

Roxie: O Motor de Entrega Rápida de Dados

Equipe de documentação de Boca Raton



Roxie: O Motor de Entrega Rápida de Dados

Equipe de documentação de Boca Raton

Copyright © 2022 HPCC Systems®. All rights reserved

Sua opinião e comentários sobre este documento são muito bem-vindos e podem ser enviados por e-mail para <docfeedback@hpccsystems.com>

Inclua a frase **Feedback sobre documentação** na linha de assunto e indique o nome do documento, o número das páginas e número da versão atual no corpo da mensagem.

LexisNexis e o logotipo Knowledge Burst são marcas comerciais registradas da Reed Elsevier Properties Inc., usadas sob licença.

SSystems® HPCC Systems® é uma marca registrada da LexisNexis Risk Data Management Inc.

Os demais produtos, logotipos e serviços podem ser marcas comerciais ou registradas de suas respectivas empresas.

Todos os nomes e dados de exemplo usados neste manual são fictícios. Qualquer semelhança com pessoas reais, vivas ou mortas, é mera coincidência.

2022 Version 8.8.34-1

| | |
|---|----|
| Introdução | 4 |
| Visão geral do Roxie | 5 |
| Índices de Payload | 6 |
| Superarquivos Roxie | 7 |
| Como o Roxie Funciona | 8 |
| Backup de Dados do Roxie | 16 |
| Afinidade do processador | 18 |
| Desenvolvendo Consultas Roxie | 19 |
| Caminho de Desenvolvimento | 19 |
| Métodos para Aplicar Jobs a um Cluster Roxie | 20 |
| Gerenciando Consultas | 23 |
| Adicionando uma consulta Roxie ao Grupo de Consultas | 24 |
| Visualizando Grupo de Consultas utilizando o ECL Watch | 25 |
| Utilizando o WsECL para executar uma consulta Roxie | 26 |
| Packages e Package Maps | 27 |
| Exemplo | 28 |
| Atualizando Dados | 29 |
| Sintaxe de Arquivos Package | 30 |
| Multi-Part package maps | 32 |
| Trabalhando com Packages através de linhas de comando ECL | 35 |
| Acesso Direto ao Roxie | 48 |
| Visão Geral do Acesso a uma Consulta Roxie | 48 |
| Acessando suas Consultas Roxie | 49 |
| Disponibilizando Dados para um Cluster Roxie utilizando DFU | 56 |
| Cópia DFU | 57 |
| Cópia Remota | 58 |
| Planejamento de Capacidade para Clusters Roxie | 60 |
| Planejamento de Capacidade | 60 |
| Preflight | 62 |
| Preflight do cluster Roxie | 63 |
| Página de Consultas no ECL Watch | 65 |

Introdução

O mecanismo Roxie – também conhecido como motor de entrega rápida de dados ou RDDE – usa uma combinação de tecnologias e técnicas que produzem um resultado extremamente rápido para consultas a dados indexados e hospedados no HPCC.

As consultas no Roxie podem ultrapassar milhares por segundo, em comparação com aquelas no Thor que tendem a levar de alguns segundos até alguns minutos cada (de extremidade a extremidade) dependendo da complexidade da consulta.

Para entender plenamente esse conceito, é melhor analisar a finalidade para a qual cada um desses processos foi projetado:

- A plataforma Thor foi projetada para realizar operações em cada registro (ou na maioria) de um enorme dataset.
- As consultas podem ser executadas na plataforma hThor se puderem indicar rapidamente pequenos conjuntos de registros nos dados.
- As consultas Roxie normalmente são usadas para identificar rapidamente pequenos conjuntos de registros repetidamente.

Se você imaginar todos os seus dados como um oceano, o Thor seria usado para realizar operações em todas no oceano como um todo.

Uma consulta hThor pode ser usada para encontrar um único peixe nesse mar de dados.

A consulta seria implementada em um cluster Roxie a ser usado para localizar centenas ou até milhares de peixes individuais, um após o outro.

As consultas Roxie são implementadas em um cluster Roxie que pré-carrega todas as consultas na memória e as prepara para que estejam prontas para execução logo que uma consulta for recebida.

As consultas são enviadas ao Roxie através do XML, SOAP ou JSON e os resultados são retornados no mesmo formato. Um Client pode se comunicar diretamente com o cluster Roxie ao abrir um soquete para um dos servidores no cluster, ou ele pode se comunicar por um ESP Service como o WsECL.

Normalmente, os resultados do Roxie são retornados para o solicitante em vez de serem gravados em um arquivo. No entanto, o Roxie pode gravar arquivos de dados, embora normalmente não seja habitual gravar um arquivo quando uma consulta não é baseada em workunits.

Visão geral do Roxie

Há normalmente quatro aspectos sobre o uso do Roxie:

- Criação de índices em datasets
- Uso de índices em consultas
- Compilação e implementação de consultas no cluster Roxie
- Fornecimento do acesso a essas consultas para interfaces voltadas para clientes através do SOAP ou HTTP.

Quando fazer uso de Índices

A plataforma Thor foi projetada para realizar operações com rapidez em grandes datasets não indexados, onde todo o dataset (ou sua maioria) deve ser utilizado. No entanto, se apenas alguns registros forem necessários, um índice pode acessá-los de forma mais eficiente. Na linguagem do ECL, um índice se comporta exatamente como um dataset que parece ser capaz de implementar certas funções (normalmente, funções de filtrar e contar) muito mais rapidamente do que um arquivo simples ou dataset CSV .

Índices de Payload

Em sistemas convencionais de banco de dados, um índice é usado juntamente com um arquivo de dados para localizar registros nesse respectivo arquivo. É possível fazer o mesmo no ECL ao armazenar posições de arquivo no índice e usá-las em uma função `FETCH` para localizar as linhas de dados correspondentes no arquivo original.

No entanto, uma vez que se pode armazenar quaisquer campos desejados em um índice, é mais comum em consultas Roxie projetar índices que armazenam ambos os campos de busca e as informações que deseja localizar. Isso elimina a leitura extra de disco para o `FETCH`. Uma vez que índices são comprimidos, isso também pode economizar espaço em disco se o arquivo de dados original não precisar ser armazenado no cluster Roxie.

Qualquer campo em um índice que não precise ser pesquisado pode ser especificado como sendo `payload` – tais campos são armazenados apenas nos nós de folha da árvore de índice, algo que pode economizar espaço e desempenho nas consultas. Os campos `payload` podem simplesmente ser campos adicionais do dataset de base, mas também podem conter o resultado de alguma computação preliminar (campos computados).

Superarquivos Roxie

Um superarquivo ou superchave usada no Roxie pode conter mais de um único subarquivo.

No entanto, seu superarquivo não pode conter mais de um subarquivo quando o superarquivo for usado para uma operação de FETCH ou em um full-keyed JOIN.

Caso só exista um subarquivo, é possível gravar uma consulta que utilize superarquivos ou superchaves (ainda que contenham apenas um subarquivo único) e ter a vantagem de atualizar seu Roxie ao simplesmente implementar novos dados sem precisar recompilar as consultas que os usam, só porque o nome do subarquivo mudou. Isso economiza tempo de compilação e, em um ambiente de produção (normalmente o caso do Roxie) onde um arquivo de dados é usado por várias consultas, os benefícios podem ser significativos.

Consulte o *Guia do programador em ECL* para obter mais detalhes.

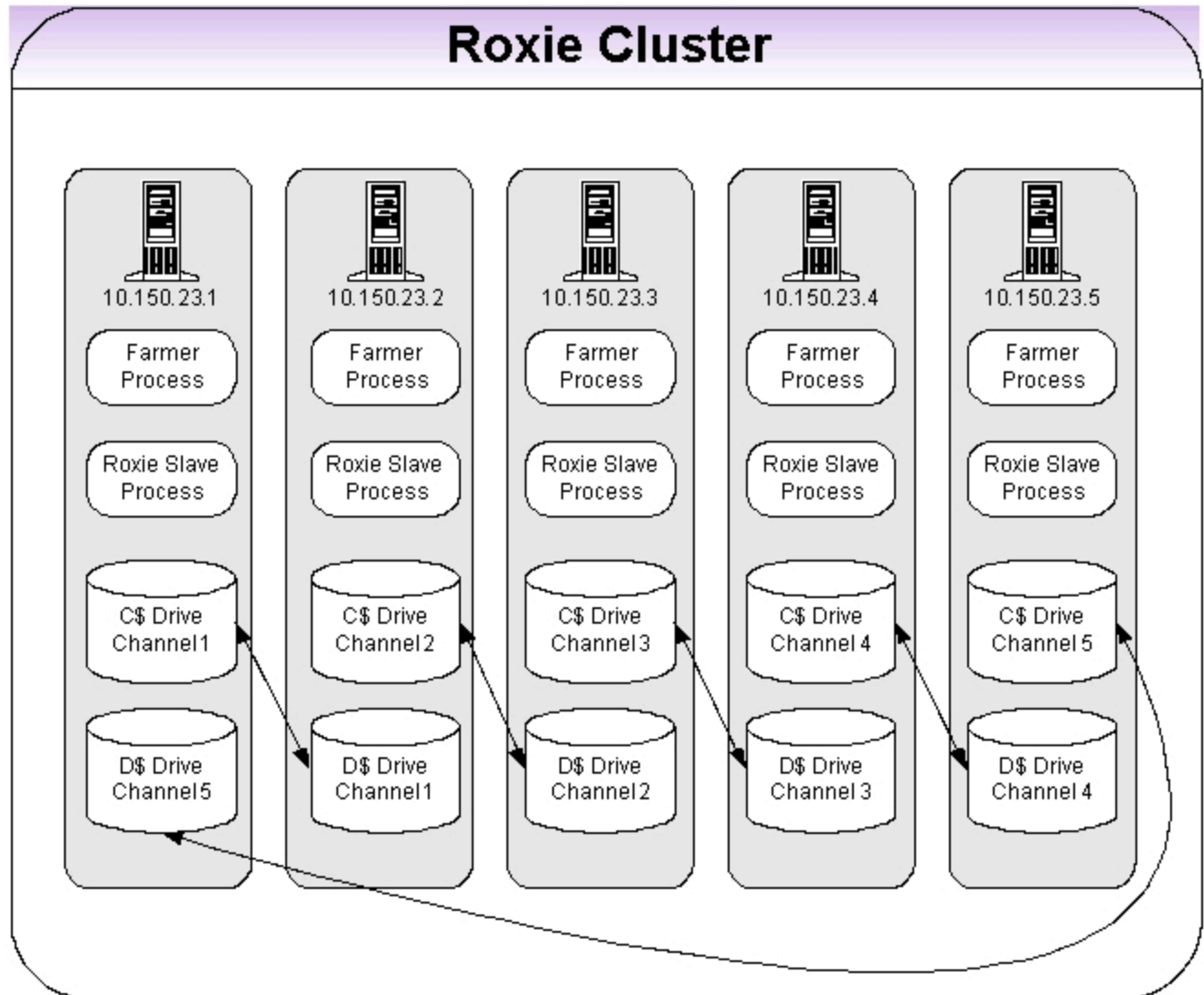
Como o Roxie Funciona

Os clusters Roxie são compostos de várias máquinas conectadas em conjunto para funcionar como uma entidade única. O código fonte do ECL para uma ou mais consultas é compilado e publicado no cluster. Depois de publicado, as consultas podem processar dados rapidamente.

Cada máquina no cluster atua em duas funções distintas – essas funções são executadas no mesmo processo e dividem boa parte do mesmo código – incluindo o código de consulta compilado – mas podem ser idealizadas como logicamente distintas.

O **servidor** aceita receber consultas, avalia o código do ECL de acordo com o comando fornecido na consulta do cliente e retorna o resultado. Quando o servidor avalia uma função ECL que exige dados do disco, ele determina os nós dos agentes que podem conter os dados adequados e envia uma solicitação para esses nós a fim de localizar quaisquer correspondências. Os resultados de múltiplos nós de agentes são condensados e se tornam a entrada para outras funções do ECL avaliadas no nó do servidor. Normalmente, aplicações solicitantes usam alguma forma de balanceamento de carga para distribuir solicitações de modo uniforme para os servidores disponíveis.

O processo **escravo** aceita solicitações apenas de outros nós do servidor no cluster. Essas solicitações correspondem a uma função única de ECL, como uma leitura de índice filtrado ou uma busca de disco. Os resultados são enviados de volta ao servidor que fez a solicitação original. Para equilibrar o desempenho e gerenciar falhas de hardware, os agentes recebem as solicitações por multicast e normalmente haverá no mínimo dois nós de agentes que receberão cada solicitação de um servidor. Os nós dos agentes se comunicam entre si a fim de evitar um esforço duplicado, de modo que o primeiro agente que conseguir processar a solicitação deverá informar aos demais que a tarefa já foi concluída. Cada nó em um cluster normalmente processa solicitações de agente em dois ou mais canais de multicast, geralmente um canal por disco. Se um nó de um agente não estiver respondendo, as solicitações nesse canal são processadas por outros nós agentes pares responsáveis pelo canal.



Esse exemplo mostra um cluster Roxie de 5 nós com cada um configurado para ser tanto um **servidor** quanto **um agente**.

As consultas que foram compiladas com a plataforma alvo especificada como cluster Roxie podem ser publicadas em um Grupo de Consultas usando o ECLWatch.

Cada cluster Roxie carrega consultas de uma ou mais listas do Grupo de Consultas.

Quando uma consulta é adicionada ao Grupo de Consultas que um Roxie está monitorando, o Roxie vai pré-carregar a consulta .so (ou .DLL) e preparar o contexto de execução ao máximo possível para que ela esteja pronta para executar solicitações recebidas para executar a consulta logo que chegar. Isso pode incluir carregar o .so (ou .DLL), solucionar as referências de arquivo e abrir arquivos (se houver identificadores de arquivos suficientes disponíveis), pré-carregar dados na memória, se solicitado, e avaliar o código ECL na consulta que foi marcada como: ONCE.

Dependendo da configuração, o Roxie pode ler dados remotamente de um cluster Thor no qual foi preparado ou, se preferir, pode ser copiado para o Roxie para que seja acessado localmente.

Normalmente, um sistema de desenvolvimento pode consultar dados in situ no cluster Thor, enquanto um sistema de produção pode preferir os benefícios de desempenho de copiar dados localmente para o Roxie.

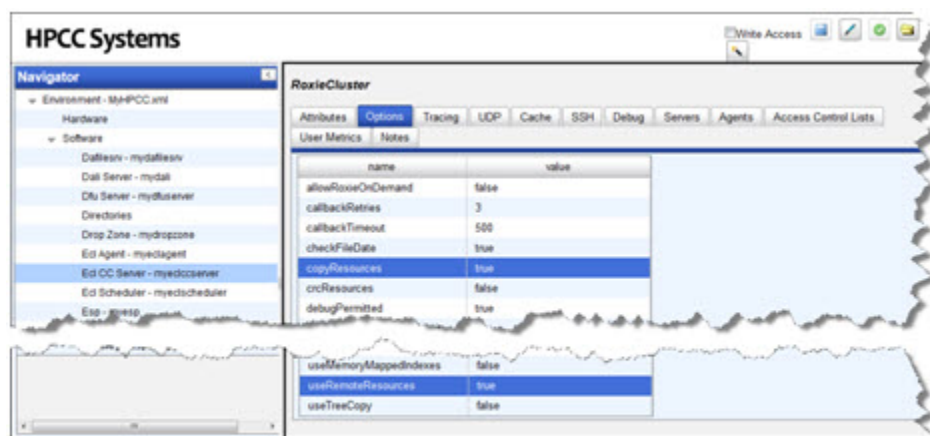
O Roxie pode ler dados remotamente enquanto são copiados e alternar para a cópia local depois que a operação de cópia for concluída. Isso oferece o benefício de disponibilidade rápida, uma vez que a consulta pode estar ativa antes que os dados sejam copiados, enquanto ainda conta com a vantagem dos benefícios de desempenho de dados locais.

Consultas e Dados

Arquivos de dados e arquivos de índices referidos pelo código ECL de uma consulta Roxie estão disponíveis de quatro maneiras, dependendo da configuração do cluster Roxie.

Há duas definições na configuração do Roxie que controlam onde o Roxie procura por arquivos de índice e dados:

| | |
|---------------------------|--|
| copyResources | Copia os arquivos de chave e de dados necessários do local atual quando uma consulta é publicada. |
| useRemoteResources | Instrui o Roxie a procurar por arquivos de chave e de dados no local atual após a consulta ter sido publicada. |



Essas opções podem parecer mutualmente exclusivas, mas o gráfico abaixo mostra o que cada uma das quatro possíveis combinações significa.

| | | | | |
|--------------------|--|--|---|---|
| copyResources | T | T | F | F |
| useRemoteResources | T | F | T | F |
| | Direciona o cluster Roxie para usar a instância remota dos dados até que possa copiar os dados localmente. Isso permite que uma consulta esteja disponível imediatamente enquanto os dados são copiados. | Direciona o cluster Roxie para copiar os dados localmente. A consulta não pode ser executada até que os dados sejam copiados. Isso garante um ótimo desempenho depois que os dados são copiados. | Direciona o cluster Roxie para carregar os dados de um local remoto. A consulta pode ser executada imediatamente, mas o desempenho é limitado pela largura de banda da rede. Isso permite que as consultas sejam executadas sem usar qualquer espaço em | Utilizará dados e índices já carregados (colocado no cluster Roxie usando DFU), mas não copia nem lê dados remotos. |

| | | | | |
|--|--|--|--|--|
| | | | disco Roxie, mas reduz seus recursos de taxa de transferência. | |
|--|--|--|--|--|

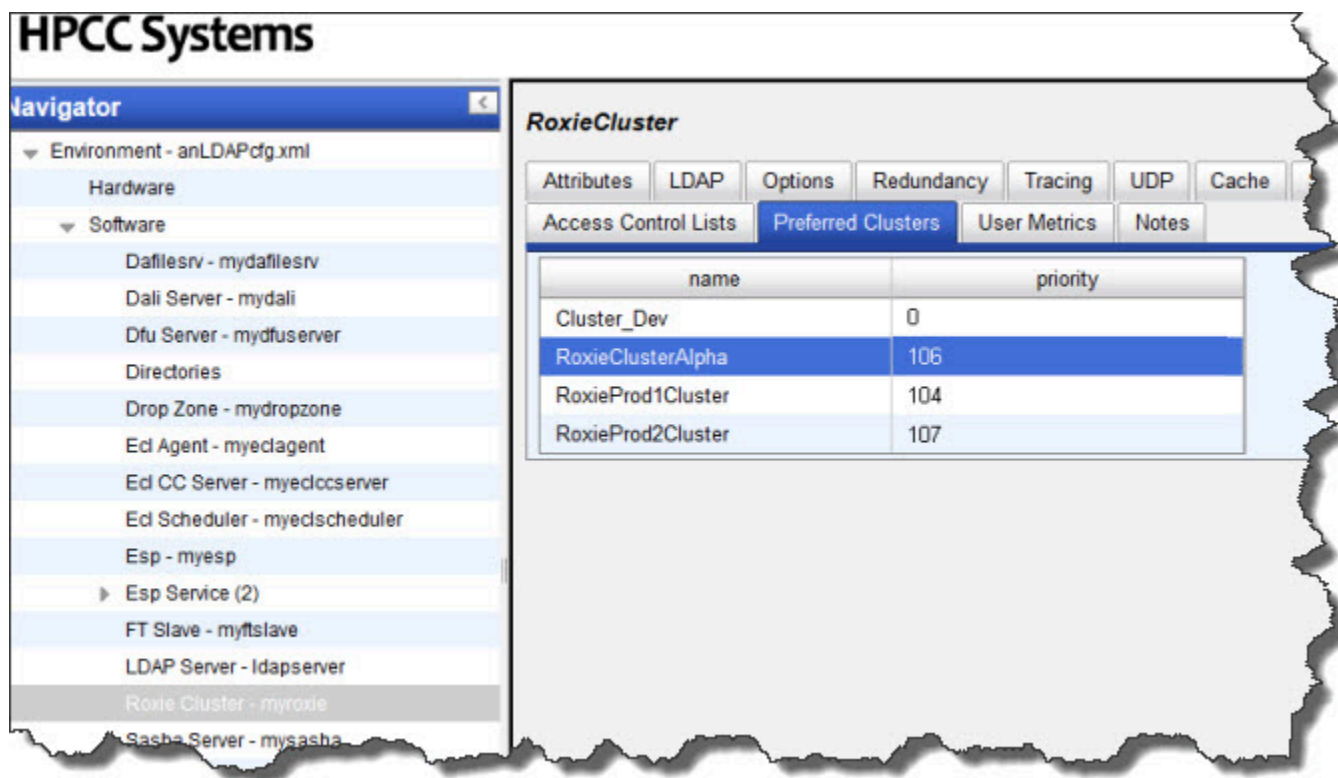
Quando **copyResources** é ativado, os arquivos de dados são copiados dos locais de origem para os nós do agente Roxie. Os arquivos de índice também são copiados, com uma parte de arquivo extra que é a "chave das chaves", ou uma metachave.

Uma cópia da metachave sempre é armazenada em cada servidor Roxie e, na maioria dos casos, é carregada na memória na inicialização para elevar o desempenho.

Clusters preferenciais

Clusters preferenciais permitem que se definam prioridades para clusters dos quais se deseja que o Roxie copie arquivos. Caso não esteja de qualquer outra forma especificado, o Roxie copia primeiramente dos clusters em seu próprio ambiente. Para ativar o recurso de clusters preferidos, é possível usar o Gerenciador de Configurações do HPCC Systems. (para obter informações adicionais sobre como configurar o HPCC System, consulte *Como instalar e executar a plataforma HPCC*)

Figure 1. Clusters preferenciais no Roxie



A aba Clusters preferidos pode ser localizada na página RoxieCluster no Gerenciador de Configurações do HPCC.

Na guia Clusters preferidos, é possível adicionar o nome e a prioridade de seus clusters Roxie. O número de mais alta prioridade é o cluster preferido. Para excluir um cluster Roxie, adicione-o à lista de clusters preferidos com uma prioridade igual a 0. O Roxie então seguirá a prioridade especificada na lista de clusters preferidos dando a mais alta prioridade ao cluster Roxie com o mais alto valor de prioridade ao copiar os dados.

Você deve numerar suas prioridades com um esquema de numeração de no mínimo (o número de clusters)+1. Use um número mais alto para permitir que você adicione clusters mais tarde.

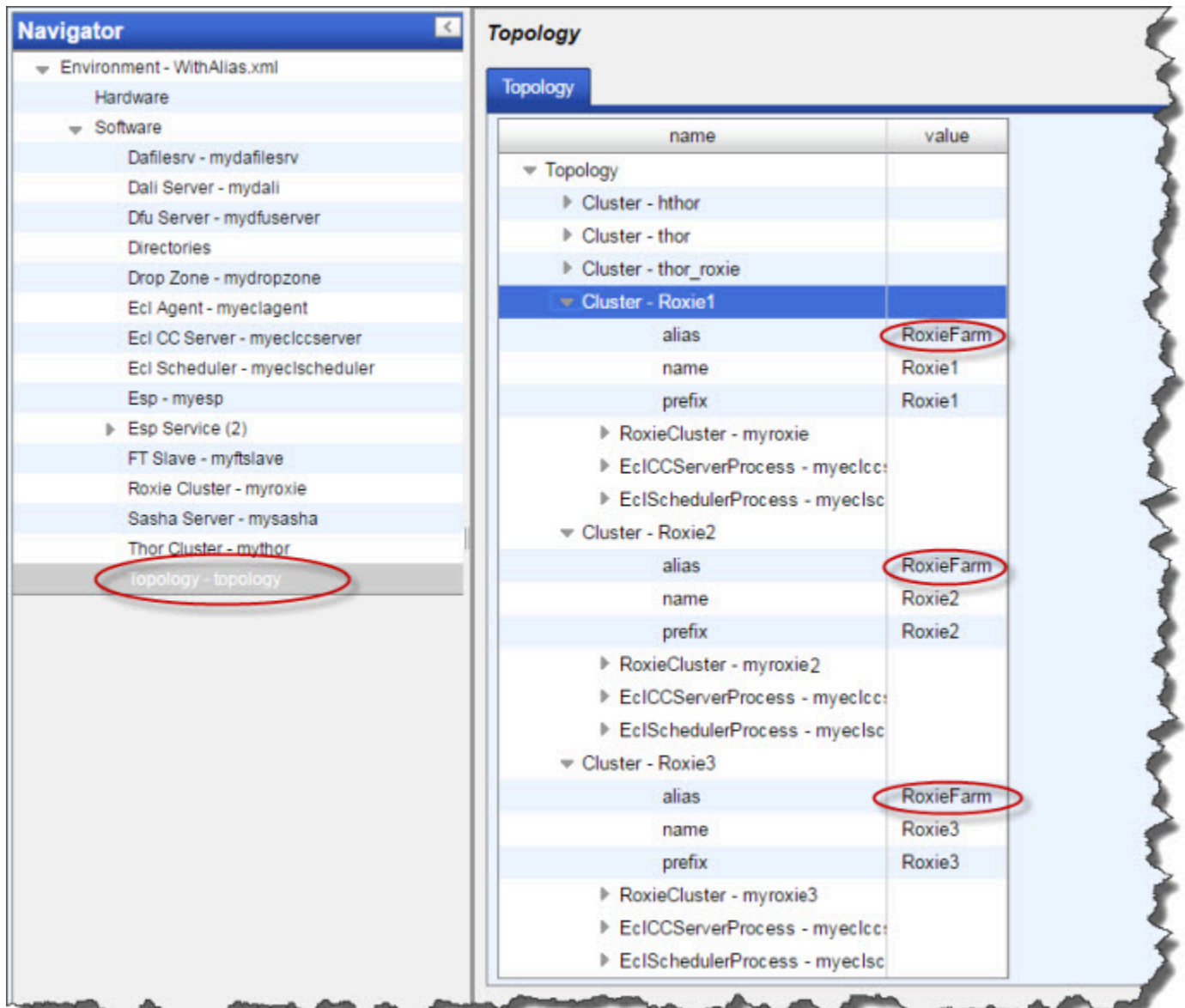
VIPs e Roxie Alias

Os clusters Roxie acessados por um VIP (IP virtual) normalmente usam nomes de destino separados para representar o mesmo conjunto de consultas. Cada Roxie também pode executar mais de um destino ao mesmo tempo.

O recurso Topology Alias (Alias de topologia) oferece uma maneira para especificar que qualquer um dos destinos que representa o mesmo conjunto de consultas pode ser usado. Permitir que os destinos que representam o mesmo conjunto de consultas tenham um alias compartilhado faz com que os acionadores selecionem o conjunto correto de consultas ao serem acionados por um VIP.

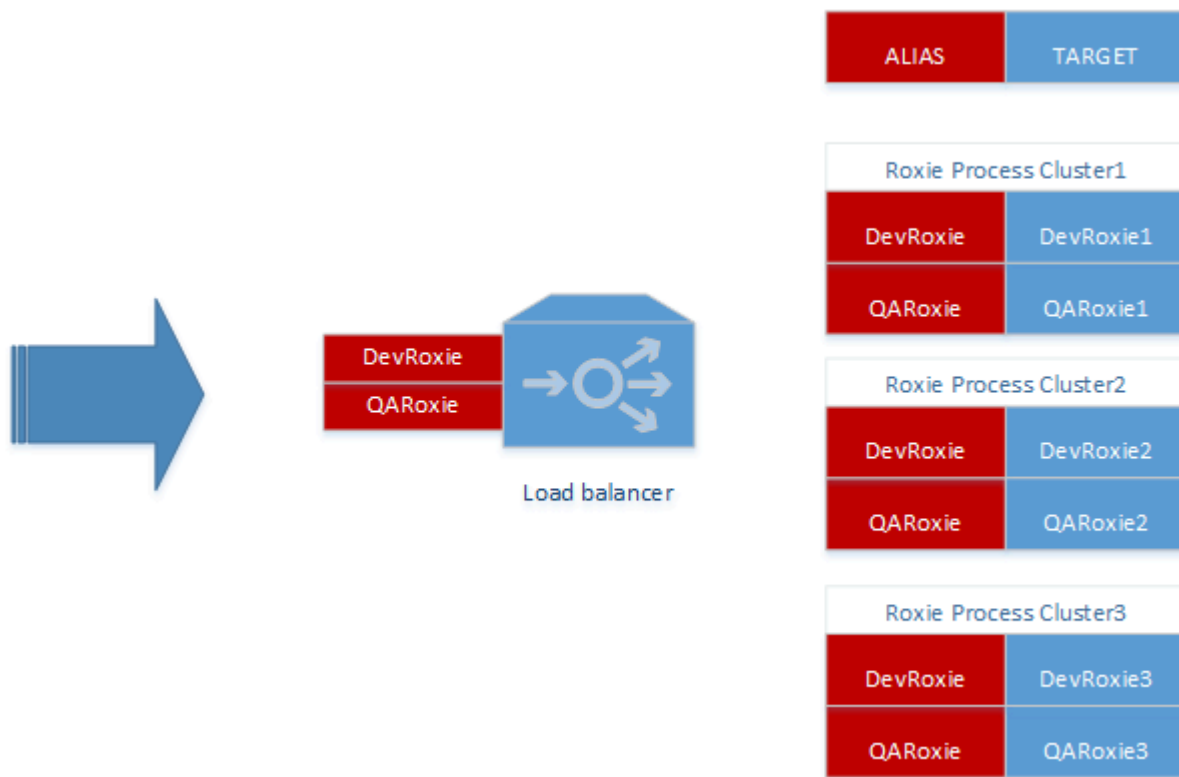
Essa configuração pode ser encontrada na seção Topology (Topologia) no Gerenciador de Configurações do HPCC.

Figure 2. Alias Roxie no Gerenciador de Configurações



O diagrama abaixo mostra o exemplo de um sistema usando aliases.

Figure 3. Roxie Aliases em ambiente de produção



Suporte a Linha de Comando

A ferramenta de linha de comando ECL oferece uma maneira de copiar consultas de um Roxie para outro. Normalmente, isso significa que arquivos de dados e índices serão copiados ou acessados remotamente pelas definições de configuração do Roxie.

Isso permite que você copie rapidamente uma consulta sem copiar os arquivos.

```
ecl queries copy --no-files
```

Isso especifica para NÃO copiar arquivos referenciados pela consulta que está sendo copiada. A consulta não pode ser executada até que os dados sejam disponibilizados.

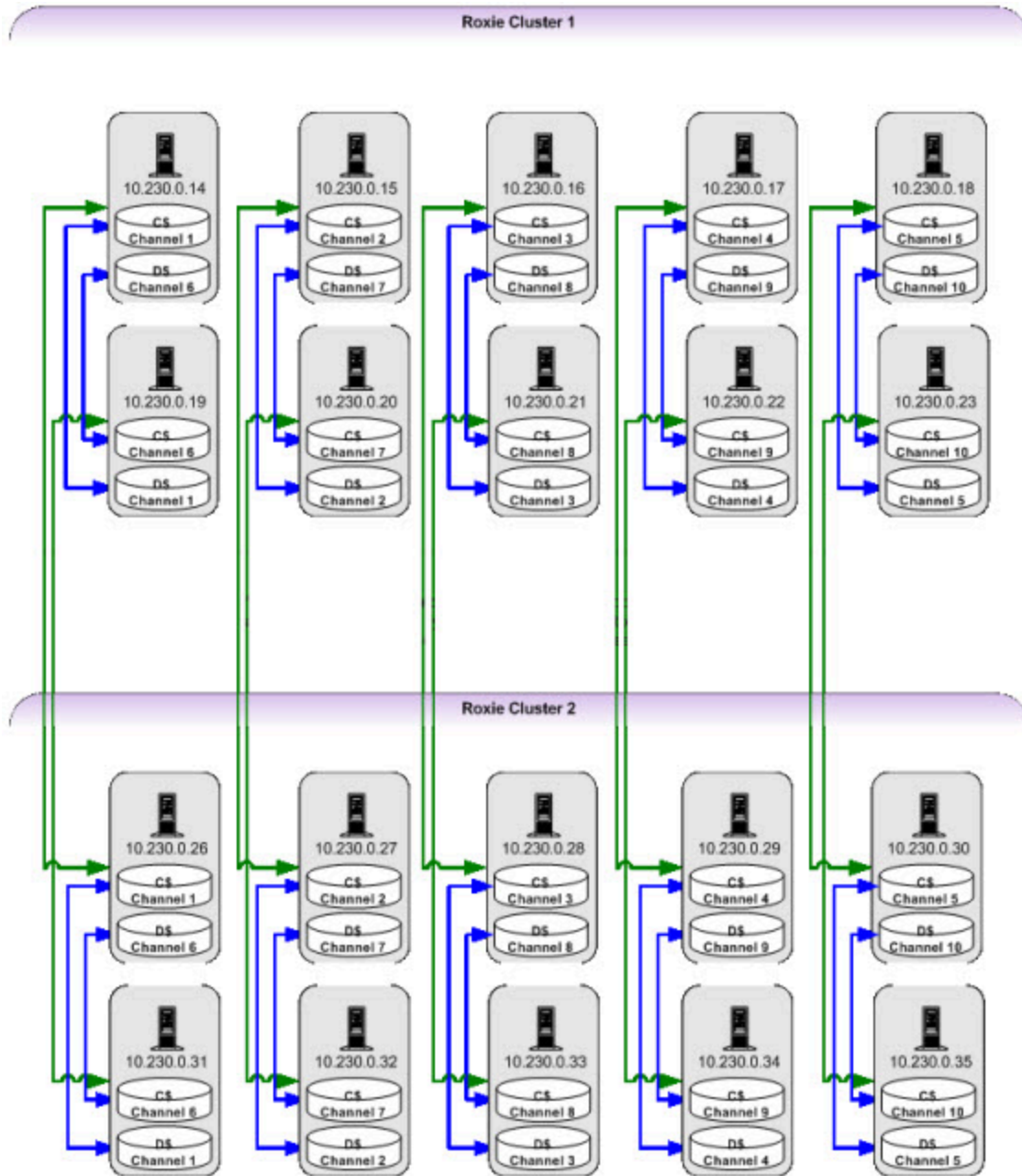
Backup de Dados do Roxie

Os dados do Roxie são protegidos por três formas de redundância:

- **Retenção de arquivo de dados de origem original:** Quando uma consulta é implementada, os dados são normalmente copiados dos discos rígidos de um cluster Thor. Dessa forma, os dados do Thor podem servir como backup, contanto que não sejam removidos nem alterados no Thor. Os dados do Thor normalmente são retidos por um período suficiente para servir como uma cópia de backup.
- **Redundância de nó par:** Cada nó o agente normalmente tem um ou mais nós pares em seu cluster. Cada par armazena uma cópia dos arquivos de dados que serão lidos.
- **Redundância de cluster irmão:** Embora não seja necessário, as implementações do Roxie podem ser executadas em múltiplos clusters Roxie configurados de modo idêntico. Quando dois clusters são implementados para produção, cada nó tem um gêmeo idêntico relacionado aos dados e/ou consultas armazenadas no nó no outro cluster.

Essa configuração oferece múltiplas cópias redundantes de arquivos de dados. Neste exemplo, há seis cópias de cada arquivo em determinado momento, eliminando a necessidade de usar procedimentos de backup tradicionais para arquivos de dados Roxie.

Roxie: O Motor de Entrega Rápida de Dados Introdução



Afinidade do processador

O Roxie pode obter um melhor desempenho em algumas circunstâncias usando afinidades de thread (linhas de execução) para restringir os threads de uma determinada consulta para um subconjunto de núcleos na máquina. Isso também é conhecido como afinidade de processador ou anexação de CPU. É uma técnica especialmente útil em máquinas de múltiplos soquetes (CPU dupla).

Isso pode ser definido no Gerenciador de Configurações usando as seguintes opções:

| | | |
|----------------------|-----------|--|
| <i>affinity</i> | Padrão: 0 | Se for diferente de zero, vincula o processo roxie a usar somente os núcleos especificados (bitmask) |
| <i>coresPerQuery</i> | Padrão: 0 | Se for diferente de zero, vincula cada consulta de entrada para usar apenas o número especificado de núcleos |

É possível substituir o *coresPerQuery* em uma única consulta usando:

```
#OPTION('bindCores', N);
```

onde N é o número de núcleos a serem usados.

Para obter melhores resultados, use um valor para *coresPerQuery* ou *bindCores* que seja divisível pelo número de núcleos em cada CPU.

Desenvolvendo Consultas Roxie

Caminho de Desenvolvimento

1. Determine as necessidades.
2. Avalie os dados e determine os campos para indexar.
3. Crie os índices.
4. Crie uma consulta hThor.
5. Teste e ajuste a consulta (usando o hThor).
6. Publique a consulta em um cluster Roxie.
7. Teste e certifique (compare os resultados com aqueles esperados).

Observação : Essas etapas são detalhadas no Tutorial de Dados e no Guia do Programador do HPCC .

Métodos para Aplicar Jobs a um Cluster Roxie

Após uma consulta ter sido compilada e implementada, há vários métodos para enviar tarefas que utilizam a consulta. Embora o uso mais comum seja através de aplicações pela interface SOAP ou XML, os outros métodos também são usuais.

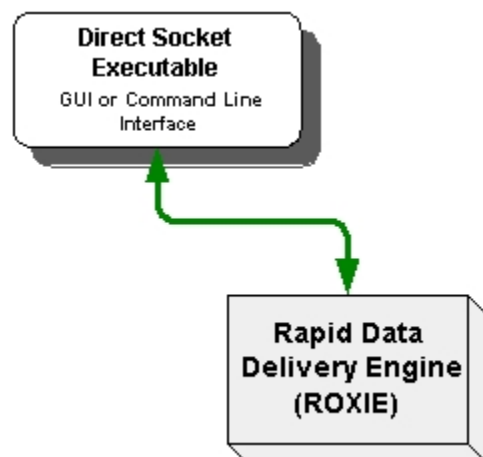
Uma conexão de soquete direta é capaz de se comunicar diretamente com o cluster Roxie, eliminando todos os outros componentes intermediários. Isso oferece uma maneira para certificar o cluster Roxie, sua configuração, a implementação da consulta e a própria consulta.

SOAPCALL permite que uma consulta do Thor acione uma consulta Roxie (consulte a *Referência de Linguagem ECL* para obter mais informações). Isso possibilita a combinação dos resultados do Roxie com outras tarefas de processamento de dados realizadas durante o ETL.

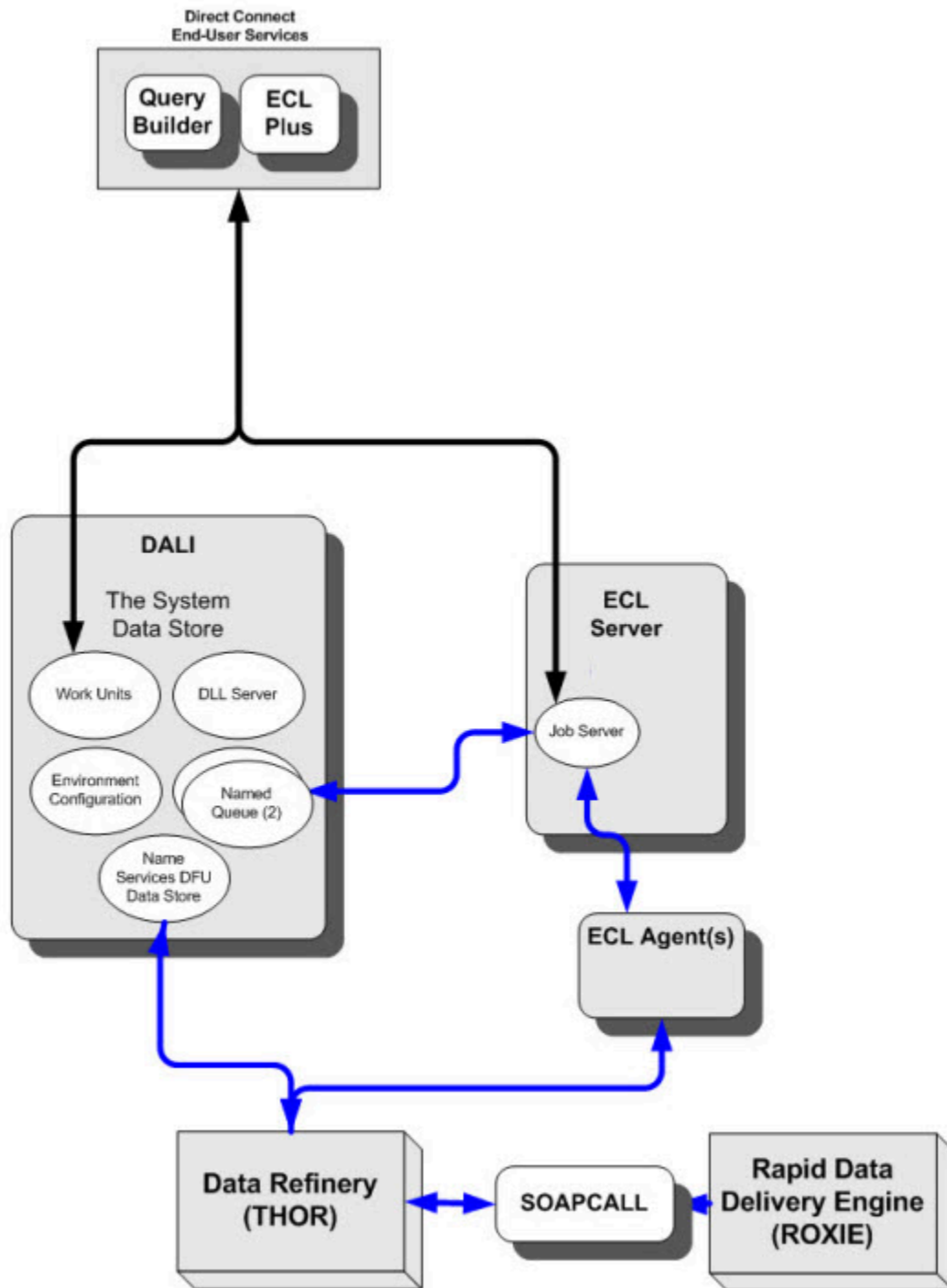
Realizar consultas por um ESP Service usando HTTP ou HTTPS permite o acesso à consultas diretamente de um navegador. O acesso pela Web permite que você conceda fácil acesso a quem desejar. Usando o HTTPS, é possível garantir a segurança de dados através da criptografia Secure Socket Layer (SSL). Isso garante que todos os dados sejam criptografados ao se deslocarem por uma rede ou Internet. Além disso, a autenticação LDAP está disponível para restringir o acesso a um conjunto de usuários.

Aplicativos personalizados que usam SOAP oferecem máxima flexibilidade e funcionalidade. O processo de desenvolvimento de aplicação é simplificado através da geração automática da Linguagem de Descrição de Serviços Web (WSDL) da Enterprise Services Platform (ESP). Muitas ferramentas de desenvolvimento (como o Microsoft .NET Studio ou NetBeans JAVA) incluem uma ferramenta para gerar códigos para criar stubs de proxy a partir de um documento WSDL. Isso simplifica o processo de desenvolvimento e garante a exposição de todos os métodos e propriedades necessários.

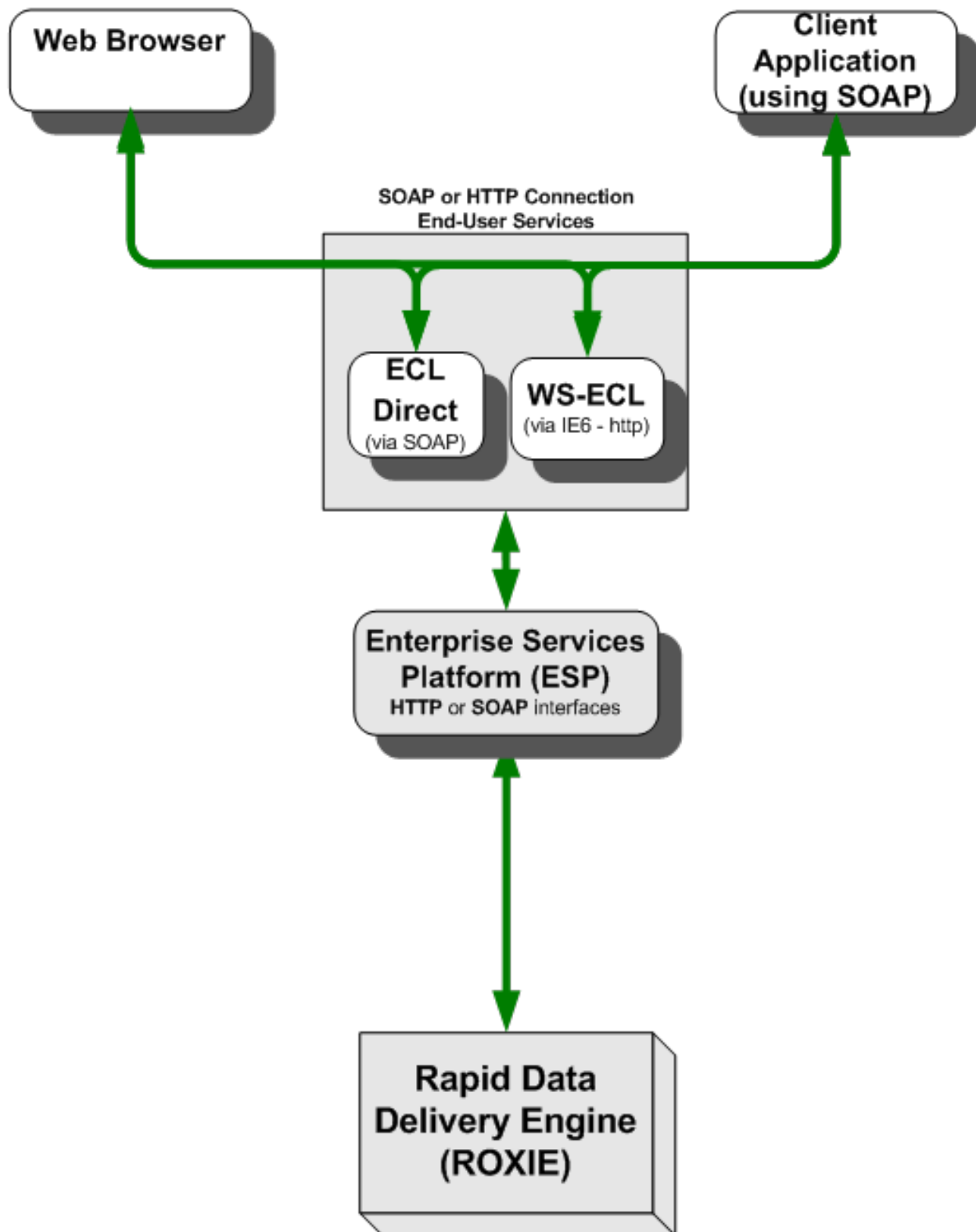
Conexão Direta com Socket (TCP/IP)



SOAPCALL via ECL



SOAP ou HTTP/HTTPS



Gerenciando Consultas

As consultas Roxie são gerenciadas através de Grupos de Consultas que são armazenados no Dali. Os Grupos de Consultas controlam quais consultas são carregadas no Roxie quando ele é iniciado e quais são manipuladas ao adicionar ou removê-las conforme necessário. A lista de consultas atualmente mantida em um Grupo de Consultas pode ser visualizada através do ECL Watch.

Depois que uma consulta é publicada no Grupo de Consultas, ela pode então ser executada no Roxie usando uma interface da Web.

Adicionando uma consulta Roxie ao Grupo de Consultas

Para adicionar uma consulta Roxie a um Grupo de Consultas:

1. Use o ECL IDE para gravar sua consulta e depois compilá-la, definindo o destino como o cluster Roxie em seu ambiente HPCC Systems.
2. Acesse a guia ECL Watch para a tarefa compilada e clique no botão Publish. Uma mensagem indicando que sua consulta foi publicada com sucesso é então exibida.

Visualizando Grupo de Consultas utilizando o ECL Watch

Usando o ECL Watch, é possível ver a consulta para todos os clusters no seu HPCC. Clique no item de menu **Queries Set/Browser** para ver os clusters que estão usando Grupo de Consultas atualmente. Clique no link **myroxie** para ver a lista de consultas atualmente disponíveis para o Roxie. Com o uso deste recurso, é possível:

- Ver a lista de consultas atualmente disponíveis em um cluster.
- Ver detalhes sobre cada consulta, incluindo o ID, o nome da consulta, o ID de tarefa, o(s) DLL(s) que ela usa e se ela está suspensa.
- Ver detalhes dos aliases existentes para cada consulta.
- Excluir uma Consulta ECL da lista.
- Excluir um alias.
- Ligar/desligar a configuração de suspensão

Utilizando o WsECL para executar uma consulta Roxie

O WsECL é a interface de serviço Web da ECL fornecida com o HPCC Systems e disponibilizada no seguinte URL: <http://nnn.nnn.nnn.nnn:8002>, onde nnn.nnn.nnn.nnn é o endereço IP do seu ESP.

O WsECL usa informações de Grupo de Consultas para exibir a lista de consultas executáveis disponíveis. Você pode usá-lo, por exemplo, para testar se sua consulta funciona como esperado.

A página da Web mostra todos os clusters que usam Grupo de Consultas. Amplie o myroxie na árvore e selecione a consulta que deseja executar. Um formulário padrão, gerado a partir dos tipos e nomes de campos de entrada, é exibido. Insira os valores e pressione Enviar para ver os resultados e testar sua consulta.

Packages e Package Maps

Uma consulta do Roxie pode ter um ou mais arquivos de dados associados. Às vezes, pode ser recomendado atualizar os dados sem alterar a consulta.

Os pacotes de dados são uma maneira de separar a definição de dados da consulta.

Um pacote de dados define uma superchave a ser usada por uma consulta. Um pacote de dados mais recente que redefine o conteúdo dessa superchave pode ser posteriormente enviado para um cluster Roxie sem recompilação de consultas. Isso permite a atualização dos dados, e garante que você esteja usando um código idêntico na consulta. Além disso, também elimina a necessidade de recompilar e republicá-lo.

Isso simplifica o controle de alterações e permite que os desenvolvedores de consultas continuem trabalhando sem correr o risco de uma consulta ser implementada para produção antes de estar pronta, embora ainda permita que os dados sejam atualizados.

Um mapa de pacote de dados oferece uma referência para o conteúdo de uma superchave usada em consultas que substitui a definição original. Isso garante maior flexibilidade e controle sobre os dados usados por uma consulta ou um conjunto de consultas.

O Roxie resolve os arquivos de dados no tempo de carregamento de consulta. Primeiro, ele vai procurar por um pacote que defina o conteúdo da superchave. Se tal pacote não for encontrado, ele irá procurar no DFU (Utilitário de arquivos distribuídos) do servidor Dali.

O resultado final é um processo mais rápido e flexível, sem a necessidade de recompilar consultas complexas. Isso permite que você atualize dados sem a chance de implementar um novo código que ainda não tenha sido aprovado para migração até a produção.

Sem os pacotes, você precisaria seguir uma das seguintes opções:

- Adicionar novos subarquivos à superchave através do código ECL, ECLWatch, ou DFUPlus, e depois recarregar o cluster.
- Revisar a consulta para usar um arquivo-chave diferente, compilar e republicá-lo.

Um arquivo de mapa de pacote pode conter um ou mais pacotes. O arquivo é usado para transferir as informações ao servidor Dali. Depois de ser enviado, o arquivo não é mais usado. Recomenda-se manter uma cópia como backup, mas isso não serve para nenhuma outra finalidade.

A definição de um superarquivo ou superchave dentro de um arquivo de pacote substitui a definição no Dali, mas NÃO muda a definição de um superarquivo ou superchave no DFU do servidor Dali.

Informações de pacote são usadas logo que são adicionadas ao Dali e o pacote é ativado. Isso pode ser feito usando o comando **ecl packagemap add --activate** (ou **ecl packagemap add & ecl packagemap activate**).

Exemplo

Neste exemplo, temos uma consulta com nome MyQuery que usa uma superchave com nome MyDataKey que inclui dois subarquivos:

- ~thor::MySubfile1
- ~thor::MySubfile2

Se quisermos adicionar um terceiro subarquivo, podemos usar um mapa de pacotes para redefinir a definição de superchave MyDataKey de modo que ela contenha três arquivos:

- ~thor::MySubfile1
- ~thor::MySubfile2
- ~thor::MySubfile3

Exemplo 1:

```
<RoxiePackages>
  <Package id="MyQuery">
    <Base id="thor::MyData_Key" />
  </Package>
  <Package id="thor::MyData_Key">
    <SuperFile id="~thor::MyData_Key">
      <SubFile value="~thor::MySubfile1"/>
      <SubFile value="~thor::MySubfile2"/>
    </SuperFile>
  </Package>
</RoxiePackages>
```

Exemplo 2:

```
<RoxiePackages>
  <Package id="MyQuery">
    <Base id="thor::MyData_Key" />
  </Package>
  <Package id="thor::MyData_Key">
    <SuperFile id="~thor::MyData_Key">
      <SubFile value="~thor::MySubfile1"/>
      <SubFile value="~thor::MySubfile2"/>
      <SubFile value="~thor::MySubfile3"/>
    </SuperFile>
  </Package>
</RoxiePackages>
```

Atualizando Dados

Esta seção detalha as etapas típicas que um desenvolvedor de consultas e um desenvolvedor de dados seguem logo que uma consulta está pronta para produção.

- Prepare os dados (neste fluxo de trabalho, os dados são definidos como uma superchave)
- Grave e teste a consulta
- Publique uma consulta usando os dados

Posteriormente, quando quiser atualizar os dados:

Implemente os dados através de uma das seguintes formas:

- Crie um mapa de pacotes que contenha um pacote redefinindo o conteúdo da superchave
- Adicione os mapas de pacotes associando as informações de pacote com um grupo de consultas.
- Ative os mapas de pacotes com as informações de pacote pertinentes voltadas para um grupo de consultas.

Use o comando:

```
ecl packagemap add --activate
```

Posteriormente, quando os novos dados chegarem, siga essas etapas para atualizar os dados usando pacotes:

- Prepare os dados e crie um novo subarquivo

Observação: É altamente recomendado não reutilizar um nome de arquivo. Normalmente é melhor criar novos arquivos. O Roxie compartilha identificadores de arquivo de forma que tentar alterar um arquivo existente enquanto ele é carregado pode causar problemas.

- Crie um pacote com uma definição de superchave que inclua o novo subarquivo
- Adicione o mapa de pacotes que redefine o conteúdo da superchave

Use o comando:

```
ecl packagemap add --activate
```

Sintaxe de Arquivos Package

Os arquivos de pacotes são formatados em XML usando as convenções detalhadas nesta seção.

Um arquivo de pacote precisa começar com **<RoxiePackages>** e terminar com **</RoxiePackages>**.

Tags de pacote possuem um atributo de **id** que especifica à que o pacote se refere.

Dentro da estrutura **<Package>**, são realizadas referências para superarquivos ou outros pacotes nomeados. Essa convenção de nomenclatura indireta permite que você agrupe definições de arquivos e referencie o pacote que contenha tal grupo.

O exemplo mostra primeiramente as referências de consulta e, em seguida, as referências de arquivos; no entanto, a exibição não precisa ser nessa ordem.

As linhas são numeradas apenas para fins de referência. Os comentários são incluídos apenas para esclarecimento, não sendo, portanto, obrigatórios.

Exemplo :

```
1. <RoxiePackages>
2. <!-- Begin Queries -->
3.   <Package id="MyQuery">
4.     <Base id="thor::MyData_Key" />
5.   </Package>
6. <!-- End Queries -->
7. <!-- Begin File references -->
8.   <Package id="thor::MyData_Key">
9.     <SuperFile id="~thor::MyData_Key">
10.      <SubFile value="~thor::Mysubfile1"/>
11.      <SubFile value="~thor::Mysubfile2"/>
12.      <SubFile value="~thor::Mysubfile3"/>
13.    </SuperFile>
14.   </Package>
15. <!--End File references -->
16. </RoxiePackages>
```

Neste exemplo, temos uma consulta: **MyQuery**. A consulta usa uma superchave definida em um pacote **thor::MyData_Key**. Isso é definido posteriormente na linha 8. O pacote **thor::MyData_Key** contém uma definição de superchave.

O exemplo mostra primeiramente as referências de consulta e, em seguida, as referências de arquivos; no entanto, a exibição não precisa ser nessa ordem.

É possível especificar IP(s) do Dali e/ou cluster(s) de origem a serem usados para resolver arquivos que serão especificados no mapa de pacotes. Isso pode ser especificado em diferentes níveis para influenciar os arquivos em um determinado escopo.

Por exemplo:

```
<PackageMap daliip="10.239.8.10">
  <Package id="MyPackage" daliip="10.239.7.1" sourceCluster="thor1" >
    <SuperFile id="thor::Mykey1_idx">
      <SubFile value="thor::mySub1"/>
    </SuperFile>
    <SuperFile id="thor::Mykey2_idx" daliip="10.239.8.1" sourceCluster="thor2">
      <SubFile value="thor::mySubKeyA"/>
      <SubFile value="thor::mySubKeyB"/>
    </SuperFile>
```

```
</Package>  
</PackageMap>
```

Multi-Part package maps

Na versão 6.0.0 ou posterior, é possível organizar seus mapeamentos do arquivo do mapa de pacote em um conjunto de arquivos que define os arquivos de alguns subconjuntos de consultas. Esses subconjuntos são chamados de partes e podem ser usados para organizar através de vários agrupamentos, como função, arquivos, desenvolvedor e afins.

Com um arquivo packag map de parte única como:

main.pkg

```
<RoxiePackages>
  <Package id="my_base" preload="1">
    <SuperFile id="thor1::people::key::person_superkey">
      <SubFile value="thor1::key::people::person_file_today"/>
    </SuperFile>
  </Package>
  <Package id="another_base" preload="1">
    <SuperFile id="thor1::people::key::contact_superkey">
      <SubFile value="thor1::key::people::contacts"/>
    </SuperFile>
  </Package>
  <Package id="person_query" 2016HondaAccordLX-23041.xhtml preload="1">
    <Base id="my_base"/>
  </Package>
  <Package id="contacts_query" preload="1">
    <Base id="another_base"/>
  </Package>
</RoxiePackages>
```

Para adicionar o packag map no destino e ativá-lo, use:

```
ecl packagemap add <target> main.pkg --activate
```

Para atualizar os mapeamentos, é necessário editar o arquivo do mapa de pacotes (ou uma cópia), adicionar e ativar o novo arquivo e depois remover o antigo.

Usando **parts**, é possível gerenciar o pacote de modo mais granular. É possível dividir os mapeamentos em arquivos de partes individuais.

contacts.pkg

```
<RoxiePackages>
  <Package id="contact_base" preload="1">
    <SuperFile id="thor1::people::key::contacts_superkey">
      <SubFile value="thor1::key::people::contacts_file_today"/>
    </SuperFile>
  </Package>
  <Package id="contact_query" preload="1">
    <Base id="contact_base"/>
  </Package>
  <Package id="contact_references_query" preload="1">
    <Base id="contact_base"/>
  </Package>
</RoxiePackages>
```

persons.pkg

```
<RoxiePackages>
  <Package id="person_base" preload="1">
    <SuperFile id="thor1::people::key::person_superkey">
      <SubFile value="thor1::key::people::person_file_today"/>
    </SuperFile>
  </Package>
</RoxiePackages>
```



```
</SuperFile>
</Package>
<Package id="person_query" preload="1">
  <Base id="person_base"/>
</Package>
</RoxiePackages>
```

Isso permite que você os adicione separadamente:

```
ecl packagemap add-part <target> target_pmid contacts.pkg
ecl packagemap add-part <target> target_pmid persons.pkg
```

E depois, ative todo o mapa de pacotes (que é composto de todas as partes adicionadas ao determinado target_pmid):

```
ecl packagemap activate <target> target_pmid
```

É possível então alterar as partes individuais e atualizá-las no mapa de pacotes:

```
ecl packagemap add-part <target> <target_pmid> persons.pkg --delete-previous
```

Observe que o target_pmid é o identificador da coleta de todas as partes do packagemap. O identificador pode ser usado para ativar ou remover todo o conjunto (mapa de pacotes).

É possível usar a função "get-part" para ver como a parte nomeada é exibida no Dali para um determinado identificador:

```
ecl packagemap get-part <target> <packagemap> contents.pkg
```

Dessa forma, é possível recuperar a parte, editá-la e substituí-la como mostrado acima. É possível usar a função **remove-part** para remover uma parte explicitamente de um mapa de pacotes.

Ainda é possível tratar todo o mapa de pacotes como um arquivo único, embora sua organização tenha sido alterada para refletir as suas partes:

multipart.pkg

```
<PackageMaps multipart="1">
  <Part id="contacts.pkg">
    <Package id="contact_base" preload="1">
      <SuperFile id="thor1::people::key::contacts_superkey">
        <SubFile value="thor1::key::people::contacts_file_today"/>
      </SuperFile>
    </Package>
    <Package id="contact_query" preload="1">
      <Base id="contact_base"/>
    </Package>
    <Package id="contact_references_query" preload="1">
      <Base id="contact_base"/>
    </Package>
  </Part>
  <Part id="persons.pkg">
    <Package id="person_base" preload="1">
      <SuperFile id="thor1::people::key::person_superkey">
        <SubFile value="thor1::key::people::person_file_today"/>
      </SuperFile>
    </Package>
    <Package id="person_query" preload="1">
      <Base id="person_base"/>
    </Package>
  </Part>
</PackageMaps>
```

Observe agora como o conteúdo está dividido em partes <Part/>. Porém, ainda é possível usar

```
ecl packagemap add <target> main.pkg
```

nesse mapa de pacotes de múltiplas partes e depois continuar gerenciando suas partes individuais. Neste caso:

```
ecl packagemap add <target> multipart.pkg  
ecl packagemap add-part <target> multipart.pkg addresses.pkg  
ecl packagemap activate <target> multipart.pkg
```

Isso seria útil, por exemplo, ao copiar todo o mapa de pacotes de um destino para outro.

Trabalhando com Packages através de linhas de comando ECL

Essa seção contém detalhes sobre as ações e opções usadas para trabalhar com pacotes. A ferramenta de linha de comando do ECL está totalmente documentada no Manual de ferramentas de cliente.

ecl packagemap add

ecl packagemap add [--daliip][options] <target> <filename>

Exemplos:

```
ecl packagemap add -s=192.168.1.10 roxie mypackagemap.pkg  
ecl packagemap add roxie mypackagemap.pkg --overwrite  
ecl packagemap add roxie mypackagemap.pkg --daliip=192.168.11.11
```

| | |
|-----------------------|--|
| ecl packagemap add | Aciona o comando adicionar Package Map |
| Ações | |
| add | Adiciona um packgemap a um cluster de destino |
| Argumentos | |
| target | O destino com o qual o package map será associado |
| filename | O nome do arquivo que contém as informações do package map. |
| --daliip= | Endereço IP ou nome do host do Dali remoto a ser usado para pesquisa de arquivos lógicos |
| Opções | |
| -O, --overwrite | Se a informação existente deve ser substituída – true se estiver presente |
| -A, --activate | Ativar o Package Map |
| --allow-foreign | Especifica a permissão do uso de arquivos remotos. Se um package map referencia arquivos remotos e se esta opção não for ativada, a adição falhará. |
| --pmid=<packagemapid> | ID do packagemap – se não for especificado, o padrão é o nome do arquivo |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |

ecl packagemap delete

ecl packagemap delete [options] <target><packagemap>

Exemplos:

```
ecl packagemap delete roxie mypackagemap
```

| | |
|-----------------------|---|
| ecl packagemap delete | Aciona o comando para remover o package map |
| Ações | |
| delete | Remove um package map |
| Opções | |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |
| --wait-connect=<Ms> | Tempo limite durante a conexão ao servidor (em milisegundos) |
| --wait-read=<Secs> | Tempo limite durante a leitura do socket (em segundos) |

ecl packagemap activate

ecl packagemap activate <target> <packagemap>

Exemplo:

```
ecl packagemap activate roxie mypackagemap.pkg
```

| | |
|-------------------------|--|
| ecl packagemap activate | O comando ativar desativará o package map atual, ativando o package map especificado. |
| Argumentos | |
| target | O destino que contém o package map que será ativado |
| packagemap | O nome do package map será atualizado |
| Opções | |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |
| --wait-connect=<Ms> | Tempo limite durante a conexão ao servidor (em milissegundos) |
| --wait-read=<Secs> | Tempo limite durante a leitura do socket (em segundos) |

ecl packagemap deactivate

ecl packagemap deactivate <target> <packagemap>

Exemplo:

```
ecl packagemap deactivate roxie mypackagemap.pkg
```

| | |
|---------------------------|--|
| ecl packagemap deactivate | O comando desativará o package map atualmente ativo. |
| Argumentos | |
| target | O destino que contém o package map que será desativado |
| packagemap | O nome do package map que será desativado |
| Opções | |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |
| --wait-connect=<Ms> | Tempo limite durante a conexão ao servidor (em milisegundos) |
| --wait-read=<Secs> | Tempo limite durante a leitura do socket (em segundos) |

ecl packagemap list

ecl packagemap list <target>

Exemplos:

```
ecl packagemap list roxie
```

| | |
|---------------------|---|
| ecl packagemap list | Aciona o comando da listagem de package maps |
| Ações | |
| list | Lista os nomes do package map carregado |
| Argumentos | |
| target | O destino que contém o package map a ser listado |
| Opções | |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |
| --wait-connect=<Ms> | Tempo limite durante a conexão ao servidor (em milisegundos) |
| --wait-read=<Secs> | Tempo limite durante a leitura do socket (em segundos) |

ecl packagemap info

ecl packagemap info [options] <target>

Exemplos:

```
ecl packagemap info roxie
```

| | |
|---------------------|---|
| ecl packagemap info | Aciona o comando de informações sobre o package map |
| Ações | |
| info | retorna informações do package map |
| Argumentos | |
| target | O destino que contém o package map a ser recuperado |
| Opções | |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |
| --wait-connect=<Ms> | Tempo limite durante a conexão ao servidor (em milisegundos) |
| --wait-read=<Secs> | Tempo limite durante a leitura do socket (em segundos) |

ecl packagemap add-part

ecl packagemap add-part <target> <pmid> <filename>

Exemplos:

```
ecl packagemap add-part roxie multipart.pkg addresses.pkg
```

O comando packagemap add-part adiciona conteúdo adicional do package a um package map existente

| | |
|--------------------------|--|
| ecl packagemap add-part | Aciona o comando packagemap add-part. |
| Ações | |
| add-part | Adiciona conteúdo adicional do package map a um existente |
| Argumentos | |
| target | Nome do destino a ser usado ao adicionar partes do package map |
| pmid | Identificador do package map para o qual a parte será adicionada |
| filename | Um ou mais arquivos de parte |
| Opções | |
| --part-name | Nome da parte que está sendo adicionada (o padrão é o nome do arquivo) |
| --delete-prev | Substitui uma parte existente de mesmo nome |
| --daliip=<ip> | IP do Dali remoto a ser usado na pesquisa de arquivos lógicos |
| --global-scope | O package map especificado é compartilhado entre vários destinos |
| --source-process=<value> | Processa o cluster para copiar arquivos |
| --allow-foreign | Não falha caso arquivos remotos sejam usados no package map |
| --preload-all | Define a opção de carregamento prévio de todos os pacotes |
| --update-super-files | Atualiza os superarquivos DFS locais se o Dali remoto tiver sido alterado |
| --update-clone-from | Atualiza o clone local da localização se o Dali remoto tiver sido alterado |
| --dont-append-cluster | Usar apenas para evitar problemas de bloqueio devido a adição de cluster ao arquivo |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |
| --wait-connect=<Ms> | Tempo limite durante a conexão ao servidor (em milissegundos) |
| --wait-read=<Secs> | Tempo limite durante a leitura do socket (em segundos) |

ecl packagemap get-part

ecl packagemap get-part <target> <packagemap> <partname>

Exemplos:

```
ecl packagemap get-part roxie multipart.pkg contacts
```

O comando get-part busca uma determinada parte de um package map

| | |
|-------------------------|---|
| ecl packagemap get-part | Aciona o comando packagemap gert-part. |
| Ações | |
| get-part | Busca uma determinada parte de um package map |
| Argumentos | |
| target | Nome do destino a ser usado ao adicionar partes do package map |
| packagemap | Nome do package map que contém a parte |
| partname | Nome da parte a ser recuperada |
| Opções | |
| --global-scope | O package map especificado é compartilhado entre vários destinos |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |
| --wait-connect=<Ms> | Tempo limite durante a conexão ao servidor (em milisegundos) |
| --wait-read=<Secs> | Tempo limite durante a leitura do socket (em segundos) |

ecl packagemap remove-part

ecl packagemap remove-part <target> <pmid> <partname>

Exemplos:

```
ecl packagemap remove-part roxie multipart.pkg contacts
```

O comando remove-part removerá determinada parte do package map

| | |
|----------------------------|--|
| ecl packagemap remove-part | Aciona o comando packagemap remove-part. |
| Ações | |
| remove-part | Remove uma determinada parte do package map |
| Argumentos | |
| target | Nome do destino a ser usado |
| packagemap | Nome do package map que contém a parte |
| partname | Nome da parte a ser removida |
| Opções | |
| --global-scope | O package map especificado é compartilhado entre vários destinos |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |
| --wait-connect=<Ms> | Tempo limite durante a conexão ao servidor (em milisegundos) |
| --wait-read=<Secs> | Tempo limite durante a leitura do socket (em segundos) |

ecl packagemap validate

ecl packagemap validate <target> [<filename>]

Exemplos:

```
ecl packagemap validate roxie mypackagemap.pkg  
ecl packagemap validate roxie --active
```

O comando packagemap validate verifica se:

- As superchaves referenciadas possuem subarquivos definidos (avisa caso não existam subarquivos)
- Todas as consultas ECL referenciadas existem no grupo de consultas Roxie atual
- Todas as consultas Roxie estão definidas no pacote

O resultado também listará qualquer arquivo usado pelas consultas ECL, mas que não foram mapeados no package map.

Filename, --active, e --pmid são mutuamente exclusivos. As opções --active ou --pmid validam um package map já adicionado em vez de um arquivo local.

A opção --queryid verifica os arquivos em uma única consulta em vez de todas as consultas no grupo de consultas de destino. É mais rápido quando apenas os arquivos de uma única consulta precisam ser validados.

| | |
|-------------------------|---|
| ecl packagemap validate | Aciona o comando packagemap validate. |
| Ações | |
| validate | Retorna informações do package map |
| Argumentos | |
| filename | O nome do arquivo que contém as informações do package map a serem validadas |
| target | O destino que contém o package map a ser validado |
| Opções | |
| -v, --verbose | Saída de informações adicionais de rastreamento |
| -s, --server | O endereço IP ou nome do host do servidor ESP que está executando os serviços do ECL Watch |
| --active | Valida o package map ativo para o destino especificado |
| --pmid=<packagemapid> | Valida o package map especificado |
| --queryid | Valida os arquivos da ID da consulta especificada, se não estiverem mapeados no package map |
| --port | A porta dos serviços do ECL Watch (o padrão é 8010) |
| -ssl | Use o protocolo SSL para proteger a conexão com o servidor. |
| -u, --username | O nome do usuário (se necessário) |
| -pw, --password | A senha (se necessário) |
| --wait-connect=<Ms> | Tempo limite durante a conexão ao servidor (em milissegundos) |

Roxie: O Motor de Entrega Rápida de Dados
Packages e Package Maps

| | |
|--------------------|--|
| --wait-read=<Secs> | Tempo limite durante a leitura do socket (em segundos) |
|--------------------|--|

Dicas:

- Sempre use superarquivos ou superchaves para se beneficiar da nomenclatura indireta e permitir o uso de pacotes. O Roxie não exige isso, mas funciona melhor dessa maneira.
- Use nomes de arquivos únicos em vez de substituir arquivos existentes. Isso evita substituições acidentais e facilita as reversões.
- Se você possui um conjunto de chaves, é mais fácil manter se usar superchaves com um arquivo de subíndice único. Múltiplos subarquivos são úteis quando é necessário adicionar os dados rapidamente, mas se houver tempo, é melhor recompilar um novo arquivo-chave único.
- Antes de remover um mapa de pacotes, verifique se você tem uma cópia como backup.

Acesso Direto ao Roxie

Visão Geral do Acesso a uma Consulta Roxie

Ele proporciona um meio imediato de teste, oferece uma interface da Web para execução de consultas únicas e oferece o WSDL e esquemas que podem ser usados para geradores de código automatizados para acesso ao SOAP. Para aproveitar esse recurso completamente, é necessário acessar o Roxie diretamente.

O WsECL ainda desempenha um papel importante ao usar essa técnica. Ele proporciona um meio imediato de teste, oferece uma interface da Web para execução de consultas únicas e oferece o WSDL e esquemas que podem ser usados para geradores de código automatizados para acesso ao SOAP. Também fornece o HTTP-GET e compatibilidade com codificações de formulários. Finalmente, também oferece a interface JSON (Notação de Objeto em JavaScript) que é semelhante à interface direta do Roxie.

Web Services

Um serviço Web é um componente de software baseado em padrões que pode ser acessado pela Internet. O serviço pode ser complexo ou simples.

Por exemplo, um serviço Web poderia solicitar um número de orçamentos de ações e retorná-los em um conjunto de resultados em XML.

Os serviços Web estão disponíveis para qualquer plataforma, modelo de objeto ou linguagem de programação. Isso oferece acesso aos usuários através da Internet, Intranet ou Extranet, independentemente de sua plataforma.

Além disso, também simplifica o desenvolvimento de sistemas distribuídos. O uso de componentes baseados em padrões protege o investimento de desenvolvimento, independentemente das tendências futuras. As tecnologias de serviço Web são baseadas no protocolo HTTP.

Simple Object Access Protocol (SOAP)

O SOAP é o protocolo baseado em XML mais comum para envio de solicitações e recebimento de respostas dos serviços Web. Basicamente, trata-se de um protocolo para comunicação entre aplicações. Ele foi projetado para comunicação pela Internet e é independente de plataforma e de linguagem. Baseado em XML, o SOAP é simples e extensível.

JavaScript Object Notation (JSON)

JSON é uma formato leve de troca de dados. Ele é de fácil leitura e gravação por humanos. É de fácil interpretação e geração para máquinas. O JSON é baseado em um subconjunto da Linguagem de programação JavaScript, sendo completamente independente de linguagens.

Acessando suas Consultas Roxie

Depois de desenvolver, compilar e publicar consultas, é necessário oferecer acesso aos usuários.

ESP e WsECL

Uma instalação padrão da plataforma HPCC inclui o Enterprise Service Platform (ESP) com o serviço WsECL configurado através de uma conexão de serviço para a porta 8002 (configurável).

O WsECL oferece um meio para acessar todas as suas consultas compiladas e publicadas usando uma interface da Web ou usando SOAP ou JSON.

Dica: É possível fornecer um nome de tarefa no URL usando este formulário:

```
http://ip:port/WsEcl/submit/query/<targetCluster>/<query>?_jobname=<jobname>
```

Isso é útil para o teste de consultas. Também é adequado para oferecer acesso em tempo real aos usuários. Além disso é possível usar o modo de proxy para ignorar a formatação de resposta WsECL e formatar diretamente a resposta SOAP do Roxie. Consulte WsECL e Modo proxy

Acessando WsECL VIA SOAP

WSDL

O Protocolo Simples de Acesso a Objetos (SOAP) define um padrão para comunicação entre processos usando um formato de base XML comum. As bibliotecas do SOAP estão prontamente disponíveis para várias plataformas de desenvolvimento e linguagens, incluindo Microsoft .NET (acessível por C#, VB.NET, ASP.NET, e outras linguagens CLR), Java (p.ex., JAX e Apache AXIS), PERL, e muitas outras.

A Linguagem de Descrição de Serviço Web (WSDL) é usada para proporcionar uma descrição estruturada de uma interface de serviço Web.

Muitas das bibliotecas disponíveis para permitir que aplicativos usem o protocolo SOAP também oferecem ferramentas para gerar automaticamente APIs específicas de serviço a partir de um documento WSDL.

A descrição WSDL está disponível automaticamente para qualquer consulta publicada.

A URL a seguir pode ser usada para recuperar a descrição WSDL de uma consulta publicada:

```
http://nnn.nnn.nnn.nnn:8002/WsEcl/definitions/query/roxie/<queryName>/main/<queryName>.wsdl
```

onde *nnn.nnn.nnn.nnn* é o endereço IP (ou nome do DNS) desse servidor ESP com a ligação para o WsECL.

8002 é a porta padrão. Se você modificou uma configuração de porta, use a porta selecionada para essa finalidade.

A sintaxe e a funcionalidade do próprio protocolo SOAP também estão fora do escopo deste documento. A lista a seguir de recursos externos pode ajudar você a entender as tecnologias baseadas em padrões empregadas.

SOAP

<http://www.w3.org/TR/soap12-part0>

<http://www.w3.org/TR/soap12-part1>

<http://www.w3.org/TR/soap12-part2>

Web Service Definition Language (WSDL) 1.1

<http://www.w3.org/TR/wsdl>

Acesso Direto VIA SOAP

1. Crie e publique sua consulta para um cluster de destino (Roxie).
2. Teste e valide usando o WsECL para acessar a consulta.

Use a interface padrão fornecida para testar:

<http://nnn.nnn.nnn.nnn:8002/>

, onde nnn.nnn.nnn.nnn é o endereço IP (ou nome do DNS) desse servidor ESP com a ligação para o WsECL.

8002 é a porta padrão. Se você modificou uma configuração de porta, use a porta selecionada para essa finalidade.

3. Comece a desenvolver a aplicação que consumirá o serviço Web importando o WSDL de:

<http://nnn.nnn.nnn.nnn:8002/WsEcl/definitions/query/roxie/<queryName>/main/<queryName>.wsdl>

, onde nnn.nnn.nnn.nnn é o endereço IP (ou nome do DNS) desse servidor ESP com a ligação para o WsECL.

8002 é a porta padrão. Se você modificou uma configuração de porta, use a porta selecionada para essa finalidade.

4. Ao terminar os testes, altere o URL de base para:

<http://nnn.nnn.nnn.nnn:9876/>

, onde nnn.nnn.nnn.nnn é o endereço IP (ou nome do DNS) de um servidor Roxie.

9876 é a porta padrão. Se você modificou uma configuração de porta, use a porta selecionada para essa finalidade.

Acesso Direto VIA JSON

1. Crie e publique sua consulta para um cluster de destino (Roxie).
2. Teste e valide usando o WsECL para acessar a consulta.

Use a interface padrão fornecida para testar:

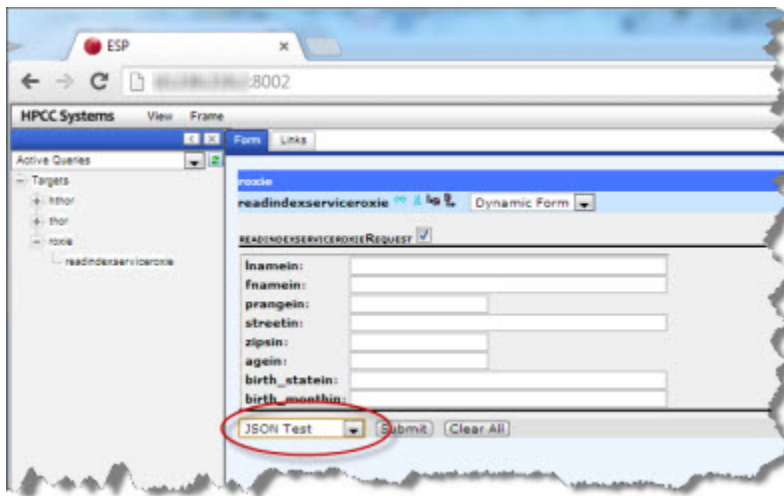
<http://nnn.nnn.nnn.nnn:8002/>

, onde nnn.nnn.nnn.nnn é o endereço IP (ou nome do DNS) desse servidor ESP com a ligação para o WsECL.

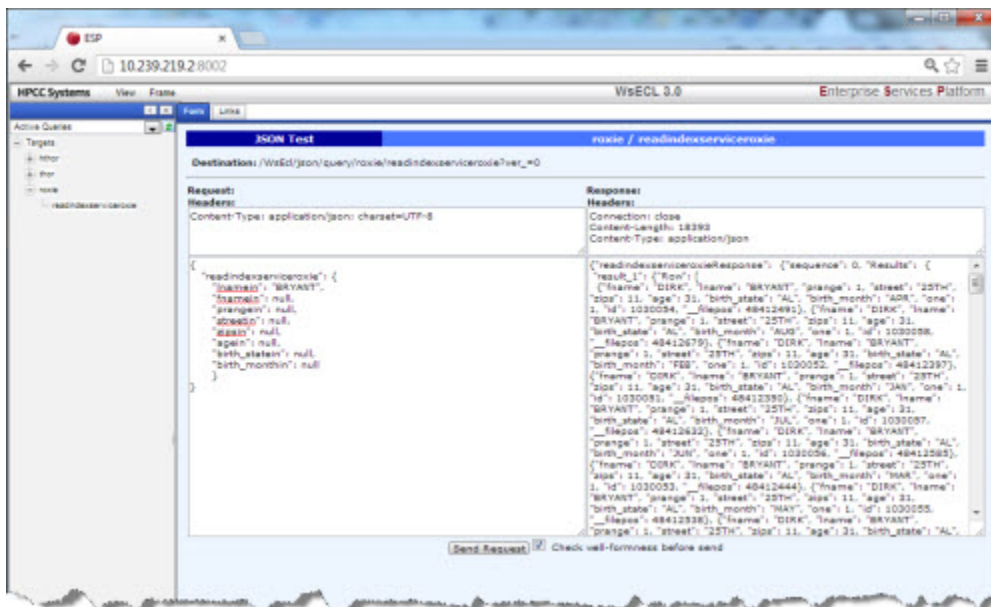
8002 é a porta padrão. Se você modificou uma configuração de porta, use a porta selecionada para essa finalidade.

3. Comece a desenvolver a aplicação que consumirá o serviço Web usando as definições JSON disponíveis a partir do formulário no WsECL.

- Na lista suspensa, selecione JSON Test e pressione o botão Submit.



- Edite os critérios de consulta à esquerda e depois pressione o botão **Send request**.
- Os resultados são exibidos no lado direito.
- Edite e reenvie (pressionando o botão **Submit Request**) para ver as respostas para os diferentes critérios.



- Defina a URL de base em sua aplicação para

<http://nnn.nnn.nnn.nnn:9876/>

, onde nnn.nnn.nnn.nnn é o endereço IP (ou nome do DNS) de um servidor Roxie.

9876 é a porta padrão. Se você modificou uma configuração de porta, use a porta selecionada para essa finalidade.

Observação: O serviço WsECL distribui automaticamente a carga enviando consultas para servidores Roxie em repetição alternada. Para utilizar todos os servidores, sua aplicação deve implementar um esquema similar de dis-

tribuição de cargas. Também é possível usar um balanceador de cargas e enviar a consulta para o IP virtual criado pelo balanceador de cargas.

Acesso Direto VIA JSON

Para obter uma resposta JSONP, adicione a decoração **&jsonp=<myfunction>** ao URL. Isso retorna uma resposta json encapsulada em JavaScript:

```
myfunction("{\"data_1\": \"hello world\", \"data_2\": [\"the\", \"sun\", \"is\", \"up\"]}");
```

WsECL e Modo proxy

Para acionar diretamente o HTTP-GET e HTTP-FORM-POST no Roxie, altere o URL para usar o modo proxy como mostrado abaixo:

```
Change:  
/WsEcl/submit/query/RoxieTargetName/QueryName  
to  
/WsEcl/proxy/query/RoxieTargetName/QueryName
```

Acionamentos proxy WsECL parametrizados para o Roxie são compatíveis com um subconjunto dos formatos de parâmetros suportados por acionamentos de modo não proxy.

```
/WsEcl/proxy/query/mytarget/myquery?param1=value
```

Para parâmetros complexos, use a notação de ponto para especificar os valores aninhados. Por exemplo:

```
<MyQuery>  
  <State>FL</State>  
  <Regs>  
    <Reg>  
      <Name>SOUTH</Name>  
      <Codes>  
        <Code>PBI</Code>  
        <Code>FLL</Code>  
      </Code>  
    </Reg>  
    <Reg>  
      <Name>NORTH</Name>  
      <Codes>  
        <Code>MCO</Code>  
      </Code>  
    </Reg>  
  </Regs>  
</MyQuery>
```

O URL ou formulário HTTP definiria os seguintes parâmetros:

```
State=FL  
Regs.Reg.0.Name=SOUTH  
Regs.Reg.0.Codes.Code.0=PBI  
Regs.Reg.0.Codes.Code.1=FLL  
Regs.Reg.1.Name=NORTH  
Regs.Reg.1.Codes.Code=MCO
```

Modo Expandido no WsECL

Para incluir os esquemas nos resultados, adicione a opção **expanded** a URL como mostrado abaixo:

```
Change:
```

```
/WsEcl/submit/query/RoxieTargetName/QueryName  
to  
/WsEcl/submit/query/RoxieTargetName/QueryName/expanded
```

Acesso RESTful Direto ao Roxie

É possível acessar suas consultas Roxie usando uma interface RESTful da seguinte maneira:

```
http://<ip>:9876/<target>/<queryid>?<stored1>=<value>
    &<storeddataset>.Row.0.name=abc&storeddataset.Row.0.id=123
```

, onde

Ip é o endereço IP ou nome de host do servidor Roxie, ou um VIP para um intervalo de IPs para uma torre de servidores Roxie

target é o nome do cluster de destino

queryid é a ID de consulta da consulta publicada.

stored1 é uma variável de entrada (usando STORED no ECL) e *valor* a ser enviado

storeddataset é um dataset a ser especificado para a consulta

Por exemplo:

```
http://127.0.0.1:9876/roxie/echotest.1?echoValue=Ziggy%20played%20guitar
```

Resultados Roxie

Por padrão, os resultados das consultas HTTP no Roxie não retornam elementos vazios (tags). Se um valor de campo estiver vazio, a tag não é retornada. No entanto, às vezes, uma aplicação pode esperar que todas as tags sejam retornadas de forma que seja possível adicionar uma decoração de URL para substituir o comportamento padrão. Isso faz com que tags vazias retornem um valor de uma string vazia.

```
.trim=0
```

A **opção .trim=0** pode ser adicionada ao SOAP, JSON, URL, ou consultas de formulário da Web para ambos WsECL e Roxie.

WsECL RESTful:

```
http://ip:port/WsEcl/submit/query/roxie/myquery?param1=abc&.trim=0
```

WsECL SOAP:

```
http://ip:port/WsEcl/soap/query/roxie/myquery?.trim=0
```

WsECL JSON:

```
http://ip:port/WsEcl/json/query/roxie/myquery?.trim=0
```

Roxie Direct SOAP ou JSON:

```
http://ip:port/target/myquery?.trim=0
```

Roxie Direct RESTful:

```
http://ip:port/roxie/myquery?param1=abc&.trim=0
```

Envio do formulário:

Para enviar o formulário RESTful para o WsECL ou Roxie, adicione a decoração .trim=0 ao seu URL.

A execução de qualquer XSLT personalizado deve definir automaticamente a opção .trim=0.

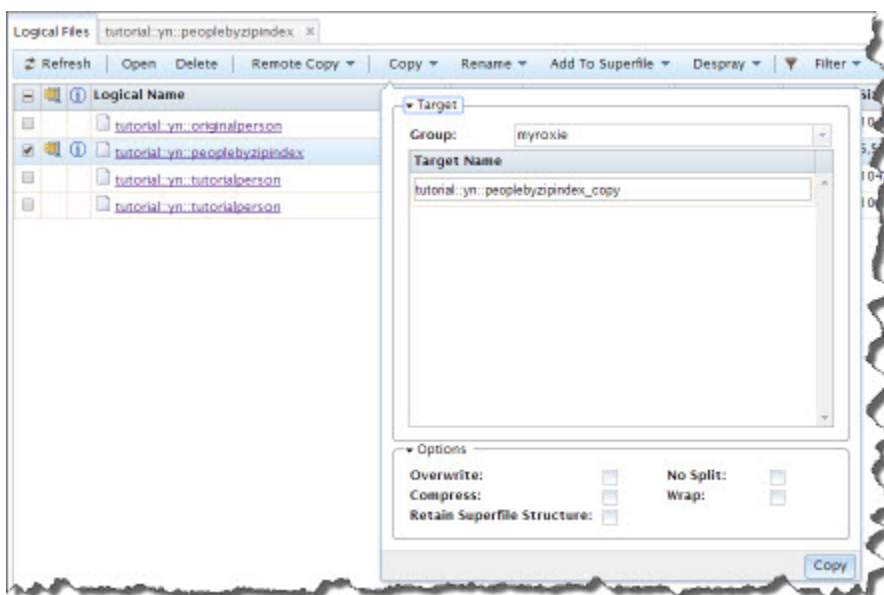
Disponibilizando Dados para um Cluster Roxie utilizando DFU

É possível usar o Utilitário de Arquivos Distribuídos (DFU) no ECL Watch para copiar ou realizar a cópia remota de arquivos de dados para um cluster Roxie. Isso permite que você copie arquivos grandes para um cluster antes da publicação de uma consulta. Se os arquivos de dados forem copiados com antecedência, uma consulta que requeira esses arquivos usará aqueles que já estão em vigor. Se você possui grandes datasets, isso permite que você prepare a implementação da consulta com antecedência.

Observação: Para usar esse recurso, o utilitário FTSlave precisa estar instalado em cada nó do cluster. Isso é feito automaticamente pelo assistente do Gerenciador de Configurações.

Copia DFU

1. Abra a página da Web do ECLWatch. É possível acessar essa página usando o seguinte URL: <http://nnn.nnn.nnn.nnn:8010>, onde nnn.nnn.nnn.nnn é o endereço IP do nó.
2. Clique no ícone **Files** e selecione **Logical Files**.
3. Localize o arquivo para copiar na lista de arquivos e marque a caixa de seleção à esquerda.



4. Pressione o botão Copy e preencha as **informações** Group, Target Name e Options.

Destino:

Group Use a lista suspensa para selecionar o cluster Roxie para o qual será copiado.

Observação: Escolha apenas entre os clusters que estão no ambiente atual.

Target Name O nome do arquivo lógico a ser criado no destino. Este campo é preenchido automaticamente com base no arquivo lógico original, mas é possível alterá-lo.

Opções:

Overwrite Marque esta caixa para substituir arquivos de mesmo nome.

No Split Marque esta caixa para impedir a divisão de partes do arquivo em várias partes de destino.

Wrap Marque esta caixa para manter o mesmo número de partes e agrupar caso o cluster de destino seja menor do que o original.

Compress Marque esta caixa para compactar os arquivos.

Retain SuperFile Structure Marque esta caixa para conservar a estrutura de um superarquivo.

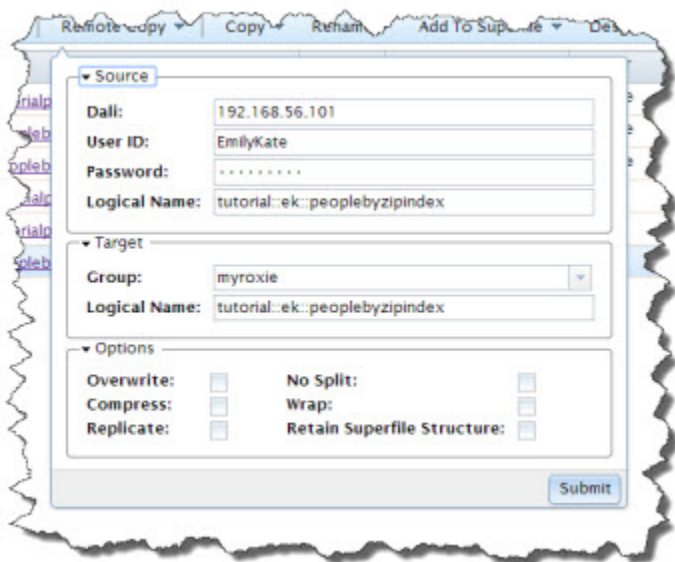
5. Pressione o botão **Copy**.

Uma nova guia é aberta, exibindo informações sobre o novo arquivo.

Cópia Remota

A cópia remota permite que você copie dados para um Roxie a partir de um cluster Roxie ou Thor fora do ambiente.

1. Abra a página da Web do ECLWatch. É possível acessar essa página usando o seguinte URL: `http://nnn.nnn.nnn.nnn:8010`, onde `nnn.nnn.nnn.nnn` é o endereço IP do nó.
2. Clique no ícone **Files** e selecione **Logical Files**.
3. Pressione o botão Cópia remota e preencha as informações.



Source:

| | |
|---------------|--|
| Dali | O IP ou nome do host do Dali para o ambiente que está sendo copiado. |
| ID do Usuário | O nome do usuário usado para ser autenticado no ambiente remoto. |
| Password | A senha que usada para ser autenticada no ambiente remoto. |
| Logical Name | O nome do arquivo lógico a ser copiado. |

Destino:

| | |
|-------|--|
| Group | Use a lista suspensa para selecionar o cluster Roxie para o qual será copiado. |
|-------|--|

Observação : Escolha apenas entre os clusters que estão no ambiente atual.

| | |
|--------------|---|
| Logical Name | O nome do arquivo lógico a ser copiado. |
|--------------|---|

Opções:

| | |
|----------------------------|--|
| Overwrite | Marque esta caixa para substituir arquivos de mesmo nome. |
| No Split | Marque esta caixa para impedir a divisão de partes do arquivo em várias partes de destino. |
| Wrap | Marque esta caixa para manter o mesmo número de partes e agrupar caso o cluster de destino seja menor do que o original. |
| Compress | Marque esta caixa para compactar os arquivos. |
| Retain SuperFile Structure | Marque esta caixa para conservar a estrutura de um superarquivo. |

4. Pressione o botão **Submit** .

Uma nova guia é aberta, exibindo informações sobre o novo arquivo.

Planejamento de Capacidade para Clusters Roxie

Planejamento de Capacidade

Os clusters Roxie são clusters de computação de alto desempenho baseado em disco (HPCC) que normalmente usam arquivos indexados. Um cluster é capaz de armazenar e manipular a mesma quantidade de dados contida em seu espaço em disco rígido combinado, porém, isso não produz um desempenho ideal.

Para maximizar o desempenho, é preciso configurar seu cluster de modo que os nós dos agentes desempenhem a maioria das jobs na memória.

Por exemplo, se uma consulta utiliza três arquivos de dados com um arquivo combinado de tamanho igual a 60 GB, um cluster de 40 canais está de bom tamanho, embora um de 60 canais provavelmente seja melhor.

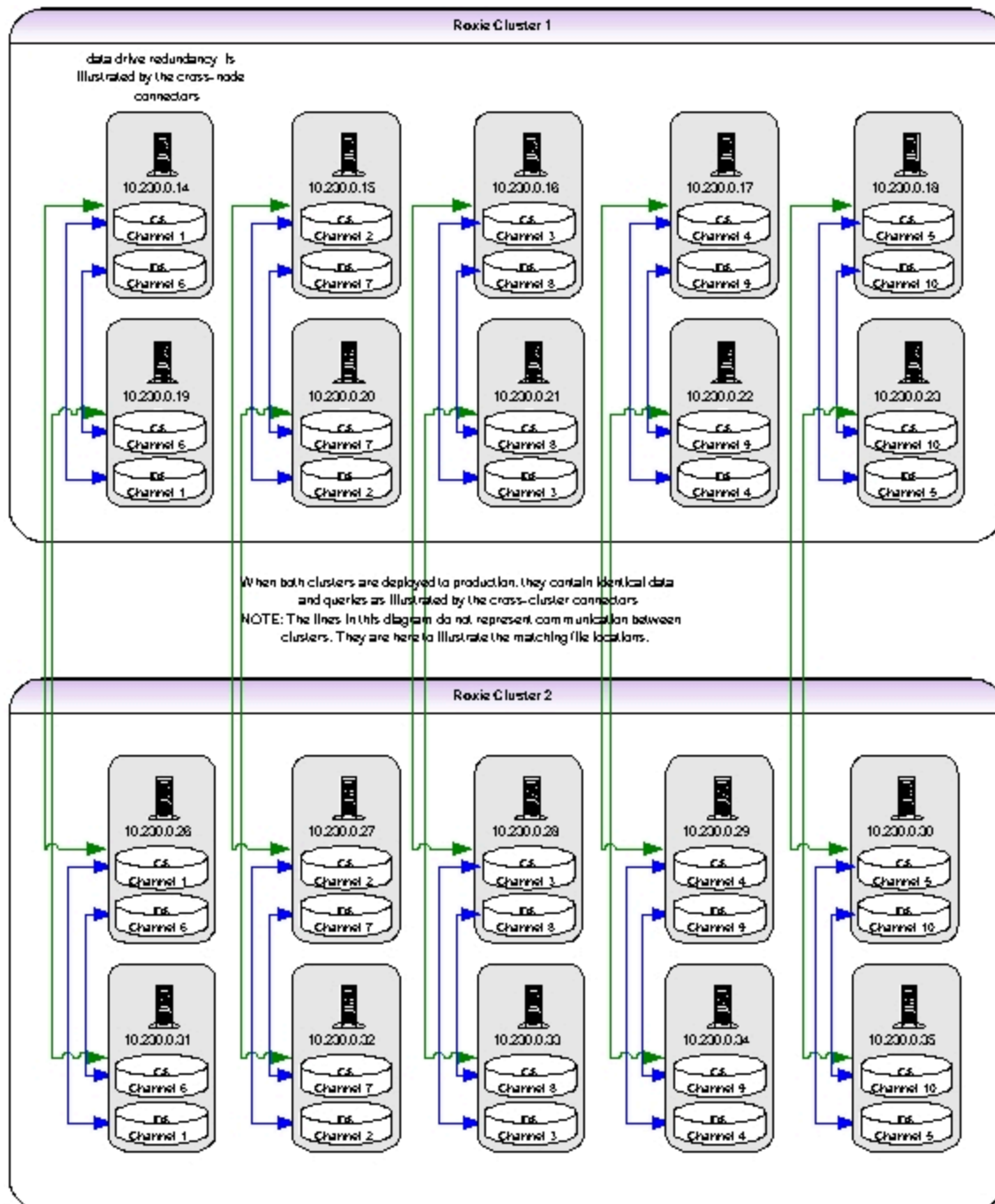
Outro ponto a considerar é o tamanho do cluster Thor que cria os arquivos de dados e que faz a indexação dos dados a serem carregados. Seu cluster Roxie de destino deve ter o mesmo tamanho do Thor, no qual os arquivos de dados e índices são criados ou um número que seja uniformemente divisível pelo tamanho do seu cluster Roxie. Por exemplo, um Thor de 100 nós para um Roxie de 20 nós seria considerado aceitável.

Isso se deve a forma na qual os dados são carregados e processados pelos agentes Roxie. Se os dados forem copiados para os nós dos agentes, as partes do arquivo são copiadas diretamente do local fonte para os locais de destino. Elas não são divididas ou redimensionadas para que caibam em um cluster com tamanho diferente. Consequentemente, ao carregar 50 partes de arquivo em um cluster de 40 canais, a parte um irá para o canal um, a parte dois para o canal dois e assim por diante. As partes 41 a 50 irão novamente para o topo, de modo que a parte 41 vá para o canal 1, a parte 42 para o canal 2, etc. O resultado será uma carga de trabalho desigualmente distribuída e que resultaria em menor desempenho. A velocidade de desempenho de um cluster é igual a de seu nó mais lento.

A consideração final é o número de processos do servidor em um cluster. Cada agente também deve ser um servidor, mas você pode dedicar nós adicionais para que sejam apenas processos do servidor. Isso é útil para consultas que exigem o processamento no servidor após os agentes terem retornado os resultados. Essas consultas intensivas de servidor poderiam ser enviadas apenas para endereços IP do servidor dedicado para que a carga seja removida dos nós atuando como servidor e agente.

Configurando os Canais

Na ilustração abaixo, os nós são configurados usando um esquema N+5 para compartilhar canais. Os canais podem ser configurados de várias formas, este é apenas um exemplo.



Nesta representação, cada chassi contém cinco blades de agentes Roxie (uma fila de servidores na imagem). Usaremos este exemplo para o restante deste manual.

Preflight

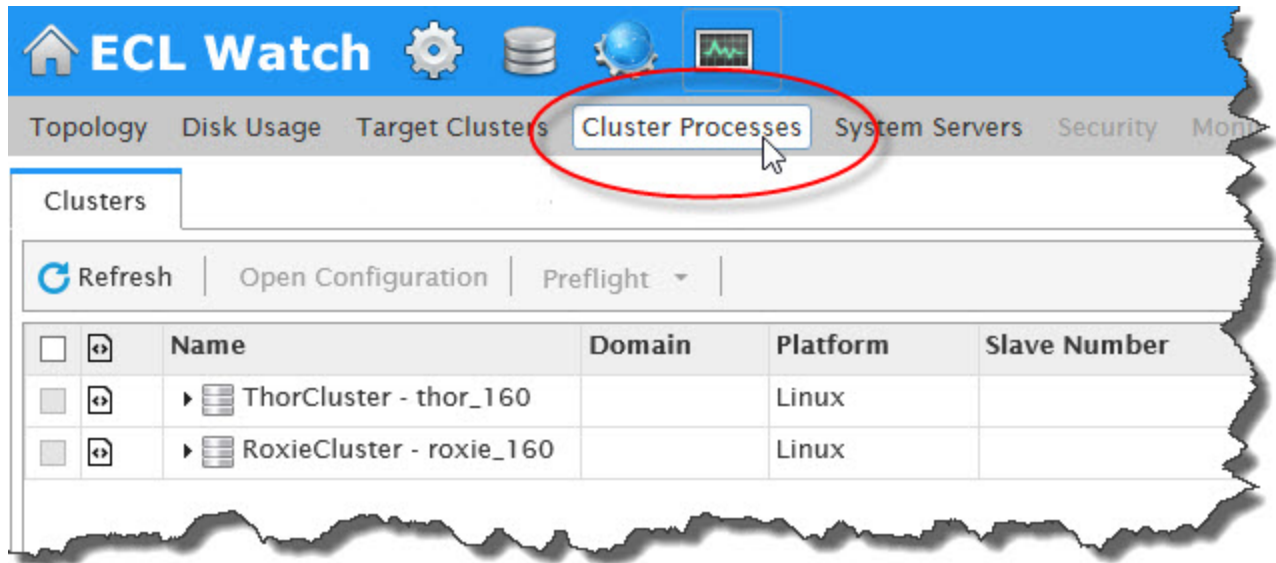
A seção Operations do ECL Watch oferece o recurso de realizar atividades Preflight. Esses utilitários preflight são usados para verificações de integridade diárias, além de evitar e solucionar problemas. O recurso oferece um local central para coleta de informações de hardware e software a partir de um conjunto remoto de máquinas, e possui várias vantagens para preparação de ambiente diário.

Essa seção contém informações para realizar verificações de preflight em um cluster Roxie. Para obter detalhes sobre outros componentes, consulte o *Guia do Administrador do HPCC System*.

Preflight do cluster Roxie

1. Clique no ícone **Operations** e clique no link **Clusters Processes**.

Figure 4. Link Cluster Processes



2. Expanda o cluster Roxie clicando na seta ao lado do link **RoxieCluster**.

Figure 5. Link RoxieCluster



3. Marque a caixa ao lado para selecionar os nós individualmente para análise, ou marque **Select All** na primeira linha.
4. Com sistema selecionado e o botão de preflight ativo, você pode exibir as opções de preflight.
5. Selecione ou desmarque qualquer uma das opções desejadas, então pressione o botão **Submit** para iniciar o preflight.

Figure 6. Submit

rs (legay

ion | Preflight ▾

Action: Machine Information ▾

Processor Information: ☒

Storage Information: ☒

Local File Systems Only: ☒

Get Software Information: ☒

Show Processes Using Filter: ☒

Additional Processes To Filter: Any Additional Processes To

Auto Refresh: ☒

Auto Refresh Increment: 5

Warn if CPU usage is over: 95

Warn if available memory is under: 95

Warn if available disk space is under: 95

% ▾

% ▾

Submit

1 - EclAgentProcess - eclagent node160101 Linux

RESULTADOS ESPERADOS

Depois que o botão Submit é pressionado, será exibida uma tela semelhante à mostrada abaixo.

Figure 7. Informações de sistema do Roxie

Clusters

Machine Information

Preflight Results

Refresh

| Location | Component | Computer Up Time | Physical |
|--|--------------------------|------------------|----------|
| 10.173.140.1 /var/lib/HPCCSystems/roxie_160/ | Roxie Server[node160001] | 4 days, 20:05 | |
| 10.173.140.2 /var/lib/HPCCSystems/roxie_160/ | Roxie Server[node160002] | 138 days, 21:37 | 31% |
| 10.173.140.3 /var/lib/HPCCSystems/roxie_160/ | Roxie Server[node160003] | 138 days, 21:37 | 31% |
| 10.173.140.4 /var/lib/HPCCSystems/roxie_160/ | Roxie Server[node160004] | 138 days, 21:37 | 31% |

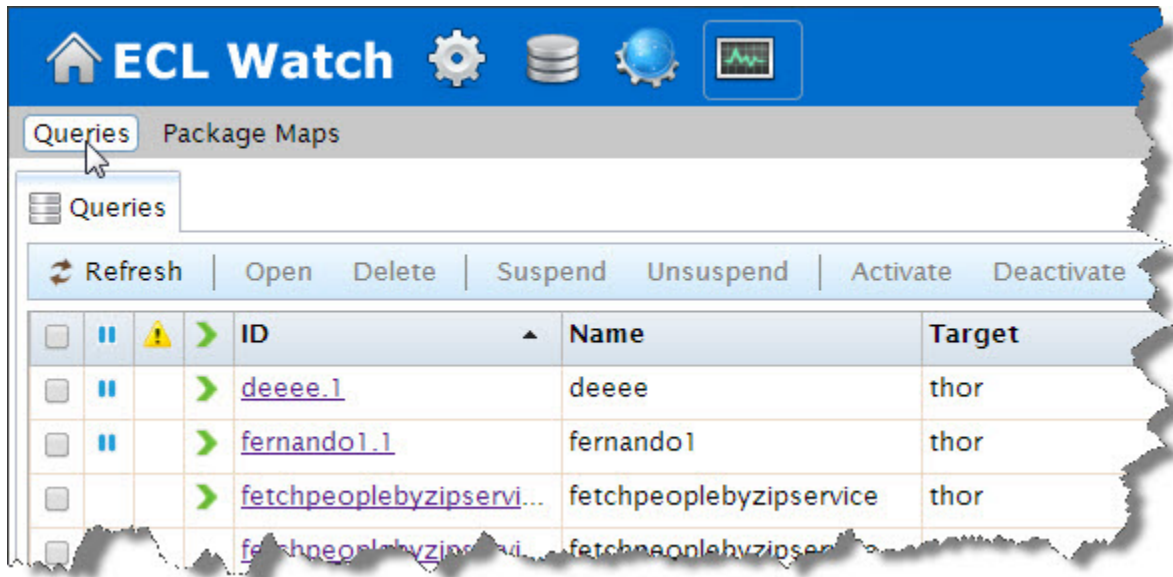
Indicam se os nós Roxie estão em execução e fornecem algumas informações adicionais sobre esses nós.

Se houver algum alerta pertinente, ele será destacado. Esses alertas exigem uma atenção adicional.




Página de Consultas no ECL Watch

A página Queries lista as consultas publicadas para cada cluster de destino. Nesta página, você pode ver as consultas publicadas. Você também pode executar algumas ações nas consultas selecionadas.

Figure 8. Procurar conjuntos de consulta



A interface “Queries” fornece algumas informações gerais; há três colunas ao lado esquerdo de cada consulta listada. Essas três colunas fornecem informações sobre cada consulta listada.

| | |
|---|---|
|  | Indica uma consulta em pausa |
|  | Indica uma consulta ativada |
|  | Indica uma consulta suspensa pelo sistema |

A página de consultas também fornece outras informações gerais sobre:

- o ID da consulta
- o nome da consulta
- o destino
- a ID da workunit (WUID)
- a dll
- Publicado por

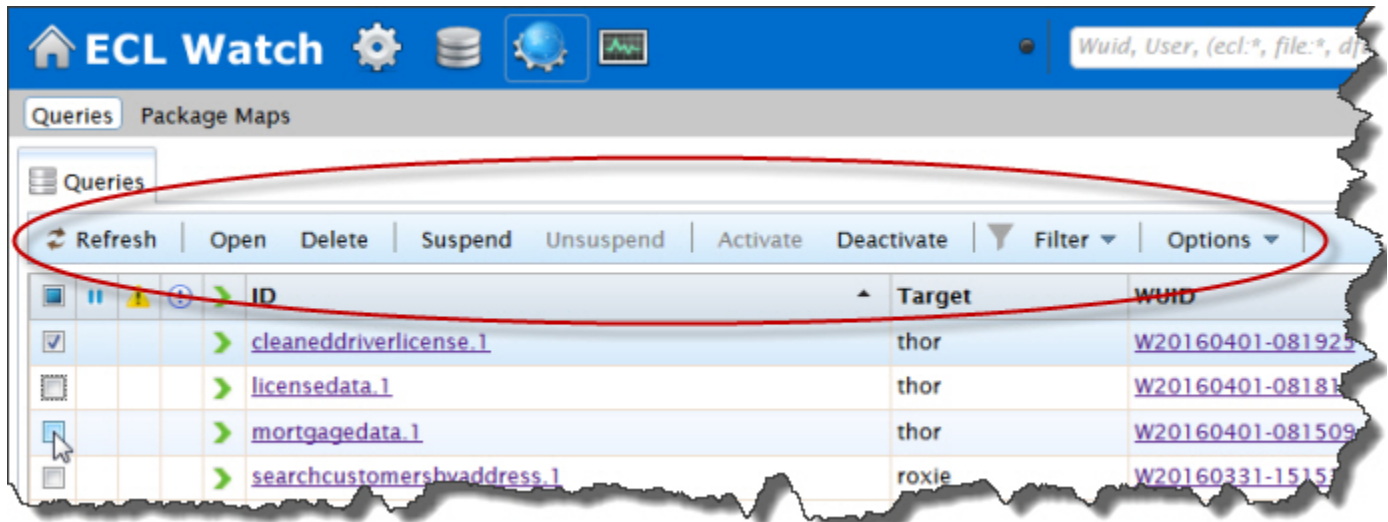
Todas as informações acima são disponibilizadas de forma resumida na página principal de consultas, porém é possível executar outras ações a partir dos botões de ação localizados no topo da aba. É possível classificar uma coluna clicando em seu título. Clique uma vez para classificar em ordem crescente e clique novamente para mudar para a ordem decrescente. A direção da seta indica a ordem da classificação.

É preciso selecionar uma consulta para ver a página detalhada ou para realizar uma ação. Você pode selecionar uma ou várias consultas clicando na caixa de seleção. Também é possível abrir uma consulta específica ao clicar duas vezes na consulta desejada.

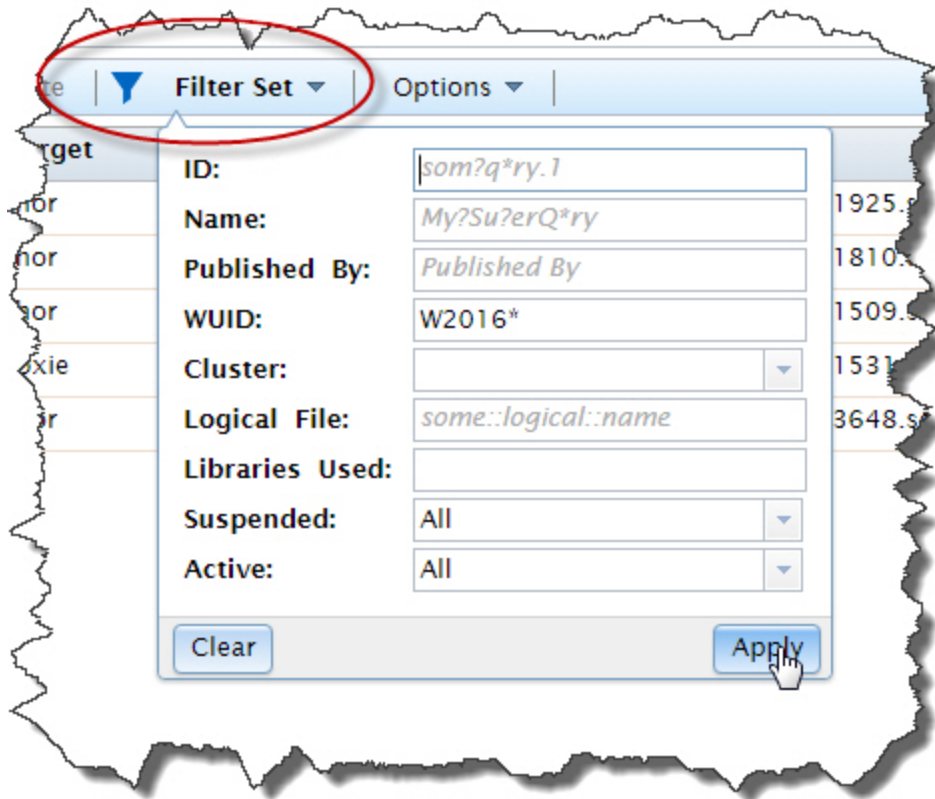
Aba Queries (Consultas)

Ao selecionar o hiperlink Publish Queries (Consultas publicadas), a aba Queries será aberta. Esta aba exibe as consultas publicadas no sistema. Os botões de ação permitem realizar operações nas consultas publicadas selecionadas.

Figure 9. Botões de ação das Consultas publicadas



| | |
|-------------------|--|
| Open | Abre a(s) consulta(s) selecionada(s). |
| Delete | Remove a(s) consulta(s) selecionada(s). |
| Suspend | Suspende a(s) consulta(s) selecionada(s). |
| Unsuspend | Reativa a(s) consulta(s) selecionada(s) suspensa(s). |
| Activate | Ativa a(s) consulta(s) selecionada(s). Essa função atribui uma consulta à sua réplica ativa de mesmo nome. |
| Deactivate | Desativa a(s) consulta(s) ativa(s) selecionada(s) removendo a réplica da consulta ativa de um dado grupo de consultas. |
| Filter | Permite filtrar consultas de acordo com os critérios inseridos. Quando o filtro é aplicado, o botão de ação exibe o Filter Set . O ícone sólido indica que as consultas publicadas exibidas são filtradas. Quando você define um filtro, um link para esse filtro é adicionado ao Recent Filters . The <i>Clique no botão New Page</i> em seguida, é exibido ao lado de Recent Filters, que abrirá uma nova página com esse conjunto de critérios de filtro. |



É possível filtrar diversos atributos de uma consulta. A filtragem pode ser feita por:

- ID
- Name (Nome)
- Priority (Prioridade)
- Publish by (Publicado por)
- WUID
- Cluster
- Logical File Name (Nome de arquivo lógico)
- Libraries Used (Bibliotecas usadas)
- Suspend (Consultas suspensas)
- Active (Consultas ativas).

A função Filter também é compatível com o uso de elementos coringa. Alguns atributos têm opções preenchidas que você pode selecionar em uma lista.

Options

Oferece a opção de busca ou exibição de consultas em um único nó ou em todos os nós. Usar esta opção pode melhorar o desempenho se você tiver um cluster maior com múltiplos nós.

| | |
|------------------------|---|
| Recreate Query | Recompila uma consulta em uma nova unidade de trabalho e publica novamente a nova workunit. Isso é útil ao atualizar para um novo compilador ECL e você deseja recompilar uma consulta da mesma fonte exata. O arquivo ECL deve estar disponível na workunit da consulta. |
| Published by me | Você pode checar a caixa <i>Published by me</i> para filtrar rapidamente somente consultas que você publicou. |

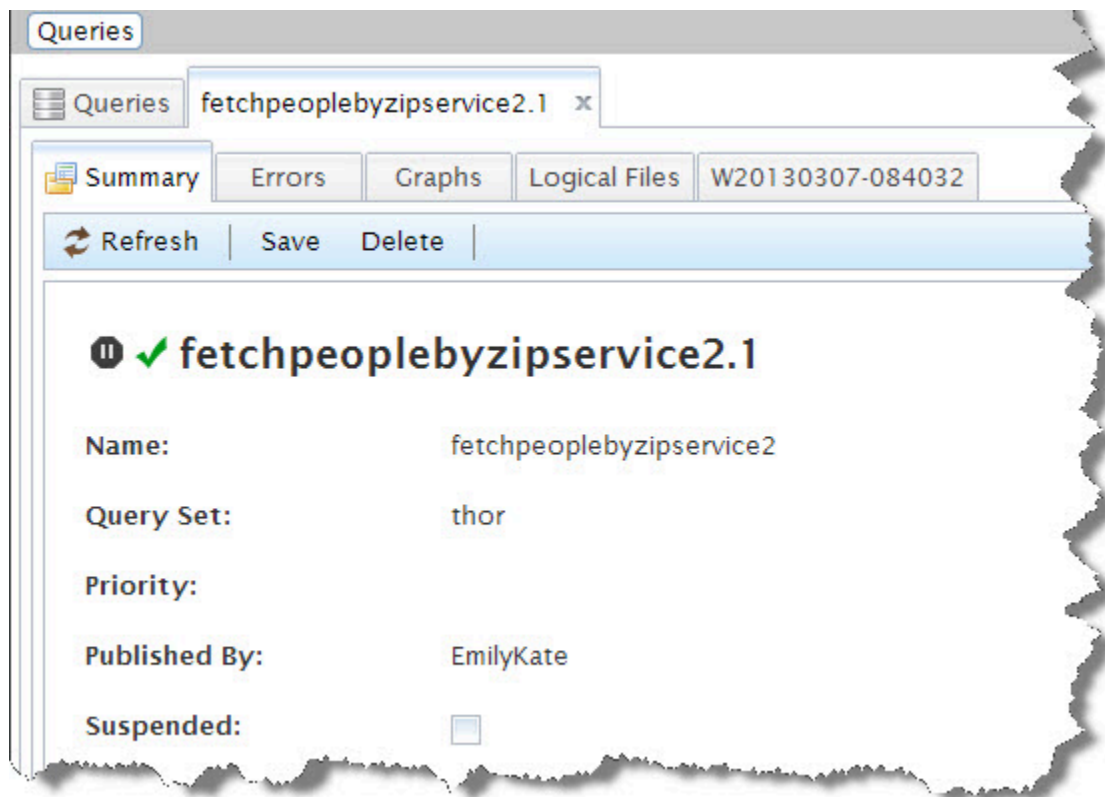
Detalhes da consulta

Para examinar a página Query Details, selecione e abra a(s) consulta(s). Esse procedimento abrirá uma aba que contém os detalhes da consulta. Na página Query Details, é possível obter mais informações sobre uma consulta específica. Você também pode executar algumas ações na consulta desejada. Há várias abas contendo informações adicionais sobre a consulta selecionada.

Aba Query Summary

A aba padrão da consulta, aberta ao selecionar uma consulta, é a aba Summary. A aba Summary exibe algumas informações detalhadas sobre a consulta.

Figure 10. Página Detalhes da consulta



Nesta aba é possível realizar algumas ações na consulta. Pressione os botões de ação para realizar a atividade desejada na consulta selecionada.

Refresh Atualiza as informações exibidas para a consulta selecionada.

Save Salva a(s) consulta(s) selecionada(s).

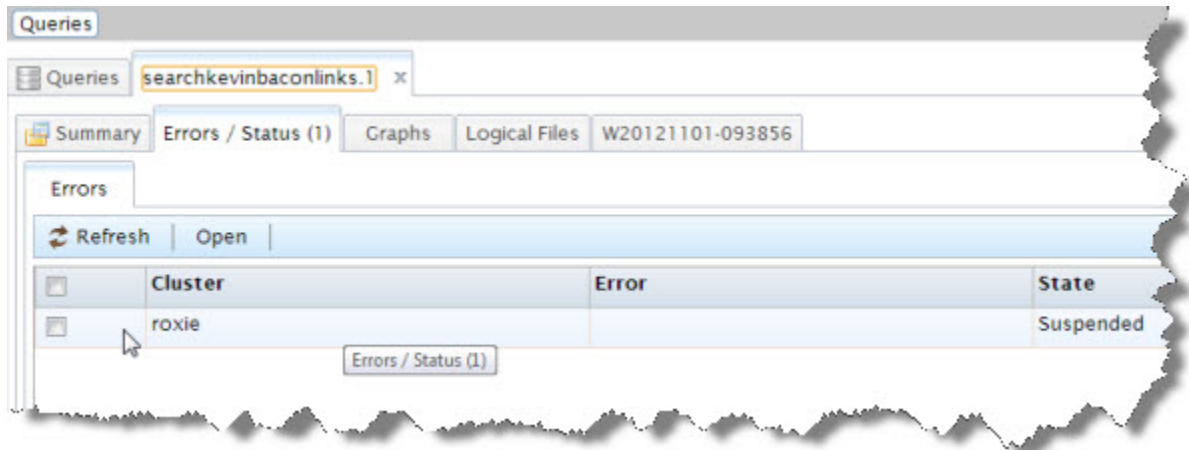
Delete Remove a(s) consulta(s) selecionada(s).

Aba Errors

Há uma aba "Erros" para cada consulta selecionada. A aba Erros exibe quaisquer erros que possam ter sido encontrados durante a compilação e a publicação daquela consulta. Se não houver erros, a aba de Erros estará em branco (vazia).

Em caso de erros, é possível analisar qualquer um específico de forma mais detalhada ao clicar na caixa de seleção, selecionar o erro desejado e pressionar o botão de ação Open. Você também pode clicar duas vezes sobre o erro selecionado.

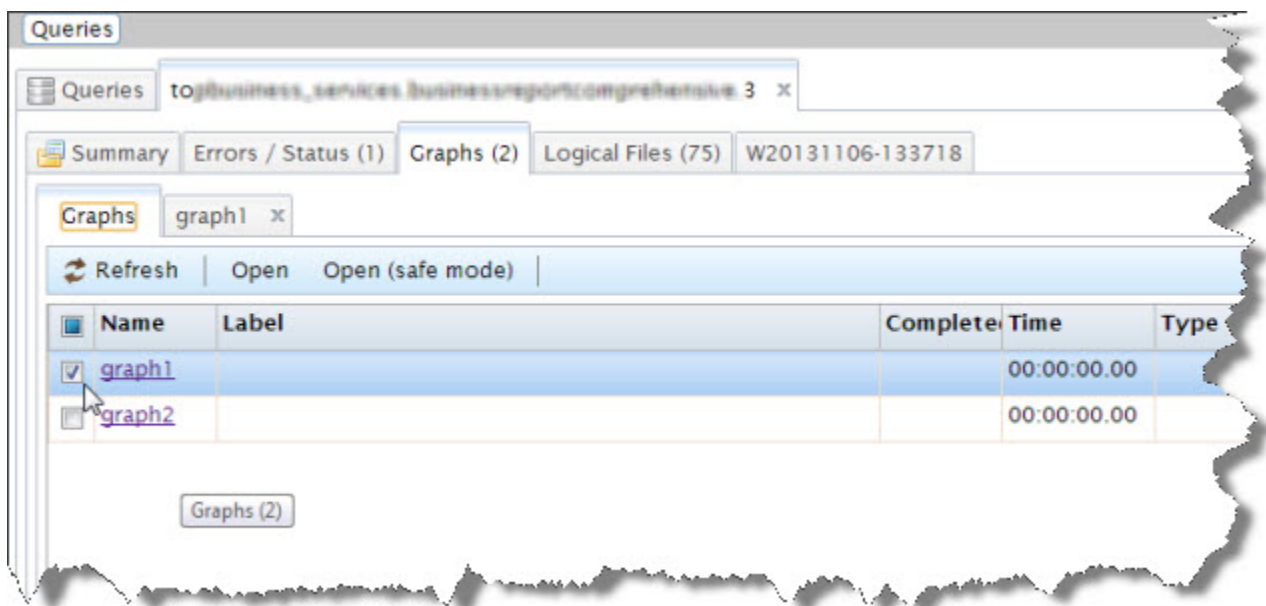
Figure 11. Erro de consulta



Aba Graphs

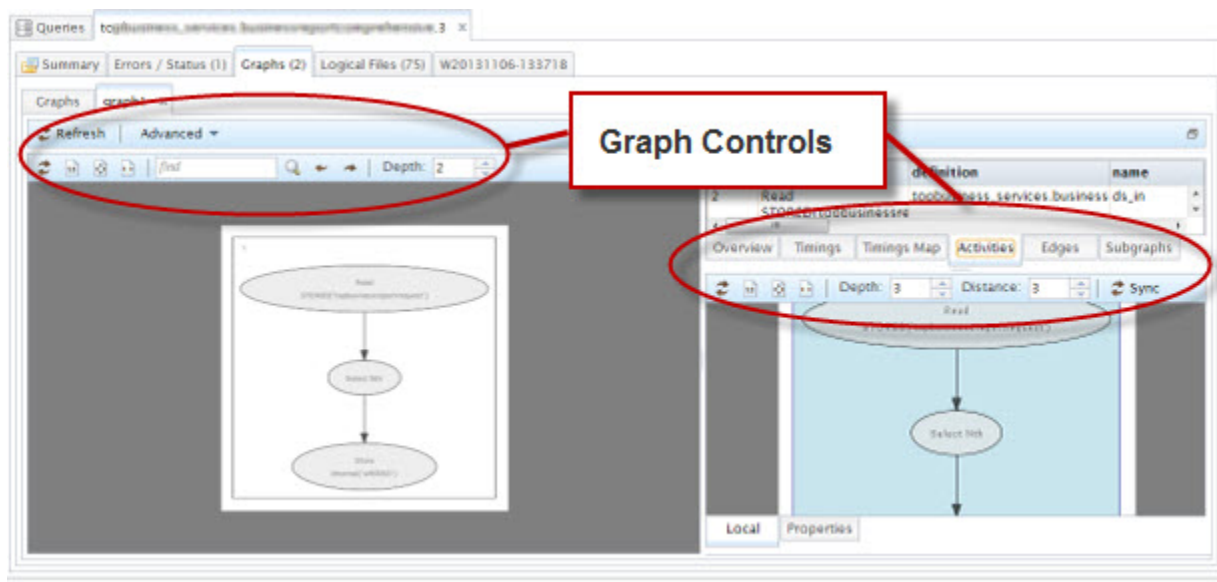
A aba Graphs oferece acesso as interpretações gráficas da consulta. Isso pode auxiliar na visualização sobre como a consulta foi executada. A aba Graphs exibe uma lista de todos os grafos gerados pela consulta selecionada, juntamente com algumas informações adicionais como medidas de tempo. Para exibir um gráfico específico, basta selecioná-lo e escolher a opção "Open" ou clicar duas vezes no gráfico listado.

Figure 12. Lista de Grafos



Ao abrir o gráfico, uma nova aba será aberta mostrando o(s) gráfico(s) selecionado(s).

Figure 13. Grafos



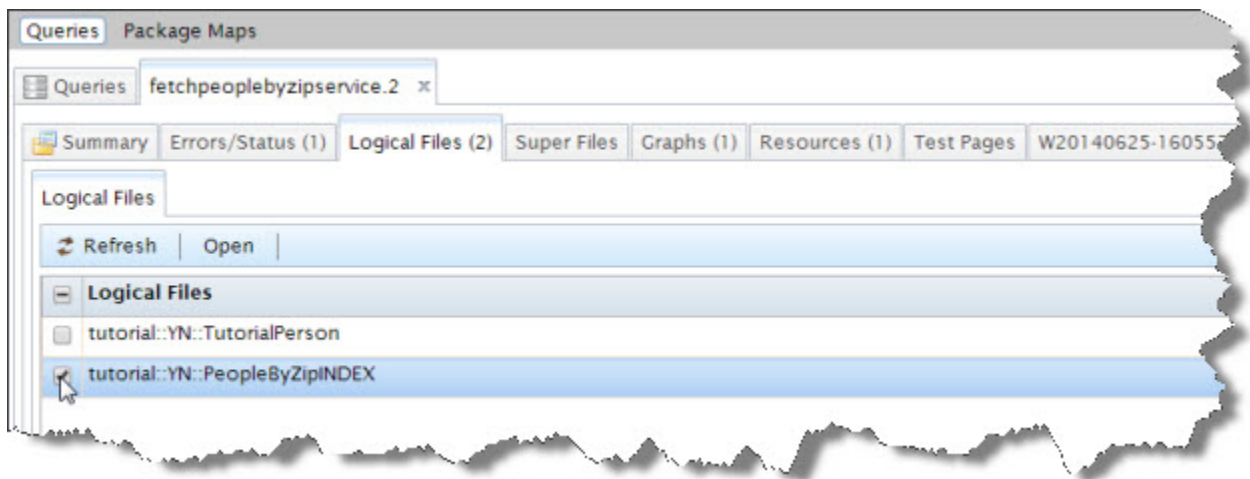
Ao abrir um gráfico, a imagem se divide em três seções menores com cada uma exibindo alguns componentes relevantes do gráfico da consulta. Observe a miríade dos controles de grafos e as abas na borda de cada aba. Use estes controles para visualizar os diferentes aspectos dos grafos.

O botão Advanced, localizado na área de controle principal do gráfico, oferece acesso a outras opções de representação gráfica ainda mais avançadas.

Aba Logical Files

A página de detalhes das consultas publicadas fornece um link para a aba dos arquivos lógicos da consulta. A aba "Logical Files" mostra todos os arquivos lógicos usados pela consulta. Para visualizar os detalhes dos arquivos lógicos de qualquer arquivo listado, selecione um ou mais arquivos marcando a caixa de seleção e pressionando o botão de ação Open. As abas de cada arquivo selecionado se abrirão e você poderá visualizar e fazer alterações nos arquivos sem precisar retornar à página de arquivos lógicos.

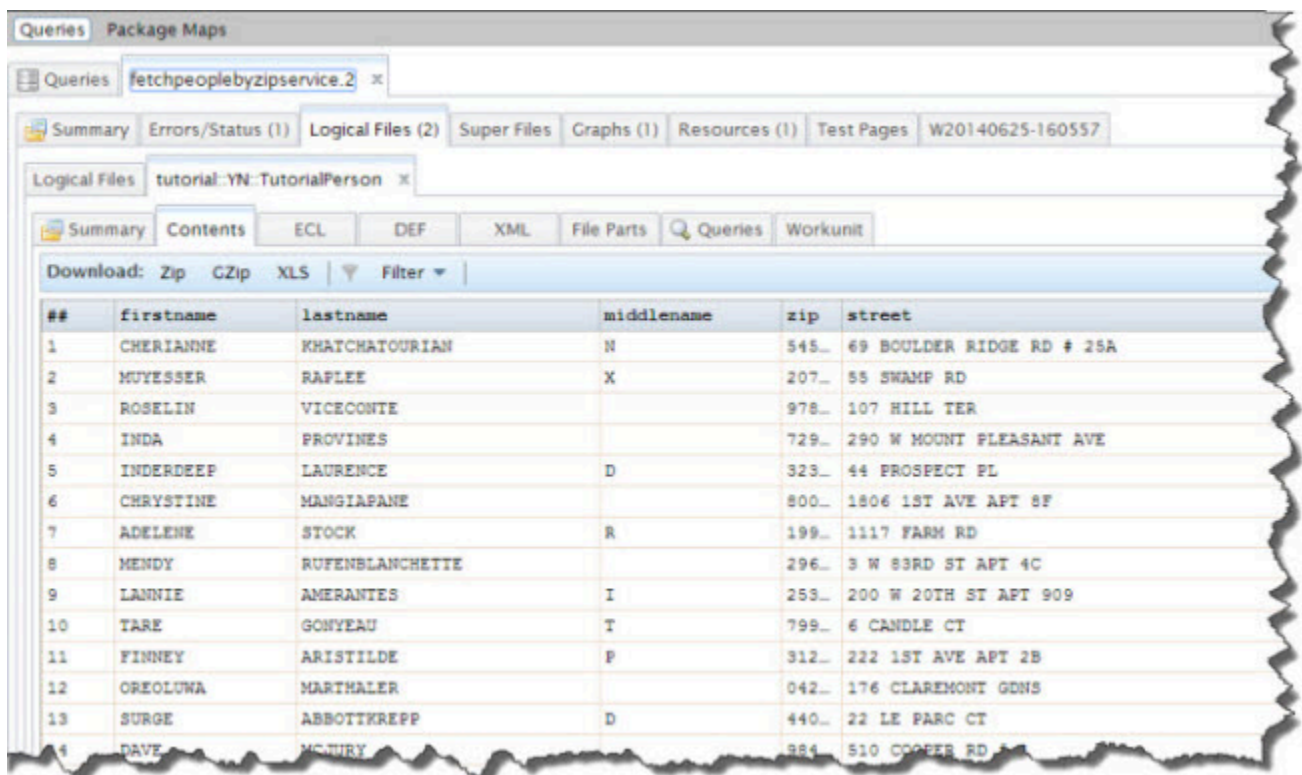
Figure 14. Aba Queries: Logical Files



A imagem acima mostra a lista de arquivos lógicos na aba Logical Files. Para ver mais detalhes sobre um arquivo lógico listado, marque a caixa próxima ao arquivo e pressione o botão **Open** fixo. Também é possível clicar duas vezes sobre o arquivo lógico que deseja visualizar.

Após aberto, é possível selecionar qualquer uma das abas que deseja visualizar: Summary, Contents, ECL, DEF, XML, File Parts, Queries, ou a Workunit.

Figure 15. Aba Queries:Logical Files:Contents

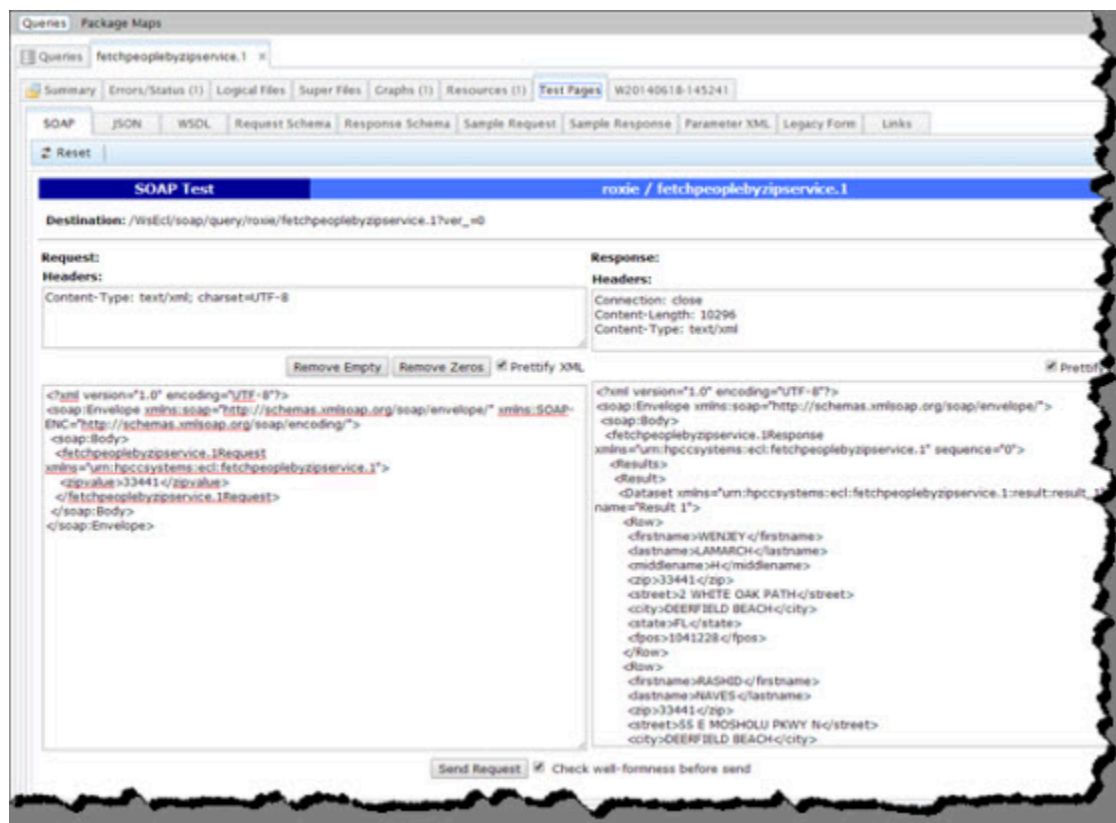


| ## | firstname | lastname | middlename | zip | street |
|----|-----------|-----------------|------------|--------|---------------------------|
| 1 | CHERIANNE | KHATCHATOURIAN | N | 545... | 69 BOULDER RIDGE RD # 25A |
| 2 | MUYESSER | RAFLEE | X | 207... | 55 SWAMP RD |
| 3 | ROSELIN | VICECONTE | | 978... | 107 HILL TER |
| 4 | INDA | PROVINES | | 729... | 290 W MOUNT PLEASANT AVE |
| 5 | INDERDEEP | LAURENCE | D | 323... | 44 PROSPECT PL |
| 6 | CHRISTINE | MANGIAPANE | | 800... | 1806 1ST AVE APT 8F |
| 7 | ADELENE | STOCK | R | 199... | 1117 FARM RD |
| 8 | MENDY | RUFENBLANCHETTE | | 296... | 3 W 83RD ST APT 4C |
| 9 | LANNIE | AMERANTES | I | 253... | 200 W 20TH ST APT 909 |
| 10 | TARE | GONYEAU | T | 799... | 6 CANDLE CT |
| 11 | FINNEY | ARISTILDE | P | 312... | 222 1ST AVE APT 2B |
| 12 | OREOLUWA | MARTHALER | | 042... | 176 CLAREMONT GDNS |
| 13 | SURGE | ABBOTTREPP | D | 440... | 22 LE PARC CT |
| 14 | DAVE | MCNURY | | 384... | 510 COOPER RD |

Aba Test Pages

A aba "Test Pages" oferece alguns recursos que podem ser usados para testar sua consulta, incluindo SOAP/JSON/WSDL e o formato antigo WS-ECL, assim como outras abas que exibem informações ou amostras úteis da consulta.

Figure 16. Aba Test Pages



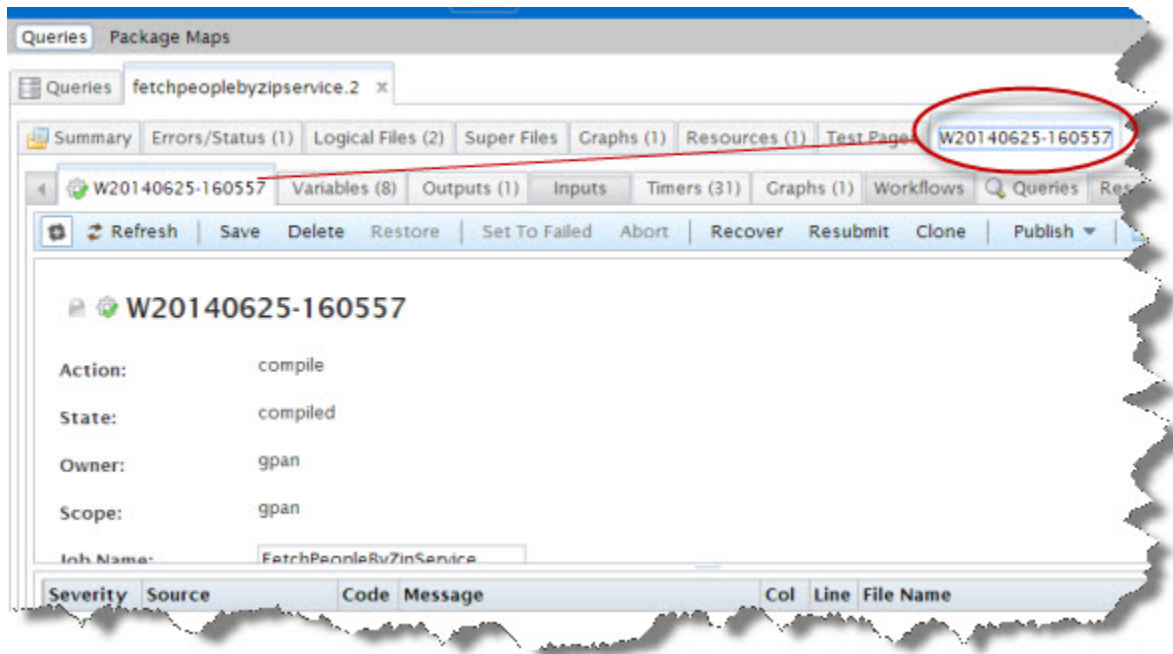
Informações disponíveis a partir da aba Test Page.

- **SOAP** Esta aba oferece uma interface interativa para o envio de consulta (com dados de entrada) e de visualização da resposta em formato XML.
- **JSON** Esta aba oferece uma interface interativa para o envio de consulta (com dados de entrada) e de visualização da resposta em formato JSON.
- **WSDL** Esta aba fornece uma definição de WSDL descrevendo a funcionalidade oferecida pela consulta (serviço Web).
- **Request Schema** Esta aba oferece um esquema em formato XSD que descreve uma solicitação para a consulta (serviço Web).
- **Response Schema** Esta aba oferece um esquema em formato XSD que descreve uma resposta para a consulta (serviço Web).
- **Sample Request** Esta aba fornece uma amostra de busca para a consulta (serviço Web) em formato XML.
- **Sample Response** Esta aba fornece uma amostra de resposta para a consulta (serviço Web) em formato XML.
- **Parameter XML** Esta aba fornece uma representação parametrizada em XML da interface da consulta.
- **Legacy Form** Esta aba fornece um formato que pode ser usado para enviar uma consulta e obter uma resposta. Isso é semelhante ao formato WsECL. Isso é semelhante ao formato WsECL.
- **Links** Fornece uma lista de links úteis como: o formato, uma amostra de URL REST, uma amostra de busca, uma amostra de resposta, o parâmetro XML, SOAP POST, WSDL, XSD e o esquema de resultado.

Workunits

A página Query Details das consultas publicadas fornece um link para página de workunit. Esta aba corresponde a um atalho que o levará para a mesma aba de workunit acessada através do menu Workunits do ECL.

Figure 17. Consultas Workunit



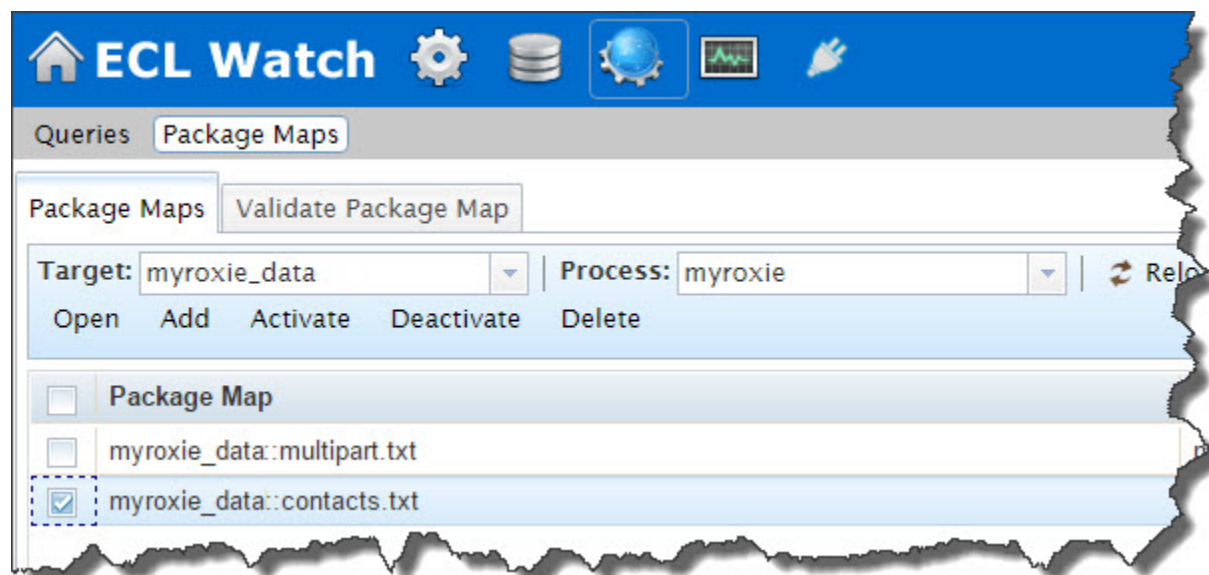
É idêntica a página **Workunits do ECL** do submenu de navegação Workunit. Aqui você pode realizar as mesmas operações. Observe que aqui também há algumas abas conhecidas – como por exemplo a aba Graphs – tanto da página Query Details da consulta como da aba workunit aqui aninhada.

Package Maps

O package maps fornece uma referência ao conteúdo de uma superchave usada em consultas que substitui a definição original. O mapeamento de arquivos do package map pode ser organizado em uma coleção de arquivos definindo alguns subconjuntos de consultas, ou organizados por vários agrupamentos tais como funções, arquivos, desenvolvedores, etc. Estes subconjuntos são denominados **parts (partes)**. Para obter mais informações sobre os package map consulte o aba *Roxie Reference*.

Você pode acessar a página de package map a partir do link do ícone "Queries". Pressione o botão **Package Maps** na barra de navegação do submenu para acessar os Package Maps em seu cluster.

Figure 18. Package Maps



A página Package Maps mostra todos os package maps carregados em seu cluster. Você pode Adicionar (Add), Ativar (Activate), Desativar (Deactivate), Remover (Delete) ou Abrir (Open) um package map. Para ver um package map, selecione um package map na lista.

Para atualizar os mapas do pacote de dados que estão sendo usados, é possível editar o arquivo do package map ou adicionar um novo e depois ativá-lo. É possível remover o arquivo antigo mais tarde.

Ações do Package Map

É possível executar ações em seus mapas do pacote de dados a partir da aba "Package Maps" no ECL Watch.

Abrir Package Map

Para visualizar um package map, selecione o mapa desejado e pressione o botão **Open**. Isso abrirá uma nova aba onde você pode acessar abas adicionais com informações sobre o pacote, o XML, e validar o mapas do pacote de dados.

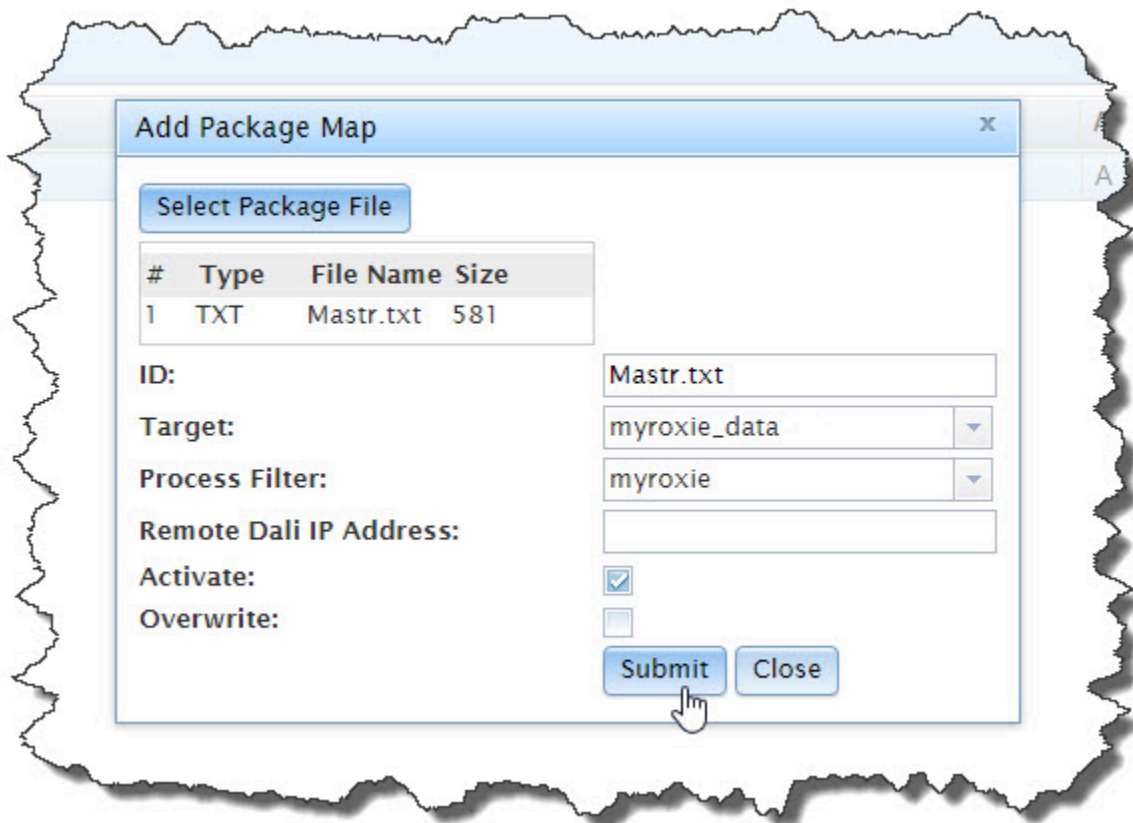
Adicionar Package Map

Para adicionar um package map ao clusters de destino:

1. Selecione um package map a ser adicionado marcando a caixa de seleção próxima a ele.

2. Pressione o botão **Add** e abra a caixa "Add Package Map"

Figure 19. Adicionar Package Map



3. Pressione o botão **Select Package File** e selecione os arquivos do pacote que deseja adicionar.
4. Selecione o **Target** para o qual deseja associar o mapeamento.
5. Selecione um **Process Filter** a partir da lista suspensa. O filtro de processo determina em qual cluster Roxie físico o package map será carregado.
6. Insira o endereço IP ou nome do host do Dali remoto que será usado para pesquisar o arquivo lógico no campo **Remote Dali IP Address**.
7. Marque as caixas para Ativar(Activate) ou Substituir (Overwrite), como desejado.

Ativar o Package Map

Pressione o botão **Activate** para desativar o Package Map de dados ativo no momento e para ativar o mapa selecionado.

Desativar o Package Map

Pressione o botão **Deactivate** para desativar o mapa do pacote de dados ativo no momento.

Remover Package Map

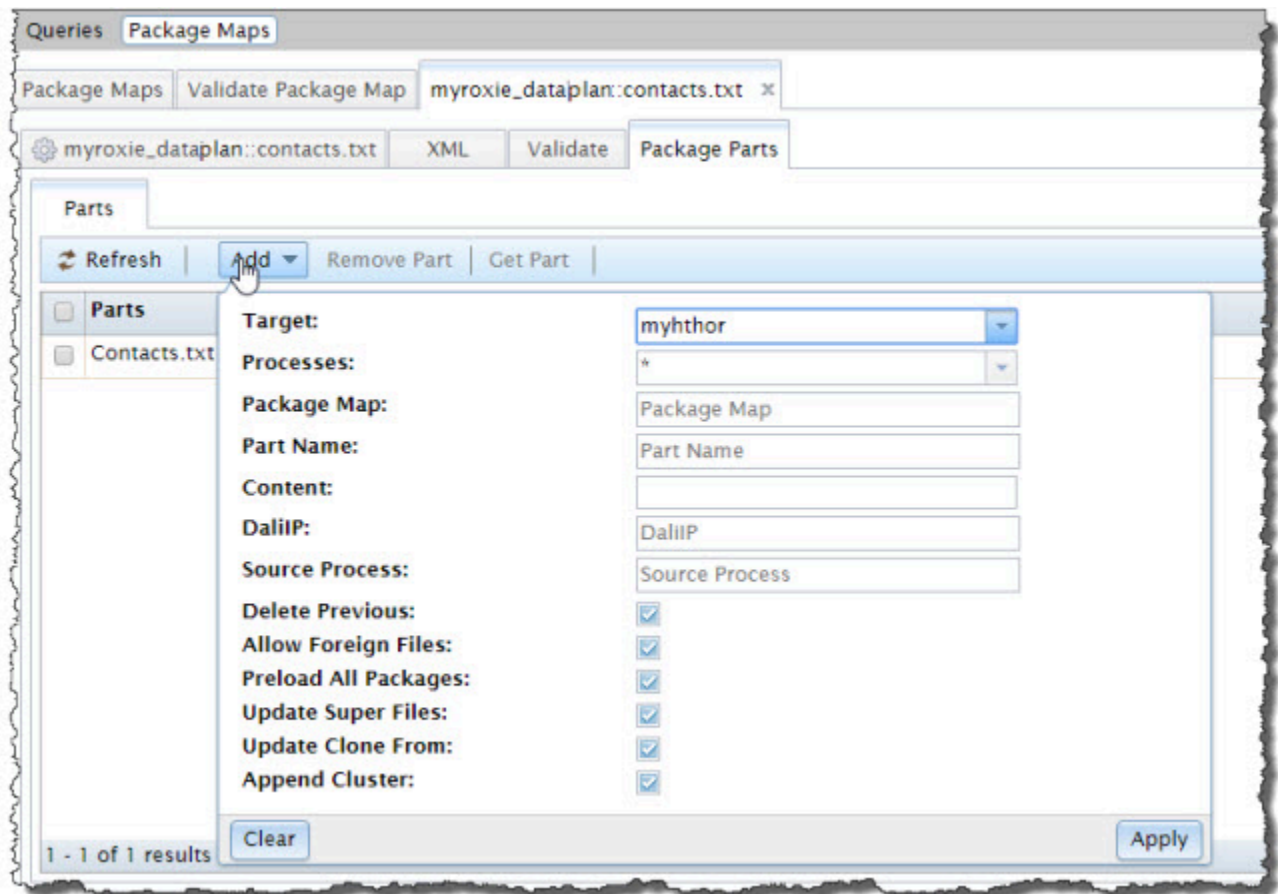
Para remover um package map:

1. Selecione o pacote a ser removido marcando a caixa de seleção próxima a ele.
2. Pressione o botão **Delete**.
3. Pressione **OK** ao solicitar a confirmação da ação.

Partes do Package Map

É possível ver mais informações e realizar algumas ações nas partes do package map. Abra o package map para visualizar a aba package maps.

Figure 20. Partes do pacote



Através desta interface no ECLWatch é possível examinar partes individuais, adicionar ou removê-las.

Adicionar Parte

Para adicionar uma parte ao mapa do pacote de dados:

1. Selecione **a aba** Package Parts.
2. Pressione o botão **Add**.
3. Preencha os campos com as informações adequadas.
4. Pressione **App**.

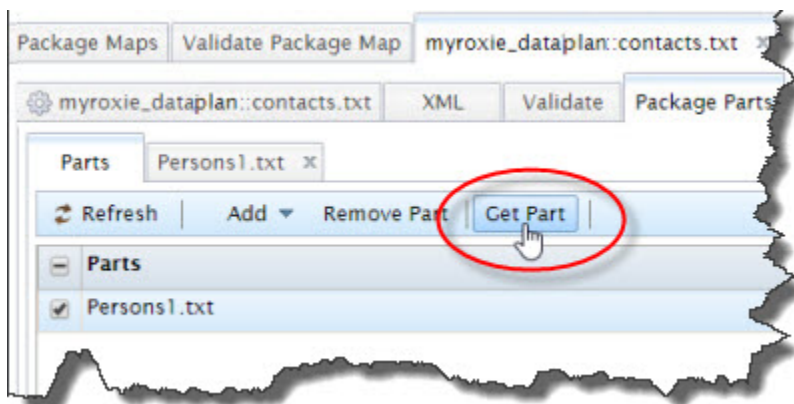
Remover Parte

Para remover uma parte do package map:

1. Selecione a **aba** Package Parts.
2. Marque a caixa de seleção próxima à parte que será removida.
3. Pressione o botão **Remove Part**.
4. Pressione **OK** ao ter a confirmação solicitada.

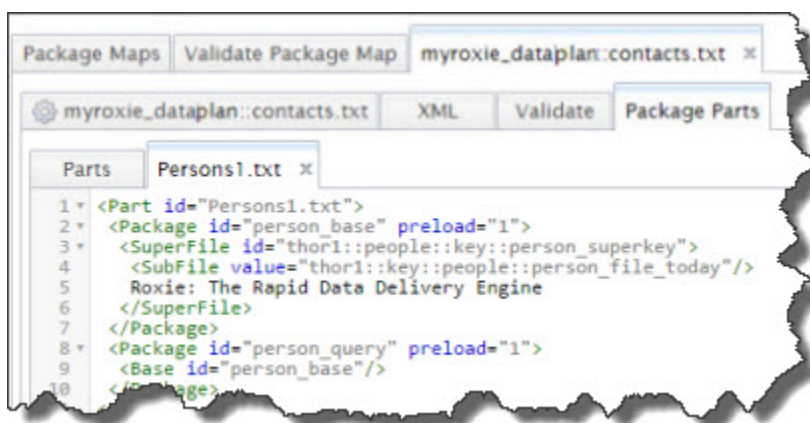
Obter parte

Figure 21. Obter parte



Pressione o botão **Get Part** para visualizar o conteúdo da parte selecionada.

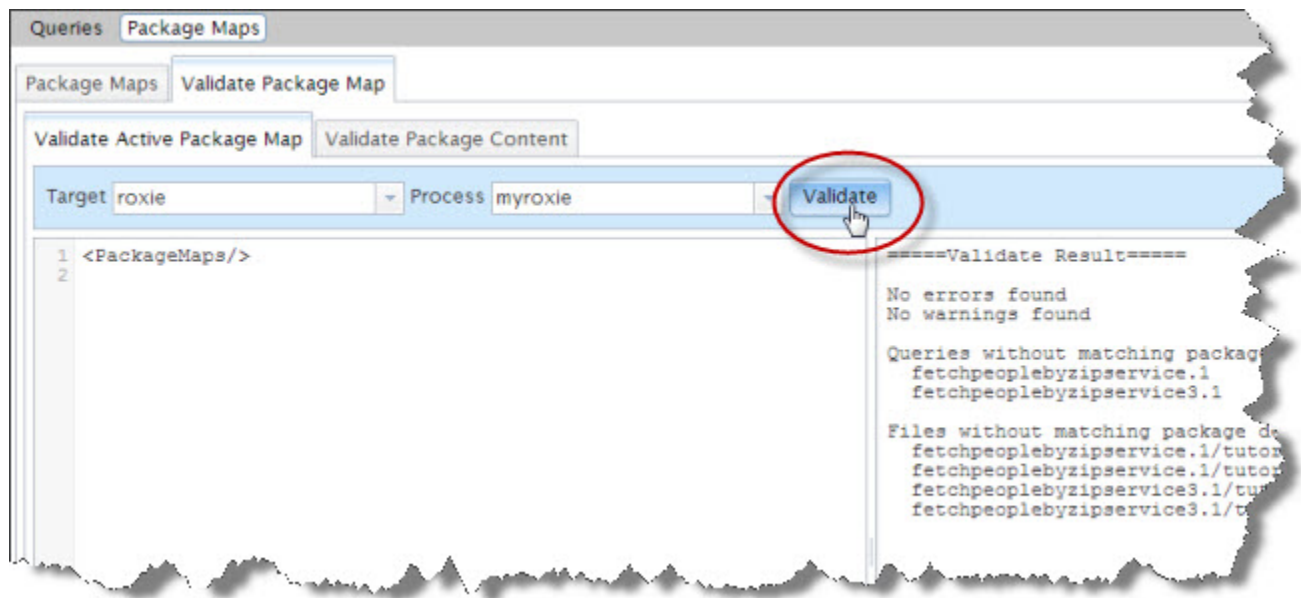
Figure 22. Conteúdo do Package Part



Validar o Package Map

A **aba** Validate Package Map é utilizada para validar package maps ativos. A **aba** Validate Package Content é utilizada para validar o conteúdo do package map que ainda não foi carregado. Para validar um package map ativo:

Figure 23. Validar Package Maps



1. Selecione a aba **Validate Package Map**.
2. Selecione o **Target** e o **Process** a partir das listas suspensas na aba **Validate Package Map**.
3. Pressione o botão **Validate** para validar o package map.

O resultado será exibido na aba **Validate Active Package Map**.

É possível validar qualquer package map ativo, inativo, externo ou aqueles que ainda não foram carregados no ambiente.

Para validar um package map externo:

1. Vá para a aba **Package Maps**.
2. Selecione o package map a ser validado.
3. Pressione o botão **Open**.
4. Selecione a aba **Validate**.

A aba **Validate Package Content** permite que você abra qualquer arquivo de package map ou insira qualquer conteúdo do pacote no formato para validá-lo. O conteúdo não precisa ser publicado no sistema.