

Dynamic ESDL

Equipe de documentação de Boca Raton



Dynamic ESDL

Equipe de documentação de Boca Raton

Copyright © 2024 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.

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.

2024 Version 9.8.94-1

Dynamic ESDL	4
Tutorial de Fluxo de Trabalho	5
Visão geral	5
Configurar e Conectar o ESDL Service	6
Escrever a Definição do Serviço ESDL	11
Gerar as definições ECL a partir das definições do Serviço ESDL	12
Publicar as Definições de Serviço ESDL e Conectar o Serviço ESDL	13
Interface de Comando do ESDL	14
Sintaxe de comando ESDL	14

Dynamic ESDL

A Dynamic ESDL é uma metodologia que ajuda você a desenvolver e gerenciar interfaces de consulta baseadas na Web de forma rápida e consistente.

A Dynamic ESDL adota uma abordagem de desenvolvimento "interface-first". Ela aproveita a linguagem ESDL para criar um contrato de interface comum de modo que ambos os desenvolvedores de consultas Roxie e de interface Web possam aderir. O objetivo é permitir que os desenvolvedores criem web-services de produção com interfaces limpas que capazes de evoluir e crescer ao longo do tempo sem interromper as aplicações existentes.

O suporte de versões embutidas do ESDL ajuda a garantir que aplicações implementadas e compiladas continuem a operar mesmo com mudanças na interface do serviço implementado para novas funcionalidades.

A capacidade do ESDL de definir e reutilizar estruturas comuns ajuda a manter as interfaces consistentes entre os métodos.

O serviço Dynamic ESDL foi desenvolvido para escalar horizontalmente, e webhooks são fornecidos para adicionar processos de registro (logging) e segurança personalizados para ajudar a criar web services totalmente "prontos para produção".

Depois que um serviço é implementado, desenvolvedores de aplicações e usuários finais podem utilizá-lo usando REST, JSON, XML, SOAP ou postagens codificadas por formulário. O Dynamic ESDL oferece acesso rápido e fácil a um WSDL, formulários dinâmicos, amostras de busca e de respostas, e interfaces de testes para permitir que os desenvolvedores testem as mudanças lógicas, as alterações de dados ou os novos recursos, além de interagirem com o serviço diretamente usando SOAP, XML ou JSON.

Tutorial de Fluxo de Trabalho

Visão geral

Nesta seção vamos:

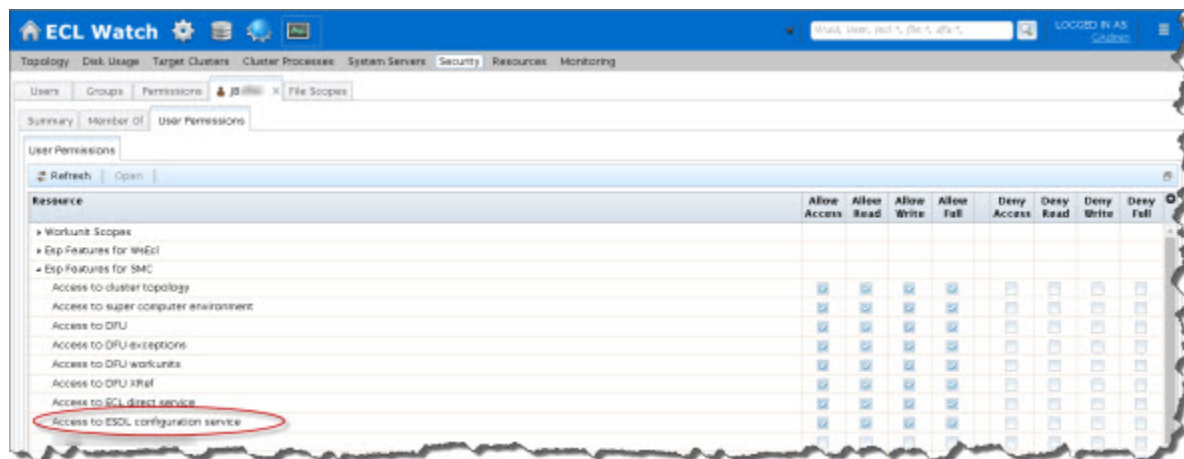
- Usar o Configuration Manager (Gerenciador de Configurações) para adicionar um ESP Service baseado em Dynamic ESDL e vinculá-lo a uma porta em um servidor ESP
- Criar uma definição do ESDL Service usando o editor no ECL IDE.
- Gerar um ECL a partir da definição de ESDL Service no ECL IDE. Esta etapa gera automaticamente um arquivo ECL no repositório ECL. Você usará as definições neste arquivo ECL ao gravar a consulta ECL que apresentará o resultado (a lógica de negócios).
- Compilar a consulta da lógica ECL e depois publicá-la em um cluster Roxie.

Neste ponto, você pode testar a consulta usando o WsECL.

- Publicar a definição de Dynamic ESDL do ECL IDE.
- Vincular os métodos de serviço às consultas Roxie usando uma configuração formatada em XML.

Segurança DESDL e LDAP

Se sua plataforma HPCC está configurada para usar a segurança LDAP, é necessário assegurar que todos os usuários que publicarão definições ESDL tenham acesso direto ao **ESDL configuration service** definido para **Allow Full**, como mostrado abaixo.



Configurar e Conectar o ESDL Service

Nesta parte do tutorial, vamos adicionar um serviço ESP e uma ligação de serviço que reserva uma porta para o serviço Dynamic ESDL. Essa etapa não depende do desenvolvimento e da publicação da consulta Roxie real, de forma que você pode configurá-la antes ou depois que a consulta estiver pronta.

1. Caso esteja em execução, pare o HPCC System usando este comando em uma janela de terminal:

Centos/Red Hat

```
sudo /sbin/service hpcc-init stop
```

Ubuntu

```
sudo service hpcc-init stop
```

Debian 6 (Squeeze)

```
sudo /etc/init.d/hpcc-init stop
```



Este comando pode ser usado para confirmar que os processos do HPCC foram interrompidos (Centos/Red Hat).

```
sudo /sbin/service hpcc-init status
```

Para Ubuntu

```
sudo service hpcc-init status
```

Debian 6 (Squeeze)

```
sudo /etc/init.d/hpcc-init status
```

2. Inicie o serviço do Gerenciador de Configurações.

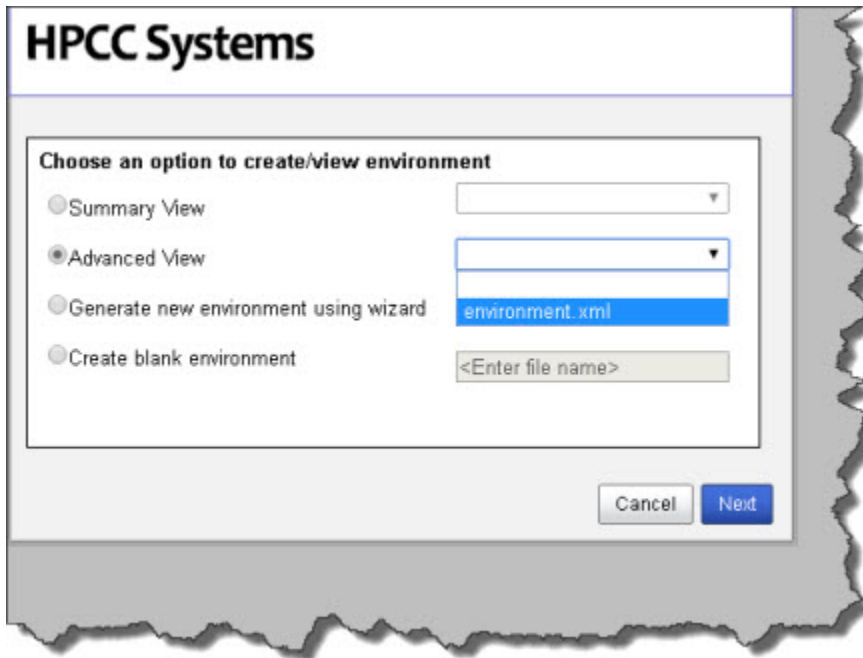
```
sudo /opt/HPCCSystems/sbin/configmgr
```

3. Utilizando um navegador de Internet, acesse a interface do Configuration Manager:

```
http://<node ip>:8015
```

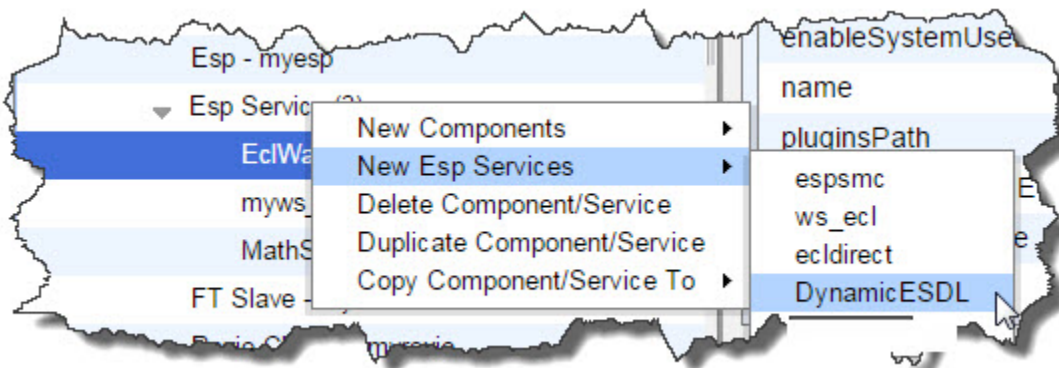
O assistente de inicialização do Configuration Manager é exibido.

4. Selecione **Advanced View** e depois selecione o arquivo XML do ambiente de origem para editar.

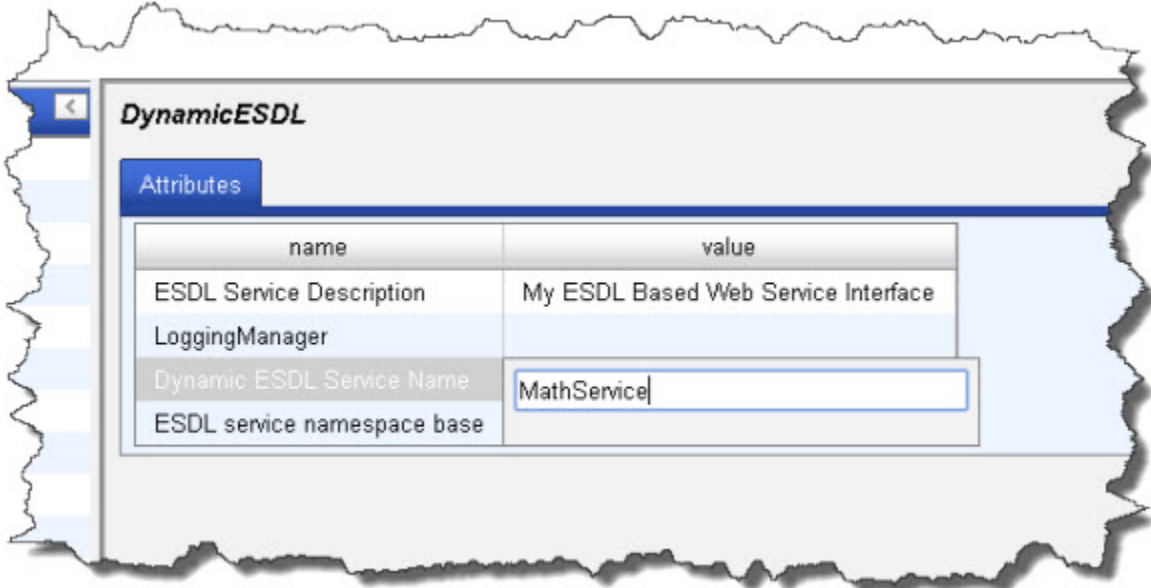


5. Pressione o botão **Next** .

6. Clique com o botão direito em **Esp Services** e selecione **Dynamic ESDL** em **New ESP Services**.



7. Forneça um nome ao arquivo do ambiente.

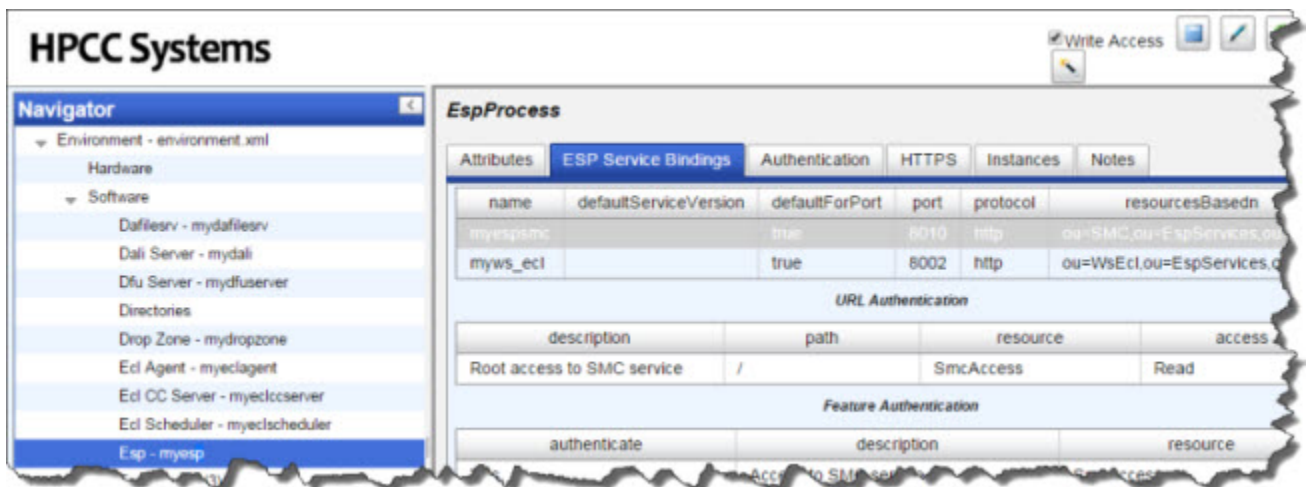


DynamicESDL

Attributes

name	value
ESDL Service Description	My ESDL Based Web Service Interface
LoggingManager	
Dynamic ESDL Service Name	MathService
ESDL service namespace base	

8. Selecione seu ESP, e depois selecione a guia ESP Service Bindings.



HPCC Systems

Write Access

Navigator

- Environment - environment.xml
 - Hardware
 - Software
 - Daflesrv - mydaflesrv
 - Dali Server - mydali
 - Dfu Server - mydfuserver
 - Directories
 - Drop Zone - mydropzone
 - Ecl Agent - myeclagent
 - Ecl CC Server - myeclccserver
 - Ecl Scheduler - myeclscheduler
 - Esp - myesp**

EspProcess

Attributes | **ESP Service Bindings** | Authentication | HTTPS | Instances | Notes

name	defaultServiceVersion	defaultForPort	port	protocol	resourcesBasedn
myespsmc		true	8010	http	ou=SMC,ou=EspServices,ou=
myws_ecl		true	8002	http	ou=WsEcl,ou=EspServices,ou=

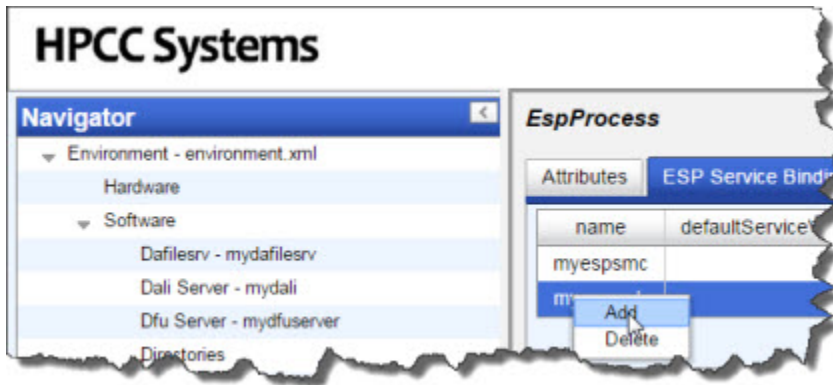
URL Authentication

description	path	resource	access
Root access to SMC service	/	SmcAccess	Read

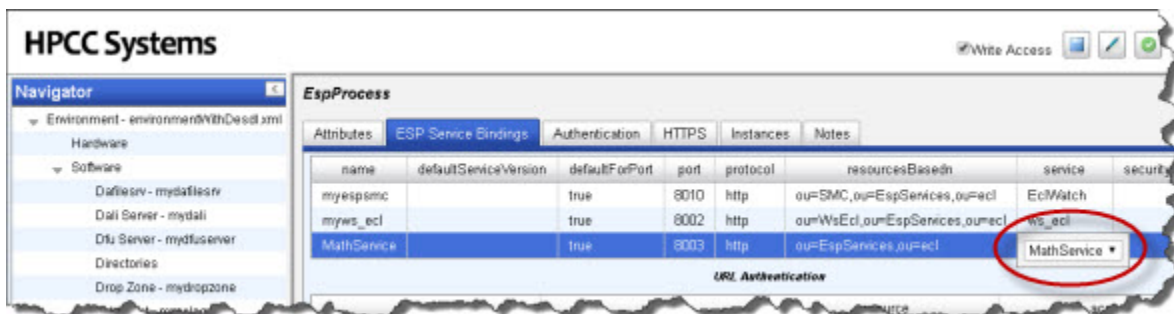
Feature Authentication

authenticate	description	resource
Access to SMC service	Root access to SMC service	SmcAccess

9. Clique com o botão direito na lista de ligações e selecione **Add**.



10. Informe um nome, porta e selecione o serviço na lista suspensa. A definição de serviço que você adicionou será exibida na lista de serviços disponíveis.



11. Copie o novo arquivo NewEnvironment.xml do diretório de origem para /etc/HPCCSystems e renomeie o arquivo para environment.xml

```
# for example
sudo cp /etc/HPCCSystems/source/NewEnvironment.xml /etc/HPCCSystems/environment.xml
```



Certifique-se de que você possua direitos suficientes para gravar arquivos no diretório de destino antes de copiar. Se solicitado a substituir o arquivo de destino, responda **yes**.

12. Copie o arquivo **/etc/HPCCSystems/environment.xml** para **/etc/HPCCSystems/** em **cada** nó.

Você pode criar um script para forçar o arquivo XML para todos os nós. Um script de amostra é fornecido com o HPCC. Os comandos a seguir copiam os arquivos XML para todos os nós como exigido:

```
sudo /opt/HPCCSystems/sbin/hpcc-push.sh <sourcefile> <destinationfile>
```

13 Reinicie o HPCC System em **todos os** nós. O comando a seguir inicia o HPCC System em um nó individual:

Centos/Red Hat

```
sudo /sbin/service hpcc-init start
```

Ubuntu

```
sudo service hpcc-init start
```

Debian 6 (Squeeze)

```
sudo /etc/init.d/hpcc-init start
```



Você pode criar um script para forçar esse arquivo para todos os nós. Um script de amostra é fornecido com o HPCC. Use o comando a seguir para iniciar o HPCC em todos os nós.

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a hpcc-init start
```

Escrever a Definição do Serviço ESDL

Nesta parte do tutorial, vamos gravar as Definições de serviço no ECL IDE. A listagem de programa abaixo mostra um serviço ESDL chamado *MathService*. Ele contém um método, *AddThis*, com uma solicitação e uma resposta definida.

```
ESPservice [auth_feature("None")] MathService
{
    ESPmethod AddThis(AddThisRequest, AddThisResponse);
};

ESPrequest AddThisRequest
{
    int FirstNumber;
    int SecondNumber;
};

ESPresponse AddThisResponse
{
    int Answer;
};
```

1. Salve o arquivo como MathService.ecm.

Gerar as definições ECL a partir das definições do Serviço ESDL

Nesta seção vamos gerar um ECL a partir do arquivo de definição do serviço ESDL. Isso usará o executável esdl instalado com as ferramentas de cliente.

Você encontrará isso em C:\Program Files (x86)\HPCCSystems\5.2.0\clienttools\bin em uma máquina Windows

ou /opt/HPCCSystems/bin/ em uma máquina Linux.

1. Na linha de comando, execute:

```
esdl ecl MathService.ecm .
```

Ao Salvar foi gerado automaticamente um arquivo com o nome MathService.ecl no diretório atual.

```
/** Not to be hand edited (changes will be lost on re-generation) */  
/** ECL Interface generated by esdl2ecl version 1.0 from MathService.xml. */  
/*=====*/  
  
export MathService := MODULE  
  
export t_AddThisRequest := record  
  integer FirstNumber {xpath('FirstNumber')};  
  integer SecondNumber {xpath('SecondNumber')};  
end;  
  
export t_AddThisResponse := record  
  integer Answer {xpath('Answer')};  
end;  
end;  
  
/** Not to be hand edited (changes will be lost on re-generation) */  
/** ECL Interface generated by esdl2ecl version 1.0 from MathService.xml. */  
/*=====*/
```

2. Copie o arquivo MathService.ecl para um módulo no seu Repositório ECL. Por exemplo, "myMathService."

3. Escreva ECL para dar suporte ao serviço:

```
IMPORT MathService;  
rec_in := MathService.MathService.t_AddThisRequest;  
  
First_Row := ROW ([], rec_in) : STORED ('AddThisRequest', FEW);  
  
res:= first_row.FirstNumber + first_row.SecondNumber;  
ds_out := ROW ({res},MathService.MathService.t_AddThisResponse);  
OUTPUT(ds_out, NAMED('AddThisResponse'));
```

4. Compile e Publique no seu cluster Roxie.

5. Teste o serviço usando o WsECL:

```
http://<node ip >:8002
```

Publicar as Definições de Serviço ESDL e Conectar o Serviço ESDL

Nesta parte do tutorial, vamos publicar as definições de ESDL Service no armazém de dados do sistema e ligar os métodos à consulta Roxie publicada.

1. Publicar a definição de Dynamic ESDL do ECL IDE.

```
esdl publish MathService MathService.ecm -s nnn.nnn.nnn.nnn -p 8010 --version 1
```

Substitua nnn.nnn.nnn.nnn pelo endereço IP do seu servidor ECL Watch ESP.

2. Crie o arquivo xml de configuração

```
<Methods>
  <Method name="AddThis" url="<RoxieIPRange>:9876" querytype="roxie" queryname="AddThis">
    <!--Optional Method Context Information start-->
    <Gateways>
      <Gateway name="mygateway" url="1.1.1.1:2222/someservice/somemethod/>
      <Gateway name="anothergateway" url="2.2.2.2:9999/someservice/somemethod/>
    </Gateways>
    <!--Optional Method Context Information end-->
  </Method>
</Methods>
```

Onde nome é o nome do(s) seu(s) método(s) e url é o endereço IP e porta do servidor Roxie, finalmente, o queryname é o nome publicado (alias) da consulta. Para um Roxie de vários nós, você pode usar um intervalo na forma de nnn.nnn.nnn.n-nnn. O ESP agenda pedidos para o destino Roxie usando lógica round-robin.

Opicionalmente, seu método pode incluir informações de contexto conforme ilustrado no exemplo a seguir. As informações de contexto devem ser formatadas de forma que possam ser consumidas pela consulta ECL de destino. O HPCC DESDL ESP não impõe nenhuma restrição nas informações de contexto transmitidas no arquivo de configuração, desde que seja um XML válido.

3. Save o arquivo como MathSvcCfg.xml
4. Vincular os métodos de serviço às consultas Roxie usando uma configuração formatada em XML.

```
esdl bind-service myesp 8003 MathService.1 MathService --config
MathSvcCfg.xml -s nnn.nnn.nnn.nnn -p 8010
```

Onde myesp é o nome do seu processo ESP, 8003 é a porta que você reservou para seu serviço Dynamic ESDL.

5. Teste o serviço usando a nova interface:

```
http://<node ip >:8003
```

Interface de Comando do ESDL

Sintaxe de comando ESDL

esdl [--version] <command> [<options>]

<i>--version</i>	exibe informações de versão.
<i>help</i> <command>	mostra ajuda para um comando específico.
xml	Gerar XML da definição de ESDL.
ecl	Gera um ECL a partir da definição ESDL.
xsd	Gera um XSD a partir da definição ESDL.
wsdl	Gera um WSDL a partir da definição ESDL.
publish	Publica definição ESDL para usar em ESP .
list-definitions	Listar todas as definições de ESDL.
delete	Remove definição ESDL.
bind-service	Configura serviço baseado em ESDL no ESP de destino (com ligação ESP existente).
list-bindings	Lista todas as ligações ESDL.
unbind-service	Remove serviço baseado em ESDL com ligação no ESP de destino.
bind-method	Configura método associado com ligação ESDL existente.
unbind-method	Remove método de uma ligação ESDL em um ESP de destino.
get-binding	Obtém a ligação ESDL.

esdl xml

esdl xml [options] filename.ecm [<outdir>]

<i>filename.ecm</i>	O arquivo que contém as definições ESDL
<i>-r --recursive</i>	Processa todas as inclusões
<i>-I, --include-path</i>	Locais para procurar arquivos ESDL incluídos. Eles podem ser caminhos absolutos ou relativos. Se precisar especificar vários diretórios, você pode usar várias opções <i>-I</i> ou usar uma única entrada com os diretórios separados pelo caractere separador de ambiente. Para Linux, use dois pontos (:) e para Windows, use ponto e vírgula (;). Os caminhos também podem ser definidos usando uma variável de ambiente--ESDL_INCLUDE_PATH.
<i>-v --verbose</i>	Exibe informações detalhadas
<i>-?/-h/--help</i>	Mostra a página de uso
Output	(srcdir <outdir>)/filename.xml

Gera XML a partir da definição ESDL. Esse XML é uma entidade intermediária usada pelo mecanismo ESDL para criar as definições de serviço de tempo de execução. Este comando é raramente usado por si próprio.

Exemplos:

```
esdl xml MathService.ecm .
```

esdl ecl

esdl ecl sourcePath outputPath [options].

<i>sourcePath</i>	O caminho absoluto do arquivo da definição ESDL que contém a definição EsdlService para o serviço.
<i>outputPath</i>	O caminho absoluto para o local onde o resultado ECL será gravado.
<i>-x, --expandedxml</i>	Saída de arquivos XML expandidos.
<i>--includes</i>	Se estiver presente, processa todos os arquivos incluídos
<i>--rollup</i>	Se estiver presente, acumula todas as inclusões processadas em um único arquivo de resultados do ECL.
<i>-cde</i>	Especifica o diretório de arquivos do componente do HPCC (localização dos arquivos xslt).
<i>--ecl-imports</i>	Lista de importação, delimitada por vírgula, a ser anexada ao resultado do ECL. Cada entrada gera um demonstrativo IMPORT correspondente.
<i>--ecl-header</i>	Texto a ser incluído no cabeçalho ou no arquivo de destino (gerado) – deve ser um ECL válido.
<i>-I, --include-path</i>	Locais para procurar arquivos ESDL incluídos. Eles podem ser caminhos absolutos ou relativos. Se precisar especificar vários diretórios, você pode usar várias opções -I ou usar uma única entrada com os diretórios separados pelo caractere separador de ambiente. Para Linux, use dois pontos (:) e para Windows, use ponto e vírgula (;). Os caminhos também podem ser definidos usando uma variável de ambiente--ESDL_INCLUDE_PATH.
Output	(sourcePath outputPath>)/filename.ecl

Isso gera estruturas do ECL a partir da definição ESDL. Essas estruturas criam a interface (pontos de entrada e de saída) da consulta Roxie.

Exemplos:

```
esdl ecl MathService.ecm.
```


esdl xsd

esdl xsd sourcePath serviceName [options]

<i>sourcePath</i>	O caminho absoluto do arquivo da definição ESDL que contém a definição EsdlService para o serviço.
<i>serviceName</i>	Nome do serviço ESDL definido em um dado arquivo ESDL.
<i>--version <version number></i>	Restringe à versão de interface
<i>--method <method name>[;<method name>]*</i>	Restringe à lista de método(s) específico(s)
<i>--xslt <xslt file path></i>	Caminho para o arquivo '/xslt/esxdl2xsd.xslt' para transformar EsdlDef em XSD
<i>--preprocess-output <raw output directory></i>	Resultados pré-processados do arquivo XML para um diretório especificado antes de aplicar a transformação XSLT
<i>--annotate <all none></i>	Sinalizador orientado para todas as anotações ou para nenhuma. Por padrão, as anotações são geradas para Enumerações. Definir o sinalizador para "nenhum" também desabilitará essa função. Definir para "todas" habilita anotações adicionais, como recolhido, cols, form_ui, html_head e linhas.
<i>--noopt</i>	Desativa a imposição de atributos "opcionais" nos elementos. Se -noopt não for especificado, todos os elementos contendo "opcional" serão incluídos nos resultados. Por padrão, o filtro "opcional" está ativado.
<i>-opt,--optional <param value></i>	Valor a ser usado para o filtro de tag opcional ao coletar dependências. Por exemplo, especificar "interno" quando alguns objetos da definição ESDL apresentam o atributo opcional ("interno") assegura que eles aparecerão no XSD, caso contrário eles seriam filtrados.
<i>-tns,--target-namespace <target namespace></i>	O namespace de destino foi especificado para transformação através do parâmetro 'tnsParam' usado no resultado final do XSD.
<i>-n <int></i>	Número de vezes que a transformação é executada após carregar o XSLT. O padrão é 1
<i>--show-inheritance</i>	Desabilita o recurso recolher. O recolhimento otimiza o resultado XML removendo estruturas usadas apenas para herança, e recolhe seus elementos na forma de elementos secundários. Isso simplifica o estilo da página. Por padrão, essa opção está ativada.
<i>--no-arrayof</i>	Elimina o uso do elemento arrayOf. O arrayOf otimiza o resultado XML para incluir definições da estrutura 'ArrayOf...' para os elementos EsdlArray que não possuem o atributo item_tag. Trabalha conjuntamente com um estilo de página otimizado que não é capaz de gerar essas definições sozinho. Por padrão, esse recurso está habilitado.
<i>-v --verbose</i>	Exibe informações detalhadas
<i>-?/-h/--help</i>	mostra página de uso
Output	(srcdir <outdir>)/filename.ecf

Gera XSD a partir da definição ESDL.

Exemplos:

```
esdl xsd MathService.ecm MathService
```

esdl wsdl

esdl wsdl sourcePath serviceName [options]

<i>sourcePath</i>	O caminho absoluto do arquivo da definição ESDL que contém a definição EsdlService para o serviço.
<i>serviceName</i>	Nome do serviço ESDL definido em um dado arquivo ESDL.
<i>--version <version number></i>	Restringe à versão de interface
<i>--method <method name>[;<method name>]*</i>	Restringe à lista de método(s) específico(s)
<i>--xslt <xslt file path></i>	Caminho para o arquivo '/xslt/esxdl2xsd.xslt' para transformar EsdlDef em XSD
<i>--preprocess-output <raw output directory></i>	Resultados pré-processados do arquivo XML para um diretório especificado antes de aplicar a transformação XSLT
<i>--annotate <all none></i>	Sinalizador orientado para todas as anotações ou para nenhuma. Por padrão, as anotações são geradas para Enumerações. Definir o sinalizador para "nenhum" também desabilitará essa função. Definir para "todas" habilita anotações adicionais, como recolhido, cols, form_ui, html_head e linhas.
<i>--noopt</i>	Desativa a imposição de atributos "opcionais" nos elementos. Se -noopt não for especificado, todos os elementos contendo "opcional" serão incluídos nos resultados. Por padrão, o filtro "opcional" está ativado.
<i>-opt,--optional <param value></i>	Valor a ser usado para o filtro de tag opcional ao coletar dependências. Por exemplo, especificar "interno" quando alguns objetos da definição ESDL apresentam o atributo opcional ("interno") assegura que eles aparecerão no XSD, caso contrário eles seriam filtrados.
<i>-tns,--target-namespace <target namespace></i>	O namespace de destino foi especificado para transformação através do parâmetro 'tnsParam' usado no resultado final do XSD.
<i>-n <int></i>	Número de vezes que a transformação é executada após carregar o XSLT. O padrão é 1
<i>--show-inheritance</i>	Desabilita o recurso recolher. O recolhimento otimiza o resultado XML removendo estruturas usadas apenas para herança, e recolhe seus elementos na forma de elementos secundários. Isso simplifica o estilo da página. Por padrão, essa opção está ativada.
<i>--no-arrayof</i>	Elimina o uso do elemento arrayOf. O arrayOf otimiza o resultado XML para incluir definições da estrutura 'ArrayOf...' para os elementos EsdlArray que não possuem o atributo item_tag. Trabalha conjuntamente com um estilo de página otimizado que não é capaz de gerar essas definições sozinho. Por padrão, esse recurso está habilitado.
<i>--wsdladdress</i>	Define o endereço de localização do arquivo de resultado WSDL
<i>-v --verbose</i>	Exibe informações detalhadas
<i>-?/-h/--help</i>	mostra página de uso

Dynamic ESDL
Interface de Comando do ESDL

Output	(srcdir <outdir>)/filename.ecl
--------	--------------------------------

Isso gera WSDL a partir da definição ESDL.

Exemplos:

```
esdl wsd1 MathService.ecm MathService
```

esdl publish

esdl publish <filename.(ecm|esdl|xml)> <servicename> [options]

filename	O arquivo ESDL (*.ecm, *.esdl, ou *.xml) que contém as definições de serviço.
servicename	O nome do serviço a ser publicado. É opcional caso a definição ESDL contenha apenas um serviço.
--overwrite	Substitui a última versão desta definição ESDL
Locais para procurar arquivos ESDL incluídos. Eles podem ser caminhos absolutos ou relativos. Se precisar especificar vários diretórios, você pode usar várias opções -I ou usar uma única entrada com os diretórios separados pelo caractere separador de ambiente. Para Linux, use dois pontos (:) e para Windows, use ponto e vírgula (;). Os caminhos também podem ser definidos usando uma variável de ambiente--ESDL_INCLUDE_PATH.	
-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)
-u, --username	O nome do usuário (se necessário)
-pw, --password	A senha (se necessário)
--version <ver>	Versão do serviço ESDL
--help	mostra as informações de uso para um determinado comando
-v, --verbose	Saída de informações adicionais de rastreamento

Publica uma definição de serviço ESDL no armazenamento de dados do sistema.

Exemplos:

```
esdl publish mathservice.ecm mathservice -s nnn.nnn.nnn.nnn --port 8010
```

esdl list-definitions

esdl list-definitions [options]

-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)
-u, --username	O nome do usuário (se necessário)
-pw, --password	A senha (se necessário)
--version <ver>	Versão do serviço ESDL
--help	mostra as informações de uso para um determinado comando
-v, --verbose	Saída de informações adicionais de rastreamento

Este comando lista as definições publicadas

Exemplo:

```
esdl list-definitions -s nnn.nnn.nnn.nnn --port 8010
```

esdl delete

esdl delete <ESDLServiceDefinitionName> <ESDLServiceDefinitionVersion> [options]

ESDLServiceDefinitionName	O nome de definição do serviço ESDL a ser removido
ESDLServiceDefinitionVersion	A versão da definição do serviço ESDL a ser removida
-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)
-u, --username	O nome do usuário (se necessário)
-pw, --password	A senha (se necessário)
--version <ver>	Versão do serviço ESDL
--help	mostra as informações de uso para um determinado comando
-v, --verbose	Saída de informações adicionais de rastreamento

Use este comando para remover uma definição do serviço ESDL. Se a definição do serviço estiver vinculada, é necessário desvincular antes de usar o comando para removê-la.

Exemplo:

```
esdl delete mathservice 2 -s nnn.nnn.nnn.nnn --port 8010
```

esdl bind-service

esdl bind-service <TargetESPProcessName> <TargetESPBindingPort | TargetESPServiceName> <ESDLDefinitionId> (<ESDLServiceName>) [command options]

TargetESPProcessName	O nome do processo ESP de destino
TargetESPBindingPort TargetESPServiceName	A porta da ligação ESP de destino ou o nome do serviço ESP de destino
ESDLDefinitionId	O nome e a versão da definição ESDL para qual o serviço será vinculado (já devem ter sido definidos no Dali)
ESDLServiceName	Nome do serviço ESDL (como definido na Definição ESDL). Obrigatório se a definição ESDL tiver vários serviços
--config <file XML>	Configuração XML (em linha ou como referência do arquivo)
--overwrite	Substitui a última versão desta definição ESDL
-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)
-u, --username	O nome do usuário (se necessário)
-pw, --password	A senha (se necessário)
--version <ver>	Versão do serviço ESDL
--help	mostra as informações de uso para um determinado comando
-v, --verbose	Saída de informações adicionais de rastreamento

Use este comando para vincular um serviço ESP baseado em Dynamic ESDL a uma definição ESDL .

Para vincular um serviço ESDL , forneça o nome do processo ESP de destino (processo ESP que hospedará o serviço ESP, como definido na definição ESDL.)

Também é preciso fornecer a porta na qual este serviço está configurado para ser executado (Ligação ESP) ou o nome do serviço que está sendo vinculado.

Também existe a opção de fornecer informações de configuração diretamente em linha ou usando um arquivo de configuração XML na seguinte sintaxe:

```
<Methods>
  <Method name="myMthd1"
    url="<RoxieIPRange>:9876/path?param=value" user="me" password="mypw" />
  <Method name="myMthd2"
    url="<RoxieIPRange>:9876/path?param=value" user="me" password="mypw" />
</Methods>
```

Exemplo:

```
esdl bind-service myesp 8003 MathSvc.1 MathSvc --config MathSvcCfg.xml
-s nnn.nnn.nnn.nnn -p 8010
```

Configurando os métodos ESDL binding

Os métodos de vinculação DESDL podem, opcionalmente, fornecer informações contextuais à consulta ECL de destino. Essas informações são configuradas anexando os elementos secundários à parte de Método (<Method>...</Method>) da ligação ESDL .

Por exemplo, o XML a seguir fornece uma amostra de ligação ESDL.

```
<Methods>
  <Method name="AddThis" url="<RoxieIPRange>:9876" querytype="roxie" queryname="AddThis"/>
</Methods>
```

Se este Método exigisse informação contextual, por exemplo sobre gateway, você poderia incluir a estrutura do gateway (<Gateways>...</Gateways>) descrita da seguinte forma.

```
<Methods>
  <Method name="AddThis" url="<RoxieIPRange>:9876" querytype="roxie" queryname="AddThis">
    <!--Optional Method Context Information start-->
    <Gateways>
      <Gateway name="mygateway" url="1.1.1.1:2222/someservice/somemethod/>
      <Gateway name="anothergateway" url="2.2.2.2:9999/someservice/somemethod/>
    </Gateways>
    <!--Optional Method Context Information end-->
  </Method>
</Methods>
```

O DESDL ESP não apresenta nenhuma restrição no layout dessa informação, apenas que é um XML válido. Isso oferece flexibilidade para incluir informações contextuais em qualquer formato XML válido.

Desenvolvedores Roxie (consulta) ECL precisam decidir quais informações irão precisar da busca ESP e projetar como essas informações serão apresentadas na configuração de busca ESP e de ligação ESDL.

No exemplo a seguir, toda busca "AddThis" processada pelo ESP e enviada ao Roxie teria as informações do gateway de amostra no contexto da busca.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
  <roxie.AddThis>
    <Context>
      <Row>
        <Common>
          <ESP>
            <ServiceName>wsmath</ServiceName>
            <Config>
              <Method name="AddThis" url="<RoxieIPRange>:9876"
                querytype="roxie" queryname="AddThis">
                <Gateways>
                  <Gateway name="mygateway" url="1.1.1.1:2222/someservice/somemethod/>
                  <Gateway name="anothergateway" url="2.2.2.2:9999/someservice/somemethod/>
                </Gateways>
              </Method>
            </Config>
          </ESP>
          <TransactionId>sometrxd</TransactionId>
        </Common>
      </Row>
    </Context>
    <AddThisRequest>
      <Row>
        <Number1>34</Number1>
        <Number2>232</Number2>
      </Row>
    </AddThisRequest>
  </roxie.AddThis>
</soap:Body>
</soap:Envelope>
```

A consulta ECL absorve essas informações e está livre para fazer o que precisa ser feito com elas. Em algumas instâncias, a consulta precisa enviar uma solicitação para um gateway a fim de processar a busca

atual de forma apropriada. Ele pode interrogar as informações contextuais para obter as informações adequadas de conexão do gateway e, em seguida, usar essas informações para criar a verdadeira conexão de solicitação de gateway.

esdl list-bindings

esdl list-bindings [options]

-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)
-u, --username	O nome do usuário (se necessário)
-pw, --password	A senha (se necessário)
--version <ver>	Versão do serviço ESDL
--help	mostra as informações de uso para um determinado comando
-v, --verbose	Saída de informações adicionais de rastreamento

Use este comando para listar as ligações em um servidor.

Exemplo:

```
esdl list-bindings -s nnn.nnn.nnn.nnn -p 8010
```

esdl unbind-service

esdl unbind-service <ESPProcessName> <ESPBindingName> [options]

ESPProcessName	O nome do processo ESP de destino
ESPBindingName	O nome da ligação do ESP
-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)
-u, --username	O nome do usuário (se necessário)
-pw, --password	A senha (se necessário)
--version <ver>	Versão do serviço ESDL
--help	mostra as informações de uso para um determinado comando
-v, --verbose	Saída de informações adicionais de rastreamento

Use este comando para desvincular as ligações baseadas em serviço ESDL .

Para desvincular uma determinada ligação ESDL, forneça o nome do processo ESP e a ligação ESP que compõe a ligação ESDL .

As ligações ESDL disponíveis para serem desvinculadas podem ser encontradas ao usar o comando "esdl list-bindings"

Exemplo:

```
esdl unbind-service myesp myServiceBinding
```

esdl bind-method

esdl bind-method <TargetESPProcessName> <TargetESPBindingName> <TargetServiceName> <TargetServiceDefVersion> <TargetMethodName> [options]

TargetESPProcessName	O nome do processo ESP de destino
TargetESPBindingName	Nome da ligação ESP de destino
TargetServiceName	Nome do serviço a ser vinculado (já deve ter sido definido no dali.)
TargetServiceDefVersion	A versão da definição ESDL do serviço de destino (deve estar presente no dali)
TargetMethodName	O nome do método de destino (deve existir na definição ESDL de serviço)
--config <file XML>	Configuração XML (em linha ou como referência do arquivo)
--overwrite	Substitui a última versão desta definição ESDL
-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)
-u, --username	O nome do usuário (se necessário)
-pw, --password	A senha (se necessário)
--version <ver>	Versão do serviço ESDL
--help	mostra as informações de uso para um determinado comando
-v, --verbose	Saída de informações adicionais de rastreamento

Use este comando para publicar as ligações baseadas em serviço ESDL .

Para vincular um serviço ESDL , forneça o nome do processo ESP de destino (processo ESP que hospedará o serviço ESP, como definido na definição ESDL.)

Também é preciso fornecer a porta na qual este serviço está configurado para ser executado (Ligação ESP) ou o nome do serviço que está sendo vinculado.

Também existe a opção de fornecer informações de configuração diretamente em linha ou usando um arquivo de configuração XML na seguinte sintaxe:

```
<Methods>
  <Method name="myMthd1"
    url="http://<RoxieIPRange>:9876/path?param=value" user="me" password="mypw"/>
  <Method name="myMthd2"
    url="http://<RoxieIPRange>:9876/path?param=value" user="me" password="mypw"/>
</Methods>
```

Exemplo:

```
esdl bind-service myesp 8003 MathSvc.1
  MathSvc --config MathSvcCfg.xml -s nnn.nnn.nnn.nnn -p 8010
```

esdl unbind-method

esdl unbind-method <ESPProcessName> <ESPBindingName> <ESDLServiceName> <MethodName> [options]

ESPProcessName	O nome do processo ESP de destino
ESPBindingName	O nome da ligação ESP de destino associado a este serviço.
ESDLServiceName	O nome do serviço ESDL associado com o método de destino.
MethodName	O nome do método de destino (deve existir na definição ESDL de serviço)
-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)
-u, --username	O nome do usuário (se necessário)
-pw, --password	A senha (se necessário)
--version <ver>	Versão do serviço ESDL
--help	mostra as informações de uso para um determinado comando
-v, --verbose	Saída de informações adicionais de rastreamento

Use este comando para desvincular uma configuração de método associada com uma determinada ligação ESDL .

Para desvincular um método, forneça o nome do processo ESP de destino (o ESP que hospeda o serviço.)

Também é preciso fornecer a ligação ESP na qual este serviço está configurado para ser executado, o nome do serviço ESDL, e o nome do método que está sendo desvinculado.

Exemplo:

```
esdl unbind-method myesp myespbinding WsMyService mymethod
```

esdl list-bindings

esdl get-binding <ESDLBindingId> [options]

ESDLBindingId	A id de vinculação ESDL de destino <espprocessname>.<espbinding-name>
-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)
-u, --username	O nome do usuário (se necessário)
-pw, --password	A senha (se necessário)
--version <ver>	Versão do serviço ESDL
--help	mostra as informações de uso para um determinado comando
-v, --verbose	Saída de informações adicionais de rastreamento

Use este comando para obter ligações baseadas em serviço DESDL .

Para especificar a configuração de serviço baseada em DESDL de destino, forneça o processo ESP de destino (nome do processo esp ou endereço IP da máquina) que hospeda o serviço.

Também é preciso fornecer a porta na qual este serviço está configurado para ser executado e o nome do serviço.

Exemplo:

```
esdl get-binding myesp.dESDL_Service -s nnn.nnn.nnn.nnn -p 8010
```