme Scale. Visualize o desempenho dos seus aplicativos eXtreme Scale com o Tivoli Performance Viewer. Consulte "Recuperando Estatísticas PMI" na página 458 para obter mais informações.

### O que Fazer Depois

Para obter mais informações sobre a ferramenta wsadmin, consulte "Acessando Beans Gerenciados (MBeans) Usando a Ferramenta wsadmin" na página 426.

## Ativando a PMI

É possível utilizar a WebSphere Application Server Performance Monitoring Infrastructure (PMI) para ativar ou desativar estatísticas em qualquer nível. Por exemplo, você pode optar por ativar as estatísticas de taxa de acesso do mapa para um mapa específico, mas não o número de estatísticas de entrada ou as estatísticas de tempo de atualização de lote do utilitário de carga. É possível ativar a PMI no console administrativo ou com scripts.

## Antes de Iniciar

Seu servidor de aplicativos deve ser iniciado e ter um aplicativo ativado para o eXtreme Scale instalado. Para ativar a PMI com scripts, você também deve estar apto a efetuar login e utilizar a ferramenta wsadmin. Para obter informações adicionais sobre a ferramenta wsadmin, consulte o tópico ferramenta wsadmin no centro de informações do WebSphere Application Server.

## Sobre Esta Tarefa

Utilize a PMI do WebSphere Application Server para fornecer um mecanismo granular com o qual é possível ativar ou desativar estatísticas em qualquer nível. Por exemplo, você pode optar por ativar as estatísticas de taxa de acesso do mapa para um mapa específico, mas não o número de estatísticas de entrada ou as estatísticas de tempo de atualização de lote do utilitário de carga. Esta seção mostra como utilizar o console administrativo e os scripts wsadmin para ativar a PMI do ObjectGrid.

## Procedimento

- **Ativar a PMI no console administrativo.**

1. No console administrativo, clique em **Monitoramento e Ajuste > Performance Monitoring Infrastructure > server\_name**.
2. Verifique se opção Ativar PMI (Performance Monitoring Infrastructure) está selecionada. Essa definição é ativada por padrão. Se a configuração não estiver ativada, selecione a caixa de opções e reinicie o servidor.
3. Clique em **Customizar**. Na árvore de configuração, selecione o ObjectGrid e o módulo Mapas do ObjectGrid. Ative as estatísticas para cada módulo.

A categoria de tipo de transação para estatísticas do ObjectGrid é criada no tempo de execução. É possível visualizar apenas as subcategorias das estatísticas do ObjectGrid e do Mapa na guia **Tempo de Execução**.

- **Ativar a PMI com scripts.**

1. Abra um prompt de linha de comandos. Navegue para o diretório `was_root/bin`. Digite **wsadmin** para iniciar a ferramenta de linha de comandos wsadmin.
2. Modifique a configuração do tempo de execução do PMI do eXtreme Scale. Verifique se a PMI está ativada para o servidor, por meio dos seguintes comandos:

```
wsadmin>set s1 [$AdminConfig getid
/Cell:CELL_NAME/Node:NODE_NAME/
Server:APPLICATION_SERVER_NAME/]
wsadmin>set pmi [$AdminConfig list PMIService $s1]
wsadmin>$AdminConfig show $pmi.
```

Se a PMI não estiver ativada, execute os seguintes comandos para ativá-la:

```
wsadmin>$AdminConfig modify $pmi {{enable true}}
wsadmin> $AdminConfig save
```

Se precisar ativar a PMI, reinicie o servidor.

3. Configure as variáveis para customizar o conjunto de estatísticas utilizando os seguintes comandos:

```
wsadmin>set perfName [$AdminControl completeObjectName type=Perf,
process=APPLICATION_SERVER_NAME,*]
wsadmin>set perfOName [$AdminControl makeObjectName $perfName]
```

```
wsadmin>set params [java::new {java.lang.Object[]} 1]
wsadmin>$params set 0 [java::new java.lang.String custom]
wsadmin>set sigs [java::new {java.lang.String[]} 1]
wsadmin>$sigs set 0 java.lang.String
```

- Configure o conjunto de estatísticas para customizar o seguinte comando:

```
wsadmin>$AdminControl invoke_jmx $perf0Name setStatisticSet $params $sigs
```

- Utilize os comandos a seguir para configurar as variáveis de modo a ativar a estatística da PMI objectGridModule:

```
wsadmin>set params [java::new {java.lang.Object[]} 2]
wsadmin>$params set 0 [java::new java.lang.String objectGridModule=1]
wsadmin>$params set 1 [java::new java.lang.Boolean false]
wsadmin>set sigs [java::new {java.lang.String[]} 2]
wsadmin>$sigs set 0 java.lang.String
wsadmin>$sigs set 1 java.lang.Boolean
```

- Configure a cadeia de estatísticas por meio deste comando:

```
wsadmin>set params2 [java::new {java.lang.Object[]} 2]
wsadmin>$params2 set 0 [java::new java.lang.String mapModule=*]
wsadmin>$params2 set 1 [java::new java.lang.Boolean false]
wsadmin>set sigs2 [java::new {java.lang.String[]} 2]
wsadmin>$sigs2 set 0 java.lang.String
wsadmin>$sigs2 set 1 java.lang.Boolean
```

- Configure a cadeia de estatísticas por meio deste comando:

```
wsadmin>$AdminControl invoke_jmx $perf0Name setCustomSetString $params2 $sigs2
```

Essas etapas ativam o PMI do tempo de execução do eXtreme Scale, mas não modificam a configuração do PMI. Se você reiniciar o servidor de aplicativos, as configurações da PMI serão perdidas, exceto para a ativação da PMI principal.

## Exemplo

Você pode executar as seguintes etapas para ativar as estatísticas da PMI para o aplicativo de amostra:

- Ative o aplicativo utilizando o endereço da Web `http://host:port/ObjectGridSample`, em que `host` e `porta` são o nome do host e número de porta HTTP do servidor no qual a amostra está instalada.
- No aplicativo de amostra, clique consecutivamente em `ObjectGridCreationServlet` e nos botões de ação 1, 2, 3, 4 e 5 para gerar ações para o `ObjectGrid` e os mapas. Não feche esta página do servlet neste momento.
- No console administrativo, clique em **Monitoramento e Ajuste > Performance Monitoring Infrastructure > server\_name**. Clique na guia **Tempo de Execução**.
- Clique no botão de rádio **Customizado**.
- Expanda o módulo Mapas do `ObjectGrid` na árvore de tempo de execução e clique no link `clusterObjectGrid`. No grupo Mapas do `ObjectGrid`, há uma instância do `ObjectGrid` denominada `clusterObjectGrid`; há quatro mapas debaixo do grupo `clusterObjectGrid`: contadores, funcionários, escritórios e sites. Na instância do `ObjectGrids`, existe uma instância do `clusterObjectGrid` e sob esta há um tipo de transação denominado `DEFAULT`.
- Você pode ativar as estatísticas de seu interesse. Por exemplo, é possível ativar o número de entradas para o mapa funcionários e o tipo de resposta de transação para o tipo de transação `DEFAULT`.

## O que Fazer Depois

Quando ativar a PMI, será possível visualizar estatísticas de PMI com o console administrativo ou por meio de scripts.

## Recuperando Estatísticas PMI

Ao recuperar estatísticas PMI, é possível visualizar o desempenho dos aplicativos eXtreme Scale.

### Antes de Iniciar

- Ative o controle de estatísticas PMI para o seu ambiente. Consulte “Ativando a PMI” na página 455 para obter mais informações.
- Os caminhos nesta tarefa assumem que você está recuperando estatísticas para o aplicativo de amostra, mas é possível utilizar estas estatísticas para qualquer outro aplicativo com etapas semelhantes.
- Se estiver utilizando o console administrativo para recuperar estatísticas, você deve estar apto a efetuar login no console administrativo. Se estiver utilizando scripts, você deve estar apto a efetuar login no wsadmin.

### Sobre Esta Tarefa

É possível recuperar estatísticas PMI para visualização no Tivoli Performance Viewer concluindo etapas no console administrativo ou com scripts.

- Etapas do console administrativo
- Etapas de scripts

Para obter mais informações sobre as estatísticas que podem ser recuperadas, consulte “Módulos PMI” na página 459.

### Procedimento

- Recupere estatísticas PMI no console administrativo.
  1. No console administrativo, clique em **Monitoramento e Ajuste > Performance Viewer > Atividade Atual**
  2. Selecione o servidor que deseja monitorar utilizando o Tivoli Performance Viewer, em seguida, ative o monitoramento.
  3. Clique no servidor para visualizar a página do Performance Viewer.
  4. Expanda a árvore de configuração. Clique em **Mapas do ObjectGrid > clusterObjectGrid** e selecione **employees**. Expanda **ObjectGrids > clusterObjectGrid** e selecione **DEFAULT**.
  5. No aplicativo de amostra do ObjectGrid, retorne ao servlet `ObjectGridCreationServlet`, clique no botão 1, em seguida, ocupar mapas. Você pode visualizar a estatística no visualizador.
- Recupere estatísticas PMI com scripts.
  1. Em um prompt de linha de comandos, navegue até o diretório `was_root/bin`. Digite `wsadmin` para iniciar a ferramenta `wsadmin`.
  2. Configure as variáveis para o ambiente utilizando os seguintes comandos:

```
wsadmin>set perfName [$AdminControl completeObjectName type=Perf,*]
wsadmin>set perfOName [$AdminControl makeObjectName $perfName]
wsadmin>set mySrvName [$AdminControl completeObjectName type=Server,
name=APPLICATION_SERVER_NAME,*]
```
  3. Configure as variáveis para obter o ambiente `mapModule` utilizando os seguintes comandos:

```
wsadmin>set params [java::new {java.lang.Object[]} 3]
wsadmin>$params set 0 [$AdminControl makeObjectName $mySrvName]
wsadmin>$params set 1 [java::new java.lang.String mapModule]
wsadmin>$params set 2 [java::new java.lang.Boolean true]
```

```

wsadmin>set sigs [java::new {java.lang.String[]} 3]
wsadmin>$sigs set 0 javax.management.ObjectName
wsadmin>$sigs set 1 java.lang.String
wsadmin>$sigs set 2 java.lang.Boolean

```

- Obtenha a estatística mapModule utilizando os seguintes comandos:

```
wsadmin>$AdminControl invoke_jmx $perf0Name getStatsString $params $sigs
```

- Configure as variáveis para obter a estatística objectGridModule utilizando os seguintes comandos:

```

wsadmin>set params2 [java::new {java.lang.Object[]} 3]
wsadmin>$params2 set 0 [$AdminControl makeObjectName $mySrvName]
wsadmin>$params2 set 1 [java::new java.lang.String objectGridModule]
wsadmin>$params2 set 2 [java::new java.lang.Boolean true]
wsadmin>set sigs2 [java::new {java.lang.String[]} 3]
wsadmin>$sigs2 set 0 javax.management.ObjectName
wsadmin>$sigs2 set 1 java.lang.String
wsadmin>$sigs2 set 2 java.lang.Boolean

```

- Obtenha a estatística objectGridModule utilizando os seguintes comandos:
- ```
wsadmin>$AdminControl invoke_jmx $perf0Name getStatsString $params2 $sigs2
```

## Resultados

É possível visualizar estatísticas no Tivoli Performance Viewer.

## Módulos PMI

É possível monitorar o desempenho dos seus aplicativos com os módulos Performance Monitoring Infrastructure (PMI).

### objectGridModule

O objectGridModule contém uma estatística de tempo: tempo de resposta de transação. Uma transação é definida como uma duração entre a chamada de método Session.begin e a chamada de método Session.commit. Esta duração é rastreada como o tempo de resposta da transação. O elemento-raiz do objectGridModule, "root", serve como ponto de entrada para as estatísticas do WebSphere eXtreme Scale. Este elemento-raiz possui ObjectGrids como seus elementos-filhos, que possuem tipos de transações como seus elementos-filhos. A estatística de tempo de resposta está associada a cada tipo de transação.

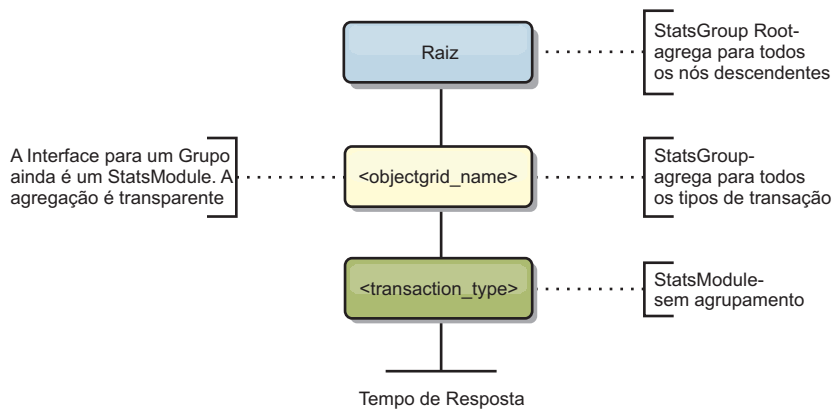


Figura 57. Estrutura do Módulo ObjectGridModule

O seguinte diagrama mostra um exemplo da estruturaObjectGridModule. Neste exemplo, duas instâncias ObjectGrid existem no sistema: ObjectGrid A e ObjectGrid

B. A instância ObjectGrid A possui dois tipos de transações: A e padrão. A instância ObjectGrid B possui apenas o tipo de transação padrão.

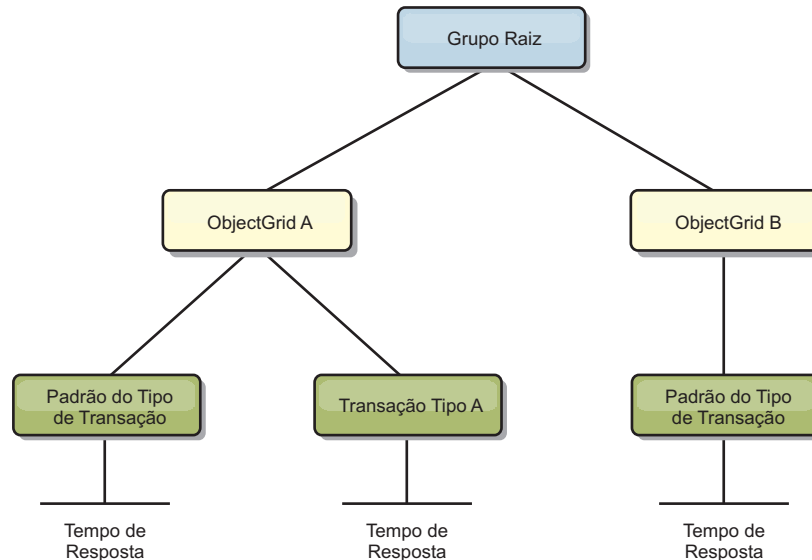


Figura 58. Exemplo da Estrutura do Módulo ObjectGridModule

Os tipos de transações são definidos por desenvolvedores de aplicativos, porque eles sabem quais tipos de transações seus aplicativos utilizam. O tipo de transação é configurado utilizando o método `Session.setTransactionType(String)` a seguir:

```

/**
 * Configura o tipo de transação para futuras transações.
 *
 * Quando este método é chamado, todas as futuras transações terão o mesmo tipo
 * até que outro tipo de transação seja configurado. Se nenhum tipo de transação
 * estiver configurado, será utilizado o tipo de transação TRANSACTION_TYPE_DEFAULT
 * padrão.
 *
 * Os tipos de transações são utilizados principalmente para finalidade
 * de rastreamento de dados estatísticos.
 * Os usuários podem predefinir tipos de transações que são executadas em um
 * application. A idéia é categorizar transações com as mesmas características
 * para uma categoria (tipo), portanto, uma estatística de tempo de resposta de
 * transação pode ser utilizada para rastrear cada tipo de transação.
 *
 * Este rastreamento é útil quando seu aplicativo tem diferentes tipos de
 * transações.
 * Entre eles, alguns tipos de transações, como transações de atualização, têm
 * um processamento mais longo que outras transações, como transações de leitura.
 * Utilizando o tipo de transação, diferentes transações são rastreadas por
 * diferentes estatísticas, portanto, as estatísticas podem ser mais úteis.
 *
 * @param tranType o tipo de transação para futuras transações.
 */
void setTransactionType(String tranType);
  
```

O exemplo a seguir configura o tipo de transação como `updatePrice`:

```

// Set the transaction type
// to updatePrice
// The time between session.begin() and session.commit() will be
// tracked in the time statistic for "updatePrice".
  
```



```

session.setTransactionType("updatePrice");
session.begin();
map.update(stockId, new Integer(100));
session.commit();

```

A primeira linha indica que o tipo de transação subsequente é updatePrice. Uma estatística updatePrice existe na instância ObjectGrid que corresponde à sessão no exemplo. Utilizando interfaces Java Management Extensions (JMX), você pode obter o tempo de resposta da transação para transações updatePrice. Também é possível obter a estatística agregada para todos os tipos de transações na instância ObjectGrid especificada.

## mapModule

O mapModule contém três estatísticas relacionadas aos mapas eXtreme Scale:

- **Taxa de ocorrências do mapa** - *BoundedRangeStatistic*: Controla a taxa de ocorrências de um mapa. A taxa de ocorrência é um valor flutuante entre 0 e 100 inclusivamente, que representa a porcentagem de ocorrências do mapa em relação a operações get do mapa.
- **Número de entradas**-*CountStatistic*: Controla o número de entradas no mapa.
- **Tempo de resposta de atualização de lote do utilitário de carga**-*TimeStatistic*: Controla o tempo de resposta que é utilizada para a operação de atualização de lote do utilitário de carga.

O elemento-raiz do mapModule, "root", serve como ponto de entrada para as estatísticas do Mapa ObjectGrid. Este elemento-raiz possui ObjectGrids como seus elementos-filhos, que possuem mapas como seus elementos-filhos. Cada instância do mapa possui três estatísticas listadas. A estrutura mapModule é mostrada no diagrama a seguir:

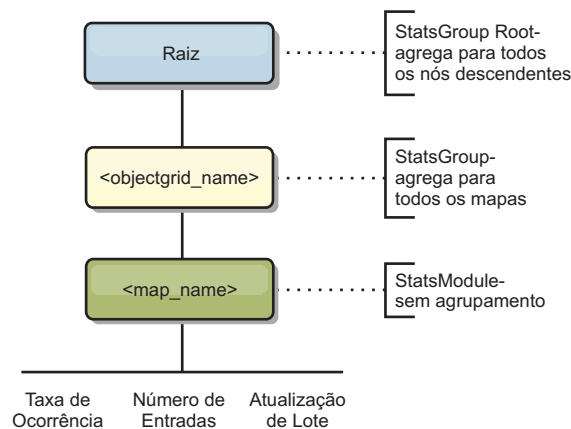
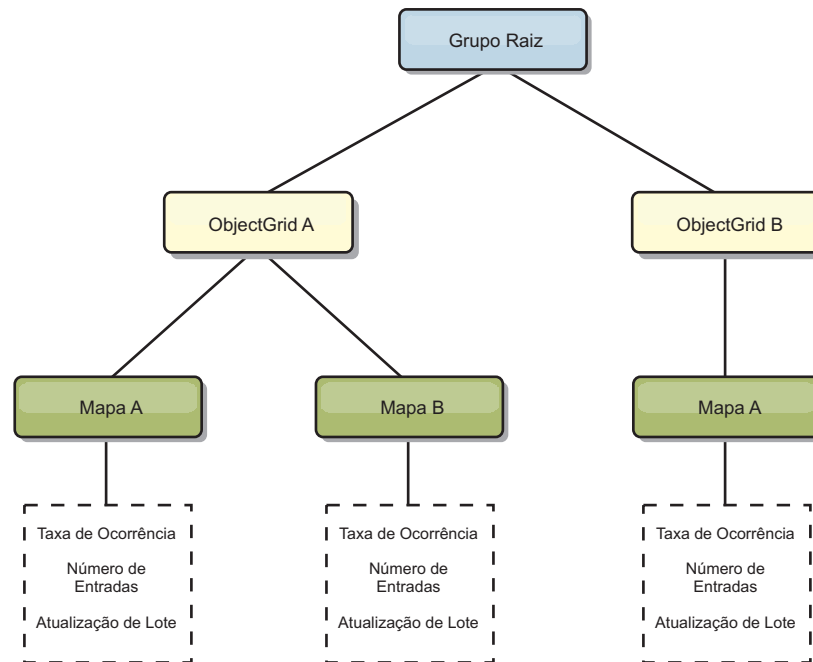


Figura 59. Estrutura do mapModule

O diagrama a seguir mostra um exemplo da estrutura mapModule:

Figura 60. Exemplo de Estrutura do Módulo mapModule



## hashIndexModule

O hashIndexModule contém as seguintes estatísticas que são relacionadas aos índices de nível de Mapa:

- **Contagem de Localizações-CountStatistic:** O número de chamadas para a operação find do índice.
- **Contagem de Colisões-CountStatistic:** O número de colisões para a operação find.
- **Contagem de Falhas-CountStatistic:** O número de falhas para a operação find.
- **Contagem de Resultados-CountStatistic:** O número de chaves retornadas da operação find.
- **Contagem de BatchUpdate-CountStatistic:** O número de atualizações de lote junto a este índice. Quando o mapa correspondente é alterado de qualquer maneira, o método doBatchUpdate() do índice será chamado. Esta estatística informará com que frequência seu índice está sendo alterado ou atualizado.
- **Tempo de Duração da Operação Find-TimeStatistic:** A quantidade de tempo que a operação find leva para ser concluída

O elemento-raiz do hashIndexModule, "root", serve como ponto de entrada para as estatísticas do HashIndex. Este elemento-raiz possui ObjectGrids como seus elementos-filhos, os ObjectGrids possuem mapas como seus elementos-filhos, que, finalmente, possuem HashIndexes como seus elementos-filhos e nós-folhas da árvore. Cada instância do HashIndex possui três estatísticas listadas. A estrutura hashIndexModule é mostrada no diagrama a seguir:

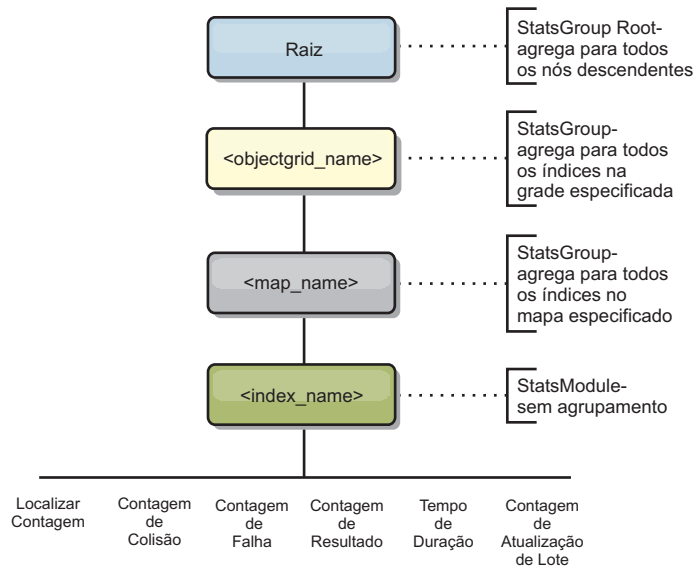


Figura 61. Estrutura do Módulo hashIndexModule

O diagrama a seguir mostra um exemplo da estrutura hashIndexModule:

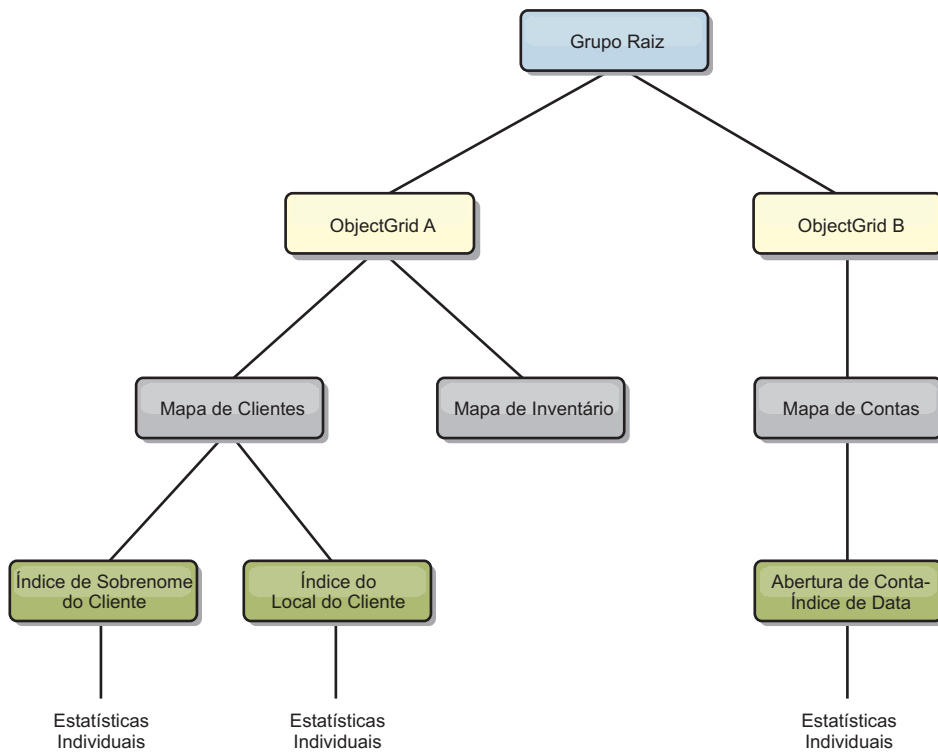


Figura 62. Exemplo da Estrutura do Módulo hashIndexModule

### agentManagerModule

O agentManagerModule contém as estatísticas que são relacionadas aos agentes de nível de mapa:

- **Tempo de Redução:** *TimeStatistic* - A quantidade de tempo para o agente concluir a operação reduce.

- **Tempo de Duração Total:** *TimeStatistic* - A quantidade total de tempo para o agente concluir todas as operações.
- **Tempo de Serialização do Agente:** *TimeStatistic* - A quantidade de tempo para serialização do agente.
- **Tempo de Aumento do Agente:** *TimeStatistic* - A quantidade de tempo para aumentar o agente no servidor.
- **Tempo de Serialização do Resultado:** *TimeStatistic* - A quantidade de tempo para serializar os resultados do agente.
- **Tempo de Aumento do Resultado:** *TimeStatistic* - A quantidade de tempo para aumentar os resultados do agente.
- **Contagem de Falhas:** *CountStatistic* - O número de vezes que o agente falhou.
- **Contagem de Chamada:** *CountStatistic* - O número de vezes que o AgentManager foi chamado.
- **Contagem de Partições:** *CountStatistic* - O número de partições para as quais o agente é enviado.

O elemento-raiz do agentManagerModule, "root", serve como ponto de entrada para as estatísticas do AgentManager. O elemento-raiz possui ObjectGrids como seus elementos-filhos, os ObjectGrids possuem seus elementos-filhos, que, finalmente, possuem instâncias do AgentManager como seus elementos-filhos e nós-folhas da árvore. Cada instância do AgentManager possui estatísticas.

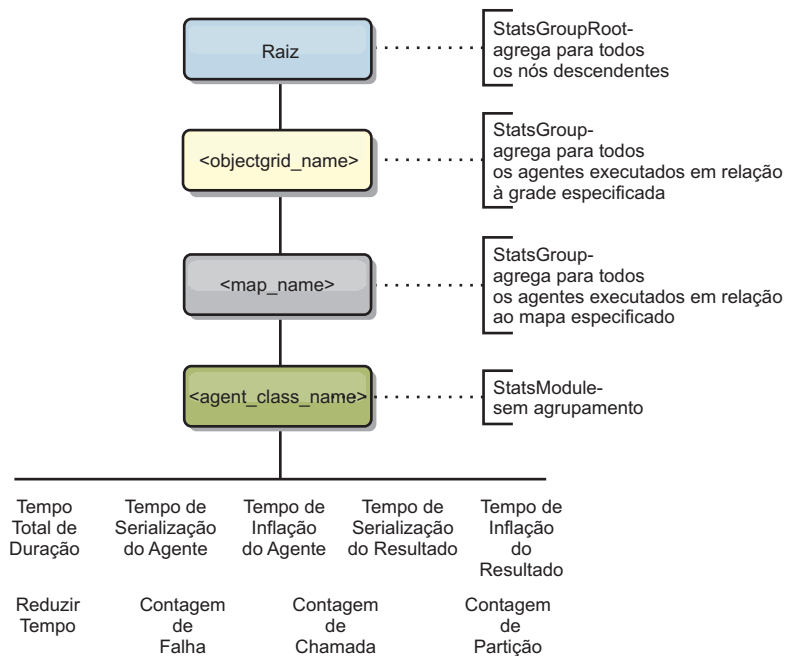


Figura 63. Estrutura agentManagerModule

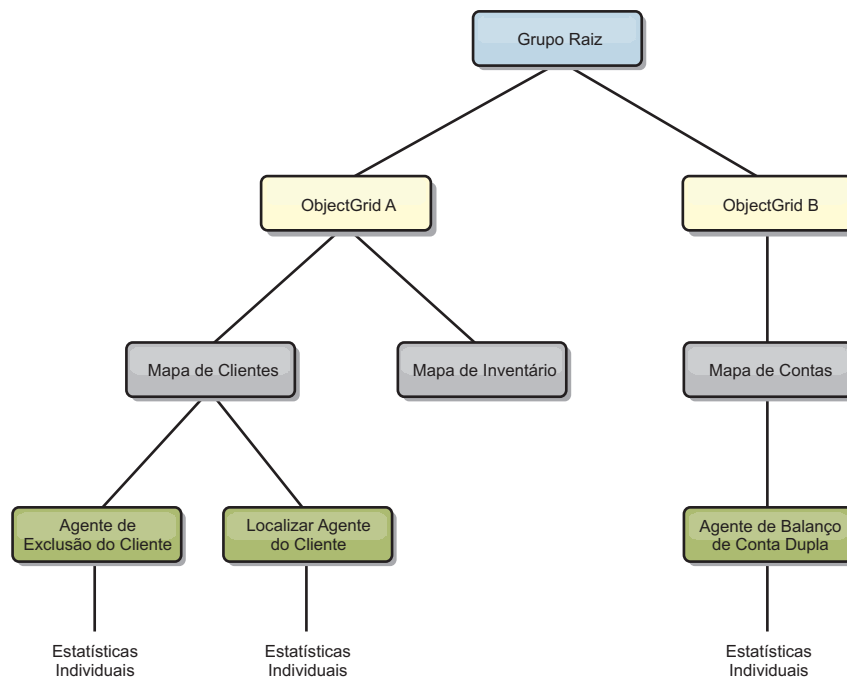


Figura 64. Exemplo da Estrutura agentManagerModule

## queryModule

O queryModule contém estatísticas relacionadas às consultas do eXtreme Scale:

- **Tempo de Criação do Plano:** *TimeStatistic* - A quantidade de tempo para criar o plano de consulta.
- **Tempo de Execução:** *TimeStatistic* - A quantidade de tempo para executar a consulta.
- **Contagem de Execução:** *CountStatistic* - O número de vezes que a consulta foi executada.
- **Contagem de Resultados:** *CountStatistic* - A contagem para cada conjunto de resultados de cada execução de consulta.
- **FailureCount:** *CountStatistic* - O número de vezes que a consulta falhou.

O elemento-raiz do queryModule, "root", serve como ponto de entrada para as estatísticas do Query. Este elemento-raiz possui ObjectGrids como seus elementos-filhos, que possuem objetos Query como seus elementos-filhos e nós-folhas da árvore. Cada instância do Query possui as três estatísticas listadas.

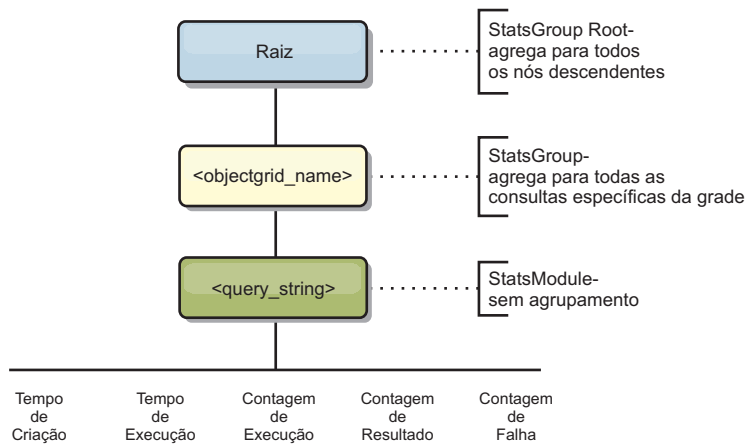


Figura 65. Estrutura queryModule

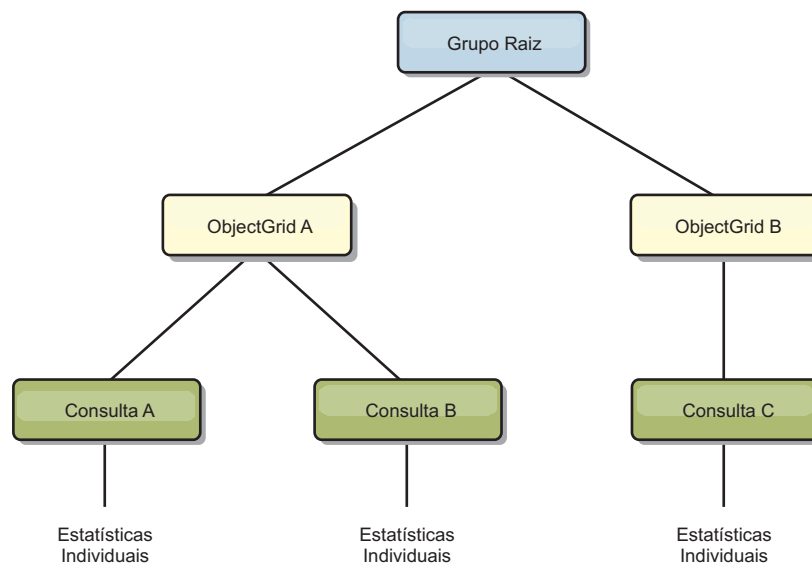


Figura 66. Exemplo da Estrutura queryModule QueryStats.jpg

## Acessando Beans Gerenciados (MBeans) Usando a Ferramenta wsadmin

É possível usar o utilitário wsadmin fornecido no WebSphere Application Server para acessar as informações do bean gerenciado (MBean).

### Procedimento

Execute a ferramenta wsadmin a partir do diretório bin em sua instalação do WebSphere Application Server. O exemplo a seguir recupera uma visualização da disposição do shard atual em um eXtreme Scale dinâmico. A ferramenta wsadmin pode ser executada a partir de qualquer instalação na qual o eXtreme Scale está em execução. Não é necessário executar a ferramenta wsadmin no serviço de catálogo.

```
$ wsadmin.sh -lang jython
wsadmin>placementService = AdminControl.queryNames
("com.ibm.websphere.objectgrid:*,type=PlacementService")
wsadmin>print AdminControl.invoke(placementService,
"listObjectGridPlacement","library ms1")
```

```

<objectGrid name="library" mapSetName="ms1">
  <container name="container-0" zoneName="DefaultDomain"
    hostname="host1.company.org" serverName="server1">
    <shard type="Primary" partitionName="0"/>
    <shard type="SynchronousReplica" partitionName="1"/>
  </container>
  <container name="container-1" zoneName="DefaultDomain"
    hostname="host2.company.org" serverName="server2">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="Primary" partitionName="1"/>
  </container>
  <container name="UNASSIGNED" zoneName=" ibm_SYSTEM"
    hostname="UNASSIGNED" serverName="UNNAMED">
    <shard type="SynchronousReplica" partitionName="0"/>
    <shard type="AsynchronousReplica" partitionName="0"/>
  </container>
</objectGrid>

```

---

## Monitoramento com Beans Gerenciados (MBeans)

É possível usar os beans gerenciados (MBeans) para controlar as estatísticas no seu ambiente.

### Antes de Iniciar

Para que os atributos sejam registrados, é necessário ativar as estatísticas. É possível ativar as estatísticas de uma das seguintes formas:

- **Com o arquivo de propriedades do servidor:**

É possível ativar as estatísticas no arquivo de propriedades do servidor com uma entrada de valor de chave `statsSpec=<StatsSpec>`. Alguns exemplos das possíveis configurações são:

- Para ativar todas as estatísticas, use `statsSpec=all=enabled`
- Para ativar apenas as estatísticas do ObjectGrid, use `statsSpec=og.all=enabled`. Para visualizar uma descrição de todas as especificações de estatísticas possíveis, consulte o StatsSpec API na documentação da API.

Para obter mais informações sobre o arquivo de propriedades do servidor, consulte Arquivo de Propriedades do Servidor.

- **Com um bean gerenciado:**

É possível ativar as estatísticas usando o atributo `StatsSpec` no MBean do ObjectGrid. Para obter mais informações, consulte o API StatsSpec na Documentação da API.

- **Programaticamente:**

Também é possível ativar as estatísticas programaticamente com a interface `StatsAccessor`, que é recuperada com a classe `StatsAccessorFactory`. Use esta interface em um ambiente do cliente ou quando precisar monitorar uma grade de dados que está em execução no processo atual.

### Procedimento

- **Acesse as estatísticas do MBean usando a ferramenta wsadmin.**

Para obter informações adicionais, consulte “Acessando Beans Gerenciados (MBeans) Usando a Ferramenta wsadmin” na página 426.

- **Acesse as estatísticas do MBean programaticamente.**



Para obter informações adicionais, consulte “Acessando Beans Gerenciados (MBeans) Programaticamente” na página 426.

## Exemplo

Para obter um exemplo de como usar os beans gerenciados, consulte Amostra: Utilitário `xsadmin`.

---

## Monitorando com Ferramentas do Fornecedor

O WebSphere eXtreme Scale pode ser monitorado usando diversas soluções populares de monitoramento corporativo. Os agentes do plug-in estão incluídos no IBM Tivoli Monitoring and Hyperic HQ, os quais monitoram o WebSphere eXtreme Scale usando os beans de gerenciamento acessíveis. O CA Wily Introscope usa a instrumentação do método Java para capturar estatísticas.

## Monitorando com o IBM Tivoli Enterprise Monitoring Agent para WebSphere eXtreme Scale

O IBM Tivoli Enterprise Monitoring Agent é uma solução de monitoramento cheia de recursos que pode ser usada para monitorar bancos de dados, sistemas operacionais e servidores em ambientes distribuídos e de host. O WebSphere eXtreme Scale inclui um agente customizado que pode ser usado para fazer introspecção dos beans de gerenciamento do eXtreme Scale. Essa solução trabalha eficientemente para ambas as implementações do eXtreme Scale e do WebSphere Application Server independentes.

### Antes de Iniciar

- Instale o WebSphere eXtreme Scale Versão 7.0.0 ou superior.  
Além disso, as estatísticas devem ser ativadas para coletar dados estatísticos de servidores WebSphere eXtreme Scale. Várias opções para ativar estatísticas são descritas no “Monitoramento com Beans Gerenciados (MBeans)” na página 467 e no Amostra: Utilitário `xsadmin`
- Instale o IBM Tivoli Monitoring Versão 6.2.1 com o fix pack 2 ou superior.
- Instale o agente de S.O. Tivoli em cada servidor ou host no qual o servidor do eXtreme Scale é executado.
- Instale o agente do WebSphere eXtreme Scale, que pode ser obtido por download gratuitamente a partir do Site do IBM Open Process Automation Library (OPAL).

Conclua as seguintes etapas para instalar e configurar o Tivoli Monitoring Agent:

### Procedimento

1. Instale o Tivoli Monitoring Agent para WebSphere eXtreme Scale.  
Faça download da imagem de instalação do Tivoli e extraia os arquivos para um diretório temporário.
2. Instale os arquivos de suporte do aplicativo eXtreme Scale.  
Instale o suporte do aplicativo eXtreme Scale em cada uma das seguintes implementações.
  - Tivoli Enterprise Portal Server (TEPS)
  - Enterprise Desktop client (TEPD)
  - Tivoli Enterprise Monitoring Server (TEMS)

- a. No diretório temporário que você criou, inicie uma nova janela de comando e execute o arquivo executável apropriado para sua plataforma. O script de instalação detecta automaticamente o tipo de implementação do Tivoli (TEMS, TEPD ou TEPS). É possível instalar qualquer tipo em um único host ou em vários hosts e todos os três tipos de implementação requerem a instalação dos arquivos de suporte de aplicativo do agente eXtreme Scale.
- b. Na janela do **Instalador**, verifique se as seleções para os componentes Tivoli implementados estão corretos. Clique em **Avançar**.
- c. Se for solicitado, envie o nome do host e as credenciais administrativas. Clique em **Avançar**.
- d. Selecione **Monitoring Agent para WebSphere eXtreme Scale**. Clique em **Avançar**.
- e. Você será notificado sobre quais ações de instalação deverão ser executadas. Clique em **Avançar** para visualizar o progresso da instalação até a conclusão.

Depois de concluir o procedimento, todos os arquivos de suporte do aplicativo necessários pelo agente do WebSphere eXtreme Scale serão instalados.

### 3. Instale o agente em cada um dos nós do eXtreme Scale.

Instale um agente do S.O. Tivoli em cada um dos computadores. Não é necessário configurar ou iniciar este agente. Use a mesma imagem de instalação da etapa anterior para executar o arquivo executável específico da plataforma.

Como uma recomendação, você precisa instalar apenas um agente por host. Cada agente é capaz de suportar várias instâncias dos servidores eXtreme Scale. Para obter melhor desempenho, utiliza uma instância do agente para monitorar cerca de 50 servidores eXtreme Scale.

- a. Na tela de boas-vindas do assistente de instalação, clique em **Avançar** para abrir a tela para especificar as informações do caminho de instalação.
- b. Para o campo **Diretório de Instalação do Tivoli Monitoring**, insira ou procure por C:\IBM\ITM (ou /opt/IBM/ITM). Em seguida para o campo **Local para a Mídia de Instalação**, verifique se o valor exibido está correto e clique em **Avançar**.
- c. Selecione os componentes que deseja incluir, como **Executar uma instalação local da solução** e clique em **Avançar**.
- d. Selecione os aplicativos para os quais o suporte será incluído ao selecionar o aplicativo, como **Monitoring Agent para WebSphere eXtreme Scale** e clique em **Avançar**.
- e. É possível visualizar o progresso até o suporte do aplicativo ser incluído com êxito.

**Nota:** Repita essas etapas em cada um dos nós do eXtreme Scale. Também é possível utilizar a instalação silenciosa. Consulte o IBM Tivoli Monitoring Information Center para obter mais informações sobre a instalação silenciosa.

### 4. Configure o agente do WebSphere eXtreme Scale.

Cada um dos agentes instalado precisa ser configurado para monitorar qualquer servidor de catálogos, servidor eXtreme Scale ou ambos.

As etapas para configurar as plataformas Windows e UNIX são diferentes. A configuração para a plataforma Windows é feita com a interface com o usuário **Gerenciar o Tivoli Monitoring Services**. A configuração para as plataformas UNIX baseia-se na linha de comandos.

**Windows** Use as seguintes etapas para configurar inicialmente o agente no Windows

- a. Na janela **Gerenciar o Tivoli Enterprise Monitoring Services**, clique em **Iniciar > Todos os Programas > IBM Tivoli Monitoring > Gerenciar o Tivoli Monitoring Services**.
  - b. Clique com o botão direito do mouse em **Monitoring Agent para WebSphere eXtreme Scale** e selecione **Configurar usando padrão**, que abre uma janela para criar uma instância exclusiva do agente.
  - c. Escolha um nome exclusivo, como por exemplo, `instance1` e clique em **Avançar**.
- Se planejar monitorar servidores do eXtreme Scale independentes, conclua as seguintes etapas:
    - a. Atualize os parâmetros Java e certifique-se de que o valor **Java Home** esteja correto. Os argumentos da JVM podem ser deixados vazios. Clique em **Avançar**.
    - b. Selecione o **Tipo de Conexão do Servidor MBean** e use **Servidor Compatível com JSR-160** para servidores eXtreme Scale independentes. Clique em **Avançar**.
    - c. Se a segurança estiver ativada, atualize os valores de **ID do Usuário** e **Senha**. Deixe o valor **URL de Serviço JMX** como está. Substitua esse valor depois. Deixe o campo **Informações do Caminho da Classe JMX** como está. Clique em **Avançar**.

Para configurar os servidores para o agente no Windows, conclua as seguintes etapas:

- a. Configure as instâncias de subnó dos servidores do eXtreme Scale na área de janela dos **Servidores de Grade do WebSphere eXtreme Scale**. Se nenhum servidor de contêiner existir no computador, clique em **Avançar** para continuar com a área de janela de serviço de catálogo.
  - b. Se vários servidores de contêiner do eXtreme Scale existirem no computador, configure o agente para monitorar cada um dos servidores.
  - c. É possível incluir quantos servidores do eXtreme Scale forem necessários se os nomes e portas forem exclusivos ao clicar em **Novo**. (Quando um servidor do eXtreme Scale for iniciado, um valor `JMXPort` deverá ser especificado).
  - d. Depois de configurar os servidores de contêiner, clique em **Avançar**, para levá-lo para a área de janela **Servidores de Catálogo do WebSphere eXtreme Scale**.
  - e. Se você não possuir nenhum servidor de catálogo, clique em **OK**. Se você tiver servidores de catálogo, inclua uma nova configuração para cada servidor, conforme foi feito com os servidores de contêiner. Novamente, escolha um nome exclusivo, de preferência o mesmo nome que foi usado ao iniciar o serviço de catálogo. Clique **OK** para concluir.
- Se você planejar monitorar servidores para o agente nos servidores do eXtreme Scale que são integrados em um processo do WebSphere Application Server, conclua as seguintes etapas:
    - a. Atualize os parâmetros Java e certifique-se de que o valor **Java Home** esteja correto. Os argumentos da JVM podem ser deixados vazios. Clique em **Avançar**.
    - b. Selecione o **Tipo de Conexão do Servidor MBean**. Selecione a versão do WebSphere Application Server que seja apropriada para seu ambiente. Clique em **Avançar**.
    - c. Certifique-se de que as informações do WebSphere Application Server no painel estejam corretas. Clique em **Avançar**.

- d. Inclua apenas uma definição de subnó. Forneça um nome para a definição de subnó, mas não atualize a definição de porta. No ambiente do WebSphere Application Server, os dados podem ser coletados de todos os servidores de aplicativos que forem gerenciados pelo agente do nó que estiver em execução no computador. Clique em **Avançar**.
- e. Se nenhum servidor de catálogo existir no ambiente, clique em **OK**. Se você tiver servidores de catálogo, inclua uma nova configuração para cada servidor, conforme foi feito com os servidores de contêiner. Escolha um nome exclusivo para o serviço de catálogo, de preferência o mesmo nome que foi usado ao iniciar o serviço de catálogo. Clique **OK** para concluir.

**Nota:** Os servidores de contêineres não precisam ser colocados junto com o serviço de catálogo.

Agora que o agente e os servidores estão configurados e prontos, na próxima janela, clique com o botão direito do mouse em `instance1` para iniciar o agente.

**UNIX** Para configurar o agente na plataforma UNIX na linha de comandos, conclua as seguintes etapas:

A seguir há um exemplo de servidores independentes que usam um tipo de conexão Compatível com JSR160. O exemplo mostra três contêineres do eXtreme Scale em um único host (`rhea00b02`) e os endereços de listener JMX são 15000, 15001 e 15002, respectivamente. Não há nenhum servidor de catálogo.

A saída do utilitário de configuração é exibida em *itálico de espaço simples*, enquanto que a resposta do usuário está em **negrito de espaço simples**. (Se nenhuma resposta do usuário era necessária, o padrão foi selecionado pressionando a tecla enter).

```
rhea00b02 # ./itmcmd config -A xt
Agent configuration started...
Enter instance name (default is ): inst1
Edit "Monitoring Agent for WebSphere eXtreme Scale" settings? [ 1=Yes, 2=No ] (default is: 1):
Edit 'Java' settings? [ 1=Yes, 2=No ] (default is: 1):
Java home (default is: C:\Program Files\IBM\Java50): /opt/OG61/java
Java trace level [ 1=Error, 2=Warning, 3=Information, 4=Minimum Debug, 5=Medium Debug, 6=Maximum Debug,
7=All ] (default is: 1):
JVM arguments (default is: ):
Edit 'Connection' settings? [ 1=Yes, 2=No ] (default is: 1):
MBean server connection type [ 1=JSR-160-Compliant Server, 2=WebSphere Application Server version 6.0,
3=WebSphere Application Server version 6.1, 4=WebSphere Application Server version 7.0 ] (default is: 1): 1
Edit 'JSR-160-Compliant Server' settings? [ 1=Yes, 2=No ] (default is: 1):
JMX user ID (default is: ):
Enter JMX password (default is: ):
Re-type : JMX password (default is: ):
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:port/objectgrid/MBeanServer):
-----
JMX Class Path Information
JMX base paths (default is: ):
JMX class path (default is: ):
JMX JAR directories (default is: ):
Edit 'WebSphere eXtreme Scale Catalog Service' settings? [ 1=Yes, 2=No ] (default is: 1): 2
Edit 'WebSphere eXtreme Scale Grid Servers' settings? [ 1=Yes, 2=No ] (default is: 1): 1
No 'WebSphere eXtreme Scale Grid Servers' settings available?
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 1
WebSphere eXtreme Scale Grid Servers (default is: ): rhea00b02_c0
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer):
service:jmx:rmi:///jndi/rmi://localhost:15000/objectgrid/MBeanServer

'WebSphere eXtreme Scale Grid Servers' settings: WebSphere eXtreme Scale Grid Servers=ogx
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 1
WebSphere eXtreme Scale Grid Servers (default is: ): rhea00b02_c1
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer):
service:jmx:rmi:///jndi/rmi://localhost:15001/objectgrid/MBeanServer

'WebSphere eXtreme Scale Grid Servers' settings: WebSphere eXtreme Scale Grid Servers= rhea00b02_c1
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 1
```

```

WebSphere eXtreme Scale Grid Servers (default is: ): rhea00b02_c2
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer):
service:jmx:rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer

'WebSphere eXtreme Scale Grid Servers' settings: WebSphere eXtreme Scale Grid Servers= rhea00b02_c2
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 5

Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1):
TEMS Host Name (Default is: rhea00b00):

Network Protocol [ip, sna, ip.pipe or ip.spipe] (Default is: ip.pipe):

    Now choose the next protocol number from one of these:
    - ip
    - sna
    - ip.spipe
    - 0 for none
Network Protocol 2 (Default is: 0):
IP.PIPE Port Number (Default is: 1918):
Enter name of KDC_PARTITION (Default is: null):

Configure connection for a secondary TEMS? [1=YES, 2=NO] (Default is: 2):
Enter Optional Primary Network Name or 0 for "none" (Default is: 0):
Agent configuration completed...

```

O exemplo anterior cria uma instância do agente chamada “inst1” e atualiza as configurações Java Home. Os servidores de contêiner do eXtreme Scale são configurados, mas o serviço de catálogo não é configurado.

**Nota:** O procedimento anterior cria um arquivo de texto no seguinte formato no diretório: <ITM\_install>/config/<host>\_xt\_<instance name>.cfg.

**Exemplo:** rhea00b02\_xt\_inst1.cfg

É recomendado editar esse arquivo com seu editor de texto simples escolhido. A seguir há um exemplo do conteúdo desse arquivo:

```

INSTANCE=inst2 [ SECTION=KQZ JAVA [ { JAVA_HOME=/opt/OG61/java } { JAVA_TRACE_LEVEL=ERROR } ]
SECTION=KQZ_JMX_CONNECTION_SECTION [ { KQZ_JMX_CONNECTION_PROPERTY=KQZ_JMX_JSR160_JSR160 } ]
SECTION=KQZ_JMX_JSR160_JSR160 [ { KQZ_JMX_JSR160_JSR160_CLASS_PATH_TITLE= }
{ KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:rmi:///jndi/rmi://localho
st:port/objectgrid/MBeanServer } { KQZ_JMX_JSR160_JSR160_CLASS_PATH_SEPARATOR= } ]
SECTION=OGS:rhea00b02_c1 [ { KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:
rmi:///jndi/rmi://localhost:15001/objectgrid/MBeanServer } ]
SECTION=OGS:rhea00b02_c0 [ { KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:
rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer } ]
SECTION=OGS:rhea00b02_c2 [ { KQZ_JMX_JSR160_JSR160_SERVICE_URL=service:jmx:
rmi:///jndi/rmi://localhost:15002/objectgrid/MBeanServer } ] ]

```

A seguir há um exemplo que mostra uma configuração de uma implementação do WebSphere Application Server:

```

rhea00b02 # ./itmcmd config -A xt
Agent configuration started...
Enter instance name (default is: ): inst1
Edit "Monitoring Agent for WebSphere eXtreme Scale" settings? [ 1=Yes, 2=No ] (default is: 1): 1
Edit 'Java' settings? [ 1=Yes, 2=No ] (default is: 1): 1
Java home (default is: C:\Program Files\IBM\Java50): /opt/WAS61/java
Java trace level [ 1=Error, 2=Warning, 3=Information, 4=Minimum Debug, 5=Medium Debug, 6=Maximum Debug,
7=All ] (default is: 1):
JVM arguments (default is: ):
Edit 'Connection' settings? [ 1=Yes, 2=No ] (default is: 1):
MBean server connection type [ 1=JSR-160-Compliant Server, 2=WebSphere Application Server version 6.0,
3=WebSphere Application Server version 6.1, 4=WebSphere Application Server version 7.0 ] (default is: 1): 4
Edit 'WebSphere Application Server version 7.0' settings? [ 1=Yes, 2=No ] (default is: 1): WAS user ID (default is: ):
Enter WAS password (default is: ):
Re-type : WAS password (default is: ):
WAS host name (default is: localhost): rhea00b02
WAS port (default is: 2809):
WAS connector protocol [ 1=rmi, 2=soap ] (default is: 1):
WAS profile name (default is: ): default
-----
WAS Class Path Information
WAS base paths (default is: C:\Program Files\IBM\WebSphere\AppServer;opt/IBM/WebSphere/AppServer): /opt/WAS61
WAS class path (default is: runtimes/com.ibm.ws.admin.client_6.1.0.jar;runtimes/com.ibm.ws.ejb.thinclient_7.0.0.jar):
WAS JAR directories (default is: lib;plugins):
Edit 'WebSphere eXtreme Scale Grid Servers' settings? [ 1=Yes, 2=No ] (default is: 1):
No 'WebSphere eXtreme Scale Grid Servers' settings available?

```

```

Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 1
WebSphere eXtreme Scale Grid Servers (default is: ): rhea00b02
JMX service URL (default is: service:jmx:rmi:///jndi/rmi://localhost:<port>/objectgrid/MBeanServer):

'WebSphere eXtreme Scale Grid Servers' settings: WebSphere eXtreme Scale Grid Servers=rhea00b02
Edit 'WebSphere eXtreme Scale Grid Servers' settings, [1=Add, 2=Edit, 3=Del, 4=Next, 5=Exit] (default is: 4): 5
Edit 'WebSphere eXtreme Scale Catalog Service' settings? [ 1=Yes, 2=No ] (default is: 1): 2
Will this agent connect to a TEMS? [1=YES, 2=NO] (Default is: 1):
TEMS Host Name (Default is: rhea00b02):

Network Protocol [ip, sna, ip.pipe or ip.spipe] (Default is: ip.pipe):

    Now choose the next protocol number from one of these:
    - ip
    - sna
    - ip.spipe
    - 0 for none
Network Protocol 2 (Default is: 0):
IP.PIPE Port Number (Default is: 1918):
Enter name of KDC_PARTITION (Default is: null):

Configure connection for a secondary TEMS? [1=YES, 2=NO] (Default is: 2):
Enter Optional Primary Network Name or 0 for "none" (Default is: 0):
Agent configuration completed...
rhea00b02 #

```

Para implementações do WebSphere Application Server, não é necessário criar vários subnós. O agente do eXtreme Scale se conecta como agente de nó para reunir todas as informações a partir dos servidores de aplicativos para os quais ele é responsável.

SECTION=CAT significa uma linha de serviço de catálogo, enquanto SECTION=OGS significa uma linha de configuração do servidor eXtreme Scale.

#### 5. Configure a porta JMX para todos os servidores de contêiner do eXtreme Scale.

Quando os servidores de contêiner do eXtreme Scale forem iniciados sem especificar o argumento **-JMXServicePort**, um servidor MBean será designado a uma porta dinâmica. O agente precisa saber antecipadamente com qual porta JMX ele se comunicará. O agente não trabalha com portas dinâmicas.

Ao iniciar os servidores, é necessário especificar o argumento **-JMXServicePort <port\_number>** ao iniciar o servidor eXtreme Scale usando o comando `startOgServer.sh | .bat`. Executar esse comando garante que o servidor JMX dentro do processo atenda em uma porta estática predefinida.

Para os exemplos anteriores em uma instalação UNIX, dois servidores eXtreme Scale precisarão ser iniciados com as portas configuradas:

- a. "-JMXServicePort" "15000" (para rhea00b02\_c0)
- b. "-JMXServicePort" "15001" (para rhea00b02\_c1)

#### a. Inicie o agente eXtreme Scale.

Supondo que a instância `inst1` foi criada, como no exemplo anterior, emita os seguintes comandos.

- 1) `cd <ITM_install>/bin`
- 2) `itmcmd agent -o inst1 start xt`

#### b. Pare o agente do eXtreme Scale.

Supondo que a instância "inst1" foi criada, como no exemplo anterior, emita os seguintes comandos.

- 1) `cd <ITM_install>/bin`
- 2) `itmcmd agent -o inst1 stop xt`

#### 6. Ative Estatísticas para todos os servidores de contêiner do eXtreme Scale.

O agente usa MBeans de estatísticas do eXtreme Scale para registrar estatísticas. A especificação de estatísticas do eXtreme Scale deve ser ativada usando um dos métodos a seguir.



- Configure propriedades do servidor para ativar todas as estatísticas quando os servidores de contêiner foram iniciados: `all=enabled`.
- Use o utilitário de amostra `xsadmin` para ativar estatísticas para todos os contêineres ativos usando os parâmetros `-setstatsspec all=enabled`.

## Resultados

Após todos os servidores serem configurados e iniciados, os dados do MBeans são exibidos no console do IBM Tivoli Portal. As áreas de trabalho predefinidas mostram gráficos e métricas de dados em cada nível de nó.

As seguintes áreas de trabalho são definidas: nó de **Servidores de Grade do eXtreme Scale** para todos os nós monitorados.

- Visualização Transações do eXtreme Scale
- Visualização Shard Primário do eXtreme Scale
- Visualização Memória do eXtreme Scale
- Visualização ObjectMap do eXtreme Scale

Também é possível configurar sua própria área de trabalho. Para obter mais informações, consulte as informações sobre a customização das áreas de trabalho no IBM Tivoli Monitoring Information Center.

## Monitorando Aplicativos eXtreme Scale com o CA Wily Introscope

O CA Wily Introscope é um produto de gerenciamento de terceiro que pode ser usado para detectar e diagnosticar problemas de desempenho em ambientes de aplicativos corporativos. O eXtreme Scale inclui detalhes sobre a configuração do CA Wily Introscope para fazer introspecção de partes selecionadas do tempo de execução do eXtreme Scale para visualizar e validar rapidamente os aplicativos do eXtreme Scale. O CA Wily Introscope funciona eficientemente para as implementações do WebSphere Application Server e independentes.

### Visão Geral

Para monitorar aplicativos eXtreme Scale com o CA Wily Introscope, é necessário colocar as configurações nos arquivos `ProbeBuilderDirective` (PBD) que fornecem acesso às informações de monitoramento para o eXtreme Scale.

**Atenção:** Os pontos de instrumentação para o Introscope podem ser alterados com cada fix pack ou release. Ao instalar um novo fix pack ou release, consulte a documentação para saber se há alguma alteração nos pontos de instrumentação.

É possível configurar os arquivos `ProbeBuilderDirective` (PBD) do CA Wily Introscope para monitorar seus aplicativos eXtreme Scale. O CA Wily Introscope é um produto de gerenciamento de aplicativos com o qual você pode detectar, selecionar e diagnosticar proativamente problemas de desempenho nos seus ambientes complexos, compostos e de aplicativos da Web.

### Configurações de Arquivos PBD para Monitoramento do Serviço de Catálogo

É possível utilizar uma ou mais das seguintes configurações em seu arquivo PBD para monitorar o serviço de catálogo.



```

TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl changeDefinedCompleted
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewChangeCompleted
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewAboutToChange
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeat
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCluster
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCurrentLeader
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatDeadServer
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewLeader
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewServer
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl
importRouteInfo BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl heartbeat
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl joinPlacementGroup
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}" TraceOneMethodOfClass:
com.ibm.ws.objectgrid.catalog.placement.PlacementServiceImpl classifyServer
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.BalanceGridEventListener shardActivated
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.catalog.placement.BalanceGridEventListener shardDeactivate
BlamePointTracerDifferentMethods "OGcatalog|{classname}|{method}"

```

## Classes para monitoramento do serviço de catálogo

### HAControllerImpl

A classe HAControllerImpl manipula eventos de ciclo de vida e feedback do grupo principal. É possível monitorar esta classe para obter uma indicação da estrutura e das alterações do grupo principal.

### ServerAgent

A classe ServerAgent é responsável pela comunicação de eventos do grupo principal com o serviço de catálogo. É possível monitorar as diversas chamadas de pulsação para marcar eventos graves.

### PlacementServiceImpl

A classe PlacementServiceImpl coordena os contêineres. É possível utilizar os métodos nesta classe para monitorar eventos de junção e disposição do servidor.

### BalanceGridEventListener

A classe BalanceGridEventListener controla a liderança de catálogo. É possível monitorar esta classe para obter uma indicação de qual serviço de catálogo está atualmente atuando como o líder.

## Configurações de Arquivos PBD para Monitoramento de Contêineres

É possível utilizar uma ou mais das seguintes configurações em seu arquivo PBD para monitorar os contêineres.

```

TraceOneMethodOfClass: com.ibm.ws.objectgrid.ShardImpl processMessage
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.plugins.CommittedLogSequenceListenerProxy applyCommitted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.plugins.CommittedLogSequenceListenerProxy sendApplyCommitted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.map.BaseMap evictMapEntries
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.checkpoint.CheckpointMapImpl$CheckpointIterator activateListener
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl changeDefinedCompleted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewChangeCompleted
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.hamanager.HAControllerImpl viewAboutToChange
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent batchProcess
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"

```

```

TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeat
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCluster
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatCurrentLeader
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatDeadServer
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewLeader
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.container.ServerAgent heartbeatNewServer
BlamePointTracerDifferentMethods "OGcontainer|{classname}|{method}"

```

## Classes para monitoramento dos contêineres

### ShardImpl

A classe `ShardImpl` possui o método `processMessage`. O método `processMessage` é o método para pedidos do cliente. Com este método, é possível obter contagens de tempo de resposta e de pedidos do lado do servidor. Ao observar as contas em todos os servidores e monitorar a utilização do heap, é possível determinar se a grade está equilibrada.

### CheckpointIterator

A classe `CheckpointIterator` possui a chamada de método `activateListener` que coloca primários no modo peer. Quando os primários são colocadas no modo peer, a réplica é atualizada com o primário após a conclusão do método. Quando uma réplica está se regenerando a partir de um primário completo, esta operação pode levar um período estendido de tempo. O sistema não é totalmente recuperado até a conclusão da operação, portanto, você pode utilizar esta classe para monitorar o progresso da operação.

### CommittedLogSequenceListenerProxy

A classe `CommittedLogSequenceListenerProxy` possui dois métodos de interesse. O método `applyCommitted` é executado para cada transação e o `sendApplyCommitted` é executado enquanto a réplica está executando pull de informações. A proporção de com que frequência estes dois métodos são executados pode fornecer a você alguma indicação de quão bem a réplica é capaz de continuar com o primário.

## Configurações de Arquivos PBD para Monitoramento dos Clientes

É possível utilizar uma ou mais das seguintes configurações em seu arquivo PBD para monitorar os clientes.

```

TraceOneMethodOfClass: com.ibm.ws.objectgrid.client.ORBClientCoreMessageHandler sendMessage
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.corba.cluster.ClusterStore bootstrap
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.corba.cluster.ClusterStore epochChangeBootstrap
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.map.BaseMap evictMapEntries
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.cluster.orb.routing.SelectionServiceImpl routeFailed
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.cluster.orb.routing.SelectionServiceImpl routeFailed
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.SessionImpl getMap
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TraceOneMethodOfClass: com.ibm.ws.objectgrid.ObjectGridImpl getSession
BlamePointTracerDifferentMethods "OGclient|{classname}|{method}"
TurnOn: ObjectMap
SetFlag: ObjectMap
IdentifyClassAs: com.ibm.ws.objectgrid.ObjectMapImpl ObjectMap
TraceComplexMethodsiffFlagged: ObjectMap BlamePointTracerDifferentMethods
"OGclient|{classname}|{method}"

```

## Classes para monitoramento de clientes

### ORBClientCoreMessageHandler

A classe `ORBClientCoreMessageHandler` é responsável por enviar pedidos

de aplicativos para os contêineres. É possível monitorar o método `sendMessage` para o tempo de resposta do cliente e o número de pedidos.

#### **ClusterStore**

A classe `ClusterStore` contém as informações de roteamento no lado do cliente.

#### **BaseMap**

A classe `BaseMap` possui o método `evictMapEntries` que é chamado quando o evictor deseja remover entradas do mapa.

#### **SelectionServiceImpl**

A classe `SelectionServiceImpl` toma as decisões de roteamento. Se o cliente está tomando decisões de failover, é possível utilizar esta classe para ver as ações que são concluídas a partir das decisões.

#### **ObjectGridImpl**

A classe `ObjectGridImpl` possui o método `getSession` que você pode monitorar para ver o número de pedidos para este método.

## **Monitorando o eXtreme Scale com o Hyperic HQ**

O Hyperic HQ é uma solução de monitoramento de terceiro que está disponível gratuitamente como uma solução de software livre ou como um produto corporativo. O WebSphere eXtreme Scale inclui um plug-in que permite que os agentes do Hyperic HQ descubram os servidores de contêiner do eXtreme Scale e relatem e agreguem as estatísticas usando os beans de gerenciamento do eXtreme Scale. É possível usar o Hyperic HQ para monitorar as implementações do eXtreme Scale independentes.

### **Antes de Iniciar**

- Esse conjunto de instruções destina-se para o Hyperic Versão 4.0. Se você tiver uma versão mais recente do Hyperic, consulte a Documentação do Hyperic para obter informações sobre os nomes de caminho e como iniciar os agentes e servidores.
- Download do servidor Hyperic e das instalações do agente. Uma instalação do servidor deve estar em execução. Para detectar todos os servidores eXtreme Scale, um agente Hyperic deve estar em execução em cada máquina na qual um servidor eXtreme Scale está em execução. Consulte o Web site Hyperic para obter informações de download e suporte para documentação.
- É necessário ter acesso aos arquivos `objectgrid-plugin.xml` e `hqplugin.jar`. Esses arquivos estão no diretório `wxs_install_root/hyperic/etc`.

### **Sobre Esta Tarefa**

Ao integrar o eXtreme Scale com o software de monitoramento Hyperic HQ, poderá monitorar e exibir graficamente as métricas sobre o desempenho do seu ambiente. Configure essa integração ao usar uma implementação de plug-in em cada agente.

### **Procedimento**

1. Inicie os servidores eXtreme Scale. O plug-in Hyperic procura nos processos locais para se conectar ao Java Virtual Machines que está executando o eXtreme Scale. Para se conectar corretamente com o Java Virtual Machines, cada servidor deve ser iniciado com a opção `-jmxServicePort`. Para obter informações sobre como iniciar os servidores com a opção `-jmxServicePort`, consulte “Script `start0gServer`” na página 393.

- Coloque os arquivos `extremescale-plugin.xml` e `wshyperic.jar` nos diretórios de plug-in do servidor e do agente apropriados na configuração Hyperic. Para integração com o Hyperic, ambas as instalações do agente e do servidor devem ter acesso ao plug-in e aos arquivos Java archive (JAR). Embora o servidor possa trocar as configurações dinamicamente, é necessário concluir a integração antes de iniciar qualquer um dos agentes.

- Coloque o arquivo `extremescale-plugin.xml` no diretório `plugin` do servidor, que está no seguinte local:

```
hyperic_home/server_home/hq-engine/server/default/deploy/hq.ear/hq-plugins
```

- Coloque o arquivo `extremescale-plugin.xml` no diretório `plugin` do agente, que está no seguinte local:

```
agent_home/bundles/gent-4.0.2-939/pdk/plugins
```

- Coloque o arquivo `wshyperic.jar` no diretório `lib` do agente, que está no seguinte local:

```
agent_home/bundles/gent-4.0.2-939/pdk/lib
```

- Configure o agente. O arquivo `agent.properties` atua como um ponto de configuração para o tempo de execução do agente. Essa propriedade está no diretório `agent_home/conf`. As seguintes chaves são opcionais, mas importantes para o plug-in do eXtreme Scale:

- `autoinventory.defaultScan.interval.millis=<time_in_milliseconds>`

Configura o intervalo em milissegundos entre as descobertas do Agente.

- `log4j.logger.org.hyperic.hq.plugin.extremescale.XSServerDetector=DEBUG`

: Ativa as instruções de depuração detalhadas a partir do plug-in eXtreme Scale.

- `username=<username>`: Configura o nome do usuário do Java Management Extensions (JMX) se a segurança estiver ativada.
- `password=<password>`: Configura a senha JMX se a segurança estiver ativada.
- `sslEnabled=<true|false>`: Informa se o plug-in deve usar ou não o Secure Sockets Layer (SSL). Por padrão, o valor é `false`.
- `trustPath=<path>`: Configura um caminho de confiança para a conexão SSL.
- `trustType=<type>`: Configura um tipo de confiança para a conexão SSL.
- `trustPass=<password>`: Configura uma senha de confiança para a conexão SSL.

- Inicie a descoberta do agente. Os agentes Hyperic enviam informações de descobertas e de métricas para o servidor. Utilize o servidor para customizar visualizações de dados e de objetos de inventário lógicos do grupo para gerar informações úteis. Depois que o servidor estiver disponível, é necessário executar o script de ativação ou iniciar o serviço do Windows para o agente:

- Linux** `agent_home/bin/hq-agent.sh start`

- Windows** Inicie o agente com o serviço do Windows.

Depois de iniciar os agentes, os servidores são detectados e os grupos configurados. É possível efetuar login no console do servidor e escolher quais recursos incluir no banco de dados de inventário para o servidor. Por padrão, o console do servidor está no seguinte URL: `http://<server_host_name>:7080/`

- As estatísticas devem ser ativadas para Hyperic para coletar dados estatísticos. Use a ação de controle `SetStatsSpec` no console do Hyperic para o eXtreme Scale. Navegue para o recurso e, em seguida, use a lista suspensa **Ação de**

**Controle** na página tabulada **Controle** para especificar uma configuração SetStatsSpec com ALL=enabled na caixa de texto **Argumentos de Controle**.

Os servidores de catálogos não são detectados pelo filtro configurado no console do Hyperic. Consulte as informações sobre a propriedade **statsSpec** no Arquivo de Propriedades do Servidor, que ativa as estatísticas assim que os contêineres forem iniciados. Várias opções para ativar estatísticas são descritas no “Monitoramento com Beans Gerenciados (MBeans)” na página 467 e no Amostra: Utilitário **xsadmin**

6. Monitorar servidores com o console do Hyperic. Depois que os servidores forem incluídos no modelo de inventário, os serviços não serão mais necessários.
  - **Visualização Painel:** Quando você visualizou os eventos de detecção de recursos, efetuou login na visualização de painel principal. A visualização de painel é uma visualização genérica que atua como um centro de mensagens que pode ser customizado. É possível exportar gráficos ou objetos de inventário para este painel principal.
  - **Visualização Recursos:** É possível consultar e visualizar o modelo de inventário inteiro a partir desta página. Após os serviços terem sido incluídos, é possível visualizar todos os servidores eXtreme Scale adequadamente rotulados e listados juntos sob a seção Servidores. É possível clicar nos servidores individuais para visualizar as métricas básicas.
7. Visualize o inventário do servidor inteiro na página Visualização de Recursos. Nesta página, você poderá selecionar vários servidores ObjectGrid e agrupá-los. Depois de agrupar um conjunto de recursos, as métricas comuns podem ser exibidas em gráfico para mostrar as sobreposições e diferenças entre os membros de grupo. Para exibir uma sobreposição, selecione as métricas na exibição do Grupo de Servidor. Em seguida, a métrica é exibida na área de gráfico. Para exibir uma sobreposição para todos os membros de grupo, clique no nome da métrica sublinhada. É possível exportar qualquer gráfico, visualizações de nó e sobreposições comparativas no painel principal com o menu **Ferramentas**.

---

## Monitorando as Informações do eXtreme Scale no DB2

Quando o JPALoader ou JPAEntityLoader é usado com DB2 como o banco de dados backend, as informações específicas do eXtreme Scale podem ser passadas para o DB2. É possível visualizar essas informações por meio de uma ferramenta do monitor de desempenho, como DB2 Performance Expert, para monitorar os aplicativos do eXtreme Scale que acessam o banco de dados.

### Antes de Iniciar

Consulte o “Coletando Rastreo” na página 526 para obter mais informações sobre os diferentes métodos que podem ser usados para configurar o rastreo.

### Sobre Esta Tarefa

Quando o carregador é configurado para usar o DB2 como o banco de dados de backend, as seguintes informações do eXtreme Scale podem ser passadas para o DB2 para fins de monitoramento:

- **Usuário:** Especifica o nome do usuário que se autentica no eXtreme Scale. Quando a autenticação básica não é usada, os principais da autenticação são usados.

- **Nome da Estação de Trabalho:** Especifica o nome do host, IP do servidor de contêiner do eXtreme Scale.
- **Nome do Aplicativo :** Especifica o nome da Unidade de Persistência do ObjectGrid (se configurado).
- **Informações da Conta:** Especifica o ID do encadeamento, o tipo de transação, o ID da transação e a sequência de conexão.

Leia sobre DB2 Performance Expert para saber como monitorar o acesso ao banco de dados.

## Procedimento

- Para ativar todas as informações do cliente do eXtreme Scale, configure as seguintes cadeias de rastreo:

```
ObjectGridClientInfo*=event=enabled
```

- Para ativar todas as informações, exceto as informações do usuário, use uma das seguintes configurações:

```
ObjectGridClientInfo*=event=enabled,ObjectGridClientInfoUser=event=disabled
```

ou

```
ObjectGridClientInfo=event=enabled
```

## Resultados

Depois de ativar a função de rastreo, os dados são exibidos na ferramenta do monitor de desempenho, como DB2 Performance Expert.

## Exemplo

No exemplo a seguir, o usuário bob é autenticado como um usuário do eXtreme Scale. O aplicativo acessa a grade de dados mygrid usando a unidade de persistência DB2Hibernate. O servidor de contêiner é XS\_Server1. As informações resultantes são:

- **Usuário**=bob
- **Nome da Estação de Trabalho**=XS\_Server1,192.168.1.101
- **Nome do Aplicativo**=mygrid,DB2Hibernate
- **Dados da Conta**=1, DEFAULT, FE7954BD-0126-4000-E000-2298094151DB,com.ibm.db2.jcc.t4.b@71787178

No exemplo a seguir, o usuário bob é autenticado usando um token WebSphere Application Server. O aplicativo acessa a grade de dados mygrid usando o nome da unidade de persistência DB2OpenJPA. O servidor de contêiner é XS\_Server2. As informações resultantes são:

- **Usuário**

```
=acme.principal.UserPrincipal[Bob],acme.principal.GroupPrincipal[admin]
```
- **Nome da Estação de Trabalho**=XS\_Server2,192.168.1.102
- **Nome do Aplicativo**=mygrid,DB2OpenJPA
- **Dados da Conta**=188,DEFAULT, FE72BC63-0126-4000-E000-851C092A4E33,com.ibm.ws.rsadapter.jdbc.WSJccSQLJConnection@2b432b43



---

## Capítulo 9. Ajuste do desempenho



É possível ajustar as configurações em seu ambiente para aumentar o desempenho geral de seu ambiente do WebSphere eXtreme Scale.

---

### Ajustando Sistemas Operacionais e Configurações de Rede

O ajuste da rede pode reduzir o atraso da pilha do Protocolo de Controle de Transmissões (TCP) por meio da mudança das configurações de conexão e melhorar o rendimento por meio da mudança dos buffers TCP.

#### Sistemas Operacionais

Um sistema Windows precisa do mínimo de ajuste enquanto que o sistema Solaris precisa do máximo de ajuste. As informações a seguir referem-se a cada sistema especificado e podem aumentar o desempenho do WebSphere eXtreme Scale. Você deve ajustar de acordo com sua carga de rede e do aplicativo.

##### Windows

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
Tcpip\Parameters
MaxFreeTcbs = dword:00011940
MaxHashTableSize = dword:00010000
MaxUserPort = dword:0000ffff
TcpTimedWaitDelay = dword:0000001e
```

##### Solaris

```
nnd -set /dev/tcp tcp_time_wait_interval 60000
fndd -set /dev/tcp tcp_keepalive_interval 15000
nnd -set /dev/tcp tcp_fin_wait_2_flush_interval 67500
nnd -set /dev/tcp tcp_conn_req_max_q 16384
nnd -set /dev/tcp tcp_conn_req_max_q0 16384
nnd -set /dev/tcp tcp_xmit_hiwat 400000
nnd -set /dev/tcp tcp_recv_hiwat 400000
nnd -set /dev/tcp tcp_cwnd_max 2097152
nnd -set /dev/tcp tcp_ip_abort_interval 20000
nnd -set /dev/tcp tcp_rexmit_interval_initial 4000
nnd -set /dev/tcp tcp_rexmit_interval_max 10000
nnd -set /dev/tcp tcp_rexmit_interval_min 3000
nnd -set /dev/tcp tcp_max_buf 4194304
```

##### AIX

```
/usr/sbin/no -o tcp_sendspace=65536
/usr/sbin/no -o tcp_recvspace=65536
/usr/sbin/no -o udp_sendspace=65536
/usr/sbin/no -o udp_recvspace=65536
/usr/sbin/no -o somaxconn=10000
/usr/sbin/no -o tcp_nodelayack=1
/usr/sbin/no -o tcp_keepinit=40
/usr/sbin/no -o tcp_keepintvl=10
```

##### LINUX

```
sysctl -w net.ipv4.tcp_timestamps=0
sysctl -w net.ipv4.tcp_tw_reuse=1
sysctl -w net.ipv4.tcp_tw_recycle=1
sysctl -w net.ipv4.tcp_fin_timeout=30
sysctl -w net.ipv4.tcp_keepalive_time=1800
sysctl -w net.ipv4.tcp_rmem="4096 87380 8388608"
sysctl -w net.ipv4.tcp_wmem="4096 87380 8388608"
sysctl -w net.ipv4.tcp_max_syn_backlog=4096
```

##### HP-UX

```
nnd -set /dev/tcp tcp_ip_abort_cinterval 20000
```



---

## Propriedades do ORB

As propriedades do Object Request Broker (ORB) modificam o comportamento do transporte da grade de dados. Essas propriedades podem ser configuradas com um arquivo `orb.properties`, como configurações no console administrativo do WebSphere Application Server ou como propriedades customizadas no ORB no console administrativo do WebSphere Application Server.

### **orb.properties**

O arquivo `orb.properties` está no diretório `java/jre/lib`. Quando você modifica o arquivo `orb.properties` em um diretório `java/jre/lib` do WebSphere Application Server, as propriedades do ORB são atualizadas no agente do nó e quaisquer outras Java virtual machines (JVMs) que estão utilizando o Java runtime environment (JRE). Se você desejar esse comportamento, use as propriedades customizadas ou as configurações do ORB no console administrativo do WebSphere Application Server.

### **Configurações Padrão do WebSphere Application Server**

O WebSphere Application Server possui algumas propriedades definidas no ORB por padrão. Essas configurações estão nos serviços de contêiner do servidor de aplicativos e no gerenciador de implementação. Essas definições padrão substituem quaisquer configurações criadas no arquivo `orb.properties`. Para cada propriedade descrita, consulte a seção **Onde especificar** para determinar o local para definir o valor sugerido.

### **Configurações do Descritor de Arquivo**

Para sistemas UNIX e Linux, existe um limite para o número de arquivos abertos que são permitidos por processo. O sistema operacional especifica a quantidade de arquivos abertos permitidos. Se este valor for configurado muito baixo, ocorrerá um erro de alocação de memória no AIX e muitos arquivos abertos serão registrados no log.

Na janela de terminal do sistema UNIX, configure este valor para um valor mais alto que o valor do sistema padrão. Para máquinas SMP maiores com clones, configure para ilimitado.

Para configurações AIX, configure este valor como -1 (ilimitado) com o comando: `ulimit -n -1`.

Para configurações Solaris, configure este valor como 16384 com o comando: `ulimit -n 16384`.

Para exibir o valor atual, use o comando: `ulimit -a`.

### **Configurações de Linha de Base**

As seguintes configurações são uma linha de base ideal mas não necessariamente as melhores configurações para cada ambiente. Entenda as configurações para ajudar a tomar uma boa decisão sobre quais valores são apropriados no seu ambiente.

```
com.ibm.CORBA.RequestTimeout=30
com.ibm.CORBA.ConnectTimeout=10
com.ibm.CORBA.FragmentTimeout=30
```

```
com.ibm.CORBA.LocateRequestTimeout=10
com.ibm.CORBA.ThreadPool.MinimumSize=256
com.ibm.CORBA.ThreadPool.MaximumSize=256
com.ibm.CORBA.ThreadPool.IsGrowable=false
com.ibm.CORBA.ConnectionMultiplicity=1
com.ibm.CORBA.MinOpenConnections=1024
com.ibm.CORBA.MaxOpenConnections=1024
com.ibm.CORBA.ServerSocketQueueDepth=1024
com.ibm.CORBA.FragmentSize=0
com.ibm.CORBA.iiop.NoLocalCopies=true
com.ibm.CORBA.NoLocalInterceptors=true
```

## Descrições de Propriedades

### Configurações de Tempo Limite

As seguintes configurações referem-se à quantia de tempo em que o ORB aguardará antes que as operações sobre pedidos sejam canceladas. Use essas configurações para evitar que encadeamentos em excesso sejam criados em uma situação anormal.

#### Tempo Limite do Pedido

**Nome da Propriedade:** com.ibm.CORBA.RequestTimeout

**Valor válido:** Valor de número inteiro para o número de segundos.

**Valor sugerido :** 30

**Onde especificar:** Console administrativo do WebSphere Application Server

**Descrição:** Indica quantos segundos uma solicitação aguarda por uma resposta antes de ser cancelada. Essa propriedade influencia a quantia de tempo em que um cliente precisa para efetuar failover se ocorrer uma falha de interrupção de rede. Se você configurar essa propriedade muito baixo, os pedidos poderão atingir o tempo limite involuntariamente. Considere cuidadosamente o valor dessa propriedade para evitar tempos limites acidentais.

#### Tempo Limite de Conexão

**Nome da Propriedade:** com.ibm.CORBA.ConnectTimeout

**Valor válido:** Valor de número inteiro para o número de segundos.

**Valor sugerido :** 10

**Onde especificar:** arquivo orb.properties

**Descrição:** Indica quantos segundos uma tentativa de conexão de soquete aguarda antes de ser cancelada. Essa propriedade, como o tempo limite de pedido, pode influenciar no tempo em que um cliente precisa para efetuar failover se ocorrer uma falha de interrupção de rede. Em geral, configure essa propriedade para um valor menor do que o valor do tempo limite de solicitação porque a quantia de tempo para estabelecer conexões é relativamente constante.

#### Tempo Limite de Fragmento

**Nome da Propriedade:** com.ibm.CORBA.FragmentTimeout

**Valor válido:** Valor de número inteiro para o número de segundos.

**Valor sugerido :** 30

**Onde especificar:** arquivo orb.properties

**Descrição:** Indica quantos segundos uma solicitação de fragmento aguarda antes de ser cancelada. Essa propriedade é semelhante à propriedade de tempo limite de pedido.

### Configurações do Conjunto de Encadeamentos

Essas propriedades restringem o tamanho do conjunto de encadeamentos a um número específico de encadeamentos. Os encadeamentos são usados pelo ORB para distribuir os pedidos do servidor depois de serem recebidos no soquete. Configurar esses valores da propriedade muito baixo resulta em um aumento de profundidade da fila de soquete e possivelmente em tempos limites.

#### Multiplicidade de Conexão

**Nome da Propriedade:** com.ibm.CORBA.ConnectionMultiplicity

**Valor válido:** Valor de número inteiro para número de conexões entre o cliente e o servidor. O valor padrão é 1. Configurar um valor maior define uma multiplicidade entre várias conexões.

**Valor sugerido :** 1

**Onde especificar:** arquivo orb.properties **Descrição:** Permite que o ORB use várias conexões em qualquer servidor. Na teoria, configurar esse valor promove paralelismo sobre as conexões. Na prática, o desempenho não é beneficiado pela configuração da multiplicidade de conexão. Não configure esse parâmetro.

#### Conexões Abertas

**Nomes da Propriedade:** com.ibm.CORBA.MinOpenConnections, com.ibm.CORBA.MaxOpenConnections

**Valor válido:** Um valor de número inteiro para número de conexões.

**Valor sugerido :** 1024

**Onde especificar:** Console administrativo do WebSphere Application Server **Descrição:** Especifica um número mínimo e máximo de conexões abertas. O ORB mantém um cache de conexões que foram estabelecidas com os clientes. Essas conexões são limpas quando este valor for transmitido. A limpeza das conexões pode prejudicar o comportamento na grade de dados.

#### É Aumentável

**Nome da Propriedade:** com.ibm.CORBA.ThreadPool.IsGrowable

**Valor válido:** booleano; configure para true ou false.

**Valor sugerido :** false

**Onde especificar:** arquivo orb.properties **Descrição:** Se configurado como true, o conjunto de encadeamentos que o ORB usa para solicitações recebidas pode crescer além do que o conjunto suporta. Se o tamanho do conjunto for excedido, novos encadeamentos serão criados para manipular o pedido, mas os encadeamentos não serão agrupados. Evite o crescimento do conjunto de encadeamentos ao definir o valor para false.

#### Profundidade da Fila de Soquete do Servidor

**Nome da Propriedade:** com.ibm.CORBA.ServerSocketQueueDepth

**Valor válido:** Um valor de número inteiro para número de conexões.

**Valor sugerido :** 1024

**Onde especificar:** arquivo orb.properties **Descrição:** Especifica o comprimento da fila para conexões recebidas dos clientes. As filas ORB recebem conexões dos clientes. Se a fila estiver cheia, as conexões serão recusadas. Recusar conexões pode prejudicar o comportamento na grade de dados.

### Tamanho do Fragmento

**Nome da Propriedade:** com.ibm.CORBA.FragmentSize

**Valor válido:** Um número inteiro que especifica o número de bytes. O padrão é 1024.

**Valor sugerido :** 0

**Onde especificar:** arquivo orb.properties **Descrição:** Especifica o tamanho máximo do pacote que o ORB usa ao enviar um pedido. Se um pedido for maior que o limite de tamanho de fragmento, esse pedido será dividido em fragmentos de pedido e enviados, cada um, separadamente e remontados no servidor. Os pedidos de fragmentação são úteis em redes não-confiáveis onde os pacotes podem precisar ser reenviados. No entanto, se a rede for confiável, dividir as solicitações em fragmentos pode causar processamento desnecessário.

### Nenhuma Cópia Local

**Nome da Propriedade:** com.ibm.CORBA.iiop.NoLocalCopies

**Valor válido:** booleano; configure para true ou false.

**Valor sugerido :** true

**Onde especificar:** No console administrativo do WebSphere Application Server, configuração **Transmitir por Referência**. **Descrição:** Especifica se o ORB é transmitido por referência. O ORB usa a chamada transmitir por valor por padrão. A chamada Transmitir por valor causa custos extras de lixo e de serialização para o caminho quando uma interface é iniciada localmente. Ao configurar esse valor para true, o ORB usa um método transmitir por referência que é mais eficiente do que a chamada transmitir por valor.

### Nenhum Interceptor Local

**Nome da Propriedade:** com.ibm.CORBA.NoLocalInterceptors

**Valor válido:** booleano; configure para true ou false.

**Valor sugerido :** true

**Onde especificar:** arquivo orb.properties **Descrição:** Especifica se o ORB inicia os interceptores de solicitação mesmo ao fazer solicitações locais (intraprocessos). Os interceptores que o WebSphere eXtreme Scale usa são para segurança e os identificadores de rota não serão necessários se o pedido for tratado dentro do processo. Os interceptores que estão entre os processos são necessários apenas para as operações de chamada de procedimento remoto (RPC). Ao configurar 'Nenhum interceptor local', é possível evitar o processamento extra ao usar introduções de interceptores locais.

**Atenção:** Se a segurança do WebSphere eXtreme Scale estiver sendo usada, configure o valor da propriedade `com.ibm.CORBA.NoLocalInterceptors` para `false`. A infraestrutura de segurança usa interceptores para autenticação.

---

## Ajustando as Java virtual machines

Você deve levar em conta vários aspectos específicos do ajuste da Java Virtual Machine (JVM) para melhorar o desempenho do WebSphere eXtreme Scale. Na maioria dos casos, poucas ou nenhuma configuração da JVM é necessária. Se muitos objetos estiverem sendo armazenados na grade de dados, ajuste o tamanho de heap para um nível apropriado a fim de evitar execução sem memória.

**7.1.1+** Ao configurar o `eXtremeMemory`, é possível armazenar os objetos na memória nativa em vez de armazenar no heap Java. Configurar o `eXtremeMemory` ativa o `eXtremeIO` como um novo mecanismo de transporte. Movendo objetos para fora do heap Java, é possível evitar pausas da coleta de lixo, levando ao desempenho mais constante e tempos de resposta atribuíveis. Para obter informações adicionais, consulte “Configurando o IBM eXtremeMemory e o IBM eXtremeIO” na página 273.

### Plataformas Testadas

Os testes de desempenho ocorreram principalmente no AIX (32 vias), Linux (quatro vias) e computadores Windows (oito vias). Com computadores AIX sofisticados, é possível testar fortemente cenários multiencadeados para identificar e corrigir os pontos de contenção.

### Coleta de Lixo

O WebSphere eXtreme Scale cria objetos temporários associados a cada transação, como pedido e resposta e sequência de log. Como esses objetos afetam a eficiência da coleta de lixo, o ajuste da coleta de lixo é fundamental.

Todas as JVMs modernas usam algoritmos de coleta de lixo paralela, o que significa que o uso de mais núcleos pode reduzir pausas na coleta de lixo. A coleta de lixo em um servidor físico com oito núcleos é mais rápida do que em um servidor físico com quatro núcleos.

Quando o aplicativo deve gerenciar uma grande quantidade de dados para cada partição, a coleta de lixo pode ser um fator. Na maioria das vezes, um cenário de leitura tem um bom desempenho mesmo com heaps grandes (20 GB ou mais) se um coletor geracional é utilizado. Contudo, depois que o heap de estabilidade é preenchido, ocorre uma pausa proporcional ao tamanho do heap ativo e ao número de processadores no computador. Essa pausa pode ser maior ou menor em máquinas com grandes heaps.

### Máquina Virtual IBM para Coleta de Lixo Java

Para a máquina virtual IBM para Java, use o coletor `optavgpause` para cenários com alta taxa de atualização (100% das entradas de modificação de transações). O coletor `gencon` funciona muito melhor que o coletor `optavgpause` nos cenários em que os dados são atualizados relativamente com pouca frequência (10% do tempo ou menos). Experimente ambos os coletores para ver qual funciona melhor no seu cenário. Executar com a coleta de lixo detalhada ativada para verificar a

porcentagem de tempo gasto na coleta de lixo. Ocorreram cenários em que 80% do tempo foi gasto na coleta de lixo até que um que o ajuste resolvesse o problema.

Use o parâmetro **-Xgcpolicy** para alterar o mecanismo de coleta de lixo. O valor do parâmetro **-Xgcpolicy** pode ser configurado como: **-Xgcpolicy:gencon** ou **-Xgcpolicy:optavgpause**, dependendo de qual coletor de lixo você deseja usar.

- Em uma configuração do WebSphere Application Server, configure o parâmetro **-Xgcpolicy** no console administrativo. Clique em **Servidores > Servidores de Aplicativos > server\_name > Definição de Processo > Java Virtual Machine**. Inclua o parâmetro no campo **Argumentos da JVM Genéricos**.
- Em uma configuração independente, transmita o parâmetro **-jvmArgs** para o script **startOgServer** para especificar o coletor de lixo. O parâmetro **-jvmArgs** deve ser o último parâmetro que é transmitido ao script.

## Outras Opções de Coleta de Lixo

**Atenção:** Se estiver usando uma JVM Sun, ajustes podem ser necessários na coleta de lixo padrão e na política.

O WebSphere eXtreme Scale suporta o WebSphere Real Time Java. Com o WebSphere Real Time Java, a resposta de processamento de transação do WebSphere eXtreme Scale é mais consistente e previsível. Como resultado, o impacto da coleta de lixo e do planejamento do encadeamento é enormemente minimizado. O impacto é reduzido ao nível no qual o tempo de desvio de resposta padrão é inferior a 10% da Java regular.

## Desempenho da JVM

O WebSphere eXtreme Scale pode ser executado em versões diferentes do Java Platform, Standard Edition. O WebSphere eXtreme Scale suporta o Java SE Versão 5 e posterior. Para obter produtividade e desempenho de desenvolvedor aprimorados, use o Java SE Versão 5 ou posterior para obter vantagem das anotações e da coleta de lixo aprimorada. O WebSphere eXtreme Scale funciona em Java virtual machines de 32 e 64 bits.

O WebSphere eXtreme Scale é testado com um subconjunto das máquinas virtuais disponíveis, todavia, a lista suportada não é exclusiva. É possível executar o WebSphere eXtreme Scale em qualquer JVM do fornecedor na Edição 5 ou posterior. No entanto, se ocorrer um problema com uma JVM do fornecedor, você deverá contatar o fornecedor da JVM para obter suporte. Se possível, use a JVM a partir do tempo de execução do WebSphere em qualquer plataforma que o WebSphere Application Server suporta.

Para a maioria dos cenários nos quais o WebSphere eXtreme Scale é usado, o Java SE Versão 6 da JVM funciona melhor do que a Edição 5. Em geral, use a versão mais recente disponível do Java Platform, Standard Edition para melhor desempenho.

## Tamanho de Heap

A recomendação é heaps de 1 a 2 GB com uma JVM para cada quatro núcleos. O número do tamanho de heap ideal depende dos seguintes fatores:

- Quantidade de objetos ativos no heap.
- Complexidade dos objetos ativos no heap.
- Quantidade de núcleos disponíveis para a JVM.

Por exemplo, um aplicativo que armazena matrizes de 10 KB pode executar um heap muito maior do que um aplicativo que usa gráficos complexos de POJOs.

## Contagem de Encadeamentos

A contagem de encadeamentos depende de poucos fatores. Há um limite de quantos encadeamentos um único shard pode gerenciar. Um shard é uma instância de uma partição e pode ser primário ou de réplica. Com mais shards para cada JVM, você possui mais encadeamentos com cada shard adicional fornecendo mais caminhos simultâneos para os dados. Embora cada shard é simultâneo o máximo possível, essa simultaneidade ainda é limitada.

## Requisitos do Object Request Broker (ORB)

O IBM SDK inclui uma implementação IBM ORB que foi testada com o WebSphere Application Server e o WebSphere eXtreme Scale. Para facilitar o processo de suporte, use um JVM fornecido pela IBM. Outras implementações de JVM utilizam um ORB diferente. O IBM ORB é fornecido apenas com Java virtual machines fornecidas pela IBM. O WebSphere eXtreme Scale requer um ORB em funcionamento para operar. O WebSphere eXtreme Scale pode ser usado com ORBs a partir de outros fornecedores. No entanto, se você tiver um problema com um ORB do fornecedor, você deve entrar em contato com o fornecedor do ORB para obter suporte. A implementação do IBM ORB é compatível com Java virtual machines de terceiros e pode ser substituído, se necessário.

## Ajuste de orb.properties

No laboratório, o seguinte arquivo foi usado nas grades de dados de até 1500 JVMs. O arquivo orb.properties está na pasta lib do ambiente de tempo de execução.

```
# IBM JDK properties for ORB
org.omg.CORBA.ORBClass=com.ibm.CORBA.iiop.ORB
org.omg.CORBA.ORBSingletonClass=com.ibm.rmi.corba.ORBSingleton

# WS Interceptors
org.omg.PortableInterceptor.ORBInitializerClass=com.ibm.ws.objectgrid.corba.ObjectGridInitializer

# WS ORB & Plugins properties
com.ibm.CORBA.ForceTunnel=never
com.ibm.CORBA.RequestTimeout=10
com.ibm.CORBA.ConnectTimeout=10

# Needed when lots of JVMs connect to the catalog at the same time
com.ibm.CORBA.ServerSocketQueueDepth=2048

# Clients and the catalog server can have sockets open to all JVMs
com.ibm.CORBA.MaxOpenConnections=1016

# Thread Pool for handling incoming requests, 200 threads here
com.ibm.CORBA.ThreadPool.IsGrowable=false
com.ibm.CORBA.ThreadPool.MaximumSize=200
com.ibm.CORBA.ThreadPool.MinimumSize=200
com.ibm.CORBA.ThreadPool.InactivityTimeout=180000

# No splitting up large requests/responses in to smaller chunks
com.ibm.CORBA.FragmentSize=0
```

---

## Ajustando a Configuração do Intervalo de Pulsação para Detecção de Failover

É possível configurar a quantidade de tempo entre verificações do sistema para servidores com falha com a configuração do intervalo de pulsação.



## Sobre Esta Tarefa

A configuração de failover varia dependendo do tipo de ambiente que você está usando. Se você estiver utilizando um ambiente independente, é possível configurar o failover com a linha de comandos. Se você estiver usando um ambiente do WebSphere Application Server Network Deployment, é necessário configurar o failover no console administrativo do WebSphere Application Server Network Deployment.

## Procedimento

- Configurar o failover para ambientes independentes.

É possível configurar intervalos de pulsação na linha de comandos ao usar o parâmetro **-heartbeat** no arquivo de script **startOgServer**. Configure esse parâmetro para um dos seguintes valores:

Tabela 30. Intervalos de Pulsações

| Valor | Ação            | Descrição                                            |
|-------|-----------------|------------------------------------------------------|
| 0     | Típica (padrão) | Failovers são tipicamente detectados em 30 segundos. |
| -1    | Agressiva       | Failovers são tipicamente detectados em 5 segundos.  |
| 1     | Moderada        | Failovers são tipicamente detectado em 180 segundos. |

Um intervalo de pulsação agressivo pode ser útil quando os processos e a rede estão estáveis. Se a rede ou os processos não são configurados de maneira ideal, as pulsações podem ser perdidas, o que pode resultar em uma falsa detecção de falhas.

- Configurar o failover para ambientes do WebSphere Application Server.

O WebSphere Application Server Network Deployment Versão 6.0.2 e superior pode ser configurado para permitir que o WebSphere eXtreme Scale execute failover muito rapidamente. O tempo de failover padrão para falhas "hard" é de aproximadamente 200 segundos. Uma falha de disco rígido é um travamento de computador ou de um servidor físico, uma desconexão de cabo de rede ou um erro do sistema operacional. As falhas causadas por travamentos de processo ou por falhas de software normalmente executam failover em menos de um segundo. A detecção de falha para falhas "soft" ocorre quando os soquetes de rede a partir de um processo inativo são fechados automaticamente pelo sistema operacional para o servidor que hospeda o processo.

### Configuração de pulsação do grupo principal

O WebSphere eXtreme Scale que executa um processo do WebSphere Application Server herda as características de failover a partir das configurações de grupo principal do servidor de aplicativos. As seguintes seções descrevem como definir as configurações de pulsação do grupo principal para versões diferentes do WebSphere Application Server Network Deployment:

- **Atualize as configurações do grupo principal para o WebSphere Application Server Network Deployment Versão 6.x e 7.x:**

O intervalo de pulsação pode ser especificado em segundos nas versões do WebSphere Application Server da Versão 6.0 a Versão 6.1.0.12, ou em milissegundos iniciando na Versão 6.1.0.13. Também é necessário especificar o número de pulsações perdidas. Esse valor indica quantas pulsações podem estar ausentes antes que um Java Virtual Machine (JVM) equivalente seja considerado uma falha. O tempo de detecção de falha "hard" é, aproximadamente, o produto do intervalo de pulsações e o número de pulsações ausentes.

Essas propriedades são especificadas usando as propriedades customizadas no grupo principal usando o console administrativo do WebSphere. Consulte Propriedades customizadas do grupo principal para obter detalhes da configuração. Estas propriedades devem ser especificadas para todos os grupos principais utilizados pelo aplicativo:

- O intervalo de pulsação é especificado usando a propriedade customizada IBM\_CS\_FD\_PERIOD\_SEC para segundos ou a propriedade customizada IBM\_CS\_FD\_PERIOD\_MILLIS para milissegundos (requer a Versão 6.1.0.13 ou posterior).
- O número de pulsações ausentes é especificado usando a propriedade customizada IBM\_CS\_FD\_CONSECUTIVE\_MISSED.

O valor padrão para a propriedade IBM\_CS\_FD\_PERIOD\_SEC é 20 e para a propriedade IBM\_CS\_FD\_CONSECUTIVE\_MISSED é 10. Se a propriedade IBM\_CS\_FD\_PERIOD\_MILLIS for especificada, ela substituirá qualquer conjunto de propriedades customizadas IBM\_CS\_FD\_PERIOD\_SEC. Os valores destas propriedades são valores de número inteiro positivo.

Use as seguintes configurações para alcançar um tempo de detecção de falha de 1500 milissegundos para os servidores WebSphere Application Server Network Deployment Versão 6.x:

- Configure IBM\_CS\_FD\_PERIOD\_MILLIS = 750 (WebSphere Application Server Network Deployment V6.1.0.13 e superior)
  - Configure IBM\_CS\_FD\_CONSECUTIVE\_MISSED = 2
- **Atualize as configurações do grupo principal para o WebSphere Application Server Network Deployment Versão 7.0:**

O WebSphere Application Server Network Deployment Versão 7.0 fornece duas configurações de grupo principal que podem ser ajustadas para aumentar ou diminuir a detecção de failover:

- **Período de transmissão de pulsação.** O padrão é 30000 milissegundos.
- **Período de tempo limite de pulsação.** O padrão é 180000 milissegundos.

Para obter mais detalhes sobre como alterar essas configurações, consulte o Centro de Informações do WebSphere Application Server Network Deployment: Configurações de Falha e Detecção de Descoberta.

Use as seguintes configurações para alcançar um tempo de detecção de falha de 1500 milissegundos para os servidores WebSphere Application Server Network Deployment Versão 7:

- Configure o período de transmissão de pulsação para 750 milissegundos.
- Configure o tempo limite da pulsação para 1500 milissegundos.

## O que Fazer Depois

Quando estas configurações são modificadas para fornecer tempos de failover curtos, há alguns problemas de ajuste de sistema a considerar. Primeiro, Java não é um ambiente em tempo real. É possível que os encadeamentos sejam atrasados se a JVM estiver experimentando longos tempos de coleta de lixo. Os encadeamentos também podem ser atrasados se a máquina que hospeda o JVM estiver sobrecarregada (devido ao próprio JVM ou outros processos que são executados na máquina). Se os encadeamentos forem atrasados, as pulsações talvez não sejam enviadas a tempo. No pior dos casos, elas podem ser atrasadas pelo tempo necessário de failover. Se os encadeamentos forem atrasados, ocorrerão falsas detecções de falhas. O sistema deve ser ativado e dimensionado para garantir que falsas detecções de falhas não aconteçam na produção. O teste de carregamento adequado é a melhor maneira de garantir isto.

**Nota:** A versão atual do eXtreme Scale suporta o WebSphere Real Time.

---

## Ajustando a Coleta de Lixo com o WebSphere Real Time

O uso do WebSphere eXtreme Scale com WebSphere Real Time aumenta a consistência e a previsibilidade a um custo de rendimento de desempenho em comparação com a política de coleta de lixo padrão empregada no IBM Java™ SE Runtime Environment (JRE) padrão. O custo versus a proposta de benefício pode variar. WebSphere eXtreme Scale cria vários objetos temporários que são associados a cada transação. Esses objetos temporários lidam com pedidos, respostas, sequências de log e sessões. Sem o WebSphere Real Time, o tempo de resposta de transação pode ultrapassar centenas de milissegundos. Entretanto, usar o WebSphere Real Time com o WebSphere eXtreme Scale pode aumentar a eficiência da coleta de lixo e reduzir o tempo de resposta em 10% do tempo de resposta de configuração independente.

### WebSphere Real Time em um Ambiente Independente

É possível usar o WebSphere Real Time com o WebSphere eXtreme Scale. Ao ativar o WebSphere Real Time, é possível obter uma coleta de lixo mais previsível com um tempo de resposta e rendimento estável e consistente em um ambiente independente do eXtreme Scale.

#### Vantagens do WebSphere Real Time

WebSphere eXtreme Scale cria vários objetos temporários que são associados a cada transação. Esses objetos temporários lidam com pedidos, respostas, sequências de log e sessões. Sem o WebSphere Real Time, o tempo de resposta de transação pode ultrapassar centenas de milissegundos. Entretanto, usar o WebSphere Real Time com o WebSphere eXtreme Scale pode aumentar a eficiência da coleta de lixo e reduzir o tempo de resposta em 10% do tempo de resposta de configuração independente.

#### Ativando o WebSphere Real Time

Instale o WebSphere Real Time e o WebSphere eXtreme Scale independente nos computadores onde você planeja executar o eXtreme Scale . Configure a variável de ambiente JAVA\_HOME para apontar para o Java SE Runtime Environment (JRE) padrão.

Configure a variável de ambiente JAVA\_HOME para apontar para o WebSphere Real Time instalado. Em seguida, ative o WebSphere Real Time da seguinte forma.

1. Edite o arquivo de instalação padrão `objectgridRoot/bin/setupCmdLine.sh | .bat` ao remover o comentário da seguinte linha.  

```
WXS_REAL_TIME_JAVA="-Xrealtime -Xgcpolicy:metronome  
-Xgc:targetUtilization=80"
```
2. Salve o arquivo.

Agora você ativou o WebSphere Real Time. Se desejar desativar o WebSphere Real Time, poderá incluir o comentário de volta para a mesma linha.

#### Práticas Recomendáveis

WebSphere Real Time permite que transações do eXtreme Scale tenham um tempo de resposta mais previsível. Os resultados mostram que o desvio de um tempo de resposta de uma transação do eXtreme Scale aumenta significativamente com o

WebSphere Real Time em comparação com o Java padrão com seu coletor de lixo padrão. A ativação do WebSphere Real Time com o eXtreme Scale é ótima se a estabilidade e o tempo de resposta de seu aplicativo forem essenciais.

As boas práticas descritas nesta seção explicam como tornar o WebSphere eXtreme Scale mais eficiente por meio do ajuste e de práticas de código dependendo de sua carga esperada.

- Configure o nível de uso correto do processador para o aplicativo e coletor de lixo.

O WebSphere Real Time fornece a capacidade de controlar o uso do processador para que o impacto da coleta de lixo em seu aplicativo seja controlado e minimizado. Use o parâmetro `-Xgc:targetUtilization=NN` para especificar a porcentagem NN do processador que é usado pelo aplicativo a cada 20 segundos. O padrão para WebSphere eXtreme Scale é 80%, mas é possível modificar o script no arquivo `objectgridRoot/bin/setupCmdLine.sh` para configurar um número diferente, como 70, que fornece mais capacidade de processador para o coletor de lixo. Implemente servidores suficientes para manter a carga do processador abaixo de 80% para seus aplicativos.

- Configure um tamanho maior de memória de heap.

O WebSphere Real Time usa mais memória do que o Java comum, assim, planeje seu WebSphere eXtreme Scale com uma memória de heap e configure o tamanho de heap ao iniciar os servidores de catálogo e os contêineres com o parâmetro `-jvmArgs -XmxNNNM` no comando **ogStartServer**. Por exemplo, é possível usar o parâmetro `-jvmArgs -Xmx500M` para iniciar os servidores de catálogo e usar o tamanho de memória apropriado para iniciar os contêineres. O tamanho de memória pode ser configurado de 60 a 70% do tamanho de dados esperado por JVM. Se você não configurar esse valor, ocorrerá um erro `OutOfMemoryError`. Opcionalmente, use o parâmetro `-jvmArgs -Xgc:noSynchronousGCOnOOM` para evitar um comportamento indesejável quando a JVM ficar sem memória.

- Ajuste os encadeamentos para a coleta de lixo.

O WebSphere eXtreme Scale cria muitos objetos temporários associados a cada transação e encadeamentos Remote Procedure Call (RPC). A coleta de lixo possui benefícios de desempenho se o computador tiver ciclos de processadores suficientes. O número padrão de encadeamentos é 1. É possível alterar o número de encadeamentos com o argumento `-Xgcthreads n`. O valor sugerido para esse argumento é o número de núcleos que estão disponíveis em relação ao número de Java virtual machines por computador.

- Ajuste o desempenho para aplicativos de curta duração com o WebSphere eXtreme Scale.

O WebSphere Real Time é ajustado para aplicativos de execução longa. Geralmente, é necessário executar transações contínuas do WebSphere eXtreme Scale por duas horas para obter dados de desempenho confiáveis. É possível usar o parâmetro `-Xquickstart` para melhorar o desempenho de aplicativos de curta duração. Esse parâmetro informa ao compilador just-in-time (JIT) para usar um nível menor de otimização.

- Minimize a fila do cliente do WebSphere eXtreme Scale e a retransmissão do cliente do WebSphere eXtreme Scale.

A principal vantagem de usar o WebSphere eXtreme Scale com o WebSphere Real Time é que o tempo de resposta da transação é altamente confiável, o que normalmente representa várias vezes melhorias na magnitude de ordem no desvio do tempo de resposta da transação. Todos os pedidos do cliente enfileirados e retransmissões de pedido do cliente por meio de outro software afetam o tempo de resposta que está além do controle do WebSphere Real Time

e do WebSphere eXtreme Scale. É necessário alterar seus encadeamentos e parâmetros de soquete para manter uma carga estável e leve sem nenhum atraso significativo e diminuição da profundidade da fila.

- Escreva aplicativos WebSphere eXtreme Scale para usar o encadeamento do WebSphere Real Time.

Sem modificar seu aplicativo, é possível obter tempo de reposta de transação WebSphere eXtreme Scale altamente confiável com diversas melhorias da magnitude de ordem no desvio do tempo de resposta. É possível explorar ainda mais a vantagem do encadeamento de seus aplicativos transacionais a partir do encadeamento Java normal para RealtimeThread o que oferece melhor controle sobre a prioridade de encadeamento e controle de planejamento.

Seu aplicativo atualmente inclui o código a seguir.

```
public class WXSCacheAppImpl extends Thread implements WXSCacheAppIF
```

É possível substituir opcionalmente este código pelo seguinte.

```
public class WXSCacheAppImpl extends RealtimeThread implements  
WXSCacheAppIF
```

## WebSphere Real Time no WebSphere Application Server

É possível utilizar o WebSphere® Real Time com eXtreme Scale em um ambiente do WebSphere Application Server Network Deployment versão 7.0. Ao ativar o WebSphere Real Time, é possível obter uma coleta de lixo mais previsível com tempo de resposta e rendimento estáveis e consistentes das transações.

### Vantagens

O uso do WebSphere eXtreme Scale com WebSphere Real Time aumenta a consistência e a previsibilidade a um custo de rendimento de desempenho em comparação com a política de coleta de lixo padrão empregada no IBM Java™ SE Runtime Environment (JRE) padrão. O custo versus a proposta de benefício pode variar com base em vários critérios. A seguir estão alguns dos principais critérios:

- Recursos do servidor - Memória disponível, tamanho e velocidade da CPU e velocidade e uso da rede
- Carregamentos do servidor - Carregamento de CPU sustentado, carregamento de CPU de pico
- Configuração Java – Tamanhos de heap, uso de destino, encadeamentos de coleta de lixo
- Configuração do modo de cópia do WebSphere eXtreme Scale - Matriz de byte vs. armazenamento POJO
- Características específicas do aplicativo – Uso de encadeamento, requisitos de resposta e tolerância, tamanho do objeto, entre outras.

Além dessa política de coleta de lixo de metrônomo disponível no WebSphere Real Time, existem políticas de coleta de lixo opcionais disponíveis no IBM Java™ SE Runtime Environment (JRE) padrão. Essas políticas, optthruput (padrão), gencon, optavgpause e subpool são projetadas especificamente para solucionar ambientes e requisitos de aplicativo diferentes. Para obter mais informações sobre essas políticas, consulte “Ajustando as Java virtual machines” na página 486. Dependendo dos requisitos, recursos e restrições do aplicativo e do ambiente, a criação de um protótipo de uma ou mais dessas políticas de coleta de lixo pode garantir que você atenda aos seus requisitos e determine a política ideal.

## Recursos com o WebSphere Application Server Network Deployment

1. A seguir estão algumas versões suportadas.
  - WebSphere Application Server Network Deployment versão 7.0.0.5 e acima.
  - WebSphere Real Time V2 SR2 para Linux e acima. Consulte IBM WebSphere Real Time V2 para Linux para obter mais informações.
  - WebSphere eXtreme Scale versão 7.0.0.0 e acima.
  - Sistemas operacionais Linux de 32 e 64 bits.
2. Servidores WebSphere eXtreme Scale não podem ser colocados junto com um WebSphere Application Server DMGr.
3. Real Time não suporta DMGr.
4. Real Time não suporta WebSphere Node Agents.

### Ativando o WebSphere Real Time

Instale o WebSphere Real Time e o WebSphere eXtreme Scale nos computadores nos quais você pretende executar o eXtreme Scale. Atualize o WebSphere Real Time Java para SR2.

É possível especificar as configurações de JVM para cada servidor por meio do console do WebSphere Application Server versão 7.0 da seguinte forma.

Escolha **Servidores > Tipos de Servidor > WebSphere Application Servers > <servidor instalado necessário>**

Na página resultante, escolha "Definição de Processo".

Na próxima página, clique em Java Virtual Machine na parte superior da coluna à direita. (Aqui é possível configurar tamanhos de heap, coleta de lixo e outros sinalizadores para cada servidor.)

Configure os seguintes sinalizadores no campo "Argumentos de JVM Genéricos":

```
-Xrealtime -Xgcpolicy:metronome  
-Xnocompressedrefs -Xgc:targetUtilization=80
```

Aplique e salve as mudanças.

Para utilizar o Real Time no WebSphere Application Server 7.0 com servidores eXtreme Scale, incluindo os sinalizadores JVM acima, você deverá criar a variável de ambiente JAVA\_HOME.

Configure JAVA\_HOME da seguinte forma.

1. Expanda "Ambiente".
2. Selecione "Variáveis do WebSphere".
3. Certifique-se de "Todos os Escopos" esteja marcado em "Mostrar Escopo".
4. Selecione o servidor necessário na lista suspensa. (Não selecione DMGr ou servidores de agente do nó.)
5. Se a variável de ambiente JAVA\_HOME não estiver listada, selecione "Novo" e especifique JAVA\_HOME para o nome da variável. No campo "Valor", insira o nome do caminho completo para o Real Time.
6. Aplique e salve suas mudanças.



## Práticas Recomendáveis

Para um conjunto de boas práticas, consulte a seção de boas práticas em “Ajustando a Coleta de Lixo com o WebSphere Real Time” na página 491. Existem algumas modificações importantes que devem ser observadas nesta lista de boas práticas para um ambiente WebSphere eXtreme Scale independente durante a implementação em um ambiente WebSphere Application Server Network Deployment.

Você deve colocar todos os parâmetros da linha de comandos adicionais da JVM no mesmo local que os parâmetros da política de coleta de lixo especificados na seção anterior.

Um destino inicial aceitável para carregamentos do processador sustentados é 50% com carregamentos de pico de duração curta atingindo até 75%. Além disso, você deve incluir capacidade adicional antes de ver a degradação mensurável na previsibilidade e consistência. É possível aumentar um pouco o desempenho se você puder tolerar tempos de resposta mais longos. Se você exceder 80% do limite, isso pode levar à degradação significativa da consistência e da previsibilidade.

---

## Ajustando o Provedor de Cache Dinâmico

O provedor de cache dinâmico do WebSphere eXtreme Scale suporta os seguintes parâmetros de configuração para o ajuste de desempenho.

### Sobre Esta Tarefa

- **com.ibm.websphere.xs.dynacache.ignore\_value\_in\_change\_event:** Ao registrar um listener de evento de mudança com o provedor de cache dinâmico e gerar uma instância `ChangeEvent`, haverá um gasto adicional associado à desserialização da entrada do cache para que o valor possa ser colocado dentro o `ChangeEvent`. Configurar este parâmetro opcional na instância do cache para `true` ignorará a desserialização da entrada do cache ao gerar o `ChangeEvents`. O valor retornado é nulo para uma operação de remoção ou uma matriz de bytes contendo o formulário serializado do objeto. As instâncias `InvalidationEvent` possuem uma penalidade de desempenho semelhante que pode ser evitada ao configurar `com.ibm.ws.cache.CacheConfig.ignoreValueInInvalidationEvent` para `true`.
- **com.ibm.websphere.xs.dynacache.enable\_compression:** Por padrão, o provedor de cache dinâmico do eXtreme Scale compacta as entradas de cache na memória para aumentar a densidade do cache, o que pode salvar uma quantidade significativa de memória para aplicativos como o armazenamento em cache do servlet. Se você souber que a maioria de seus dados em cache não deve ser compactável, considere configurar este valor como `false`.





---

## Capítulo 10. Segurança



WebSphere eXtreme Scale pode proteger o acesso a dados, incluindo permissão para integração com provedores de segurança externos. Os aspectos de segurança incluem autenticação, autorização, segurança de transporte, segurança da grade de dados, segurança local e segurança JMX (MBean).

---

### Autenticação de Cliente do Aplicativo

A autenticação de cliente do aplicativo consiste em ativar a autenticação de credencial e de segurança cliente/servidor e de configurar um autenticador e um gerador de credencial de sistema.

#### Ativação da Segurança do Cliente-Servidor

A segurança deve ser ativada no cliente e no servidor para poder autenticar-se com êxito com o ObjectGrid.

#### Ativar a Segurança do Cliente

O WebSphere eXtreme Scale fornece um arquivo de amostra de propriedade do cliente, o arquivo `sampleClient.properties`, no diretório de instalação `was_root/optionalLibraries/ObjectGrid/properties` do WebSphere Application Server ou no diretório `/ObjectGrid/properties` em uma instalação do servidor combinada. É possível modificar este arquivo de gabarito com valores apropriados. Configure a propriedade `securityEnabled` no arquivo `objectgridClient.properties` para `true`. A propriedade `securityEnabled` indica se a segurança está ativada. Quando um cliente se conecta a um servidor, os valores no lado do cliente e do servidor devem ser ambos `true` ou ambos `false`. Por exemplo, se a segurança do servidor conectado estiver ativada, o valor da propriedade deverá ser configurado como `true` na lado do cliente para que o cliente se conecte ao servidor.

A interface

`com.ibm.websphere.objectgrid.security.config.ClientSecurityConfiguration` representa o arquivo `security.ogclient.props`. É possível usar a API pública `com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory` para criar uma instância desta interface com valores padrão ou criar uma instância passando o perfil de propriedade de segurança do cliente do ObjectGrid. O arquivo `security.ogclient.props` contém outras propriedades. Consulte a Documentação da API `ClientSecurityConfiguration` e a Documentação da API `ClientSecurityConfigurationFactory` para obter mais detalhes.

#### Ativação da Segurança do Servidor

Para ativar a segurança no lado do servidor, é possível configurar a propriedade **`securityEnabled`** no arquivo `security.xml` para `true`. Use um arquivo XML do descritor de segurança para especificar a configuração de segurança da grade de dados para isolar a configuração de segurança de toda a grade da configuração sem segurança.

## Ativando a Autenticação de Credencial

Após o cliente do eXtreme Scale recuperar o objeto Credential usando o objeto CredentialGenerator, o objeto Credential é enviado juntamente com o pedido do cliente para o servidor eXtreme Scale. O servidor autentica o objeto Credential antes de processar o pedido. Se o objeto Credential for autenticado com êxito, um objeto Subject será retornado para representar este objeto Credential. Este objeto Subject é, então, usado para autorizar o pedido.

Configure a propriedade **credentialAuthentication** nos arquivos de propriedades do cliente e do servidor para ativar a autenticação de credencial. Para obter mais informações, consulte Arquivo de Propriedades do Cliente e Arquivo de Propriedades do Servidor.

As tabelas a seguir exibem qual mecanismo de autenticação será utilizado em diferentes configurações.

*Tabela 31. Autenticação de credencial nas configurações do cliente e do servidor*

| Autenticação de Credencial do Cliente | Autenticação de Credencial do Servidor | Resultado  |
|---------------------------------------|----------------------------------------|------------|
| Não                                   | Nunca                                  | Desativada |
| Não                                   | Suportado                              | Desativada |
| Não                                   | Requerido                              | Error case |
| Suportado                             | Nunca                                  | Desativada |
| Suportado                             | Suportado                              | Ativado    |
| Suportado                             | Requerido                              | Ativado    |
| Requerido                             | Nunca                                  | Error case |
| Requerido                             | Suportado                              | Ativado    |
| Requerido                             | Requerido                              | Ativado    |

## Configuração de um autenticador

O servidor eXtreme Scale usa o plug-in do Autenticador para autenticar o objeto Credential. Uma implementação da interface do Autenticador obtém o objeto Credential e, em seguida, autentica-o para um registro do usuário, por exemplo, um servidor Lightweight Directory Access Protocol (LDAP) e assim por diante. O eXtreme Scale não fornece uma configuração de registro. A conexão com um registro do usuário e a autenticação nele devem ser implementadas neste plug-in.

Por exemplo, uma implementação do Autenticador extrai o ID do usuário e a senha da credencial, usa-os para conectar e validar um servidor LDAP e cria um objeto Subject como resultado da autenticação. A implementação pode utilizar os módulos de login do Java Authentication and Authorization Service (JAAS). Um objeto Subject é retornado como resultado da autenticação.

É possível configurar o autenticador no arquivo XML descritor de segurança, como mostra o seguinte exemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<securityConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config/security ../objectGridSecurity.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config/security">
  <security securityEnabled="true" loginSessionExpirationTime="300" >
```

```

<authenticator className ="com.ibm.websphere.objectgrid.security.plugins.builtins.KeyStoreLoginAuthenticator">
  </authenticator>
</security>
</securityConfig>

```

Use a opção **-clusterSecurityFile** ao iniciar um servidor seguro para configurar o arquivo XML de segurança. Consulte o tutorial de segurança do Java SE no *Visão Geral do Produto* para obter mais informações.

## Configuração de um gerador de credencial do sistema

O gerador de credenciais do sistema é utilizado para representar um factory para a credencial do sistema. Uma credencial de sistema é semelhante para uma credencial de administrador. É possível configurar o elemento SystemCredentialGenerator no arquivo XML de segurança de catálogo, conforme mostrado no seguinte exemplo:

```

<systemCredentialGenerator className ="com.ibm.websphere.objectgrid.security.plugins.
  builtins.UserPasswordCredentialGenerator">
  <property name="properties" type="java.lang.String" value="manager manager1"
    description="username password" />
</systemCredentialGenerator>

```

Para demonstração, o nome de usuário e senha são armazenados em texto limpo. Não armazene o nome de usuário e senha em texto limpo em um ambiente de produção.

O WebSphere eXtreme Scale fornece um gerador de credencial de sistema padrão, que usa as credenciais do servidor. Se você não especificar explicitamente o gerador de credencial de sistema, este gerador de credencial de sistema padrão é usado.

---

## Autorização do Aplicativo Cliente

A autorização do cliente do aplicativo consiste de classes de permissão do ObjectGrid, de mecanismos de autorização, de um período de verificação de permissão e de autorização de acesso apenas pelo criador.

Para o eXtreme Scale, a autorização se baseia no objeto Subject e nas permissões. O produto suporta dois tipos de mecanismos de autorização: Java Authentication and Authorization Service (JAAS) e autorização customizada.

### Classes de permissão do ObjectGrid

A autorização é baseada em permissões. A seguir há quatro tipos diferentes de classes de permissão.

- A classe MapPermission representa permissões para acessar os dados nos mapas de ObjectGrid.
- A classe ObjectGridPermission representa permissões para acessar o ObjectGrid.
- A classe ServerMapPermission representa permissões para acessar mapas de ObjectGrid no lado do servidor a partir de um cliente.
- A classe AgentPermission representa permissões para iniciar um agente no lado do servidor.

Para obter mais informações sobre as APIs e permissões associadas, consulte o tópico sobre a programação de autorização do cliente no *Guia de Programação*.

## Período de verificação de permissão

O eXtreme Scale suporta armazenamento em cache dos resultados da verificação de permissão de mapa por motivo de desempenho. Sem este mecanismo, quando um método que está na lista de métodos para sua classe de permissão específica é chamado, o tempo de execução chama o mecanismo de autorização configurado para autorizar o acesso. Com este período de verificação de permissão configurado, o mecanismo de autorização é chamado periodicamente com base no período de verificação de permissão. Para obter uma lista de métodos para cada classe de permissão, consulte o tópico sobre a programação de autorização do cliente no *Guia de Programação*.

As informações de autorização da permissão são baseadas no objeto Subject. Quando um cliente tenta acessar os métodos, o tempo de execução do eXtreme Scale consulta o cache com base no objeto Subject. Se o objeto não puder ser localizado no cache, o tempo de execução verifica as permissões concedidas para este objeto Subject, e, então, armazena as permissões em um cache.

O período de verificação de permissão deve ser definido antes da inicialização do ObjectGrid. O período de verificação de permissão pode ser configurado de duas maneiras:

É possível utilizar o arquivo XML do ObjectGrid para definir um ObjectGrid e configurar o período de verificação de permissão. No exemplo a seguir, o período de verificação de permissão é configurado para 45 segundos:

```
<objectGrids>
<objectGrid name="secureClusterObjectGrid" securityEnabled="true"
authorizationMechanism="AUTHORIZATION_MECHANISM_JAAS"
permissionCheckPeriod="45">
  <bean id="bean id="TransactionCallback"
className="com.ibm.websphere.samples.objectgrid.HeapTransactionCallback" />
  ...
</objectGrids>
```

Se desejar criar um ObjectGrid com APIs, chame o seguinte método para configurar o período de verificação de permissão. Este método pode ser chamado apenas antes da inicialização da instância do ObjectGrid. Este método se aplica apenas ao modelo de programação do eXtreme Scale local quando você utiliza a instância do ObjectGrid diretamente.

```
/**
 * This method takes a single parameter indicating how often you
 * want to check the permission used to allow a client access. If the
 * parameter is 0 then every single get/put/update/remove/evict call
 * asks the authorization mechanism, either JAAS authorization or custom
 * authorization, to check if the current subject has permission. This might be
 * prohibitively expensive from a performance point of view depending on
 * the authorization implementation, but if you need to have ever call check the
 * authorization mechanism, then set the parameter to 0.
 * Alternatively, if the parameter is > 0 then it indicates the number
 * of seconds to cache a set of permissions before returning to
 * the authorization mechanism to refresh them. This value provides much
 * better performance, but if the back-end
 * permissions are changed during this time then the ObjectGrid can
 * allow or prevent access even though the back-end security
 * provider was modified.
 *
 * @param period the permission check period in seconds.
 */
void setPermissionCheckPeriod(int period);
```

## Autorização de acesso apenas pelo criador

A autorização de acesso apenas pelo criador garante que apenas o usuário (representado pelos objetos Principal associados a ele) que insere a entidade no mapa ObjectGrid possa acessar (ler, atualizar, invalidar e remover tal entrada).

O modelo existente de autorização do mapa do ObjectGrid é baseado no tipo de acesso mas não em entradas de dados. Em outras palavras, um usuário possui um tipo de acesso específico, tal como read, write, insert, delete ou invalidate, para todos os dados no mapa ou em nenhum dos dados. Entretanto, o eXtreme Scale não autoriza usuários para a entrada de dados individual. Este recurso oferece uma nova maneira de autorizar usuários para entradas de dados.

Em um cenário onde usuários diferentes acessam diferentes conjuntos de dados, este modelo pode ser útil. Quando o usuário carrega dados do armazenamento persistente nos mapas do ObjectGrid, o acesso pode ser autorizado pelo armazenamento persistente. Neste caso, não há necessidade de outra autorização na camada do mapa do ObjectGrid. É necessário apenas garantir que a pessoa que carrega os dados no mapa possa acessá-lo, permitindo o recurso de acesso apenas pelo criador.

#### Valores de atributos do modo apenas do criador:

##### desativada

O recurso de acesso apenas pelo criador é desativado.

##### complement

O recurso de acesso apenas pelo criador é ativado para complementar a autorização do mapa. Em outras palavras, a autorização de mapa e o recurso de acesso apenas pelo criador serão efetivados. Portanto, é possível limitar ainda mais as operações nos dados. Por exemplo, o criador não pode invalidar os dados.

##### supersede

O recurso de acesso apenas pelo criador é ativado para substituir a autorização do mapa. Em outras palavras, o recurso de acesso apenas pelo criador substituirá a autorização do mapa e nenhuma autorização de mapa será feita.

É possível configurar o modo de acesso apenas pelo criador de duas maneiras:

#### Pelo arquivo XML:

É possível utilizar o arquivo XML do ObjectGrid para definir um ObjectGrid e configurar o acesso de modo apenas pelo criador para disabled, complement ou supersede, como mostra o seguinte exemplo:

```
<objectGrids>
  <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
    accessByCreatorOnlyMode="supersede"
    <bean id="TransactionCallback"
      classname="com.ibm.websphere.samples.objectgrid.HeapTransactionCallback" />
  ...
</objectGrids>
```

#### Programaticamente:

Se desejar criar um ObjectGrid programaticamente, é possível chamar o método a seguir para configurar o modo de acesso apenas pelo criador. A chamada deste método aplica-se somente ao modelo de programação local do eXtreme Scale quando você instancia diretamente a instância do ObjectGrid:

```
/**
 * Set the "access by creator only" mode.
 * Enabling "access by creator only" mode ensures that only the user (represented
 * by the Principals associated with it), who inserts the record into the map,
 * can access (read, update, invalidate, and remove) the record.
 * The "access by creator only" mode can be disabled, or can complement the
 * ObjectGrid authorization model, or it can supersede the ObjectGrid
 * authorization model. The default value is disabled:
 * {@link SecurityConstants#ACCESS_BY_CREATOR_ONLY_DISABLED}.
```

```

* @see SecurityConstants#ACCESS_BY_CREATOR_ONLY_DISABLED
* @see SecurityConstants#ACCESS_BY_CREATOR_ONLY_COMPLEMENT
* @see SecurityConstants#ACCESS_BY_CREATOR_ONLY_SUPERSEDE
*
* @param accessByCreatorOnlyMode the access by creator mode.
*
* @since WAS XD 6.1 FIX3
*/
void setAccessByCreatorOnlyMode(int accessByCreatorOnlyMode);

```

Para ilustrar ainda mais, considere um cenário no qual uma conta de mapa do ObjectGrid está em uma grade financeira e Manager1 e o Employee1 são os dois usuários. A política de autorização do eXtreme Scale concede todas as permissões de acesso para o Manager1 e apenas a permissão de acesso de leitura para o Employee1. A política do JAAS para a autorização do mapa do ObjectGrid é mostrada no seguinte exemplo:

```

grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    Principal com.acme.PrincipalImpl "Manager1" {
        permission com.ibm.websphere.objectgrid.security.MapPermission
            "banking.account", "all"
    };
grant codebase "http://www.ibm.com/com/ibm/ws/objectgrid/security/PrivilegedAction"
    Principal com.acme.PrincipalImpl "Employee1" {
        permission com.ibm.websphere.objectgrid.security.MapPermission
            "banking.account", "read, insert"
    };

```

Considere como o recurso de acesso apenas pelo criador afeta a autorização:

- **disabled** Se o recurso de acesso apenas pelo criador estiver desativado, a autorização de mapa não é diferente. O usuário "Manager1" pode acessar todos os dados no mapa "account". O usuário "Employee1" pode ler e inserir todos os dados no mapa, mas não pode atualizar, invalidar e remover nenhum dado no mapa.
- **complement** Se o recurso de acesso apenas pelo criador estiver ativado com a opção "complement", a autorização de mapa e a autorização de acesso apenas pelo criador estarão em vigor. O usuário "Manager1" pode acessar os dados no mapa "account", mas apenas se o usuário sozinho os carregou no mapa. O usuário "Employee1" pode ler os dados no mapa "account", mas apenas se tal usuário sozinho os carregou no mapa. (Entretanto, este usuário não pode atualizar, invalidar ou remover dados no mapa).
- **supersede** Se o recurso de acesso apenas pelo criador estiver ativado com a opção "supersede", a autorização do mapa não será imposta. A autorização de acesso apenas pelo criador será a política apenas de autorização. O usuário "Manager1" possui o mesmo privilégio que no modo "complement": este usuário pode acessar os dados no mapa "account" apenas se o mesmo usuário carregou os dados no mapa. Entretanto, o usuário "Employee1" agora possui acesso total aos dados no mapa "account" se este usuários os carregou no mapa. Em outras palavras, a política de autorização definida na política do Java Authentication and Authorization Service (JAAS) não será imposta.

---

## Autenticação da Grade de Dados

É possível utilizar o plug-in do gerenciador de token seguro para ativar a autenticação servidor-para-servidor, que requer que você implemente a interface SecureTokenManager.

O método generateToken(Object) obtém uma proteção de objeto, e depois gera um token que não pode ser compreendido pelos outros. O método verifyTokens(byte[]) faz o processo inverso: converte o token de volta ao objeto original.



Uma implementação SecureTokenManager simples usa um algoritmo de codificação simples, como um algoritmo XOR, para codificar o objeto na forma serializada e depois usa o algoritmo de codificação correspondente para decodificar o token. Esta implementação não é segura e é fácil de ser interrompida.

### Implementação padrão do **WebSphere eXtreme Scale**

O WebSphere eXtreme Scale fornece uma implementação imediatamente disponível para esta interface. Esta implementação padrão utiliza um par de chaves para assinar e verificar a assinatura e utiliza uma chave secreta para criptografar o conteúdo. Cada servidor tem um armazenamento de chaves de tipo JCKES para armazenar o par de chaves, uma chave privada e uma chave pública e uma chave secreta. O armazenamento de chaves tem que ser do tipo JCKES para armazenar as chaves secretas. Estas chaves são utilizadas para criptografar e assinar ou verificar a cadeia de segredo na extremidade de envio. Além disso, o token está associado ao tempo de expiração. Na extremidade de recebimento, os dados são verificados, descriptografados e comparados com a cadeia de segredo do receptor. Os protocolos de comunicação Secure Sockets Layer (SSL) não são necessários entre um par de servidores para autenticação, porque as chaves privadas e as chaves públicas servem para a mesma finalidade. No entanto, se a comunicação do servidor não for criptografada, os dados poderão ser roubados por violação na comunicação. Como o token expira em breve, a ameaça de ataque à reprodução é minimizada. Esta possibilidade é significativamente reduzida se todos os servidores forem implementados atrás de um firewall.

A desvantagem desta abordagem é que os administradores do WebSphere eXtreme Scale precisam gerar chaves e transportá-las para todos os servidores, o que pode causar violação de segurança durante o transporte.

---

## Segurança da Grade de Dados

A segurança da grade de dados garante que um servidor que está se juntando tenha as credenciais certas, de modo que um servidor malicioso não possa juntar-se à grade de dados. A segurança da grade de dados usa um mecanismo de sequência de segredo compartilhado.

Todos os servidores WebSphere eXtreme Scale, incluindo servidores de catálogo, concordam quanto a uma cadeia de segredo compartilhado. Quando um servidor junta a grade de dados, ele é desafiado a apresentar a sequência secreta. Se a cadeia secreta do servidor que está se juntando corresponder à cadeia no servidor presidente ou servidor de catálogo, o servidor que está se juntando é aceito. Se a cadeia não corresponder, o pedido de junção é rejeitado.

Não é seguro enviar um segredo em texto não-criptografado. A infraestrutura de segurança do WebSphere eXtreme Scale fornece um plug-in do gerenciador de token seguro para permitir que o servidor proteja este segredo antes de enviar. Você deve decidir como implementar a operação de proteção. O WebSphere eXtreme Scale fornece uma implementação pronta para usar, na qual a operação segura é implementada para criptografar e assinar o segredo.

A cadeia do segredo é configurada no arquivo `server.properties`. Consulte o Arquivo de Propriedades do Servidor para obter mais informações sobre a propriedade `authenticationSecret`.

## Plug-in SecureTokenManager

Um plug-in do gerenciador de token de segurança é representado pela interface `com.ibm.websphere.objectgrid.security.plugins.SecureTokenManager`.

Para obter mais informações sobre o plug-in `SecureTokenManager`, consulte a documentação da API `SecureTokenManager`.

O método `generateToken(Object)` obtém um objeto, e depois gera um token que não pode ser compreendido pelos outros. O método `verifyTokens(byte[])` faz o processo inverso: o método converte o token de volta ao objeto original.

Uma implementação `SecureTokenManager` simples usa um algoritmo de codificação simples, como um algoritmo exclusivo ou (XOR), para codificar o objeto na forma serializada e depois usa o algoritmo de codificação correspondente para codificar o token. Essa implementação não é segura.

O WebSphere eXtreme Scale fornece uma implementação imediatamente disponível para esta interface.

A implementação padrão utiliza um par de chaves para assinar e verificar a assinatura e utiliza uma chave secreta para criptografar o conteúdo. Cada servidor tem um armazenamento de chaves de tipo JCKES para armazenar o par de chaves, uma chave privada e uma chave pública e uma chave secreta. O armazenamento de chaves tem que ser do tipo JCKES para armazenar as chaves secretas.

Estas chaves são utilizadas para criptografar e assinar ou verificar a cadeia de segredo na extremidade de envio. Além disso, o token está associado ao tempo de expiração. Na extremidade de recebimento, os dados são verificados, decifrados e comparados com a cadeia de segredo do receptor. Os protocolos de comunicação Secure Sockets Layer (SSL) não são necessários entre um par de servidores para autenticação, porque as chaves privadas e as chaves públicas servem para a mesma finalidade. No entanto, se a comunicação do servidor não for criptografada, os dados poderão ser roubados por violação na comunicação. Como o token expira em breve, a ameaça de ataque à reprodução é minimizada. Esta possibilidade é significativamente reduzida se todos os servidores forem implementados atrás de um firewall.

A desvantagem desta abordagem é que os administradores do WebSphere eXtreme Scale precisam gerar chaves e transportá-las para todos os servidores, o que pode causar violação de segurança durante o transporte.

## Scripts de Amostra para Criar Propriedades do Gerenciador de Token Seguro

Conforme observado na seção anterior, é possível criar um keystore contendo um par de chaves para assinar e verificar a assinatura e uma chave secreta para criptografar o conteúdo.

Por exemplo, é possível utilizar o comando JDK 6 `keytool` para criar a chave da seguinte forma:

```
keytool -genkeypair -alias keypair1 -keystore key1.jck -storetype
JCEKS -keyalg
rsa -dname "CN=sample.ibm.com, OU=WebSphere eXtreme Scale" -storepass
key111 -keypass
keypair1 -validity 10000
```

```
keytool -genseckey -alias seckey1 -keystore key1.jck -storetype JCEKS  
-keyalg  
DES -storepass key111 -keypass seckey1 -validity 1000
```

Esses dois comandos criam um par de chaves "keypair1" e uma chave secreta "seckey1". É possível então configurar o seguinte no arquivo de propriedades do servidor:

```
secureTokenKeyStore=key1.jck  
secureTokenKeyStorePassword=key111  
secureTokenKeyStoreType=JCEKS  
secureTokenKeyPairAlias=keypair1  
secureTokenKeyPairPassword=keypair1  
secureTokenSecretKeyAlias=seckey1  
secureTokenSecretKeyPassword=seckey1  
secureTokenCipherAlgorithm=DES  
secureTokenSignAlgorithm=RSA
```

## Configuração

Consulte Propriedades do servidor para obter mais informações sobre as propriedades que você usa para configurar o gerenciador de token seguro.

---

## Transport Layer Security e Secure Sockets Layer

O WebSphere eXtreme Scale suporta TCP/IP e Transport Layer Security/Secure Sockets Layer (TLS/SSL) para a comunicação segura entre clientes e servidores.

### Ativar TLS/SSL em Ambas as Direções

O TLS/SSL às vezes é ativado em uma direção. Por exemplo, apenas o certificado público do servidor é importado no armazenamento confiável do cliente, porém o certificado público do cliente não é importado no armazenamento confiável do servidor. Entretanto, o WebSphere eXtreme Scale usa amplamente os agentes da grade de dados. Uma característica de um agente da grade de dados é que quando o servidor envia respostas de volta para o cliente, uma nova conexão é criada. O servidor eXtreme Scale, em seguida, atua como um cliente. Portanto, você deve importar o certificado público do cliente no armazenamento confiável do servidor.

### Ativar Segurança de Transporte para Sun JDK

WebSphere eXtreme Scale requer o ExtensionJava Secure Sockets IBM (IBMJSSE) ou o IBM Java Secure Sockets Extension 2 (IBMJSSE2). Os provedores IBMJSSE e IBMJSSE2 contêm uma implementação de referência que suporta os protocolos SSL e TLS (Transport Layer Security) e uma estrutura de API (Application Programming Interface).

O Sun JDK não fornece os servidores IBM JSSE e IBM JSSE2, portanto, a segurança do transporte não pode ser ativada com um Sun JDK. Para realizar esse trabalho, um Sun JDK fornecido com o WebSphere Application Server é necessário. O Sun SDK fornecido pelo WebSphere Application Server contém os provedores IBM JSSE e IBM JSSE2.

Consulte o "Configurando um Object Request Broker Customizado" na página 286 para obter informações sobre o uso de um não IBM JDK para WebSphere eXtreme Scale. Se -Djava.endorsed.dirs estiver configurado, ele apontará para os diretórios objectgridRoot/lib/endorsed e JRE/lib/endorsed. O diretório

objectgridRoot/lib/endorsed é necessário de modo que o IBM ORB seja utilizado e o diretório JRE/lib/endorsed é necessário para carregar os provedores IBM JSSE e IBM JSSE2.

Trabalhe com a etapa 4 do tutorial de segurança no *Visão Geral do Produto* para obter informações sobre como configurar suas propriedades SSL necessárias, criar keystores e armazenamentos confiáveis e iniciar servidores seguros no WebSphere eXtreme Scale.

## Configurando Tipos de Transportes Seguros

A Segurança da Camada de Transporte (TLS) fornece comunicação segura entre o cliente e o servidor. O mecanismo de comunicação que é usado depende do valor do parâmetro **transportType** que é especificado nos arquivos de configuração do cliente e do servidor.

### Sobre Esta Tarefa

Quando o Secure Sockets Layer (SSL) é usado, os parâmetros de configuração SSL devem ser fornecidos para o lado do cliente e também para o lado do servidor. Em um ambiente Java SE, a configuração SSL é configurada nos arquivos de propriedades do cliente ou servidor. Se o cliente ou servidor estiver no WebSphere Application Server, será possível usar as configurações de transporte do WebSphere Application Server CSIV2 existentes para os servidores de contêiner e clientes. Consulte o “Integração de Segurança com o WebSphere Application Server” na página 515 para obter informações adicionais.

Tabela 32. Protocolo de Transporte a Ser Utilizado nas Configurações de Transporte do Cliente e Transporte do Servidor.

Se as configurações transportType forem diferentes entre o cliente e o servidor, o protocolo resultante pode variar ou resultar em um erro.

Propriedade transportType do cliente	Propriedade transportType do servidor	Protocolo resultante
TCP/IP	TCP/IP	TCP/IP
TCP/IP	Suportado pelo SSL	TCP/IP
TCP/IP	Requerido pelo SSL	Erro
Suportado pelo SSL	TCP/IP	TCP/IP
Suportado pelo SSL	Suportado pelo SSL	SSL (se o SSL falhar, TCP/IP)
Suportado pelo SSL	Requerido pelo SSL	SSL
Requerido pelo SSL	TCP/IP	Erro
Requerido pelo SSL	Suportado pelo SSL	SSL
Requerido pelo SSL	Requerido pelo SSL	SSL

### Procedimento

1. Para configurar a propriedade **transportType** na configuração de segurança do cliente, consulte Arquivo de Propriedades do Cliente.
2. Para configurar a propriedade **transportType** na configuração de segurança do contêiner e do servidor de catálogos, consulte Arquivo de Propriedades do Servidor.

## Configurando os Parâmetros do Secure Sockets Layer (SSL) para Clientes ou Servidores

O modo com que os parâmetros SSL são configurados varia entre clientes e servidores.

## Sobre Esta Tarefa

O TLS/SSL às vezes é ativado em uma direção. Por exemplo, o certificado público do servidor é importado no armazenamento confiável do cliente, mas o certificado público do cliente não é importado para o armazenamento confiável do servidor. Entretanto, o WebSphere eXtreme Scale usa amplamente os agentes da grade de dados. Uma característica de um agente de grade de dados é que quando o servidor envia resposta de volta para o cliente, ele cria uma conexão. O servidor eXtreme Scale, em seguida, atua como um cliente. Portanto, você deve importar o certificado público do cliente no armazenamento confiável do servidor.

## Procedimento

- Configure os parâmetros SSL do cliente.  
Use uma das opções a seguir para configurar os parâmetros SSL no cliente:
  - Crie um objeto `com.ibm.websphere.objectgrid.security.config.SSLConfiguration` usando o `com.ibm.websphere.objectgrid.security.config.ClientSecurityConfigurationFactory`.
  - Configure os parâmetros no arquivo `client.properties`. Em seguida, é possível configurar o valor da propriedade como uma propriedade do cliente JVM ou usar as APIs do WebSphere eXtreme Scale. Passe o arquivo de propriedades no método `ClientSecurityConfigurationFactory.getClientSecurityConfiguration(String)` para o cliente e use o objeto retornado como um parâmetro para o método `ObjectGridManager.connect(String, ClientSecurityConfiguration, URL)`.
- Configure os parâmetros SSL do servidor.  
Os parâmetros SSL são configurados para servidores usando o arquivo `server.properties`. Para iniciar um servidor de contêiner ou de catálogos com um arquivo de propriedades específico, use o parâmetro **-serverProps** no script **startOgServer**. Para obter mais informações sobre os parâmetros SSL, é possível configurar os servidores do eXtreme Scale, consulte Propriedades do Servidor de Segurança.

---

## Segurança do Java Management Extensions (JMX)

É possível proteger chamadas de beans gerenciados (MBean) em um ambiente distribuído.

Para obter mais informações sobre MBeans disponíveis, consulte “Administrando com Beans Gerenciados (MBeans)” na página 425.

Na topologia de implementação distribuída, MBeans são hospedados diretamente nos servidores de catálogos e nos servidores de contêiner. Em geral, a segurança de JMX em uma topologia distribuída segue a especificação de segurança do JMX conforme especificado na Especificação da Java Management Extensions (JMX). Ela consiste nas três partes a seguir:

1. Autenticação - O cliente remoto precisa ser autenticado no servidor do conector.
2. Controle de acesso - O controle de acesso de MBean limita quem pode acessar as informações de MBean e quem pode executar as operações de MBean.
3. Transporte seguro - O transporte entre o cliente JMX e o servidor pode ser protegido utilizando TLS/SSL.

## Autenticação

O JMX fornece métodos para os servidores conectores para autenticar os clientes remotos. Para o conector RMI, a autenticação é concluída ao fornecer um objeto que implementa a interface `JMXAuthenticator` quando o servidor conector é criado. Assim, o eXtreme Scale implementa esta interface `JMXAuthenticator` para usar o plug-in do `ObjectGrid Authenticator` para autenticar os clientes remotos. Consulte "Tutorial de Segurança do Java SE - Etapa 2" na página 71o tutorial de segurança no *Visão Geral do Produto* para obter detalhes sobre como o eXtreme Scale autentica um cliente.

O cliente JMX segue as APIs do JMX para fornecer credenciais para conectar-se com o servidor conector. A estrutura JMX passa a credencial para o servidor conector e, então, chama a implementação do `JMXAuthenticator` para autenticação. Conforme descrito anteriormente, a implementação do `JMXAuthenticator` então delega a autenticação para a implementação do `Autenticador` do `ObjectGrid`.

Revise o exemplo a seguir que descreve como conectar-se a um servidor conector com uma credencial:

```
javax.management.remote.JMXServiceURL jmxUrl = new JMXServiceURL(
    "service:jmx:rmi:///jndi/rmi://localhost:1099/objectgrid/MBeanServer");

environment.put(JMXConnector.CREDENTIALS, new UserPasswordCredential("admin", "xxxxx"));

// Create the JMXConnectorServer
JMXConnector cntor = JMXConnectorFactory.newJMXConnector(jmxUrl, null);

// Connect and invoke an operation on the remote MBeanServer
cntor.connect(environment);
```

No exemplo anterior, um objeto `UserPasswordCredential` é fornecido com o ID do usuário configurado para `admin` e a senha configurada para `xxxxx`. Este objeto `UserPasswordCredential` é configurado no mapa do ambiente, que é usado no método `JMXConnector.connect(Map)`. Este objeto `UserPasswordCredential` é então passado para o servidor pela estrutura JMX e finalmente passado para a estrutura de autenticação do `ObjectGrid` para a autenticação.

O modelo de programação do cliente segue estritamente a especificação JMX.

## Controle de Acesso

Um servidor MBean JMX pode ter acesso a informações sensíveis e pode executar operações sensíveis. O JMX fornece o controle de acesso necessário que identifica quais clientes podem acessar tais informações e quem pode executar tais operações. o controle de acesso é integrado no modelo de segurança Java padrão por meio da definição de permissões que controlam o acesso ao servidor MBean e às suas operações.

Para controle de acesso de operação ou autorização do JMX, o eXtreme Scale conta com o suporte de JAAS fornecido pela implementação JMX. Em qualquer ponto determinado na execução de um programa, há um conjunto atual de permissões que um encadeamento de execuções contém. Quando um destes encadeamentos chama uma operação de especificação JMX, estes são conhecidos como as permissões contida. Quando uma operação JMX é executada, uma verificação de segurança é feita para verificar se a permissão necessária é incluída pela permissão contida.

A definição de política do MBean segue o formato da política Java. Por exemplo, a seguinte política concede a todos os assinantes e todas as bases de código o direito



de recuperar o endereço JMX do servidor para o PlacementServiceMBean, mas com restrição ao domínio com.ibm.websphere.objectgrid.

```
grant {
    permission javax.management.MBeanPermission
        "com.ibm.websphere.objectgrid.management.PlacementServiceMBean#retrieveServerJMXAddress
    [com.ibm.websphere.objectgrid:*,type=PlacementService]",
        "invoke";
}
```

O exemplo de política a seguir pode ser utilizado para concluir a autorização baseada na identidade do cliente remoto. A política concede a mesma permissão MBean, conforme mostrado no exemplo anterior, exceto apenas para usuários com o nome X500Principal como

CN=Administrator,OU=software,O=IBM,L=Rochester,ST=MN,C=US.

```
grant principal javax.security.auth.x500.X500Principal
    "CN=Administrator,OU=software,O=IBM,
    L=Rochester,ST=MN,C=US" {permission
    javax.management.MBeanPermission
        "com.ibm.websphere.objectgrid.management.PlacementServiceMBean#retrieveServerJMXAddress
    [com.ibm.websphere.objectgrid:*,type=PlacementService]",
        "invoke";
}
```

As políticas do Java são verificadas somente se o gerenciador de segurança estiver ativado. Inicie os servidores de catálogos e servidores de contêineres com o argumento JVM `-Djava.security.manager` para forçar o controle de acesso da operação MBean.

## Transporte Seguro

O transporte entre o cliente e o servidor JMX pode ser protegido utilizando TLS/SSL. Se o `transportType` do servidor de catálogos ou servidor de contêineres for configurado como `SSL_Required` ou `SSL_Supported`, então, é necessário utilizar o SSL para conectar-se ao servidor JMX.

Para usar o SSL, é necessário configurar o armazenamento confiável, o tipo e a senha do armazenamento confiável no cliente MBean usando propriedades do sistema `-D`:

1. `-Djavax.net.ssl.trustStore=TRUST_STORE_LOCATION`
2. `-Djavax.net.ssl.trustStorePassword=TRUST_STORE_PASSWORD`
3. `-Djavax.net.ssl.trustStoreType=TRUST_STORE_TYPE`

Se `com.ibm.websphere.ssl.protocol.SSLSocketFactory` for usado como seu `factory` de soquete SSL em seu arquivo `java_home/jre/lib/security/java.security`, use as seguintes propriedades:

1. `-Dcom.ibm.ssl.trustStore=TRUST_STORE_LOCATION`
2. `-Dcom.ibm.ssl.trustStorePassword=TRUST_STORE_PASSWORD`
3. `-Dcom.ibm.ssl.trustStoreType=TRUST_STORE_TYPE`

Para obter essas informações quando o protocolo Segurança da Camada de Transporte/Secure Sockets Layer (TLS/SSL) estiver ativado, você deve iniciar os servidores de catálogos e de contêiner com o conjunto de portas do serviço JMX. Para configurar a porta de serviço JMX, é possível usar a opção `-JMXServicePort` no script `startOgServer` ou chamar o método `setJMXServicePort` na interface `ServerProperties`.

Para ativar o transporte seguro do JMX para o servidor de contêiner, você deve configurar a porta de serviço do JMX. Configurar a porta do serviço do JMX é



necessário quando você estiver usando o Transport Layer Security/Secure Sockets Layer (TLS/SSL) e desejar exibir informações do servidor de contêiner a partir do servidor de catálogos. Por exemplo, a porta é necessária quando estiver usando o comando `xscmd -c showMapSizes`. Use um dos seguintes métodos para configurar a porta de serviço JMX:

- Use a opção `-JMXServicePort` no script `startOgServer`.
- Se você estiver usando um servidor integrado, chame o método `setJMXServicePort` na interface `ServerProperties` para configurar a porta de serviço JMX.

Você deve usar um número da porta diferente para cada JVM na configuração. Se desejar usar o JMX/RMI, especifique explicitamente a opção `-JMXServicePort` e o número da porta, mesmo se desejar usar o valor da porta padrão.

---

## Integração de Segurança com Provedores Externos

Para proteger seus dados, o produto pode se integrar com vários provedores de segurança.

O WebSphere eXtreme Scale pode integrar-se a uma implementação de segurança externa. Esta implementação externa deve fornecer serviços de autenticação e autorização para o WebSphere eXtreme Scale. O WebSphere eXtreme Scale possui pontos de plug-in para integrar em uma implementação de segurança. O WebSphere eXtreme Scale foi integrado com sucesso nos seguintes componentes:

- LDAP (Lightweight Directory Access Protocol)
- Kerberos
- Segurança do ObjectGrid
- Tivoli Access Manager
- JAAS (Java Authentication and Authorization Service)

O eXtreme Scale utiliza o provedor de segurança para as seguintes tarefas:

- Autenticar clientes para servidores.
- Autorizar clientes para acessar determinados artefatos do eXtreme Scale ou para especificar o que pode ser feito com os artefatos do eXtreme Scale.

O eXtreme Scale possui os seguintes tipos de autorizações:

### **Autorização de mapa**

Clientes ou grupos podem ser autorizados a executar operações de inserção, leitura, atualização, despejo ou exclusão nos mapas.

### **Autorização do ObjectGrid**

Clientes ou grupos podem ser autorizados a executar consultas de objetos ou de entidades nos objectGrids.

### **Autorização do agente do DataGrid**

Clientes ou grupos podem ser autorizados a permitir que os agentes DataGrid sejam implementados a um ObjectGrid.

### **Autorização de Mapa do Lado do Servidor**

Clientes ou grupos podem ser autorizados a replicar um mapa do servidor no lado do cliente ou criar um índice dinâmico para o mapa do servidor.

### **Autorização de administração**

Clientes ou grupos podem ser autorizados a executar tarefas de administração.

**Nota:** Se você já tiver a segurança ativada para seu backend, lembre-se de que estas configurações de segurança não são mais suficientes para proteger seus dados. As configurações de segurança do seu banco de dados ou outro datastore não são, de forma alguma, transferidas para o seu cache. É necessário proteger separadamente os dados que agora são armazenados em cache utilizando o mecanismo de segurança do eXtreme Scale, incluindo segurança no nível de autenticação, autorização e transporte.

**Restrição:** Não use um Kit de Desenvolvimento ou Ambiente de Tempo de Execução na Versão 1.6 e acima quando você também está usando a Segurança da Camada de Transporte SSL com uma configuração independente do WebSphere eXtreme Scale. A Versão 1.6 e posterior não suporta as interfaces de programação de aplicativos do WebSphere eXtreme Scale Versão 7.1. Use a Versão 1.5 ou anterior para as configurações que requerem a segurança Transporte SSL para instalações independentes do eXtreme Scale. Essa restrição se aplica apenas quando estiver usando a segurança SSL em uma configuração independente do eXtreme Scale. A Versão 1.6 e posterior é suportada para configurações de transporte não SSL.

---

## Protegendo o Serviço de Dados REST

Proteja vários aspectos do serviço de dados REST. O acesso ao serviço de dados REST do eXtreme Scale pode ser protegido por meio da autenticação e autorização. O acesso também pode ser controlado por regras de configuração com escopo definido de serviço, conhecidas como regras de acesso. A segurança de transporte é a terceira consideração.

### Sobre Esta Tarefa

O acesso ao serviço de dados REST do eXtreme Scale pode ser protegido por meio da autenticação e autorização. A autenticação e a autorização são realizadas fazendo a integração com a segurança do eXtreme Scale.

O acesso também pode ser controlado por regras de configuração com escopo definido de serviço, conhecidas como regras de acesso. Existem dois tipos de regras de acesso: direitos de operação de serviço, que controlam as operações CRUD que são permitidas pelo serviço, e direitos de acesso de entidade, que controlam as operações CRUD que são permitidas para um tipo de entidade particular.

A segurança de transporte é fornecida pela configuração de contêiner hosting para conexões entre o cliente da web e o serviço REST. E a segurança de transporte é fornecida pela configuração de cliente eXtreme Scale (para serviço REST para conexões de grade de dados eXtreme Scale).

### Procedimento

- Autenticação e autorização de controle.

O acesso ao serviço de dados REST do eXtreme Scale pode ser protegido por meio da autenticação e autorização. A autenticação e a autorização são realizadas pela integração com a segurança do eXtreme Scale.

O serviço de dados REST do eXtreme Scale usa a segurança do eXtreme Scale para autenticação e autorização, a fim de controlar quais usuários podem acessar o serviço e as operações que um usuário tem permissão para executar por meio do serviço. O serviço de dados REST do eXtreme Scale usa uma credencial global configurada, com usuário e senha, ou uma credencial derivada de um

desafio HTTP BASIC que é enviado com cada transação para a grade do eXtreme Scale na qual a autenticação e a autorização são executadas.

1. Configure a autenticação e a autorização do cliente do eXtreme Scale na grade Consulte “Integração de Segurança com Provedores Externos” na página 510 para obter detalhes sobre como configurar a autenticação e a autorização do cliente do eXtreme Scale.
2. Configure o cliente do eXtreme Scale usado pelo serviço REST para segurança.

O serviço de dados REST do eXtreme Scale chama a biblioteca do cliente do eXtreme Scale ao se comunicar com a grade do eXtreme Scale. Portanto, o cliente do eXtreme Scale deve ser configurado para a segurança do eXtreme Scale.

A autenticação de cliente do eXtreme Scale é ativada por meio de propriedades no arquivo de propriedades do cliente objectgrid. No mínimo, os seguintes atributos devem ser ativados ao usar a segurança do cliente com o serviço REST:

```
securityEnabled=true  
credentialAuthentication=Supported [-or-] Required  
credentialGeneratorProps=user:pass [-or-] {xor encoded user:pass}
```

**Lembre-se:** O usuário e a senha especificados na propriedade `credentialGeneratorProps` devem ser mapeados para um ID no registro de autenticação e ter direitos de políticas do ObjectGrid suficientes para se conectar com, e criar, os ObjectGrids.

Um arquivo de política do cliente de objectgrid de amostra está localizado em `restservice_home/security/security.ogclient.properties`. Consulte também Arquivo de Propriedades do Cliente.

3. Configure o serviço de dados REST do eXtreme Scale para segurança.

O arquivo de propriedades de configuração do serviço de dados REST do eXtreme Scale precisa conter as seguintes entradas para ser integrado com a segurança do eXtreme Scale:

```
ogClientPropertyFile=file_name
```

O `ogClientPropertyFile` é o local do arquivo de propriedades que contém propriedades do cliente de ObjectGrid mencionadas na etapa anterior. O serviço REST usa este arquivo para inicializar o cliente do eXtreme Scale para falar com a grade quando a segurança estiver ativada.

```
loginType=basic [-or-] none
```

A propriedade `loginType` configura o serviço REST para o tipo de login. Se um valor de nenhum for especificado, o ID do usuário e senha "globais" definidos por `credentialGeneratorProps` serão enviados para a grade para cada transação. Se um valor de básico for especificado, o serviço REST apresentará um desafio HTTP BASIC para o cliente solicitando credenciais que ele enviará em cada transação ao se comunicar com a grade.

Para obter mais informações sobre as propriedades `ogClientPropertyFile` e `loginType`, consulte o Arquivo de Propriedades do Serviço de Dados REST.

- Aplique as regras de acesso.

O acesso também pode ser controlado por regras de configuração com escopo definido de serviço, conhecidas como regras de acesso. Existem dois tipos de regras de acesso: direitos de operação de serviço, que controla as operações CRUD que são permitidas pelo serviço; e direitos de acesso de entidade, que controlam as operações CRUD que são permitidas para um tipo de entidade particular.

O serviço de dados REST do eXtreme Scale permite, opcionalmente, regras de acesso que podem ser configuradas para restringir o acesso ao serviço e a entidades no serviço. Essas regras de acesso são especificadas no arquivo de propriedades de direitos de acesso do serviço REST. O nome deste arquivo é especificado no arquivo de propriedades do serviço de dados REST pela propriedade `wxsRestAccessRightsFile`. Para obter mais informações sobre essa propriedade, consulte Arquivo de Propriedades do Serviço de Dados REST. Este arquivo é um arquivo de propriedades Java típico com pares de chave e valor. Existem dois tipos de regras de acesso: direitos de operação de serviço, que controlam as operações CRUD que são permitidas pelo serviço; e direitos de acesso de entidade, que controlam as operações CRUD que são permitidas para um tipo de entidade particular.

#### 1. Configure os direitos de operação de serviço.

Os direitos de operações de serviço especificam direitos de acesso que se aplicam a todos os ObjectGrids expostos por meio do serviço REST ou a todas as entidades de um ObjectGrid individual, conforme especificado.

Use a seguinte sintaxe.

```
serviceOperationRights=service_operation_right  
serviceOperationRights.grid_name -OR- *=service_operation_right
```

em que

- `serviceOperationRights` pode ser uma das opções a seguir: [NONE, READSINGLE, READMULTIPLE, ALLREAD, ALL]
- `serviceOperationRights.grid_name -OR- *` implica que o direito de acesso se aplica a todos os ObjectGrids, também o nome de um ObjectGrid específico pode ser fornecido.

Por exemplo:

```
serviceOperationsRights=ALL  
serviceOperationsRights.*=NONE  
serviceOperationsRights.EMPLOYEEGRID=READSINGLE
```

O primeiro exemplo especifica que todas as operações de serviço são permitidas para todos os ObjectGrids expostos por este Serviço REST. O segundo exemplo é semelhante ao primeiro, uma vez que também se aplica a todos os ObjectGrids expostos pelo serviço REST; no entanto, ele especifica o direito de acesso como NONE, o que significa que nenhuma das operações de serviço são permitidas nos ObjectGrids. O último exemplo especifica como controlar as operações de serviço para uma grade específica, aqui apenas Reads, que resultam em um único registro, são permitidas para todas as entidades do EMPLOYEEGRID.

O padrão assumido pelo serviço REST é `serviceOperationsRights` que significa que todas as operações são permitidas para todos os ObjectGrids expostos por este serviço. Isto é diferente da implementação Microsoft para a qual o padrão é NONE, portanto, nenhuma operação é permitida no Serviço REST.

**Importante:** Os direitos de operações de serviço são avaliados na ordem em que são especificados neste arquivo, portanto, o último direito especificado substituirá os direitos que o precedem.

#### 2. Configure os direitos de acesso da entidade.

Os direitos do conjunto de entidades especifica os direitos de acesso que se aplicam a entidades do ObjectGrid específicas expostas por meio do serviço REST. Esses direitos fornecem uma maneira de impor um controle de acesso

melhor e mais refinado sobre entidades do ObjectGrid individuais que o comparado aos direitos de Operações de Serviço.

Use a seguinte sintaxe.

```
entitySetRights.grid_name.entity_name=entity_set_right
```

em que

- *entity\_set\_right* pode ser um dos direitos a seguir.

Tabela 33. Direitos de Acesso de Entidade. Valores suportados.

Direito de acesso	Descrição
NENHUM	Nega todos os direitos para acessar dados
READSINGLE	Permite a leitura de itens de dados únicos
READMULTIPLE	Permite a leitura de conjuntos de dados
ALLREAD	Permite a leitura de um único ou de vários conjuntos de dados
WRITEAPPEND	Permite a criação de novos itens de dados em conjuntos de dados
WRITEREPLACE	Permite a substituição de dados
WRITEDELETE	Permite a exclusão de itens de dados dos conjuntos de dados
WRITEMERGE	Permite a mesclagem de dados
ALLWRITE	Permite a gravação (isto é, criação, substituição, mesclagem ou exclusão) de dados
ALL	Permite a criação, leitura, atualização e exclusão de dados

- *entity\_name* é o nome de um ObjectGrid específico dentro do serviço REST.
- *grid\_name* é o nome de uma entidade específica dentro do ObjectGrid especificado.

**Nota:** Se os direitos de operação de serviço e os direitos do conjunto de entidades forem especificados para um ObjectGrid respectivo e suas entidades, o mais restritivo desses direitos será reforçado, conforme ilustrado nos exemplos a seguir. Observe também que os direitos do conjunto de entidades são avaliados na ordem em que são especificados no arquivo. O último direito especificado substituirá os direitos que o precedem.

**Exemplo 1:** Se `serviceOperationsRights.NorthwindGrid=READSINGLE` e `entitySetRights.NorthwindGrid.Customer=ALL` forem especificados. `READSINGLE` será reforçado para a entidade Cliente.

**Exemplo 2:** Se `serviceOperationsRights.NorthwindGrid=ALLREAD` for especificado e `entitySetRights.NorthwindGrid.Customer=ALLWRITE` for especificado, apenas Reads será permitido para todas as entidades de `NorthwindGrid`. No entanto, para Cliente, seus direitos do conjunto de entidades impedirão quaisquer Reads (uma vez que especificou `ALLWRITE`) e, conseqüentemente, com efeito, a entidade Cliente terá o direito de acesso como `NONE`.

- Transportes seguros.

A segurança de transporte é fornecida pela configuração de contêiner hosting para conexões entre o cliente da web e o serviço REST. A segurança de transporte é fornecida pela configuração do cliente do eXtreme Scale para conexões entre o serviço REST e a grade do eXtreme Scale .

1. Proteja a conexão do cliente e do serviço REST. A segurança de transporte para esta conexão é fornecida pelo ambiente de contêiner do hosting, não no eXtreme Scale.

2. Proteja a conexão do serviço REST e da grade do eXtreme Scale. A segurança de transporte para esta conexão é configurada no eXtreme Scale. Consulte “Transport Layer Security e Secure Sockets Layer” na página 505.

---

## **Integração de Segurança com o WebSphere Application Server**

Quando o WebSphere eXtreme Scale é implementada em um ambiente do WebSphere Application Server, é possível simplificar a configuração de segurança do fluxo de autenticação e da camada de transporte a partir do WebSphere Application Server.

### **Fluxo de Autenticação Simplificada**

Quando os clientes e servidores do eXtreme Scale estão em execução no WebSphere Application Server e no mesmo domínio de segurança, é possível usar a infraestrutura de segurança do WebSphere Application Server para propagar as credenciais de autenticação do cliente para o servidor do eXtreme Scale. Por exemplo, se um servlet atuar como um cliente do eXtreme Scale para se conectar a um servidor do eXtreme Scale no mesmo domínio de segurança, e o servlet já estiver autenticado, é possível propagar o token de autenticação do cliente (servlet) para o servidor e, em seguida, usar a infraestrutura de segurança do WebSphere Application Server para converter o token de autenticação de volta para as credenciais do cliente.

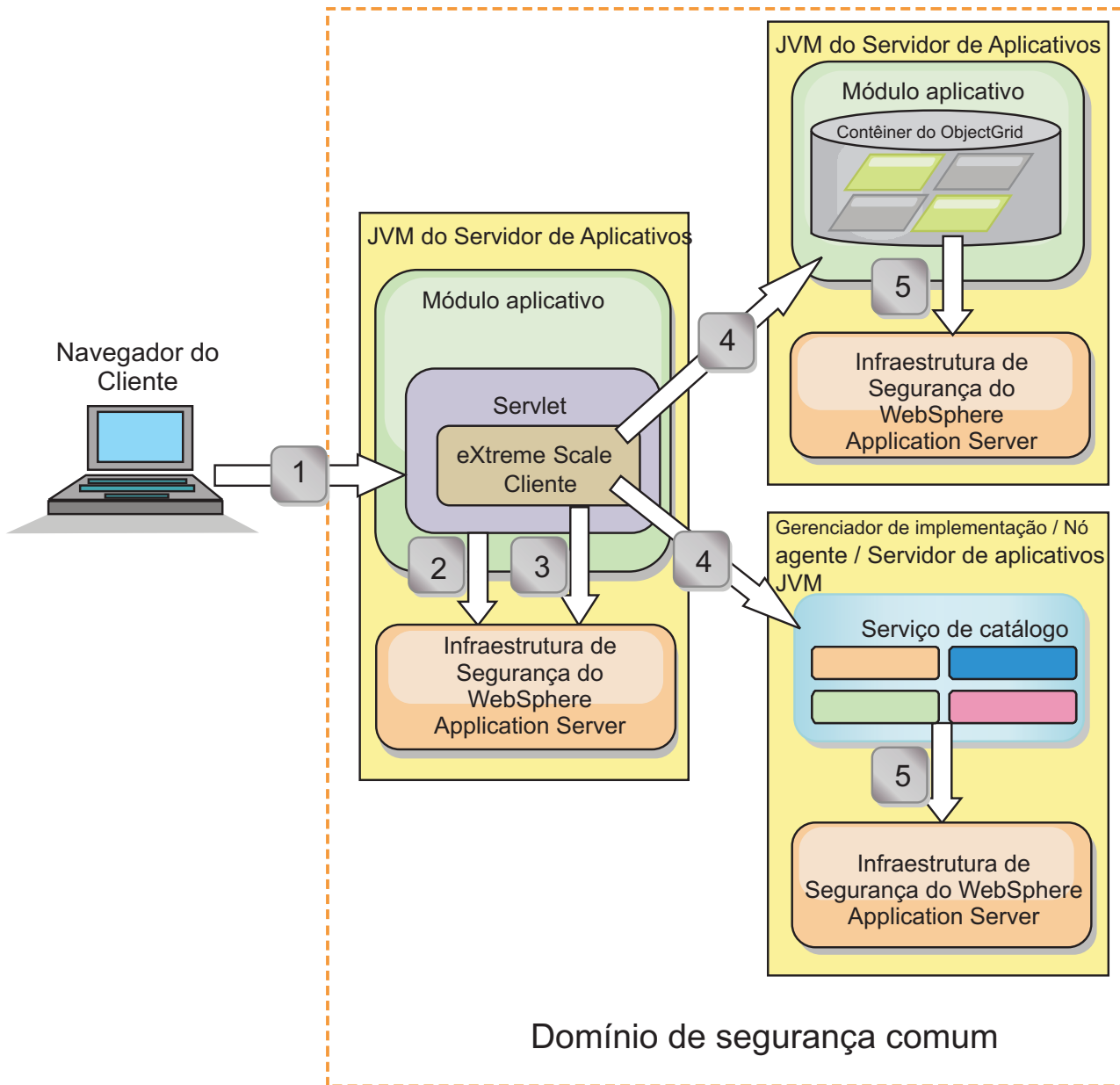


Figura 67. Fluxo de Autenticação para Servidores Dentro do Mesmo Domínio de Segurança

No diagrama anterior, os servidores de aplicativos estão no mesmo domínio de segurança. Um servidor de aplicativos hospeda o aplicativo da web, que também é um cliente do eXtreme Scale. O outro servidor de aplicativos hospeda o servidor de contêiner. A Java Virtual Machine (JVM) do gerenciador de implementação ou do agente do nó hospeda o serviço de catálogo. As setas no diagrama indicam como o processo de autenticação flui:

1. Um usuário do aplicativo corporativo usa um navegador da web para efetuar login no primeiro servidor de aplicativos com um nome de usuário e senha.
2. O primeiro servidor de aplicativos envia o nome de usuário e a senha do cliente para a infraestrutura de segurança do WebSphere Application Server para autenticar-se com o registro do usuário. Por exemplo, este registro do usuário pode ser um servidor LDAP. Como resultado, as informações de segurança são armazenadas no encadeamento do servidor de aplicativos.



3. O arquivo JavaServer Pages (JSP) age como um cliente do eXtreme Scale para recuperar as informações de segurança a partir do encadeamento do servidor. O arquivo JSP chama a infraestrutura de segurança do WebSphere Application Server para obter os tokens de segurança que representam o usuário do aplicativo corporativo.
4. O cliente ou o arquivo JSP do eXtreme Scale envia os tokens de segurança com a solicitação para o servidor de contêiner e para o serviço de catálogo hospedados nas outras JVMs. O servidor de catálogos e o servidor de contêiner usam os tokens de segurança do WebSphere Application Server como uma credencial de cliente do eXtreme Scale.
5. Os servidores de catálogos e contêiner enviam os tokens de segurança para a infraestrutura de segurança do WebSphere Application Server para converter os tokens de segurança em informações de segurança do usuário. Essas informações de segurança do usuário são representadas por um objeto Subject, que contém os principais, as credenciais públicas e credenciais privadas. Essa conversão pode ocorrer porque os servidores de aplicativos que hospedam o cliente, um servidor de catálogos e um servidor de contêiner do eXtreme Scale compartilham os mesmos tokens WebSphere Application Server Lightweight Third-Party Authentication (LTPA).

## **Integração de Autenticação**

### **Integração de segurança distribuída com o WebSphere Application Server:**

Para o modelo distribuído, use as seguintes classes:

- `com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenCredentialGenerator`
- `com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenAuthenticator`
- `com.ibm.websphere.objectgrid.security.plugins.builtins.WSTokenCredential`

Para obter exemplos sobre como usar essas classes, consulte “Tutorial: Integrar a Segurança do WebSphere eXtreme Scale com o WebSphere Application Server” na página 84.

No lado do servidor, o `WSTokenAuthentication` pode ser utilizado como o autenticador para autenticar o objeto `WSTokenCredential`.

### **Integração de segurança local com o WebSphere Application Server:**

Para o modelo de ObjectGrid local, use as seguintes classes:

- `com.ibm.websphere.objectgrid.security.plugins.builtins.WSSubjectSourceImpl`
- `com.ibm.websphere.objectgrid.security.plugins.builtins.WSSubjectValidationImpl`

Para obter mais informações sobre essas classes, consulte Programação de Segurança Local. É possível configurar a classe `WSSubjectSourceImpl` como o plug-in `SubjectSource` e a classe `WSSubjectValidationImpl` como o plug-in `SubjectValidation`.

## **Suporte à Segurança da Camada de Transporte no WebSphere Application Server**

Quando um cliente, um servidor de contêiner ou servidor de catálogos do eXtreme Scale está em execução em um processo do WebSphere Application Server, a segurança de transporte do eXtreme Scale é gerenciada pelas configurações de

transporte do WebSphere Application Server CSIV2. Para o cliente ou servidor de contêiner do eXtreme Scale, as propriedades do cliente ou de servidor do eXtreme Scale não devem ser usadas para definir as configurações SSL. Todas as configurações SSL devem ser especificadas na configuração do WebSphere Application Server.

Entretanto, o servidor de catálogos é um pouco diferente. O servidor de catálogos tem seus próprios caminhos de transporte proprietários que não podem ser gerenciados pelas configurações de transporte do WebSphere Application Server CSIV2. Portanto, as propriedades de SSL ainda precisam ser configuradas no arquivo de propriedades do servidor para o servidor de catálogos. Consulte o “Tutorial: Integrar a Segurança do WebSphere eXtreme Scale com o WebSphere Application Server” na página 84 para obter informações adicionais.

## Configurando a Segurança do Cliente em um Domínio de Serviço de Catálogo

Ao configurar a segurança do cliente em um domínio do serviço de catálogo, é possível definir as propriedades de configuração de autenticação de cliente padrão. Essas propriedades são usadas quando um arquivo de propriedades do cliente não estiver localizado na Java virtual machine (JVM) que está hospedando o cliente ou quando o cliente não especificar programaticamente as propriedades de segurança. Se um arquivo de propriedades do cliente existir, as propriedades que especificas no console substituirão os valores no arquivo. É possível substituir essas propriedades ao especificar um arquivo `splicer.properties` com a propriedade customizada `com.ibm.websphere.xs.sessionFilterProps` ou ao unir o arquivo EAR do aplicativo.

### Antes de Iniciar

- Você deve saber a implementação `CredentialGenerator` que está usando para autenticar os clientes com a grade de dados remota. É possível usar uma das implementações que são fornecidas pelo WebSphere eXtreme Scale: `UserPasswordCredentialGenerator` ou `WSTokenCredentialGenerator`. Também é possível usar uma implementação customizada da interface `CredentialGenerator`. A implementação customizada deve estar no caminho de classe do cliente de tempo de execução e do servidor. Se estiver configurando um cenário de sessão HTTP com WebSphere Application Server, você deverá colocar a implementação no caminho de classe do gerenciador de implementação e no caminho de classe do servidor de aplicativos no qual o cliente está em execução.
- Um domínio de serviço de catálogo deve ser definido. Consulte “Criando Domínios do Serviço de Catálogo no WebSphere Application Server” na página 252 para obter mais informações.

### Sobre Esta Tarefa

A segurança do cliente deve ser configurada no domínio de serviço de catálogo, quando a autenticação de credencial no lado do servidor foi ativada, ao configurar um dos seguintes cenários:

- A política de segurança no lado do servidor tem a propriedade **`credentialAuthentication`** configurada para `Obrigatório`.
- A política de segurança do lado do servidor possui a propriedade **`credentialAuthentication`** configurada para `Suportado` e (AND) um **`authorizationMechanism`** foi especificado no arquivo XML do ObjectGrid.

Nesses cenários, uma credencial deve ser passada a partir do cliente. A credencial que é passada do cliente é recuperada a partir do método `getCredential` em uma classe que implementa a interface `CredentialGenerator`. Em um cenário de configuração de sessão HTTP, o tempo de execução deve conhecer a implementação `CredentialGenerator` a ser usada para gerar uma credencial que é transmitida para uma grade de dados remota. Se a classe de implementação `CredentialGenerator` não for especificada para uso, a grade de dados remota rejeitará solicitações do cliente porque o cliente não pode ser autenticado.

## Procedimento

Defina as propriedades de segurança do cliente. No console administrativo do WebSphere Application Server, clique em **Administração do sistema > WebSphere eXtreme Scale > Domínios do serviço de catálogo > *catalog\_service\_domain\_name* > Propriedades de segurança do cliente**. Especifique as propriedades de segurança do cliente na página e salve suas mudanças. Consulte “Propriedades de Segurança do Cliente” na página 268 para obter uma lista de propriedades que podem ser configuradas.

## Resultados

As propriedades de segurança do cliente configuradas no domínio de serviço de catálogo são usadas como valores padrão. Os valores especificados substituem quaisquer propriedades que forem definidas nos arquivos `client.properties`.

## O que Fazer Depois

Configure seus aplicativos para usar o WebSphere eXtreme Scale para gerenciamento de sessão. Consulte “Unindo Automaticamente Aplicativos para Gerenciamento de Sessões HTTP no WebSphere Application Server” na página 300 para obter mais informações.

---

## Ativando a Segurança Local

O WebSphere eXtreme Scale fornece vários terminais de segurança para integrar mecanismos customizados. No modelo de programação local, a principal função de segurança é a autorização e não possui suporte à autenticação. É necessário autenticar fora do WebSphere Application Server já existente. No entanto, é possível usar os plug-ins fornecidos para obter e validar objetos Subject.

## Sobre Esta Tarefa

A segurança local pode ser ativada com o arquivo descritor XML do ObjectGrid ou programaticamente.

## Procedimento

Ative a segurança local com o arquivo descritor XML do ObjectGrid. O arquivo `secure-objectgrid-definition.xml`, que é utilizado na amostra do aplicativo corporativo `ObjectGridSample`, é mostrado no seguinte exemplo. Configure o atributo `securityEnabled` como `true` para ativar a segurança.

```
<objectGrids>
  <objectGrid name="secureClusterObjectGrid" securityEnabled="true"
    authorizationMechanism="AUTHORIZATION_MECHANISM_JAAS">
    ...
  </objectGrids>
```

## O que Fazer Depois

Inicie os servidores de contêiner e de catálogos com a segurança ativada.

---

## Iniciando e Parando Servidores Seguros

A segurança é ativada ao definir as configurações específicas de segurança quando você inicia e para servidores.

### Iniciando Servidores Seguros em um Ambiente Independente

Para iniciar os servidores seguros independentes, passe os arquivos de configuração adequados ao especificar os parâmetros para o comando **startOgServer**.

#### Antes de Iniciar

Se estiver usando um provedor de segurança do cliente externo para autenticação ou autorização, defina a variável de ambiente *CLIENT\_AUTH\_LIB*. Abra uma janela de linha de comandos ou de terminal e execute o comando que é apropriado para seu sistema operacional:

- **Windows** `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes>`
- **UNIX** `set CLIENT_AUTH_LIB=<path_to_security_JAR_or_classes> export CLIENT_AUTH_LIB`

Quando os comandos **startOgServer** e **stopOgServer** forem executados, esta variável será anexada ao caminho de classe.

#### Procedimento

- Inicie seus servidores de contêiner seguros.  
Iniciar um servidor de contêiner seguro requer o seguinte arquivo de configuração de segurança:
  - **Arquivo de propriedades do servidor:** O arquivo de propriedades do servidor configura as propriedades de segurança específicas para o servidor. Consulte o Arquivo de Propriedades do Servidor para obter mais detalhes.  
Especifique o local desse arquivo de configuração ao fornecer o seguinte argumento para o script **startOgServer**:
    - **-serverProps**  
Especifica o local do arquivo de propriedades do servidor, que contém as propriedades de segurança específicas do servidor. O nome do arquivo especificado para esta propriedade está em formato de caminho de arquivo simples, como `../security/server.properties`.
- Inicie os servidores de catálogos seguros.  
Para iniciar um serviço de catálogo seguro, você deve ter os seguintes arquivos de configuração:
  - **Arquivo descritor XML de segurança:** O arquivo descritor XML de segurança descreve as propriedades de segurança comuns para todos os servidores, incluindo servidores de catálogo e servidores de contêiner. Um exemplo de propriedade é a configuração do autenticador que representa o registro do usuário e o mecanismo de autenticação.
  - **Arquivo de propriedades do servidor:** O arquivo de propriedades do servidor configura as propriedades de segurança que são específicas para o servidor.

Especifique o local destes arquivos de configuração ao fornecer os argumentos a seguir para o script **startOgServer**:

**-clusterSecurityFile e -clusterSecurityUrl**

Estes argumentos especificam o local do arquivo descritor XML de Segurança. Use o parâmetro **-clusterSecurityFile** para especificar um arquivo local ou o parâmetro **-clusterSecurityUrl** para especificar a URL do arquivo `objectGridSecurity.xml`.

**-serverProps**

Especifica o local do arquivo de propriedades do servidor, que contém as propriedades de segurança específicas do servidor. O nome do arquivo especificado para esta propriedade está em formato de caminho de arquivo simples, tal como `c:/tmp/og/catalogserver.props`.

## Iniciando Servidores Seguros no WebSphere Application Server

Para iniciar servidores seguros no WebSphere Application Server, você deve especificar os arquivos de configuração de segurança nos argumentos genéricos da Java Virtual Machine (JVM).

### Procedimento

- Inicie um serviço de catálogo seguro no WebSphere Application Server.  
Um servidor de catálogos contém dois diferentes níveis de informações de segurança:
  - `-Dobjectgrid.cluster.security.xml.url`: Especifica o local do arquivo `objectGridSecurity.xml`, que descreve as propriedades de segurança comuns a todos os servidores, incluindo os servidores de catálogos e servidores de contêiner. Um exemplo das propriedades de segurança definidas é a configuração do autenticador, que representa o registro do usuário e o mecanismo de autenticação. O nome do arquivo especificado para esta propriedade deve estar em um formato de URL, como `file:///tmp/og/objectGridSecurity.xml`.
  - `-Dobjectgrid.server.props`: Especifica o arquivo de propriedades do servidor que contém as propriedades de segurança específicas do servidor. O nome do arquivo especificado para esta propriedade está em formato de caminho de arquivo simples, tal como `c:/tmp/og/catalogserver.props`.
- 1. No console administrativo do WebSphere Application Server, clique em **Administração do Sistema**. Clique no processo no qual o servidor de catálogos está implementado, como o gerenciador de implementação.
- 2. Clique em **Gerenciamento Java e processos > Definição de Processo > Java Virtual Machine**.
- 3. Digite as propriedades no campo **Argumentos JVM genéricos**. A seguir há um exemplo dos valores que podem ser incluídos:
  - `-Dobjectgrid.cluster.security.xml.url=file:///tmp/og/objectGridSecurity.xml`
  - `-Dobjectgrid.server.props=/tmp/og/catalog.server.props`
- 4. Clique em **OK** e salve as mudanças.
- Inicie um servidor de contêiner seguro no WebSphere Application Server.  
Um servidor de contêiner, ao se conectar com o servidor de catálogos, herda a configuração de segurança que está no arquivo `objectGridSecurity.xml`, como a configuração do autenticador ou configurações de tempo limite da sessão de login. Você também deve definir as propriedades de segurança específicas do servidor para servidores de contêiner específicos na propriedade `-Dobjectgrid.server.props`.

O nome do arquivo especificado para esta propriedade está apenas no formato de caminho de arquivo simples, como `c:/tmp/og/server.props`.

Siga as mesmas etapas acima para incluir a propriedade de segurança para os argumentos genéricos da JVM.

1. Abra a página da Java virtual machine para o servidor. No console administrativo do WebSphere Application Server, clique em **Servidores > Servidores de Aplicativos > server\_name > Gerenciamento Java e processos > Definição de Processo > Java Virtual Machine**
2. Digite a propriedade no campo **Argumentos da JVM Genéricos**. A seguir há um exemplo dos valores que podem ser incluídos:  
`-Dobjectgrid.server.props=/opt/wxs/security/server2.props`
3. Clique em **OK** e salve as mudanças.

## Parando Servidores Seguros

Parar servidores de catálogos ou servidores de contêiner seguros requer um arquivo de configuração de segurança.

### Procedimento

Pare um servidor de catálogos ou um servidor de contêiner seguro.

Parar um servidor seguro requer o seguinte arquivo de configuração de segurança:

- **Arquivo de propriedades do cliente:** O arquivo de propriedades do cliente pode ser usado para configurar as propriedades de segurança do cliente. As propriedades de segurança do cliente são necessárias para um cliente se conectar a um servidor seguro. Consulte o Arquivo de Propriedades do Cliente para obter mais detalhes.

Especifique o local desses arquivos de configuração ao fornecer o seguinte argumento para o script **stopOgServer**:

#### **-clientSecurityFile**

Especifica o caminho para o arquivo de propriedades do cliente que define as propriedades de segurança para o cliente. O nome do arquivo especificado para esta propriedade está em formato de caminho de arquivo simples, como `../security/objectGridClient.properties`.

#### **Exemplo:**

```
stopOgServer.bat|sh cs1 -catalogServiceEndPoints
cs1:MyServer1.company.com:6601:6602,
cs2:MyServer2.company.com:6601:6602,
cs3:MyServer3.company.com:6601:6602
-clientSecurityFile ../security/objectGridClient.properties
```

---

## Configurando Perfis de Segurança para o Utilitário xscmd

Criando um perfil de segurança, é possível usar parâmetros de segurança salvos para usar o utilitário **xscmd** com ambientes seguros.

### Antes de Iniciar

Para obter informações adicionais sobre como configurar o utilitário **xscmd**, consulte “Administrando com o Utilitário **xscmd**” na página 407.



## Sobre Esta Tarefa

É possível usar o parâmetro **-ssp** *profile\_name* ou **--saveSecProfile** *profile\_name* com o restante de seu comando **xscmd**. Para salvar um perfil de segurança. O perfil pode conter configurações para nomes de usuário e senhas, geradores de credencial, keystores, armazenamentos confiáveis e tipos de transporte.

O do grupo de comandos **ProfileManagement** no utilitário **xscmd** contém comandos para gerenciar seus perfis de segurança.

## Procedimento

- Salve um perfil de segurança.

Para salvar um perfil de segurança, use o parâmetro **-ssp** *profile\_name* ou **--saveSecProfile** *profile\_name* com o restante de seu comando. A inclusão deste parâmetro em seu comando salva os parâmetros a seguir:

```
-al,--alias <alias>
-arc,--authRetryCount <integer>
-ca,--credAuth <support>
-cgc,--credGenClass <className>
-cgp,--credGenProps <property>
-cxpv,--contextProvider <provider>
-ks,--keyStore <filePath>
-ksp,--keyStorePassword <password>
-kst,--keyStoreType <type>
-prot,--protocol <protocol>
-pwd,--password <password>
-ts,--trustStore <filePath>
-tsp,--trustStorePassword <password>
-tst,--trustStoreType <type>
-tt,--transportType <type>
-user,--username <username>
```

Os perfis de segurança são salvos no diretório `user_home\xscmd\profiles\security\<profile_name>.properties`.

- Use um perfil de segurança salvo.

Para usar um perfil de segurança salvo, inclua o parâmetro **-sp** *profile\_name* ou **--securityProfile** *profile\_name* no comando que você está executando. Exemplo de comando: `xscmd -c listHosts -cep myhost.mycompany.com -sp myprofile`

- Liste os comandos no grupo de comandos do **ProfileManagement**.

Execute o comando a seguir: **xscmd -lc ProfileManagement**.

- Liste os perfis de segurança existentes.

Execute o comando a seguir: **xscmd -c listProfiles -v**.

- Exiba as configurações que são salvas em um perfil de segurança.

Execute o comando a seguir: **xscmd -c showProfile -pn profile\_name**.

- Remova um perfil de segurança existente.

Execute o comando a seguir: **xscmd -c RemoveProfile -pn profile\_name**.





---

## Capítulo 11. Resolução de Problemas



Além dos logs e do rastreamento, de mensagens e notas sobre a liberação discutidos nesta seção, é possível usar as ferramentas de monitoramento para descobrir problemas, como o local de dados no ambiente, a disponibilidade de servidores na grade de dados, e assim por diante. Se você estiver executando um ambiente do WebSphere Application Server, poderá usar a Performance Monitoring Infrastructure (PMI). Se você estiver executando em um ambiente independente, poderá usar uma ferramenta de monitoramento do fornecedor, como CA Wily Introscope ou Hyperic HQ. Também é possível usar e customizar o utilitário `xscmd` para exibir informações de texto sobre o ambiente.

---

### Ativando a Criação de Log

É possível usar logs para monitorar e solucionar problemas em seu ambiente.

#### Sobre Esta Tarefa

Os logs são salvos em diferentes locais e formatos dependendo da sua configuração.

#### Procedimento

- **Ative os logs em um ambiente independente.**

Com servidores de catálogos independentes, os logs estão no local onde o comando `startOgServer` é executado. Para os servidores de contêiner, é possível usar o local padrão ou definir um local de log customizado:

- **Local de log padrão:** O logs estão no diretório onde o comando do servidor foi executado. Se você iniciar os servidores no diretório `wxs_home/bin`, os arquivos de log e de rastreamento estão nos diretórios `logs/<server_name>` no diretório `bin`.
- **Local de log customizado:** Para especificar um local alternativo para os logs do servidor de contêiner, crie um arquivo de propriedades, como `server.properties`, com o seguinte conteúdo:

```
workingDirectory=<directory>
traceSpec=
systemStreamToFileEnabled=true
```

A propriedade **workingDirectory** é o diretório-raiz para os logs e para o arquivo de rastreamento opcional. O WebSphere eXtreme Scale cria um diretório com o nome do servidor de contêiner com um arquivo `SystemOut.log`, um arquivo `SystemErr.log` e um arquivo de rastreamento. Para usar um arquivo de propriedades durante a inicialização de contêiner, use a opção **-serverProps** e forneça o local do arquivo de propriedades do servidor.

- **Ative os logs no WebSphere Application Server.**

Consulte WebSphere Application Server: Ativando e Desativando Criação de Log para obter mais informações.

- **Recupere os arquivos FFDC.**

Os arquivos FFDC servem para que o suporte IBM auxilie na depuração. Estes arquivos poderão ser solicitados pelo suporte IBM quando ocorrer um problema. Esses arquivos aparecem no diretório `ffdc` e contêm arquivos semelhantes ao seguinte:

```
server2_exception.log
server2_20802080_07.03.05_10.52.18_0.txt
```

## O que Fazer Depois

Visualize os arquivos de log e seus locais especificados. As mensagens comuns para procurar no arquivo `SystemOut.log` são as mensagens de confirmação de início, como o seguinte exemplo:

```
CWOBJ1001I: ObjectGrid Server catalogServer01 is ready to process requests.
```

Para obter mais informações sobre uma mensagem específica nos arquivos de log, consulte Mensagens.

---

## Coletando Rastreo

É possível usar o rastreo para monitorar e solucionar problemas em seu ambiente. O rastreo deve ser fornecido para um servidor quando trabalhar com o suporte IBM.

### Sobre Esta Tarefa

Coletar rastreo pode ajudá-lo a monitorar e corrigir problemas em sua implementação do WebSphere eXtreme Scale. A maneira como o rastreo é coletado depende da sua configuração. Consulte “Opções de Rastreo” na página 527 para obter a lista das especificações de rastreo diferentes que podem ser coletadas.

### Procedimento

- **Colete o rastreo dentro de um ambiente WebSphere Application Server.**  
Se os seus servidores de catálogos e de contêiner estiverem em um ambiente WebSphere Application Server, consulte WebSphere Application Server: Trabalhando com Rastreo para obter mais informações.
- **Colete o rastreo com o comando start do servidor de catálogos ou de contêiner independente.**

É possível configurar o rastreo em um servidor de serviço de catálogo ou de contêiner usando os parâmetros **-traceSpec** e **-traceFile** com o comando **startOgServer**. Por exemplo:

```
startOgServer.sh catalogServer -traceSpec ObjectGridPlacement=all=enabled -traceFile /home/user1/logs/trace.log
```

O parâmetro **-traceFile** é opcional. Se não configurar um local **-traceFile**, o arquivo de rastreo irá para o mesmo local dos arquivos de log do sistema. Para obter mais informações sobre esses parâmetros, consulte “ScriptstartOgServer” na página 393.

- **Colete o rastreo no servidor de catálogos ou de contêiner independente com um arquivo de propriedades.**

Para coletar rastreo a partir de um arquivo de propriedades, crie um arquivo, como `server.properties`, com o seguinte conteúdo:

```
workingDirectory=<directory>
traceSpec=<trace_specification>
systemStreamToFileEnabled=true
```

A propriedade **workingDirectory** é o diretório-raiz para os logs e para o arquivo de rastreo opcional. Se o valor **workingDirectory** não estiver configurado, o diretório de trabalho padrão será o local usado para iniciar os servidores, como `wxs_home/bin`. Para usar um arquivo de propriedades durante a inicialização do servidor, utilize o parâmetro **-serverProps** com o comando **startOgServer** e forneça o local do arquivo de propriedades do servidor. Para obter mais

informações sobre o arquivo de propriedades do servidor e como usar esse arquivo, consulte Arquivo de Propriedades do Servidor.

- **Colete o rastreo em um cliente independente.**

É possível iniciar a coleção de rastreo em um cliente independente ao incluir propriedades do sistema no script de inicialização para o aplicativo cliente. No exemplo a seguir, as configurações de rastreo são especificadas para o aplicativo `com.ibm.samples.MyClientProgram`:

```
java -DtraceSettingsFile=MyTraceSettings.properties  
-Djava.util.logging.manager=com.ibm.ws.bootstrap.WsLogManager  
-Djava.util.logging.configFileByServer=true com.ibm.samples.MyClientProgram
```

Consulte WebSphere Application Server: Ativando Rastreo em Aplicativos Clientes e Independentes para obter mais informações.

- **Colete o rastreo com a interface ObjectGridManager.**

Também é possível configurar o rastreo durante o tempo de execução em uma interface `ObjectGridManager`. A configuração de um rastreo em uma interface `ObjectGridManager` pode ser usada para obter rastreo em um cliente eXtreme Scale enquanto ele se conecta com um eXtreme Scale e confirma as transações. Para configurar o rastreo em uma interface `ObjectGridManager`, forneça uma especificação de rastreo e um log de rastreo.

```
ObjectGridManager manager= ObjectGridManagerFactory.getObjectGridManager();  
...  
manager.setTraceEnabled(true);  
manager.setTraceFileName("logs/myClient.log");  
manager.setTraceSpecification("ObjectGridReplication=all=enabled");
```

Para obter mais informações sobre a interface `ObjectGridManager`, consulte as informações sobre como interagir com o `ObjectGrid` usando a interface `ObjectGridManager` no *Guia de Programação*.

- **Colete o rastreo em servidores de contêiner com o utilitário xscmd.**

Para coletar o rastreo com o utilitário `xscmd`, use o comando `-c setTraceSpec`. Use o utilitário `xscmd` para coletar o rastreo em um ambiente independente durante o tempo de execução em vez de durante a inicialização. É possível coletar o rastreo em todos os servidores e serviços de catálogo ou filtrar os servidores com base no nome do `ObjectGrid`, e em outras propriedades. Por exemplo, para coletar o rastreo `ObjectGridReplication` com acesso ao servidor de serviço de catálogo, execute:

```
xscmd -c setTraceSpec "ObjectGridReplication=all=enabled"
```

Também é possível desativar o rastreo ao configurar a especificação de rastreo para `*=all=disabled..`

## Resultados

Os arquivos de rastreo são gravados no local especificado.

## Opções de Rastreo

É possível ativar o rastreo para fornecer informações sobre o seu ambiente para o suporte IBM.

### Sobre o Rastreo

O rastreo do WebSphere eXtreme Scale é dividido em vários componentes diferentes. É possível especificar o nível de rastreo a ser usado. Os níveis comuns de rastreo incluem: `all`, `debug`, `entryExit` e `event`.

Um exemplo de cadeia de rastreamento é o seguinte:

```
ObjectGridComponent=level=enabled
```

É possível concatenar as cadeias de rastreamento. Use o sinal de asterisco (\*) para especificar um valor de curinga, como `ObjectGrid*=all=enabled`. Se for necessário fornecer um rastreamento para o suporte IBM, uma cadeia de rastreamento específica será solicitada. Por exemplo, se ocorrer um problema com a replicação, a cadeia de rastreamento `ObjectGridReplication=debug=enabled` pode ser solicitada.

## Especificação de Rastreamento

### **ObjectGrid**

Mecanismo de cache principal geral.

### **ObjectGridCatalogServer**

Serviço de catálogo geral.

### **ObjectGridChannel**

Comunicações de topologia de implementação estática.

### **ObjectGridClientInfo**

Informações do cliente do DB2.

### **ObjectGridClientInfoUser**

Informações sobre o usuário do DB2.

### **ObjectgridCORBA**

Comunicações de topologia de implementação dinâmica.

### **ObjectGridDataGrid**

A API do AgentManager.

### **ObjectGridDynaCache**

O provedor de cache dinâmico do WebSphere eXtreme Scale.

### **ObjectGridEntityManager**

A API do EntityManager. Utilize com a opção Projector.

### **ObjectGridEvictors**

Evictors integrados do ObjectGrid.

### **ObjectGridJPA**

Carregadores do Java Persistence API (JPA).

### **ObjectGridJPACache**

Plug-ins do Cache JPA

### **ObjectGridLocking**

Gerenciador de bloqueios de entrada de cache do ObjectGrid.

### **ObjectGridMBean**

Beans de gerenciamento.

### **ObjectGridMonitor**

Infraestrutura de monitoramento histórico.

### **7.1.1+ ObjectGridNative**

Rastreamento do código nativo do WebSphere eXtreme Scale, incluindo o código nativo eXtremeMemory.

### **7.1.1+ ObjectGridOSGi**

Os componente de integração de OSGi do WebSphere eXtreme Scale.

### **ObjectGridPlacement**

Serviço de disposição de shards do servidor de catálogos.

- ObjectGridQuery**  
Consulte ObjectGrid.
- ObjectGridReplication**  
Serviço de replicação.
- ObjectGridRouting**  
Detalhes de roteamento do cliente/servidor.
- ObjectGridSecurity**  
Rastreio de segurança.
- 7.1.1+ ObjectGridSerializer**  
A infraestrutura do plug-in DataSerializer.
- ObjectGridStats**  
Estatísticas do ObjectGrid.
- ObjectGridStreamQuery**  
API de Consulta do Fluxo.
- 7.1.1+ ObjectGridTransactionManager**  
O gerenciador de transações do WebSphere eXtreme Scale.
- ObjectGridWriteBehind**  
Atributo write-behind do ObjectGrid.
- 7.1.1+ ObjectGridXM**  
Rastreio geral do IBM eXtremeMemory.
- 7.1.1+ ObjectGridXMEviction**  
Rastreio de despejo do eXtremeMemory.
- 7.1.1+ ObjectGridXMTransport**  
Rastreio de transporte geral do eXtremeMemory.
- 7.1.1+ ObjectGridXMTransportInbound**  
Rastreio de transporte específico da entrada do eXtremeMemory.
- 7.1.1+ ObjectGridXMTransportOutbound**  
Rastreio de transporte específico da entrada do eXtremeMemory.
- Projector**  
O mecanismo dentro da API do EntityManager.
- QueryEngine**  
O mecanismo de consulta para a API de Consulta do Objeto e a API de Consulta do EntityManager.
- QueryEnginePlan**  
Rastreio do plano de consulta.
- 7.1.1+ TCPChannel**  
O canal TCP/IP do IBM eXtremeIO.
- 7.1.1+ XsByteBuffer**  
Rastreio do buffer de bytes do WebSphere eXtreme Scale.

---

## Analizando Dados de Log e de Rastreio

É possível usar as ferramentas de análise de log para analisar como seu tempo de execução está executando e resolver problemas que ocorrem no ambiente.

## Sobre Esta Tarefa

É possível gerar relatórios a partir dos arquivos existentes de log e de rastreamento no ambiente. Esses relatórios visuais podem ser usados para as seguintes finalidades:

- **Analisar o status e o desempenho do ambiente de tempo de execução:**
  - Implementar consistência do ambiente
  - Criar log da frequência
  - Topologia de execução versus topologia configurada
  - Mudanças de topologia não planejadas
  - Status de quorum
  - Status de replicação de partição
  - Estatísticas de memória, rendimento, uso do processador, e assim por diante
- **Para resolver problemas no ambiente:**
  - Visualizações de topologia em pontos no tempo específicos
  - Estatísticas de memória, rendimento, uso do processador durante falhas do cliente
  - Níveis de fix pack atuais, ajuste de configurações
  - Status de quorum

## Visão Geral de Análise de Log

É possível usar a ferramenta **xsLogAnalyzer** para ajudar a resolver problemas no ambiente.

### Todas as Mensagens de Failover

Exibe o número total de mensagens de failover como um gráfico ao longo do tempo. Também exibe uma lista das mensagens de failover, incluindo os servidores que foram afetados.

### Todas as Mensagens Críticas do eXtreme Scale

Exibe os IDs de mensagem junto com as explicações e ações do usuário associadas, o que pode economizar o tempo de procura por mensagens.

### Todas as Exceções

Exibe as cinco principais exceções, incluindo as mensagens e quantas vezes elas ocorreram e quais servidores foram afetados pela exceção.

### Resumo de Topologia

Exibe um diagrama de como sua topologia é configurada de acordo com os arquivos de log. Este resumo pode ser usado para comparar a configuração real, possivelmente a identificação de erros de configuração.

### Consistência de Topologia: Tabela de Comparação do Object Request Broker (ORB)

Exibe configurações ORB no ambiente. É possível usar essa tabela para ajudar a determinar se as configurações estão consistentes em seu ambiente.



## Visualização de Linha de Tempo de Eventos

Exibe um diagrama de linha de tempo de diferentes ações que ocorreram na grade de dados, incluindo eventos de ciclo de vida, exceções, mensagens críticas e eventos de captura de dados de erros (FFDC).

## Executando Análise de Log

É possível executar a ferramenta **xsLogAnalyzer** em um conjunto de arquivos de log e de rastreo a partir de qualquer computador.

### Antes de Iniciar

- Ative os logs e rastreo. Consulte “Ativando a Criação de Log” na página 525 e “Coletando Rastreo” na página 526 para obter mais informações.
- Colete os arquivos de log. Os arquivos de log podem estar em vários locais dependendo de como eles foram configurados. Se você estiver usando as configurações de log padrão, os arquivos de log poderão ser obtidos a partir dos seguintes locais:
  - Em uma instalação independente: `wxs_install_root/bin/logs/<server_name>`
  - Em uma instalação integrada ao WebSphere Application Server: `was_root/logs/<server_name>`
- Colete os arquivos de rastreo. Os arquivos de rastreo podem estar em vários locais dependendo de como eles foram configurados. Se você estiver usando as configurações de rastreo padrão, os arquivos de trace poderão ser obtidos a partir dos seguintes locais:
  - Em uma instalação independente: Se nenhum valor de rastreo específico for configurado, os arquivos de rastreo serão gravados para o mesmo local que o sistema de arquivos de log.
  - Em uma instalação que é integrada com o WebSphere Application Server: `was_root/profiles/server_name/logs`.

Copie os arquivos de log e rastreo no computador a partir do qual você está planejando usar a ferramenta Log Analyzer.

- Se desejar criar scanners customizados no relatório gerado, crie um arquivo de propriedades de especificações do scanner e o arquivo de configuração antes de executar a ferramenta. Para obter informações adicionais, consulte “Criando Scanners Customizados para Análise do Log” na página 532.

### Procedimento

1. Execute a ferramenta **xsLogAnalyzer**.

O script está nos seguintes locais:

- Em uma instalação independente : `wxs_install_root/ObjectGrid/bin`
- Em uma instalação integrada ao WebSphere Application Server: `was_root/bin`

**Dica:** Se seus arquivos de log forem grandes, considere usar os parâmetros **-startTime**, **-endTime** e **-maxRecords** quando executar o relatório para restringir o número de entradas de log que são varridos. Usar esses parâmetros quando você executa o relatório torna os relatórios mais fáceis de ler e de executar com mais eficiência. É possível executar diversos relatórios no mesmo conjunto de arquivos de log.

```
xsLogAnalyzer.sh|bat -logsRoot c:\myxslogs -outDir c:\myxslogs\out  
-startTime 11.09.27_15.10.56.089 -endTime 11.09.27_16.10.56.089 -maxRecords 100
```

**-logsRoot**

Especifica o caminho absoluto para o diretório de log que você deseja avaliar (necessário).

**-outDir**

Especifica um diretório existente para gravar a saída do relatório. Se você não especificar um valor, o relatório será gravado no local raiz da ferramenta **xsLogAnalyzer**.

**-startTime**

Especifica o horário de início para avaliar nos logs. A data está no formato a seguir: *year.month.day.hour.minute.second.millisecond*

**-endTime**

Especifica o horário de encerramento para avaliar nos logs. A data está no formato a seguir: *year.month.day.hour.minute.second.millisecond*

**-trace** Especifica uma sequência de rastreamento, tal como `ObjectGrid*=all=enabled`.

**-maxRecords**

Especifica o número máximo de registros a serem gerados no relatório. O padrão é 100. Se você especificar o valor como 50, os primeiros 50 registros serão gerados para o período de tempo especificado.

- Abra os arquivos gerados. Se um diretório de saída não foi definido, os relatórios serão gerados em uma pasta chamada `report_date_time`. Para abrir a página principal dos relatórios, abra o arquivo `index.html`.
- Use os relatórios para analisar os dados do log. Use as dicas a seguir para maximizar o desempenho das exibições de relatório:
  - Para maximizar o desempenho de consultas nos dados de log, use informações o mais específicas possível. Por exemplo, uma consulta para `server` leva muito mais tempo para ser executada e retorna mais resultados do que `server_host_name`.
  - Algumas visualizações têm um número limitado de pontos de dados que são exibidos de uma vez. É possível ajustar o segmento de tempo que está sendo visualizado alterando os dados atuais, tais como a hora de início e de encerramento, na visualização.

## O que Fazer Depois

Para obter mais informações sobre resolução de problemas da ferramenta **xsLogAnalyzer** e os relatórios gerados, consulte "Resolução de Problemas da Análise do Log" na página 534.

## Criando Scanners Customizados para Análise do Log

É possível criar scanners customizados para análise do log. Depois de configurar o scanner, os resultados são gerados nos relatórios quando você executa a ferramenta **xsLogAnalyzer**. O scanner customizado varre os logs para os registros de eventos com base nas expressões regulares que você especificou.

### Procedimento

- Crie um arquivo de propriedades de especificações do scanner que especifica a expressão geral para executar o scanner customizado.
  - Crie e salve um arquivo de propriedades. O arquivo deve estar no diretório `logalyzer_root/config/custom`. É possível nomear o arquivo como: `you like`. O arquivo é usado pelo novo scanner, portanto, nomear o scanner no

arquivo de propriedades é útil, por exemplo:  
`my_new_server_scanner_spec.properties`.

- b. Inclua as propriedades a seguir no arquivo `my_new_server_scanner_spec.properties`:  
`include.regular_expression = REGULAR_EXPRESSION_TO_SCAN`

A variável `REGULAR_EXPRESSION_TO_SCAN` é uma expressão regular na qual filtrar os arquivos de log.

Exemplo: Para varrer em busca de instâncias de linhas que contêm as sequências "xception" e "rroor" independentemente da ordem, configure a propriedade **include.regular\_expression** com o valor a seguir:

```
include.regular_expression = (xception.+rroor)|(rroor.+xception)
```

Esta expressão regular faz com que os eventos sejam registrados se a sequência "rroor" vier antes ou após a sequência "xception".

Exemplo: Para varrer através de cada linha nos logs em busca de instâncias de linhas que contêm as sequências de frase "xception" ou a frase "rroor" independentemente da ordem, configure a propriedade **include.regular\_expression** com o valor a seguir:

```
include.regular_expression = (xception)|(rroor)
```

Esta expressão regular faz com que os eventos sejam registrados se a sequência "rroor" vier antes ou após a sequência "xception".

2. Crie um arquivo de configuração que a ferramenta **xsLogAnalyzer** usa para criar o scanner.
  - a. Crie e salve um arquivo de propriedades. O arquivo deve estar no diretório `loganalyzer_root/config/custom`. É possível nomear o arquivo como `scanner_nameScanner.config`, em que `scanner_name` é um nome exclusivo para o novo scanner. Por exemplo, você pode nomear o arquivo `serverScanner.config`
  - b. Inclua as propriedades a seguir no arquivo `scanner_nameScanner.config`:  
`scannerSpecificationFiles = LOCATION_OF_SCANNER_SPECIFICATION_FILE`

A variável `LOCATION_OF_SCANNER_SPECIFICATION_FILE` é o caminho e o local do arquivo de especificação que você criou na etapa anterior. Por exemplo: `loganalyzer_root/config/custom/my_new_scanner_spec.properties`. Também é possível especificar diversos arquivos de especificação de scanner usando uma lista separada por ponto e vírgula:

```
scannerSpecificationFiles = LOCATION_OF_SCANNER_SPECIFICATION_FILE1;LOCATION_OF_SCANNER_SPECIFICATION_FILE2
```

3. Execute a ferramenta **xsLogAnalyzer**. Para obter informações adicionais, consulte "Executando Análise de Log" na página 531.

## Resultados

Depois de executar a ferramenta **xsLogAnalyzer**, o relatório contém novas guias no relatório para os scanners customizados que você configurou. Cada guia contém as visualizações a seguir:

### Gráficos

Um gráfico plotado que ilustra os eventos registrados. Os eventos são exibidos na ordem na qual os eventos foram localizados.

## Tabelas

Uma representação tabular dos eventos registrados.

## Relatórios Resumo

# Resolução de Problemas da Análise do Log

Use as informações de resolução de problemas a seguir para diagnosticar e corrigir problemas com a ferramenta **xsLogAnalyzer** e seus relatórios gerados.

## Procedimento

- **Problema:** Ocorrem condições de falta de memória quando você está usando a ferramenta **xsLogAnalyzer** para gerar relatórios. Um exemplo de um erro que pode ocorrer é o seguinte: `java.lang.OutOfMemoryError: Limite de sobrecarga de GC excedido`.

**Solução:** A ferramenta **xsLogAnalyzer** é executada dentro de uma Java virtual machine (JVM). É possível configurar a JVM para aumentar o tamanho de heap antes de executar a ferramenta **xsLogAnalyzer** especificando algumas configurações quando você executa a ferramenta. O aumento do tamanho de heap permite que mais registros de eventos sejam armazenados na memória da JVM. Comece com uma configuração de 2048 M, assumindo que o sistema operacional possui memória principal suficiente. Na mesma instância de linha de comandos na qual você está planejando executar a ferramenta **xsLogAnalyzer**, configure o tamanho máximo de heap da JVM:

```
java -XmxHEAP_SIZEm
```

O valor de `HEAP_SIZE` pode ser qualquer número inteiro e representa o número de megabytes que são alocados para o heap da JVM. Por exemplo, você pode executar `java -Xmx2048m`. Se as mensagens de falta de memória continuarem, ou você não tem os recursos para alocar 2048m ou mais de memória, limite o número de eventos que estão sendo mantidos no heap. É possível limitar o número máximo de eventos no heap transmitindo o parâmetro **-maxRecords** para o comando **xsLogAnalyzer**

- **Problema:** Quando você abre um relatório gerado a partir da ferramenta **xsLogAnalyzer**, o navegador é interrompido ou não carrega a página.

**Causa:** Os arquivos HTML gerados são muito grandes e não podem ser carregados pelo navegador. Esses arquivos são grandes porque o escopo dos arquivos de log que você está analisando é muito amplo.

**Solução:** Considere usar os parâmetros **-startTime**, **-endTime** e **-maxRecords** quando você executar a ferramenta **xsLogAnalyzer** para restringir o número de entradas de log que são varridas. Usar esses parâmetros quando você executa o relatório torna os relatórios mais fáceis de ler e de executar com mais eficiência. É possível executar diversos relatórios no mesmo conjunto de arquivos de log.

---

## Instalação de Resolução de Problemas

Use estas informações para resolver problemas com sua instalação.

## Procedimento

- **Problema:** Quando você executa o comando de instalação a partir de um computador remoto, como `\\mymachine\downloads\`, a seguinte mensagem é exibida: `CMD.EXE foi iniciado com o caminho acima como o diretório atual`. Caminhos UNC não são suportados. Padronizando para o diretório Windows. Como resultado, a instalação não é concluída corretamente.

**Solução:** Mapeie o computador remoto para uma unidade de rede. Por exemplo, no Windows, é possível clicar com o botão direito em **Meu Computador**, escolher **Mapear Unidade de Rede** e incluir o caminho das convenções de nomenclatura uniformes (UNC) no computador remoto. Em seguida, é possível executar o script de instalação a partir da unidade de rede com êxito, por exemplo: `y:\mymachine\downloads\WXS\install.bat`.

- **Problema:** A instalação não é concluída com êxito.

**Solução:** Verifique os arquivos de log para ver onde a instalação falhou. Quando a instalação não é concluída com êxito, os logs estão no diretório `wxs_install_root/logs/wxs`.

- **Problema:** Uma falha catastrófica ocorre durante a instalação.

**Solução:** Verifique os arquivos de log para ver onde a instalação falhou. Quando a instalação falha quando estiver parcialmente concluída, os logs geralmente podem estar localizados no diretório `user_root/wxs_install_logs/`.

- **Windows Problema:** Se estiver instalando o WebSphere eXtreme Scale Client no Windows, o seguinte texto poderá ser exibido nos resultados da instalação:

Sucesso: A instalação do seguinte produto foi bem-sucedida:  
WebSphere eXtreme Scale Client. Algumas etapas de configuração têm erros.  
Para obter mais informações, consulte o seguinte arquivo de log:  
<WebSphere Application Server install root>\logs\wxs\_client\install\log.txt"  
Revise o log de instalação (log.txt) e o log de aumento do gerenciador de implementação

**Solução:** Se uma falha for vista com o arquivo `iscdeploy.sh`, o erro poderá ser ignorado. Este erro não causa nenhum problema.

---

## Resolvendo Problemas da Integração de Cache

Use estas informações para resolver problemas com a configuração de integração de seu cache, incluindo sessões HTTP e configurações de cache dinâmico.

### Procedimento

- **7.1.1+ Problema:** Os IDs da sessão de HTTP não estão sendo reutilizados.

**Causa:** É possível reutilizar os IDs de sessão. Se você criar uma grade de dados para persistência de sessão na Versão 7.1.1 ou posterior, a reutilização do ID de sessão será automaticamente ativada. No entanto, se você criou configurações anteriores, esta configuração já poderá ser configurada com o valor incorreto.

**Solução:** Consulte as seguintes configurações para verificar se a reutilização do ID de sessão de HTTP está ativada:

- A propriedade `reuseSessionId` no arquivo `splicer.properties` deve ser configurada para `true`.
- O valor da propriedade customizada `HttpSessionIdReuse` deve ser configurado para `true`. Essa propriedade customizada pode ser configurada em um dos seguintes caminhos no console administrativo do WebSphere Application Server:
  - **Servidores** > *server\_name* > **Gerenciamento de Sessões** > **Propriedades customizadas**
  - **Clusters Dinâmicos** > *dynamic\_cluster\_name* > **Modelo do Servidor** > **Gerenciamento de Sessão** > **Propriedades Customizadas**
  - **Servidores** > **Tipos de Servidor** > **Servidores de Aplicativos WebSphere** > *server\_name* e, em seguida, em **Infraestrutura do Servidor**, clique em

Gerenciamento Java e processos > Definição de Processo > Java virtual machine > Propriedades Customizadas

- Servidores > Tipos de Servidor > Servidores de Aplicativos do WebSphere > *server\_name* > Configurações do Contêiner da Web > Contêiner da Web.

Se você atualizar quaisquer valores de propriedade customizada, reconfigure o gerenciamento de sessão eXtreme Scale para que o arquivo `splicer.properties` reconheça a mudança.

- **Problema:** Quando você estiver usando uma grade de dados para armazenar as sessões HTTP e o carregamento de transações for alta, uma mensagem CWOBJ0006W será exibida no arquivo `SystemOut.log`.

```
CWOBJ0006W: Ocorreu uma exceção:  
com.ibm.websphere.objectgrid.ObjectGridRuntimeException:  
java.util.ConcurrentModificationException
```

Esta mensagem ocorre apenas quando o parâmetro **replicationInterval** no arquivo `splicer.properties` é configurado com um valor maior que zero e o aplicativo da Web modifica um objeto `List` que foi configurado como um atributo na `HTTPSession`.

**Solução:** Clone o atributo que contém o objeto `List` modificado e coloque o atributo clonado no objeto de sessão.

---

## Resolução de Problemas do Plug-in do Cache JPA

Use estas informações para resolver problemas com sua configuração de plug-in do cache JPA. Estes problemas podem ocorrer em ambas as configurações Hibernate e OpenJPA.

### Procedimento

- **Problema:** A exceção a seguir é exibida: `CacheException: Falha ao obter o servidor ObjectGrid`.

Com um valor de atributo **ObjectGridType** de `EMBEDDED` ou `EMBEDDED_PARTITION`, o cache eXtreme Scale tenta obter uma instância do servidor a partir do tempo de execução. Em um ambiente Java Platform, Standard Edition, um servidor eXtreme Scale com serviço de catálogo integrado é iniciado. O serviço de catálogo integrado tentará atender na porta 2809. Se essa porta estiver sendo usada por outro processo, ocorrerá um erro.

**Solução:** Se terminais de serviço de catálogo externo forem especificados, por exemplo, com o arquivo `objectGridServer.properties`, este erro ocorrerá se o nome do host ou a porta for especificada incorretamente. Corrija o conflito de porta.

- **Problema:** A exceção a seguir é exibida: `CacheException: Falha ao obter o REMOTE ObjectGrid para o REMOTE ObjectGrid configurado. objectGridName = [ObjectGridName], Nome da PU = [persistenceUnitName]`

Esse erro ocorre porque o cache não pode obter a instância do `ObjectGrid` a partir dos terminais de serviço de catálogo fornecidos.

**Solução:** Esse problema geralmente ocorre devido a um nome de host ou porta incorreto.

- **Problema:** A exceção a seguir é exibida: `CacheException: Não é possível ter duas PUs [persistenceUnitName_1, persistenceUnitName_2] configuradas com o mesmo ObjectGridName [ObjectGridName] do EMBEDDED ObjectGridType`  
Essa exceção ocorrerá se você tiver muitas unidades de persistência configuradas e os caches do eXtreme Scale destas unidades forem configurados com o mesmo



nome de ObjectGrid e valor de atributo EMBEDDED do **ObjectGridType**. Estas configurações de unidade de persistência podem estar no mesmo arquivo `persistence.xml` ou diferente.

**Solução:** É necessário verificar se o nome do ObjectGrid é exclusivo para cada unidade de persistência quando o valor do atributo **ObjectGridType** for EMBEDDED.

- **Problema:** A exceção a seguir é exibida: `CacheException: O REMOTE ObjectGrid [ObjectGridName] não inclui os BackingMaps [mapName_1, mapName_2,...] necessários`

Com um tipo de ObjectGrid REMOTE, se o ObjectGrid do lado do cliente obtido não tiver os mapas de apoio de entidade completos para suportar o cache da unidade de persistência, esta exceção ocorrerá. Por exemplo, cinco classes de entidade são listadas na configuração da unidade de persistência, mas o ObjectGrid obtido possui apenas dois BackingMaps. Embora o ObjectGrid obtido possa ter 10 BackingMaps, se algum destes cinco BackingMaps de entidade necessários não for localizado nos dez BackingMaps, esta exceção ainda ocorrerá.

**Solução:** Certifique-se de que a configuração do mapa de apoio suporte o cache da unidade de persistência.

---

## Resolução de Problemas de Administração

Use as informações a seguir para resolver problemas de administração, incluindo iniciar e reiniciar os servidores, usando o utilitário **xscmd**, e assim por diante.

### Procedimento

- **Problema:** Scripts de administração estão ausentes do diretório `profile_root/bin` de uma instalação do WebSphere Application Server.

**Causa:** Quando você atualiza a instalação, novos arquivos de script não são instalados automaticamente nos perfis.

**Solução:** Se você deseja executar um script a partir de seu diretório `profile_root/bin`, cancele o aumento e aumente novamente o perfil com a liberação mais recente. Para obter mais informações, consulte *Cancelando o Aumento de um Perfil Usando o Prompt de Comandos e “Criando e Alterando Perfis para o WebSphere eXtreme Scale”* na página 182.

- **Problema:** Quando você está executando um comando **xscmd**, a seguinte mensagem é impressa na tela:

```
java.lang.IllegalStateException: Placement service MBean not available.
[]
    at
    com.ibm.websphere.samples.objectgrid.admin.OGAdmin.main(OGAdmin.java:1449)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at
    sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:60)
    at
    sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:37)
    at java.lang.reflect.Method.invoke(Method.java:611)
    at com.ibm.ws.bootstrap.WSLauncher.main(WSLauncher.java:267)
Ending at: 2011-11-10 18:13:00.000000484
```

**Causa:** Um problema de conexão ocorreu com o servidor de catálogo.

**Solução:** Verifique se seus servidores de catálogos estão em execução e estão disponíveis pela rede. Esta mensagem também pode ocorrer quando você possui um domínio de serviço de catálogo definido, mas menos de dois servidores de catálogos estão em execução. O ambiente não está disponível até dois servidores de catálogo serem iniciados.



---

## Resolução de Problemas de Várias Configurações do Datacenter

Use estas informações para resolver problemas de diversas configurações do datacenter, incluindo vinculação entre domínios do serviço de catálogo.

### Procedimento

**Problema:** Os dados estão ausentes em um ou mais domínios do serviço de catálogo. Por exemplo, você pode executar o comando `xscmd -c establishLink`. Quando você examina os dados para cada domínio do serviço de catálogo vinculado, os dados têm uma outra aparência, por exemplo, a partir do comando `xscmd -c showMapSizes`.

**Solução:** É possível solucionar esse problema com o comando `xscmd -c showLinkedPrimaries`. Esse comando imprime cada shard primário, incluindo quais primários estrangeiros são vinculados.

No cenário descrito, é possível descobrir, a partir da execução do comando `xscmd -c showLinkedPrimaries`, que os shards primários do primeiro domínio do serviço de catálogo são vinculados aos shards primários do segundo domínio de serviço de catálogo, porém o segundo domínio de serviço de catálogo não possui links para o primeiro domínio de serviço de catálogo. Você pode considerar executar novamente o comando `xscmd -c establishLink` a partir do segundo domínio de serviço de catálogo para o primeiro domínio de serviço de catálogo.

---

## Resolução de Problemas de Carregadores

Use estas informações para resolver problemas com os carregadores de banco de dados.

### Procedimento

- **Problema:** Quando estiver usando um carregador OpenJPA com DB2 no WebSphere Application Server, uma exceção de cursor fechado ocorre.

A exceção a seguir é do DB2 no arquivo de log  
`org.apache.openjpa.persistence.PersistenceException:  
[jcc][t4][10120][10898][3.57.82] Invalid operation: result set is closed.`

**Solução:** Por padrão, o servidor de aplicativos configura a propriedade customizada `resultSetHoldability` com um valor de 2 (`CLOSE_CURSORS_AT_COMMIT`). Esta propriedade faz com o DB2 feche seu `resultSet/cursor` nos limites da transação. Para remover a exceção, altere o valor da propriedade customizada para 1 (`HOLD_CURSORS_OVER_COMMIT`). Configure a propriedade customizada `resultSetHoldability` no seguinte caminho na célula do WebSphere Application Server: **Recursos > Provedor JDBC > Provedor Driver Universal JDBC > DataSources > data\_source\_name > Propriedades Customizadas > Novo.**

- **Problema** O DB2 exibe uma exceção: A transação atual foi retrocedida devido a um conflito ou tempo limite. Código de razão "2".. `SQLCODE=-911, SQLSTATE=40001, DRIVER=3.50.152`

Essa exceção ocorre devido a um problema de contenção de bloqueio quando estiver executando com o OpenJPA com DB2 no WebSphere Application Server. O nível de isolamento padrão para a Leitura Repetida (RR) do WebSphere Application Server, que obtém bloqueios de longa duração com o DB2. **Solução:**

Configure o nível de isolamento para Leitura Confirmada para reduzir a contenção de bloqueio. Configure a propriedade customizada da origem de dados `webSphereDefaultIsolationLevel` para configurar o nível de isolamento

para 2(TRANSACTION\_READ\_COMMITTED) no seguinte caminho na célula WebSphere Application Server: **Recursos > Provedor JDBC > JDBC\_provider > Origens de Dados > data\_source\_name > Propriedades Customizadas > Novo**. Para obter informações adicionais sobre a propriedade customizada webSphereDefaultIsolationLevel e níveis de isolamento de transação, consulte Requisitos para Configurar os Níveis de Isolamento de Acesso a Dados.

- **Problema:** Quando estiver usando a função de pré-carregamento do JPALoader ou JPAEntityLoader, a seguinte mensagem CWOBJ15111 não é exibida para a partição em um servidor de contêiner: CWOBJ15111: GRID\_NAME:MAPSET\_NAME:PARTITION\_ID (primário) está aberto para negócios. Em vez disso, ocorrerá uma exceção TargetNotAvailableException no servidor de contêiner, que ativa a partição que é especificada pela propriedade preloadPartition.

**Solução:** Configure o atributo preloadMode para true se você utilizar um JPALoader ou JPAEntityLoader para pré-carregar dados no mapa. Se a propriedade preloadPartition da JPALoader ou JPAEntityLoader estiver configurada para um valor entre 0 e total\_number\_of\_partitions-1, então o JPALoader e o JPAEntityLoader tentarão pré-carregar os dados a partir do banco de dados backend no mapa. O trecho de código a seguir ilustra como o atributo preloadMode é configurado para ativar o pré-carregamento assíncrono:

```
BackingMap bm = og.defineMap( "map1" );
bm.setPreloadMode( true );
```

Também é possível configurar o atributo preloadMode usando um arquivo XML conforme ilustrado no seguinte exemplo:

```
<backingMap name="map1" preloadMode="true" pluginCollectionRef="map1"
lockStrategy="OPTIMISTIC" />
```

---

## Resolução de Problemas de Configuração XML

Quando você configura o eXtreme Scale, é possível encontrar um comportamento inesperado com seus arquivos XML. As seções a seguir descrevem problemas que podem ocorrer e soluções.

### Procedimento

- **Problema:** A política de implementação e os arquivos XML do ObjectGrid devem corresponder.

A política de implementação e os arquivos XML de ObjectGrid devem corresponder. Se eles não possuem nomes de ObjectGrid e nomes de mapas correspondentes, ocorrem erros.

Se a lista de backingMap em um arquivo XML de ObjectGrid não corresponder à lista de referências de mapas em um arquivo XML da política de implementação, ocorre um erro no servidor de catálogos.

Por exemplo, o seguinte arquivo XML de ObjectGrid e arquivo XML da política de implementação são utilizados para iniciar um processo de contêiner. O arquivo da política de implementação possui mais referências de mapas que são listados no arquivo XML de ObjectGrid.

**ObjectGrid.xml - exemplo incorreto**

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="accounting">
      <backingMap name="payroll" readOnly="false" />
    </objectGrid>
  </objectGrids>
</objectGridConfig>
```

#### deploymentPolicy.xml - exemplo incorreto

```
<?xml version="1.0" encoding="UTF-8"?>
<deploymentPolicy xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/deploymentPolicy ../deploymentPolicy.xsd"
xmlns="http://ibm.com/ws/objectgrid/deploymentPolicy">
  <objectgridDeployment objectgridName="accounting">
    <mapSet name="mapSet1" numberOfPartitions="4" minSyncReplicas="1"
maxSyncReplicas="2" maxAsyncReplicas="1">
      <map ref="payroll"/>
      <map ref="ledger"/>
    </mapSet>
  </objectgridDeployment>
</deploymentPolicy>
```

**Mensagens:** Uma mensagem de erro ocorrerá no arquivo SystemOut.log quando a política de implementação for incompatível com o arquivo XML do ObjectGrid. Para o exemplo precedente, ocorre a seguinte mensagem:

```
CW0BJ3179E: A referência do ledger do mapa no mapSet mapSet1 do arquivo do descritor de implementação de
contabilidade do ObjectGrid
não faz referência a um mapa de retorno válido do XML do ObjectGrid
.
```

Se a política de implementação estiver sem referências de mapa para os backingMaps listados no arquivo XML do ObjectGrid, ocorrerá uma mensagem de erro no arquivo SystemOut.log. Por exemplo:

```
CW0BJ3178E: O ledger do mapa na contabilidade do ObjectGrid referido no XML do ObjectGrid
não foi localizado no arquivo do descritor de implementação.
```

**Solução:** Determine qual arquivo tem a lista correta e altere o código relevante de acordo.

- **Problema:** Nomes do ObjectGrid incorretos entre arquivos XML também causam um erro.

O nome do ObjectGrid é referido no arquivo XML ObjectGrid e no arquivo XML da política de implementação.

**Mensagem:** Uma ObjectGridException ocorre causada pela exceção de IncompatibleDeploymentPolicyException. A seguir, está um exemplo.

Causado por:

```
com.ibm.websphere.objectgrid.IncompatibleDeploymentPolicyException: O
objectgridDeployment com "accountin" de objectGridName não possui um
objectGrid correspondente no XML do ObjectGrid.
```

O arquivo XML ObjectGrid é a lista principal dos nomes ObjectGrid. Ocorrerá um erro se uma política de implementação possuir um nome ObjectGrid que não está contido no arquivo XML ObjectGrid.

**Solução:** Verifique detalhes, como a ortografia do nome do ObjectGrid. Remova quaisquer nomes extras ou inclua nomes ObjectGrid ausentes nos arquivos XML ObjectGrid ou da política de implementação. Na mensagem de exemplo, o objectGridName está escrito incorretamente como "accountin" em vez de "accounting".

- **Problema:** Alguns dos atributos no arquivo XML podem receber apenas alguns valores. Estes atributos possuem valores aceitáveis enumerados pelo esquema. A lista a seguir fornece alguns dos atributos:
  - atributo authorizationMechanism no elemento objectGrid
  - atributo copyMode no elemento backingMap
  - atributo lockStrategy no elemento backingMap
  - atributo ttlEvictorType no elemento backingMap
  - atributo type no elemento property
  - initialState no elemento objectGrid
  - evictionTriggers no elemento backingMap

Se a um destes atributos for designado um valor inválido, a validação XML falhará. No arquivo XML de exemplo a seguir, é utilizado um valor incorreto de INVALID\_COPY\_MODE:

```
exemplo de INVALID_COPY_MODE
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="accounting">
      <backingMap name="payroll" copyMode="INVALID_COPY_MODE"/>
    </objectGrid/>
  </objectGrids>
</objectGridConfig>
```

A mensagem a seguir aparece no log.

```
CWOBJ2403E: O arquivo XML é inválido. Foi detectado um problema com < null > na linha 5.
A mensagem de erro é cvc-enumeration-valid: O valor 'INVALID_COPY_MODE' não possui uma faceta
válida com relação à enumeração
'[COPY_ON_READ_AND_COMMIT, COPY_ON_READ, COPY_ON_WRITE, NO_COPY, COPY_TO_BYTES]'.
Ela deve ser um valor a partir da enumeração.
```

- **Problema:** Atributos ou tags ausentes ou incorretos em um arquivo XML causa erros, como o seguinte exemplo em que, no arquivo XML do ObjectGrid, falta a tag final < /objectGrid >:

atributos ausentes - XML de exemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectGrid.xsd"
  xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="accounting">
      <backingMap name="payroll" />
  </objectGrids>
</objectGridConfig>
```

### Mensagem:

```
CWOBJ2403E: O arquivo XML é inválido. Foi detectado um
problema com < null > na linha 7. A mensagem de erro é: The
end-tag for element type "objectGrid" must end with a '>' delimiter.
```

Ocorre uma ObjectGridException sobre o arquivo XML inválido com o nome do arquivo XML.

**Solução:** Certifique-se de que as tags e os atributos necessários apareçam nos arquivos XML com o formato correto.

- **Problema:** Se um arquivo XML estiver formatado com sintaxe incorreta ou ausente, o CWOBJ2403E aparecerá no log. Por exemplo, a mensagem a seguir é exibida quando uma aspa está ausente em um dos atributos XML

```
CWOBJ2403E: O arquivo XML é inválido. Foi detectado
um problema com < null > na linha 7. A mensagem de erro é:
Open quote is expected for attribute "maxSyncReplicas"
associated with an element type "mapSet".
```

Uma ObjectGridException sobre o arquivo XML inválido também ocorre.

**Solução:** Várias soluções podem ser usadas para um determinado erro de sintaxe XML. Consulte a documentação relevante sobre a gravação de script XML.

- **Problema:** Referência a uma coleção de plug-in inexistente torna um arquivo XML inválido. Por exemplo, ao usar o XML para definir um plug-ins BackingMap, o atributo pluginCollectionRef do elemento backingMap deve referenciar um backingMapPluginCollection. O atributo pluginCollectionRef deve corresponder aos elementos backingMapPluginCollection.

### Mensagem:

Se o atributo pluginCollectionRef não corresponder a nenhum dos atributos de ID de algum dos elementos backingMapPluginConfiguration, uma mensagem semelhante à seguinte será exibida no log.

```
[7/14/05 14:02:01:971 CDT] 686c060e XmlErrorHandl E CW0BJ9002E:
This is an English only Error message: Invalid XML file. Line: 14; URI: null;
Message: Key 'pluginCollectionRef' with
value 'bookPlugins' not found for identity
constraint of element 'objectGridConfig'.
```

O arquivo XML a seguir é utilizado para produzir o erro. Observe que o nome do BackingMap book possui seu atributo pluginCollectionRef configurado como bookPlugins e o backingMapPluginCollection único possui um ID de collection1.

Referenciando um XML de atributo não-existente - exemplo

```
<?xml version="1.0" encoding="UTF-8"?>
<objectGridConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://ibm.com/ws/objectgrid/config ../objectgrid.xsd"
xmlns="http://ibm.com/ws/objectgrid/config">
  <objectGrids>
    <objectGrid name="bookstore">
      <backingMap name="book" pluginCollectionRef="bookPlugin" />
    </objectGrid>
  </objectGrids>
  <backingMapPluginCollections>
    <backingMapPluginCollection id="collection1">
      <bean id="Evictor"
        className="com.ibm.websphere.objectgrid.plugins.builtins.LRUEvictor" />
    </backingMapPluginCollection>
  </backingMapPluginCollections>
</objectGridConfig>
```

### Solução:

Para corrigir o problema, certifique-se de que o valor de cada pluginCollectionRef corresponda ao ID de um dos elementos backingMapPluginCollection. Simplesmente altere o nome de pluginCollectionRef para collection1 para não receber este erro. Outras maneiras de corrigir o problema incluem a alteração do ID do backingMapPluginCollection existente para corresponder ao pluginCollectionRef ou a inclusão de um backingMapPluginCollection adicional com um ID que corresponda ao pluginCollectionRef.

- **Problema:** O IBM Software Development Kit (SDK) Versão 5 contém uma implementação de alguma função de Java API for XML Processing (JAXP) a ser usada para validação de XML com relação a um esquema. Ao utilizar um SDK que não contém esta implementação, as tentativas de validação poderão falhar.

Ao tentar validar o XML com um SDK que não possui a implementação necessária, o log contém o seguinte erro:

```
A validação XML de XmlConfigBuild está ativada
SystemErr R com.ibm.websphere.objectgrid
SystemErr R at
com.ibm.ws.objectgrid.ObjectGridManagerImpl.getObjectGridConfigurations
(ObjectGridManagerImpl.java:182)
SystemErr R at com.ibm.ws.objectgrid.ObjectGridManagerImpl.createObjectGrid(ObjectGridManagerImpl.java:309)
SystemErr R at com.ibm.ws.objectgrid.test.config.DocTest.main(DocTest.java:128)
SystemErr R Caused by: java.lang.IllegalArgumentException: No attributes are implemented
SystemErr R at org.apache.crimson.jaxp.DocumentBuilderFactoryImpl.setAttribute(DocumentBuilderFactoryImpl.java:93)
SystemErr R at com.ibm.ws.objectgrid.config.XmlConfigBuilder.<init>(XmlConfigBuilder.java:133)
SystemErr R at com.ibm.websphere.objectgrid.ProcessConfigXML$2.runProcessConfigXML.java:99)...
```

O SDK utilizado não contém uma implementação da função JAXP necessária para validar arquivos XML em um esquema.

**Solução:** Se você desejar validar o XML usando um SDK que não contém a implementação JAXP, faça download do Apache Xerces e inclua seus arquivos Java archive (JAR) no caminho de classe. Para evitar este problema, depois de efetuar download do Xerces e incluir os arquivos JAR no caminho de classe, é possível validar o arquivo XML com êxito.

---

## Resolução de Problemas de Segurança

Use estas informações para resolver problemas com sua configuração de segurança.

## Procedimento

- **Problema:** A extremidade da conexão do cliente requer Secure Sockets Layer (SSL), com a configuração `transportType` definida para `SSL-Required`. No entanto, a extremidade da conexão do servidor não suporta SSL e possui a configuração `transportType` definida para `TCP/IP`. Como resultado, a exceção a seguir é encadeada para outra exceção nos arquivos de log:

```
java.net.ConnectException: connect: Address is invalid on local machine, or
port is not valid on remote machine
    at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:389)
    at java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:250)
    at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:237)
    at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:385)
    at java.net.Socket.connect(Socket.java:540)
    at
com.ibm.rmi.transport.TCPTransportConnection.createSocket(TCPTransportConnection.java:155)
    at
com.ibm.rmi.transport.TCPTransportConnection.createSocket(TCPTransportConnection.java:167)
```

O endereço nessa seção pode ser um servidor de catálogo, um servidor de contêiner ou um cliente.

**Solução:** Consulte “Configurando Tipos de Transportes Seguros” na página 506 para obter uma tabela com as configurações de segurança válidas entre clientes e servidores.

- Quando o agente é usado, o cliente envia a chamada do agente para o servidor e o servidor envia a resposta de volta para o cliente para reconhecer a chamada do agente. Quando o agente conclui o processamento, o servidor inicia uma conexão para enviar os resultados do agente. Isso torna o servidor de contêiner um cliente a partir do ponto de vista de conexão. Portanto, se TLS/SSL for configurado, certifique-se de que o certificado público do cliente seja importado no armazenamento confiável do servidor.

---

## IBM Support Assistant for WebSphere eXtreme Scale

É possível usar o IBM Support Assistant para coletar dados, analisar sintomas e acessar informações do produto.

### IBM Support Assistant Lite

O IBM Support Assistant Lite for WebSphere eXtreme Scale fornece uma coleta de dados automática e suporte à análise de sintomas para cenários de determinação de problemas.

O IBM Support Assistant Lite reduz o período de tempo que leva para reproduzir um problema com os níveis de rastreamento Reliability, Availability and Serviceability adequados configurados (os níveis de rastreamento são automaticamente configurados pela ferramenta) para simplificar a determinação de problemas. Se você precisar de assistência adicional, o IBM Support Assistant Lite também reduz o esforço necessário para enviar as informações de log apropriadas para o IBM Support.

O IBM Support Assistant Lite é incluído em cada instalação do WebSphere eXtreme Scale Versão 7.1.0

### IBM Support Assistant

O IBM® Support Assistant (ISA) fornece acesso rápido a recursos do produto, de educação e de suporte que podem ajudar você a responder questões e resolver sozinho problemas com os produtos de software IBM, sem precisar entrar em



contato com o IBM Support. Plug-ins diferentes específicos do produto permitem customizar o IBM Support Assistant para os produtos particulares que você instalou. O IBM Support Assistant também podem coletar dados do sistema, arquivos de log e outras informações para ajudar o IBM Support a determinar a causa de um problema particular.

O IBM Support Assistant é um utilitário a ser instalado em sua estação de trabalho, não diretamente no sistema do servidor WebSphere eXtreme Scale em si. Os requisitos de memória e de recurso para o Assistant podem afetar negativamente o desempenho do sistema do servidor WebSphere eXtreme Scale. Os componentes de diagnóstico móveis incluídos são projetados para um impacto mínimo para a operação normal de um servidor.

É possível usar o IBM Support Assistant para ajudá-lo das seguintes maneiras:

- Para pesquisar em fontes de conhecimento e de informações IBM e não IBM em vários produtos IBM para responder uma questão ou resolver um problema
- Para localizar informações adicionais por meio de recursos da web específicos do produto; incluindo páginas iniciais do produto e de suporte, grupos de notícias e fóruns de clientes, qualificações e recursos de treinamento e informações sobre como resolver problemas e as perguntas mais comuns
- Para aumentar sua capacidade para diagnosticar problemas específicos do produto com as ferramentas de diagnóstico de destino disponíveis no Support Assistant
- Para simplificar a coleta de dados de diagnóstico para ajudar você e a IBM a resolver seus problemas (coletar dados gerais ou dados específicos do produto/sintoma)
- Para ajudar no relatório de incidentes de problemas ao IBM Support por meio de uma interface online customizada, incluindo a capacidade de conectar os dados de diagnóstico mencionados acima ou qualquer outra informação a incidentes novos ou existentes

Finalmente, é possível usar o recurso Updater integrado para obter suporte para produtos de software e recursos adicionais assim que eles forem disponibilizados. Para configurar o IBM Support Assistant para ser usado com o WebSphere eXtreme Scale, primeiro instale o IBM Support Assistant usando os arquivos fornecidos na imagem transferida por download da página da web Visão Geral do IBM Support em: [http://www-947.ibm.com/support/entry/portal/Overview/Software/Other\\_Software/IBM\\_Support\\_Assistant](http://www-947.ibm.com/support/entry/portal/Overview/Software/Other_Software/IBM_Support_Assistant). A seguir, use o IBM Support Assistant para localizar e instalar qualquer atualização do produto. Também é possível optar por instalar plug-ins disponíveis para outro software IBM em seu ambiente. Mais informações e a versão mais recente do IBM Support Assistant estão disponíveis a partir da página da web do IBM Support Assistant em: <http://www.ibm.com/software/support/isa/>.



---

## Avisos

Referências nesta publicação a produtos, programas ou serviços IBM não significam que a IBM pretende torná-los disponíveis em todos os países onde opera. Qualquer referência a produtos, programas ou serviços IBM não significa que apenas produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM poderá ser utilizado em substituição a este produto, programa ou serviço. A avaliação e verificação da operação em conjunto com outros produtos, exceto aqueles expressamente designados pela IBM, são de inteira responsabilidade do Cliente.

A IBM pode ter patentes ou solicitações de patentes pendentes relativas a assuntos tratados nesta publicação. O fornecimento desta publicação não garante ao Cliente nenhum direito sobre tais patentes. Pedidos de licença devem ser enviados, por escrito, para:

Gerência de Relações Comerciais e Industriais da IBM Brasil  
Av. Pasteur, 138-146  
Botafogo  
Rio de Janeiro, RJ  
CEP 22290-240

Licenciados deste programa que desejam obter mais informações sobre este assunto com objetivo de permitir: (i) a troca de informações entre programas criados independentemente e outros programas (incluindo este) e (ii) a utilização mútua das informações trocadas, devem entrar em contato com:

Gerência de Relações Comerciais e Industriais da IBM Brasil  
Av. Pasteur, 138-146,  
Botafogo  
Rio de Janeiro, RJ  
CEP 22290-240  
Brasil

Tais informações podem estar disponíveis, sujeitas a termos e condições apropriados, incluindo em alguns casos, o pagamento de uma taxa.



---

## Marcas Registradas

Os termos a seguir são marcas registradas da IBM Corporation nos Estados Unidos e/ou em outros países:

- AIX
- CICS
- Cloudscape
- DB2
- Domino
- IBM
- Lotus
- RACF
- Redbooks
- Tivoli
- WebSphere
- z/OS

Java e todas as marcas registradas baseadas em Java são marcas registradas da Sun Microsystems, Inc. nos Estados Unidos e/ou em outros países.

LINUX é uma marca registrada de Linus Torvalds nos Estados Unidos e/ou em outros países.

Microsoft, Windows, Windows NT e o logotipo do Windows são marcas registradas da Microsoft Corporation nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada do The Open Group nos Estados Unidos e em outros países.

Outros nomes de empresas, produtos e serviços podem ser marcas registradas ou marcas de serviços de terceiros.



# Índice Remissivo

## A

- administração
  - resolução de problemas 537
  - visão geral 387
  - WebSphere Application Server 252
- ajuste
  - coleta de lixo
    - tempo real 491
  - configurações de rede 481
  - Java virtual machines 486
  - portas de rede 62
  - sistemas operacionais 481
- ajuste de desempenho 481
- análise de log
  - customizado 532
  - em execução 531
  - resolução de problemas 534
- AP 35
- API de AvailabilityState 421
- APIs
  - administração 402
  - AvailabilityState 421
  - estatísticas 450
  - MBean 450
  - servidor integrado 405
  - StateManager 421
- arquitetura
  - topologias 10
- arquivo CSV
  - definições de estatísticas 447
- arquivo de definição de construção
  - CIP 167
  - IIP 170
- arquivo de resposta 195
- arquivos csv 446
- arquivos de configuração
  - arquivo orb.properties 482
  - arquivo wxssetup.response.txt 175
  - exemplo de zonas de política de implementação 244
  - Hibernate 345
- arquivos de tempo de execução
  - independente 193
  - WebSphere Application Server 163
- atualizador de dados baseado em tempo 349
- autenticação
  - integrando a segurança em ambientes mistos 107
- autorização de cliente
  - acesso apenas do criador 499
  - customizado 499
  - JAAS 499
- autorização de grade 502

## B

- banco de dados
  - cache disperso e completo 18
  - cache read-through 19

- banco de dados (*continuação*)
  - cache secundário 19
  - cache write-behind 22
  - cache write-through 19
  - pré-carregamento de dados 28
  - preparação de dados 28
  - sincronização 30
  - técnicas de sincronização de banco de dados 30
- beans gerenciados 467
- benefícios
  - armazenamento em cache write-behind 22
- bloqueio
  - configuração com XML 222
  - configurando programaticamente 222
  - não 222
  - otimista 222
  - pessimista 222
- boas práticas
  - tempo real
    - ambiente independente 491

## C

- cache
  - distribuído 15
    - integrado 13
    - local 11
  - cache coerente 17
  - cache completo 18
  - Cache dinâmico
    - ajuste 495
  - cache disperso 18
  - cache distribuído 15
  - cache integrado 13
  - cache local
    - replicação por peer 12
  - cache secundário
    - integração com o banco de dados 19
  - cache sequencial 19
- cálculos
  - contagem de partições 53
  - dimensionamento de memória 53
- clientes
  - Configuração XML 289
  - invalidação 291
  - visão geral 289
- colocação 419
- Comando manageprofiles 182
- comando routetable 424
- comando teardown 400, 424
- Comando wasprofile 182
- comandos
  - manageprofiles 184
  - routetable 424
  - startOgServer 387
  - stopOgServer 387
  - teardown 400
- configuração 60
- métodos 219

- configuração (*continuação*)
  - topologias do datacenter 276
  - visão geral 219
- configuração inicial 205
- configurações de XML
  - resolução de problemas 539
- console da web
  - conexão do servidor de catálogos 437
  - descrições das estatísticas 440
  - estatísticas 439
  - iniciando 435
  - relatórios customizados 445
  - visão geral 435
- Console de Primeiras Etapas 183
- contêiner OSGi
  - configuração do Apache Aries Blueprint 381
- controle de acesso de segurança JMX
  - autenticação 507
  - suporte JAAS 507
  - transporte seguro 507
- convencões de diretório 51, 160

## D

- datacenters
  - configuração 276
  - configuração de topologia 276
  - gerenciamento de falhas para 424
- DB2 479
- desinstalação 206
- dimensionando CPU
  - para transações 55
  - para transações paralelas 56
- disponibilidade
  - gerenciando estados 421
- distribuindo alterações
  - JVMs peer 225
- diversas configurações do datacenter 538
- domínio do serviço de catálogo 248
- domínios de serviço de catálogo
  - tarefas administrativas 254
  - WebSphere Application Server 253

## E

- E/S do eXtreme 273
- Eclipse Equinox
  - configuração do ambiente 200
- elemento de log 225
- estatísticas
  - API de estatísticas 450
  - visão geral 433
- evictors
  - configuração com XML 221
- eXtremeIO
  - configuração 273

eXtremeMemory  
configuração 273

## F

failover  
configuração 249, 489  
ferramenta wsadmin  
domínio do serviço de catálogo 254  
MBeans 426, 466

## G

gerenciador de sessões  
persistência com a grade de dados 300  
WebSphere Application Server 296, 308  
gerenciador de sessões HTTP  
com WebSphere Virtual Enterprise 308  
configuração 296  
configuração com XML 311  
parâmetros para configuração 316  
WebSphere Application Server 296  
grade de dados  
configuração 220

## H

Hibernate  
configuração 341  
configuração com XML 345  
Hyperic HQ 477

## I

IBM Installation Factory  
arquivo de definição de construção 166  
IBM Support Assistant 543  
IBM Tivoli Monitoring 468  
IBM Update Installer para WebSphere  
desinstalando CIP 169  
IBM Update Installer para WebSphere  
Software 213  
implementação distribuída  
configuração 232  
implementações locais 220  
independente  
ORB (Agente de Pedido de Objetos) 286  
REST 360  
índices  
desempenho 33  
qualidade dos dados 33  
iniciando  
programaticamente 402  
serviço de catálogo 393  
servidores 387  
servidores de catálogos 393  
servidores de contêiner 393  
servidores para serviço de dados  
REST 360

instalação  
arquivo de resposta silenciosa 175  
assistente 161  
desinstalação 206  
IBM Installation Factory para CIP 166  
IBM Installation Factory para IIP 166  
independente 191  
pacote de instalação  
customizada 173  
planejando 48, 152  
resolução de problemas 206, 534  
Serviço de dados REST 198  
silent 173, 195, 197  
tipos 151  
topologias 151  
verificação 204  
visão geral 151  
WebSphere Application Server 161  
WebSphere Application Server  
Network Deployment 161  
instalação silenciosa 175  
instalando  
manutenção 213  
Installation Factory  
CIP  
manutenção 169  
integração com o banco de dados  
configuração 347  
integração com outros servidores 47  
integração em cache  
configuração 296  
resolução de problemas 535  
interoperabilidade do gerenciador de sessão  
com os produtos WebSphere 47  
introdução  
visão geral 1  
Introscope 474  
invalidação 228

## J

Java EE  
considerações 50, 159  
Java Message Service (JMS)  
listener de evento 228  
replicação ponto a ponto 224  
Java Persistence API (JPA)  
atualizador de dados baseado em tempo  
configuração 349  
configuração  
integrado 332  
remoto 332  
visão geral 347  
plug-in de cache  
configuração 332  
introdução 325  
topologia de cache  
integrado 325  
particionado integrado 325, 332  
remoto 325  
Java SE  
considerações 49, 158  
Java virtual machine 486

JDK  
considerações 49, 158  
JMS  
replicação ponto a ponto 224  
JVM 486

## L

lista de verificação operacional 60  
Listener  
Java Message Service (JMS) 228  
listener de evento 228  
logs 525

## M

MBean  
wsadmin 426, 466  
MBeans  
acessando com a segurança  
ativada 507  
administrando com 426  
programaticamente 426  
visão geral 467  
memória do eXtreme 273  
migração 212  
monitoramento  
agente 468  
API de estatísticas 450  
arquivos csv 446  
CA Wily Introscope 474  
com o Tivoli Enterprise Monitoring  
Agent 468  
DB2 479  
Hyperic HQ 477  
módulos estatísticos 452  
Performance Monitoring Infrastructure  
(PMI) 455  
visão geral 433  
visão geral da ferramenta do  
fornecedor 468

## O

off-line 421  
on-line 421  
OpenJPA  
arquivos XML do ObjectGrid  
exemplo 339  
plug-in de cache  
configuração 335  
ORB  
configuração 284  
customizado 286  
WebSphere Application Server 285  
ORB (Agente de Pedido de Objetos)  
arquivo orb.properties 482  
configuração 284  
configuração customizada 286  
eXtreme Scale independente 285  
propriedades 482  
WebSphere Application Server 285  
OSGi  
ambiente do Eclipse Equinox 200  
tutoriais  
arquivos de configuração 136

OSGi (*continuação*)  
  tutoriais (*continuação*)  
    atualizando classificações de  
      serviço 148  
    configurando contêineres 141  
    configurando o Eclipse para  
      executar clientes 144  
    configurando servidores 140  
    consultando classificações de  
      serviço 146  
    consultando pacotes  
      configuráveis 145  
    executando clientes 143  
    executando os pacotes  
      configuráveis 132  
    fazendo upgrade de pacotes  
      configuráveis 145  
    iniciando clientes 144  
    iniciando os pacotes  
      configuráveis 139, 143  
    instalando os buffers de  
      protocolo 142  
    instalando pacotes  
      configuráveis 138  
    localizando classificações de  
      serviço 148  
    pacotes configuráveis de  
      amostra 134  
    preparando para instalar pacotes  
      configuráveis 134  
    visão geral 133

## P

parâmetros SSL 507  
Parando  
  programaticamente 402  
parando servidores 398  
partição de disponibilidade (AP) 35  
perfil de segurança 522  
perfis  
  aumento 182  
  aumento com a UI 183  
  aumento com o comando 184  
  criação 182  
  criação com a UI 183  
  criação com o comando 184  
  usuário não-root 190  
Performance Monitoring Infrastructure  
  ativação 456  
  módulos 459  
  recuperação de estatísticas 458  
placas de rede  
  configuração 284  
planejamento de capacidade 53  
planejando 9, 481  
  configurações de rede 481  
  implementação do aplicativo 9  
  lista de verificação operacional 60  
  sistemas operacionais 481  
planejar  
  instalação 48, 152  
Plug-in do Cache JPA  
  resolução de problemas 536  
plug-in do Installation Factory  
  arquivo de definição de construção  
  modificar 172

plug-in do Installation Factory  
  (*continuação*)  
  instalando  
    CIP 167  
    IIP 171  
plug-in Profile Management Tool  
  aumento do perfil 183  
  criação de perfil 183  
  visão geral 182  
plug-ins OSGi  
  administrando com 416  
  configuração 380  
PMI  
  monitoramento 455  
PMI (Performance Monitoring  
  Infrastructure)  
  monitoramento 455  
políticas de implementação  
  configuração 232  
por partição 55  
Portal WebSphere  
  configuração 306  
portas  
  configuração 280  
  configuração independente 280  
  WebSphere Application Server 283  
portas de rede  
  planejando 62  
pré-recarregamento 421  
propriedade enableXml 273  
propriedade maxXmlSize 273  
propriedade  
  xIOContainerTCPNonSecurePort 273  
propriedades  
  ORB (Agente de Pedido de  
    Objetos) 482  
propriedades customizadas  
  Propriedades do ORB 482  
propriedades do servidor  
  enableXml 273  
  maxXmlSize 273  
  xIOContainerTCPNonSecurePort 273  
provedor de cache dinâmico  
  configurando 321  
  planejamento de capacidade 57

## Q

quiesce 421  
quorums  
  substituir 424

## R

rede 481  
replicação  
  configuração com JMS 224  
  Listener de Eventos da JMS 228  
replicação da grade de dados multimestre  
  planejando 35  
replicação de multimestre  
  planejamento de design 42  
  planejando 35  
  planejando para carregadores 40  
  plano de configuração 39  
  topologias 35

replicação ponto a ponto 224  
requisitos  
  hardware 48, 156  
  software 48, 156  
resolução de problemas 525  
  administração 537  
  configurações de XML 539  
  instalação 206, 534  
  integração em cache 535  
  sessão de HTTP 535

## S

segurança  
  autenticação 64, 497  
  Autorização 64  
  Conexão Única (SSO) 497  
  configuração 518  
  integração 510  
  integração com o WebSphere  
    Application Server 515  
  introdução 510  
  local 519  
  plug-ins 519  
  resolução de problemas 543  
  segurança do cliente 518  
  tipos de transporte 506  
  transporte seguro 64  
  visão geral 497  
segurança da grade de dados  
  gerenciador de tokens 503  
  JSE 503  
segurança do cliente-servidor  
  secure sockets layer (SSL) 505  
  TCP/IP 505  
  transport layer security (TLS) 505  
segurança local  
  ativando 519  
senhas  
  console da web 435  
sequência de log 225  
serviço de catálogo  
  alta disponibilidade 248  
  boas práticas para 248  
  cluster 248  
  domínios de serviço de catálogo 401  
  iniciando em um ambiente que não  
    está executando o WebSphere  
    Application Server 387  
  iniciando no WebSphere Application  
    Server 401  
  WebSphere Application Server 252  
serviço de dados Rest  
  protegendo 511  
Serviço de dados REST  
  Apache Tomcat  
    implementação 370  
    iniciando 372  
  ativação  
    visão geral 352  
  Cliente do Visual Studio 2008 WCF  
    configuração 378  
  cliente Java  
    configuração 376  
  configuração  
    visão geral 350



- Serviço de dados REST (*continuação*)
  - feeds ATOM
    - configuração 374
  - grade de dados
    - iniciando 358
  - grade de dados independente
    - iniciando 357
  - instalação 198
  - modelo de dados
    - visão geral 352
  - recuperar e atualizar dados
    - visão geral 354
  - servidores de aplicativos
    - configuração 360
  - WebSphere Application Server
    - implementação 360
    - iniciando 362
  - WebSphere Application Server Community Edition
    - implementação 364
    - iniciando 368
  - servidor de contêineres
    - colocação 419
  - servidores de catálogos
    - configuração 247
  - servidores de contêiner
    - configuração
      - visão geral 247
    - iniciando 390
  - WebSphere Application Server
    - configuração 270
    - iniciar automaticamente 271
  - servidores independentes
    - iniciando 387
  - servidores seguros
    - iniciando 520
    - Parando 520, 522
    - Serviço de dados REST 511
    - WebSphere Application Server 521
  - sessões HTTP
    - Arquivo splicer.properties 319
  - SIP
    - gerenciamento de sessões 306
    - sessão 306
  - sistemas operacionais
    - ajuste 481
  - SSL (Secure Sockets Layer)
    - servidores de catálogos 437
  - startOgServer 387, 390
    - options 393
  - stopOgServer 387, 399
  - Suporte 543

## T

- tarefas de pós-instalação 205
- tempo de resposta
  - ajustando a coleta de lixo
    - tempo real 491
  - tempo real
    - ambiente independente 491
- tempo limitenova tentativa de solicitação 294
- tempo real
  - ajustando a coleta de lixo 491
  - ambiente independente 491
  - WebSphere Application Server 493

- topologias
  - instalação 152
  - planejar 10
- trace
  - opções para configuração 527
- transações paralelas 56
- transporte 273
- transportes
  - configuração 284
  - eXtremeIO 273
  - ORB 284
- tutoriais 67
  - acessando aplicativos clientes
    - na estrutura do OSGi 144
  - acessando arquivos do tutorial 86, 108
  - amostra não-segura 68
  - arquivos de configuração 136
  - ativando autorização 101, 127
    - para usuários 102, 128
  - atualizando classificações de serviço 148
  - atualizando pacotes
    - configuráveis 145
  - autenticação de cliente 71
  - autenticador de clientes 67
  - Autorização 77
  - autorização de cliente 67
  - comunicação segura de terminais 80
  - configuração do WebSphere Application Server 112
  - configurando a autenticação
    - em ambientes mistos 114
  - configurando a autorização
    - para grupos 104
  - configurando a segurança de transporte 97, 123
  - configurando a segurança do cliente 115
  - configurando a segurança do servidor de catálogos 116
  - configurando a segurança do servidor de contêiner 119
  - configurando contêineres do eXtreme Scale 141
  - configurando o Eclipse
    - para OSGi 144
  - configurando o WebSphere Application Server 89, 91
  - configurando servidores do eXtreme Scale 140
  - configurando transportes
    - entrada 98, 124
    - saída 98, 124
  - consultando classificações de serviço 146
  - consultando pacotes
    - configuráveis 145
  - executando as amostras 95, 99, 121, 126
  - executando clientes de amostra
    - no OSGi 143
  - exemplo não-seguro 67
  - incluindo propriedades SSL 99, 125
  - iniciando os pacotes
    - configuráveis 132

- tutoriais (*continuação*)
  - iniciar pacotes configuráveis
    - OSGi 143
  - instalação de amostra 95
  - instalando as amostras 121
  - instalando os Google Protocol Buffers 142
  - instalando pacotes configuráveis 138
  - instalando pacotes configuráveis do eXtreme Scale 139
  - integrando a segurança
    - em ambientes mistos 106
  - integrando a segurança do produto
    - com o WebSphere Application Server 84
  - localizando classificações de serviço 148
  - monitorando grades de dados e mapas
    - com xscmd 106, 130
  - OSGi
    - arquivos de configuração 136
    - atualizando classificações de serviço 148
    - configurando contêineres 141
    - configurando o Eclipse para executar clientes 144
    - configurando servidores 140
    - consultando classificações de serviço 146
    - consultando pacotes
      - configuráveis 145
    - executando clientes 143
    - fazendo upgrade de pacotes
      - configuráveis 145
    - iniciando clientes 144
    - iniciando os pacotes
      - configuráveis 132, 139, 143
    - instalando os buffers de protocolo 142
    - instalando pacotes
      - configuráveis 138
    - localizando classificações de serviço 148
    - pacotes configuráveis de amostra 134
    - preparando para instalar pacotes
      - configuráveis 134
      - visão geral 133
    - pacotes configuráveis OSGi de amostra 134
    - planejando para ambientes
      - combinados 108
    - preparando para instalar pacotes
      - configuráveis do eXtreme Scale 134
    - segurança do servidor de catálogos
      - configuração 94
    - segurança do servidor do cliente
      - configuração 91
    - usando a autorização JAAS 100, 126
    - visão geral
      - iniciando servidores e contêineres 133
    - visão geral da topologia 86, 108
    - WebSphere Application Server 85

tutorial  
configurando a segurança do servidor  
de catálogos 92

## U

utilitário xscmd  
administração 407  
monitoramento com 453  
utilitários de carga  
banco de dados 26  
JPA 347  
resolução de problemas 538

## V

validação baseada em evento 32  
Visão Geral do eXtreme Scale 9  
visão geral do produto  
integração do produto  
com o WebSphere Application  
Server 84

## W

WebSphere Application Server 213  
configurando com o WebSphere  
eXtreme Scale 252  
WebSphere eXtreme Scale  
configurando com o WebSphere  
Application Server 252  
Wily Introscope 474  
write-behind  
integração com o banco de dados 22  
wsadmin  
MBean 426, 466

## X

xsadmin  
migração para xscmd 214  
xscmd  
migração 214  
perfil de segurança 522  
xsloganalyzer 531, 532

## Z

zonas  
arquivo XML descritor de política de  
implementação 244  
datacenter 234  
dividindo entre 234  
exemplos de zonas 234  
monitor 246  
posicionamento de shard 234  
roteamento 239  
servidores de contêiner 243  
sobre WANs 234







Impresso no Brasil