

- [Home](#)
- Específicos
- Flex
- Getting Started
- Java
- Processo Datasul
- Progress
- Tools
- Trilhas
- User
- [Page Operations](#)
- [Browse Space](#)

SDK - Software Development Kit - v.1.0

## DatasulZoom - Metadados

Added by [Raphael Luiz Nascimento](#), last edited by [Diego Silvio Zimmermann](#) on Jan 28, 2011 ([view change](#))

Labels  
[metadados](#), [zoom](#)

## Visão Geral

Para utilizar o componente de DatasulZoom do metadados, algumas propriedades deverão ser informadas. Abaixo segue uma lista de atributos que deverão ser preenchidos.

Documentação completa e referência do zoom do framework por favor acesse a seguinte url: [Datasul Zoom](#).

## Atributos

Esta tabela demonstra os atributos disponíveis para o componente, nem todos os atributos são obrigatórios.

Atributo	Tipo	Obrigatório?	Descrição
label	String	Sim	Label do Crud.
detailMode	Boolean	Não	Seta o zoom para ser utilizado em modo de detalhe. Valor padrão false.
fieldCode	String	Sim	Nome da propriedade que será a chave do objeto retornado.
fieldLabel	String	Sim	Nome da propriedade que será a descrição do zoom.
xmlName	String	Sim	Caminho completo do xml que contém o crud. Exemplo : <b>view.ems5.CRUDUnidOrgan</b>
typeColumn	String	Sim	Tipo do atributo que foi inserido na propriedade fieldCode.
queryColumnName	String	Sim	Identifica qual o nome da coluna que será efetuada a busca na base de dados.
entityHandler	String	Sim	Nome da BO que faz a busca dos dados para o zoom.
entityName	String	Sim	Nome da entidade. Exemplo: <b>emsuni.UnidOrgan</b>
id	String	Não	Id do componente na tela.
findBlankCode	Boolean	Não	Flag indicando que o zoom deverá buscar os registros mesmo se o atributo codeValue não foi informado pelo usuário.
errorMessage	String	Não	Mensagem de erro caso não tenha encontrado nenhum registro na base.
codeValue	String	Não	Valor inicial do campo zoom. Quando este atributo for informado o zoom irá automaticamente buscar um registro que corresponda ao valor setado.
cacheEnabled	Boolean	Não	Valor inicial do campo zoom para controle do cache. Quando este atributo for informado como "true" o zoom irá automaticamente gerar o controle de cache. Default é "false".
change	Function	Não	Listener para o evento change.
changeOut	Function	Não	Listener para o evento changeOut do zoom.
blankFocusOut	Function	Não	Listener para o evento blankFocusOut.
showLabel	Boolean	Não	Atributo utilizado para escolher mostrar ou não o rótulo fieldLabel ao escolher um valor para o zoom. O default é true (mostrar o rótulo).
fieldCodeMask	String	Não	Atributo que pode ser preenchido com um formato de máscara para ser aplicado ao fieldCode. Essa máscara funciona com a seguinte notação e tem apenas o intuito de apresentação em tela EX: DATE: >>/>>/>>>>, DECIMAL: >>>.>>>.>>> Lembrando que essa notação não segue formato progress ela apenas aplica a formatação no campo. Internamente no zoom tudo funciona com o valor real.

## Métodos

Método	Descrição
updateObject	Realiza o update do componente zoom. Esse método funciona da mesma forma que o método updateObjectKey, o qual, foi depreciado para o componente DatasulZoom atual.
findObject	Efetua a busca na base de dados conforme o valor passado como parâmetro. O parâmetro <b>text</b> é do tipo <b>String</b> sendo este obrigatório para busca na base de dados. Caso a busca dos dados retorne mais de um elemento ou nenhum é exibida uma mensagem de erro mapeado para o evento ZoomEvent.ZOOM_FIND_OBJECT_FAULT. Caso a busca retorne apenas um registro o valor do elemento será retornado no evento ZoomEvent.ZOOM_FIND_OBJECT_RESULT. Para utilização do método <i>findObject()</i> deve ser instanciado um objeto do tipo DatasulZoom com os seguintes atributos <b>entityHandler</b> , <b>entityName</b> , <b>queryColumnName</b> e <b>typeColumn</b> . Ex: dzTeste.findObject("10");
openZoom	Apresenta a tela de Zoom para seleção de elementos. A tela é centralizada conforme o parâmetro <b>parent</b> passado, caso o mesmo não seja atribuído é considerada a aplicação como parent do objeto. Caso seja atribuído um parent que não está visível em tela a tela de Zoom não será apresentada. Após a abertura da tela de Zoom, seleção do registro e pressionado o botão <i>Selecionar</i> o evento ZoomEvent.ZOOM_SELECT é disparado com o valor do registro selecionado pelo usuário. Caso o usuário pressione o botão <i>Cancelar</i> na tela exibida ou feche a mesma o evento ZoomEvent.ZOOM_BACK é disparado. Para a utilização do método <i>openZoom()</i> deve ser instanciado um objeto do tipo DatasulZoom com os seguintes atributos <b>entityHandler</b> , <b>entityName</b> , <b>fieldCode</b> , <b>fieldLabel</b> , <b>queryColumnName</b> , <b>typeColumn</b> e <b>xmlName</b> . Em algumas situações é necessário atribuir alguns filtros iniciais para a tela de Zoom, para isto o atributo <b>zoomExtraFields</b> deve ser preenchido. Ex: dzTeste.openZoom();
cleanZoomControl	Limpa os controles de valores já selecionados. Método utilizado quando se tem a necessidade de executar change para um valor que já foi selecionado. Ex: dzTeste.cleanZoomControl();

## Exemplo

Abaixo segue um exemplo de utilização do componente:

```
<totvs:DatasulZoom
  label="Teste DatasulZoom"
  detailMode="false"
  fieldCode="primaryKey.codUnidOrgan"
  fieldLabel="nomAbrev"
  xmlName="view.ems5.CRUDUnidOrgan"
  typeColumn="String"
  queryColumnName="primaryKey.codUnidOrgan"
  entityHandler="prgint/utb/utb936wl.py"
  entityName="emsuni.UnidOrgan"
  id="dtZoom"
  findBlankCode="true"
  errorMessage="Erro Message"
  codeValue="101"
  change="this.changeEventPerform(event)"
  changeOut="this.changeOutEventPerform(event)"
  blankFocusOut="this.blankFocusOutEventPerform(event)"
/>
```

Abaixo exemplo de utilização do método findObject():

```
import mx.controls.Alert;
import com.totvs.framework.zoom.control.ZoomEvent;
import com.totvs.framework.zoom.view.DatasulZoom;

private function onGetZoomValue():void {
  var dzZoom:DatasulZoom = new DatasulZoom();
  dzZoom.entityHandler = "prgfin/cfl/cfl908we.py";
  dzZoom.entityName = "emsfin.UnidControlFinanc";
  dzZoom.queryColumnName = "primaryKey.codUnidControlFinanc";
  dzZoom.typeColumn = "String";

  dzZoom.addEventListener(ZoomEvent.ZOOM_FIND_OBJECT_RESULT, this.onFindObjectResult);
  dzZoom.addEventListener(ZoomEvent.ZOOM_FIND_OBJECT_FAULT, this.onFindObjectFault);

  dzZoom.findObject("10");
}

private function onFindObjectResult(event:ZoomEvent):void {
  this.removeListeners(event.target);
  if ( event.object && event.object.hasOwnProperty("desUnidControlFinanc") ) {
    Alert.show(event.object["desUnidControlFinanc"] as String);
  }
}

private function onFindObjectFault(event:ZoomEvent):void {
  this.removeListeners(event.target);
  Alert.show(event.object.toString());
}

private function removeListeners(object:Object):void {
  object.removeEventListener(ZoomEvent.ZOOM_FIND_OBJECT_RESULT, this.onFindObjectResult);
}
```

```

    object.removeListener(ZoomEvent.ZOOM_FIND_OBJECT_FAULT, this.onFindObjectFault);
}

```

Abaixo exemplos de utilização do método `openZoom()`, sendo o primeiro sem atribuir o parâmetro `zoomExtraFields` e o segundo com o valor atribuído:

```

import mx.controls.Alert;
import com.totvs.framework.zoom.control.ZoomEvent;
import com.totvs.framework.zoom.view.DatasulZoom;

private function onOpenZoom():void {
    var dzZoom:DatasulZoom = new DatasulZoom();
    dzZoom.entityHandler = "prgfin/cfl/cfl1908we.py";
    dzZoom.entityName = "emsfin.UnidControlFinanc";
    dzZoom.fieldCode = "primaryKey.codUnidControlFinanc";
    dzZoom.fieldLabel = "desUnidControlFinanc";
    dzZoom.queryColumnName = "primaryKey.codUnidControlFinanc";
    dzZoom.typeColumn = "String";
    dzZoom.xmlName = "view.ems5.CRUDUnidControlFinanc";

    dzZoom.addEventListener(ZoomEvent.ZOOM_SELECT, this.onZoomSelect);
    dzZoom.addEventListener(ZoomEvent.ZOOM_BACK, this.onZoomBack);

    dzZoom.openZoom();
}

private function onZoomSelect(event:ZoomEvent):void {
    this.removeListeners(event.target);
    if ( event.object &&
        event.object.hasOwnProperty("primaryKey.codUnidControlFinanc") &&
        event.object.hasOwnProperty("desUnidControlFinanc") ) {
        var code:String = event.object["primaryKey.codUnidControlFinanc"] as String;
        var desc:String = event.object["desUnidControlFinanc"] as String;
        Alert.show("Code: " + code + "\nDesc: " + desc);
    }
}

private function onZoomBack(event:ZoomEvent):void {
    this.removeListeners(event.target);
    Alert.show("Cancel");
}

private function removeListeners(object:Object):void {
    object.removeListener(ZoomEvent.ZOOM_SELECT, this.onZoomSelect);
    object.removeListener(ZoomEvent.ZOOM_BACK, this.onZoomBack);
}

```

```

import com.totvs.framework.filter.model.FilterTypeConstants;
import com.totvs.framework.filter.model.FilterParameter;
import mx.collections.ArrayCollection;
import mx.controls.Alert;
import com.totvs.framework.zoom.control.ZoomEvent;
import com.totvs.framework.zoom.view.DatasulZoom;

private function onOpenZoom():void {
    var dzZoom:DatasulZoom = new DatasulZoom();
    dzZoom.entityHandler = "prgfin/cfl/cfl1908we.py";
    dzZoom.entityName = "emsfin.UnidControlFinanc";
    dzZoom.fieldCode = "primaryKey.codUnidControlFinanc";
    dzZoom.fieldLabel = "desUnidControlFinanc";
    dzZoom.queryColumnName = "primaryKey.codUnidControlFinanc";
    dzZoom.typeColumn = "String";
    dzZoom.xmlName = "view.ems5.CRUDUnidControlFinanc";

    var filters:ArrayCollection = new ArrayCollection();
    var filterParameter:FilterParameter = new FilterParameter();
    filterParameter.columnName = "ind_niv_unid_control_financ";
    filterParameter.columnType = FilterTypeConstants.TYPE_STRING;
    filterParameter.columnValue = "Sintético";
    filters.addItem(filterParameter);
    dzZoom.zoomExtraFields = filters;

    dzZoom.addEventListener(ZoomEvent.ZOOM_SELECT, this.onZoomSelect);
    dzZoom.addEventListener(ZoomEvent.ZOOM_BACK, this.onZoomBack);

    dzZoom.openZoom();
}

private function onZoomSelect(event:ZoomEvent):void {
    this.removeListeners(event.target);
    if ( event.object &&
        event.object.hasOwnProperty("primaryKey.codUnidControlFinanc") &&
        event.object.hasOwnProperty("desUnidControlFinanc") ) {
        var code:String = event.object["primaryKey.codUnidControlFinanc"] as String;
        var desc:String = event.object["desUnidControlFinanc"] as String;
        Alert.show("Code: " + code + "\nDesc: " + desc);
    }
}

private function onZoomBack(event:ZoomEvent):void {
    this.removeListeners(event.target);
    Alert.show("Cancel");
}

private function removeListeners(object:Object):void {
    object.removeListener(ZoomEvent.ZOOM_SELECT, this.onZoomSelect);
    object.removeListener(ZoomEvent.ZOOM_BACK, this.onZoomBack);
}

```

## Preenchendo os campos

Alguns atributos listados acima o valor se encontra no XML do crud.

Atributo		Observação
fieldCode	O valor deste atributo geralmente é o nome da coluna que é chave primária da tabela que o crud utiliza. Geralmente este campo contém a prefixo "primaryKey."	<MDItem column="primaryKey.codUnidOrgan" i18nDescription="14" itemId="140592" i18nHelp="15" type="String" mask="">
fieldLabel	O valor deste atributo é o nome da coluna que o desenvolvedor quer que seja a description do zoom.	<MDItem column="nomAbrev" i18nDescription="20" itemId="140596" i18nHelp="20" type="String" mask="">
xmlName	Caminho completo do xml do CRUD. Este xml pode ser encontrado em dentro da pasta MD que se encontra no war do framework.	Por exemplo: D:\datasul\jboss-4.2.3-GA\server\byyou\deploy\datasul-expedition-java-ear-3.2.0-SNAPSHOT.ear\datasul-expedition-common-war-3.2.0-SNAPSHOT.war\MD.
typeColumn	Tipo da coluna que foi selecionada como fieldCode	<MDItem column="primaryKey.codUnidOrgan" i18nDescription="14" itemId="140592" i18nHelp="15" type="String" mask="">
queryColumnName	Valor do atributo que irá ser realizada a query.	Mesmo conteúdo da tag fieldCode.
entityHandler	Caminho do programa progress que irá realizar a busca das informações do zoom. O valor deste atributo é encontrado no XML da tabela que este crud utiliza.	<p>Para informar este valor siga os seguintes passos:</p> <p>1) No XML do crud procure o valor table  &lt;MDForm name="CRUDUnidOrgan" table="model.emsuni.UnidOrgan" template="CRUD" i18nDescription="1" zoomColumns=""&gt;</p> <p>2) Dentro do jboss partindo da pasta MD (dentro do war) procure o seguinte caminho ..... \MD\model\emsuni.UnidOrgan.xml.</p> <p>3) No XML da table procure pelo atributo handler.  &lt;MDTable name="model.emsuni.UnidOrgan" i18nDescription="1" database="emsuni" appName="model" appVersion="5.1.1" idColumns="" voClassName="com.totvs.model.emsuni.be.UnidOrgan" handler="prgint/utb/utb936wl.py"&gt;</p>
entityName	Nome da entidade	<p>Repita os passos 1 e 2 listados acima e procure pelo atributo <b>name</b> da tag MDTable.  &lt;MDTable name="model.emsuni.UnidOrgan" i18nDescription="1" database="emsuni" appName="model" appVersion="5.1.1" idColumns="" voClassName="com.totvs.model.emsuni.be.UnidOrgan" handler="prgint/utb/utb936wl.py"&gt;</p> <div style="border: 1px solid yellow; padding: 5px; margin-top: 10px;"> <p><b>⚠ Aviso</b>  O valor do campo deverá ser preenchido sem a palavra "model.", sendo assim o valor a ser preenchido é <b>"emsuni.UnidOrgan"</b>.</p> </div>

## ZoomExtraField utilizando Query nativa do Progress

Há casos que o desenvolvedor precisa elaborar um filtro mais complexo que o exemplo acima não atendia certas possibilidades, vejamos alguns exemplos:

- *Abrir o CRUD de Pessoas filtrando somente pelos departamentos de Compra e Faturamento.*
- *Abrir o CRUD de Pessoas filtrando somente os funcionários entre 5 a 10 anos de casa.*
- *Abrir o CRUD de Pessoas com idade superior a 45 anos.*
- *etc.*

Foi incluído no filtro o tipo "Query", esta query é automaticamente passada para o programa progress sem qualquer tipo de tratamento quanto a sua sintaxe e validade. Portanto é de responsabilidade do desenvolvedor o correto preenchimento deste parâmetro.

A query passada como parâmetro será executada dinamicamente, portanto a sintaxe e operadores possíveis são os mesmos do Progress.

Abaixo segue um exemplo da sua utilização:

```
{code:ActionScript}
import com.totvs.framework.filter.model.FilterTypeConstants;
import com.totvs.framework.filter.model.FilterParameter;
import mx.collections.ArrayCollection;
import mx.controls.Alert;
import com.totvs.framework.zoom.control.ZoomEvent;
import com.totvs.framework.zoom.view.DatasulZoom;

private function onOpenZoom():void {
    var dzZoom:DatasulZoom = new DatasulZoom();
    dzZoom.entityHandler = "prgfin/cfl/cfl1908we.py";
    dzZoom.entityName = "emsfin.UnidControlFinanc";
    dzZoom.fieldCode = "primaryKey.codUnidControlFinanc";
    dzZoom.fieldLabel = "desUnidControlFinanc";
    dzZoom.queryColumnName = "primaryKey.codUnidControlFinanc";
    dzZoom.typeColumn = "String";
    dzZoom.xmlName = "view.ems5.CRUDUnidControlFinanc";

    var filters:ArrayCollection = new ArrayCollection();
    var filterParameter:FilterParameter = new FilterParameter();
    filterParameter.columnType = FilterTypeConstants.TYPE_QUERY;
    filterParameter.columnValue = "ind_niv_unid_control_financ <> 'Sintético' ";
    filters.addItem(filterParameter);
    dzZoom.zoomExtraFields = filters;

    dzZoom.addEventListener(ZoomEvent.ZOOM_SELECT, this.onZoomSelect);
    dzZoom.addEventListener(ZoomEvent.ZOOM_BACK, this.onZoomBack);

    dzZoom.openZoom();
}
private function onZoomSelect(event:ZoomEvent):void {
    this.removeListeners(event.target);
    if ( event.object &&
        event.object.hasOwnProperty("primaryKey.codUnidControlFinanc") &&
        event.object.hasOwnProperty("desUnidControlFinanc") ) {
        var code:String = event.object["primaryKey.codUnidControlFinanc"] as String;
        var desc:String = event.object["desUnidControlFinanc"] as String;
        Alert.show("Code: " + code + "\nDesc: " + desc);
    }
}
private function onZoomBack(event:ZoomEvent):void {
    this.removeListeners(event.target);
    Alert.show("Cancel");
}
private function removeListeners(object:Object):void {
    object.removeEventListener(ZoomEvent.ZOOM_SELECT, this.onZoomSelect);
    object.removeEventListener(ZoomEvent.ZOOM_BACK, this.onZoomBack);
}
```



#### Atenção

Diferentemente dos filtros citados no exemplos acima (não utilizando Query) você deve colocar o nome dos campos que estão no xml, porém neste caso a query será enviada automaticamente pelo progress e esta deve ser no formato do progress e conter uma cláusula de where válida no progress.  
O nome das colunas devem ser os mesmos das colunas que foram cadastradas para a tabela.



#### Atenção

Não é possível incluir uma query utilizando como parâmetro uma reference, visto que esta reference não existe no lado do progress, para este caso você deve incluir o campo da tabela que faz a referência.

## Ver Também

- [Datasul Zoom](#)
- [Filtrando dados através do método ADD-FILTER-PARAM no ABL Script](#)
- [Zoom Container](#)

### Informações sobre este Documento

Versão	Estado	Compatibilidade	Criador	Criação	Último Revisor	Última Revisão
<a href="#">30</a>		3.2.0	<a href="#">Raphael Luiz Nascimento</a>	Abr 29, 2010 18:00	<a href="#">Diego Silvio Zimmermann</a>	Jan 28, 2011 08:33



Your  
Rating:

Results:

0 rates

[O que é RSS?](#)

**Children (2)** [Hide Children](#) | [View in Hierarchy](#)

 [DatasulZoom - DataGrid \(Estilo ZoomManager\)](#)

 [Refactor DatasulZoom](#)

 [Add Comment](#)