

Manual do Usuário do HighPerformance Template

Sumário

1 Introdução ao HighPerformance Template	1
1.1 Arquivos da Biblioteca	1
1.2 Instalação da Biblioteca em uma Aplicação	2
1.3 Objetos com Temas Nativos	2
1.4 Associação de Valores a Objetos de Dados	2
1.5 Execução de Scripts em Objetos Instanciados	2
1.6 Acesso a Propriedades ou Métodos Não Disponíveis nas Instâncias	3
2 Objeto Chart	5
3 Objetos DateTimePicker e DateTimeCombo	7
4 Menus de Comando	9
4.1 Associação Simples	9
4.2 Menus em Grupo	9
4.3 Alinhamento Automático	9
4.4 Propriedade Behaviour	10
4.5 Adicionando Ícones em um Menu	10
5 Textos	12
6 Temas	14
6.1 Configuração de Esquemas ao Iniciar uma Aplicação	14
6.2 Carregando um Novo Esquema	15
6.3 Identificando o Esquema Atual	15
6.4 Incluindo um Objeto Criado pelo Usuário em um Tema	15
6.5 Cadastrando Cores em um Tema	16
7 Templates de Quadros e Múltiplos Monitores	17
7.1 Definindo uma Matriz de Monitores	17
7.2 Seleccionando um Template de Quadros	18
7.3 Guia Rápido de Configuração de Quadros	22
8 Telas e Navegação	23
8.1 Abrindo uma Tela em um Quadro já Existente	23
8.2 Abrindo um Novo Quadro	23
8.3 Abrindo Telas e Quadros em Outro Monitor	23
8.4 Usando os Recursos Voltar e Avançar	23
8.5 Abrindo Telas e Templates por Script	24
8.6 Abrindo Telas Pop-ups e Modais	24
9 Tradução Automática	26
9.1 Importando um Dicionário	26
9.2 Alterando um Idioma em Tempo de Execução	28
10 Conversão de Objetos para o Formato XML	29
11 Menus de Navegação	31
12 Controle TreeView	34
12.1 Modo HighPerformance ou Convencional	37

12.2 Exibindo Ícones	38
12.3 Exibindo a Barra de Rolagem	39
12.4 Seleccionando Múltiplos Nós	39
12.5 Filtrando o Conteúdo	39
12.6 Troca Dinâmica da Fonte de Dados	40
12.7 Evento Personalizado	40
13 Novidades da Versão	41

O **HighPerformance Template** para **E3** ou **Elipse Power** é um modelo de aplicação que sintetiza os resultados de diversos estudos realizados nas mais distintas áreas do conhecimento, aplicados no desenvolvimento de interfaces industriais. Os objetos, recursos visuais e diretrizes de design que compõem o **HighPerformance Template** possibilitam:

- Aumento da consciência situacional do usuário
- Maior efetividade na prevenção de falhas e redução de erros operativos
- Maior acessibilidade para os portadores de daltonismo
- Aprimoramento estético
- Melhor usabilidade
- Maior facilidade e velocidade de desenvolvimento

Para mais detalhes sobre os benefícios oferecidos pelo **HighPerformance Template**, consulte o *Elipse Knowledgebase*.

A solução completa de alto desempenho é um somatório de recursos cujo escopo vai além da criação de simples objetos. Trabalha-se não apenas objetos isolados, mas o contexto das Telas e da aplicação como um todo. Por isto, o **HighPerformance Template** contempla não apenas uma biblioteca de objetos para visualização de dados, mas diversas bibliotecas integradas entre si e que contêm ícones, menus, Esquemas de cores, Temas, objetos de formulários e categorias de textos, entre outros recursos.

1.1 Arquivos da Biblioteca

O **HighPerformance Template** está organizado nos grupos de arquivos descritos na tabela a seguir.

Grupos de arquivos disponíveis no HighPerformance Template

GRUPO	ARQUIVOS	DESCRIÇÃO
Alarmes	hpE3Alarm.lib	Contém objetos para serem integrados ao E3Alarm
Visualização de Dados	hpDisplayMedia.lib, hpTexts.lib, hpForms.lib, hpTreeview.lib e hpEPMChart.lib	Contém objetos de visualização de dados, como <i>bargraphs</i> , <i>displays</i> , <i>charts</i> , textos, objetos de formulários, <i>TreeViews</i> , etc.
Ícones	hpArchitecture2d, hpArchitecture3d e hpIcons.lib	Contém ícones adaptados ao HighPerformance Template
Menus	hpMenu.lib e hpNavigation.lib	Contém objetos para composição de menus, abas e Associações em geral
Temas	hpThemes.lib	Temas de cores
Tradutores	hpTranslator.lib	Contém recursos para tradução automática
Viewer	hpViewer.lib	Configuração de matrizes de monitores e tamanho do Viewer

GRUPO	ARQUIVOS	DESCRIÇÃO
XML	hpXMLIndexer.lib	Geração de conteúdo no formato XML
Projetos	hpMain.prj	Contém objetos necessários para o funcionamento do HighPerformance Template . Recomenda-se usar o objeto Viewer deste arquivo para toda a aplicação

1.2 Instalação da Biblioteca em uma Aplicação

Para usar o **HighPerformance Template**, copie a pasta **HighPerformance**, com todas as suas sub-pastas e arquivos, para a pasta raiz da aplicação.

Em seguida, no Studio, insira no Domínio todos os arquivos de biblioteca (.lib) e projetos (.prj) que estão nas pastas **Libraries** e **Projects**, respectivamente.

Os arquivos de bibliotecas e projetos do **HighPerformance Template** sempre utilizam o prefixo **hp**, para que possam ser facilmente diferenciados dos objetos da aplicação.

O arquivo de projeto hpMain.prj vem com um objeto Viewer previamente configurado. Este objeto contém objetos de controle do *Template* e, por este motivo, deve-se evitar sua substituição por outro objeto Viewer.

1.3 Objetos com Temas Nativos

Todos os objetos de Bibliotecas usam os Temas nativamente. Isto significa que ao instanciar um Display, um Texto, um ícone ou qualquer outro objeto destas bibliotecas em uma Tela, o objeto instanciado vem com a funcionalidade de alteração automática de cor com base no Tema selecionado.

1.4 Associação de Valores a Objetos de Dados

A Biblioteca **hpDisplayMedia** contém objetos muito semelhantes entre si, porém com a diferenciação do sufixo **IO** no nome, como por exemplo **hpDynamicValue** e **hpDynamicValueIO**.

O sufixo **IO** em um objeto significa que este objeto pode receber uma Associação direta do Tag de Comunicação (**IOTag**), por isto é mais indicado para uso com dados de processo, ao passo que o objeto sem o sufixo é indicado para valores simulados ou calculados (**DemoTag** ou **InternalTag**).

1.5 Execução de Scripts em Objetos Instanciados

Pode-se monitorar eventos de clique de mouse e escrever scripts personalizados diretamente no objeto instanciado, sem a necessidade de editar o objeto original na Biblioteca.

Para isto, muitos dos objetos possuem a propriedade **EventClick**, que é um contador incrementado a cada clique do mouse que o objeto recebe. Ao monitorar este valor, pode-se criar um evento personalizado **OnEventClick**, que é disparado quando o valor desta propriedade muda, e escrever um script na janela de edição de scripts.

Janela de criação de eventos de usuário

1.6 Acesso a Propriedades ou Métodos Não Disponíveis nas Instâncias

A principal vantagem do **HighPerformance Template** é criar um objeto já configurado, de forma que o usuário não precise se preocupar com detalhes de configuração. O usuário instancia o objeto e já pode usá-lo com mínimas (ou nenhuma) alteração. Isto preserva as características principais da metodologia de alto desempenho visual, ao mesmo tempo que garante a praticidade de uso.

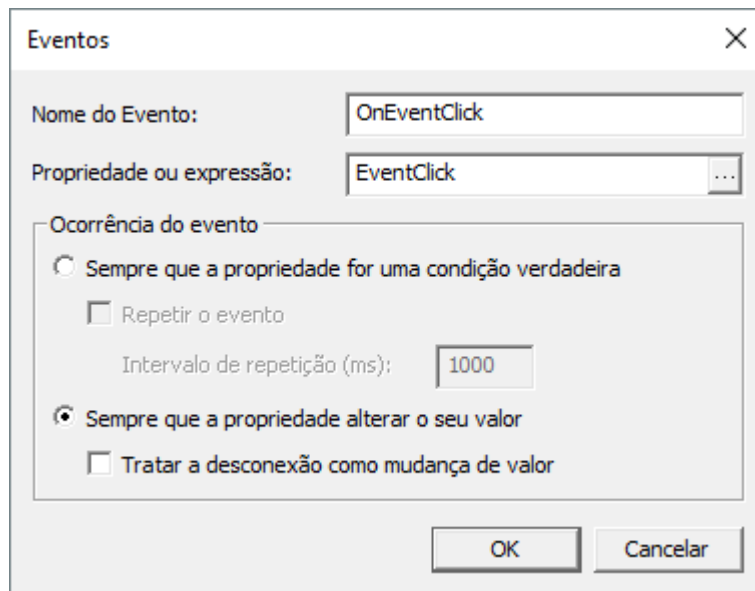
Há casos, no entanto, em que alguns recursos importantes do objeto estão indisponíveis na instância. Um exemplo são os métodos de objetos. No caso de Listas (*List Box*) ou Listas de Seleção (*Combo Box*), o funcionamento básico do objeto depende de alguns destes métodos, como por exemplo o método **AddItem**.

Nestes casos específicos, o **HighPerformance Template** fornece acesso integral ao objeto através da propriedade **Object**. Esta propriedade referencia o objeto original, tornando disponíveis todas as suas propriedades e métodos. Por exemplo, o método **AddItem** é acessado na instância, por script, usando o código a seguir.

```
Object.AddItem "Item a ser adicionado"
```

Este recurso deve ser usado apenas para garantir o funcionamento básico do controle, e não para descaracterizá-lo, alterando sua cor de fundo ou sua fonte, pois ele já se encontra harmonizado com todos os demais objetos do **HighPerformance Template**.

Já eventos como **Click**, **Change**, **MouseMove**, etc., ganham propriedades equivalentes com a adição do prefixo **Event** em seu nome. Portanto, **EventClick**, **EventChange** e **EventMouseMove** são propriedades que sofrem alteração em seu valor quando o evento original ocorre. O usuário deve criar um evento personalizado (Evento de Usuário) para executar um script na ocorrência do evento original. A figura a seguir mostra a janela de criação de um Evento de Usuário chamado **OnEventClick**.



Criação de um evento personalizado OnEventClick

No caso dos eventos, há ainda a particularidade de eventos que retornam valores e eventos que não retornam valores. Por exemplo, o evento **Click** não retorna nenhum valor ao usuário.

```
Sub ComboBox1_Click()
```

Quando um evento não retorna valores, cada ocorrência incrementa o valor da propriedade correspondente. No exemplo anterior, cada clique do mouse no objeto incrementa o valor da propriedade **EventClick**. Assim, pode-se monitorar qualquer mudança de valor e executar um script ao clicar no objeto.

Já o evento **KeyDown** retorna o código das teclas pressionadas, *KeyCode* e *Shift*.

```
Sub ComboBox1_KeyDown(KeyCode, Shift)
```

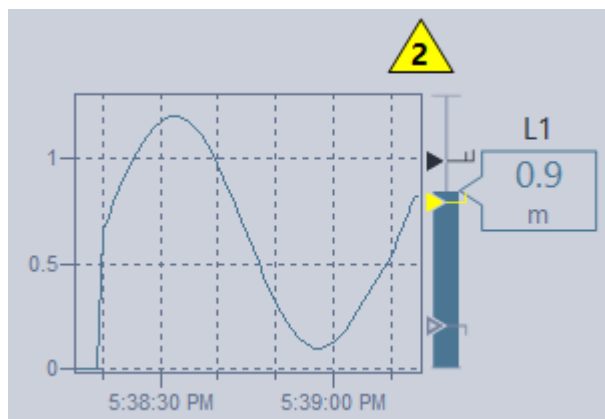
Quando o evento retorna valores, estes são passados para a propriedade correspondente na forma de um *array*. No exemplo anterior, a cada ocorrência do evento os parâmetros *KeyCode* e *Shift* são passados à propriedade **EventKeyDown** como um *array*.

```
Array(KeyCode, Shift)
```

Assim é possível monitorar alterações no valor da propriedade e recuperar esta informação na instância do objeto, por exemplo usando o código a seguir.

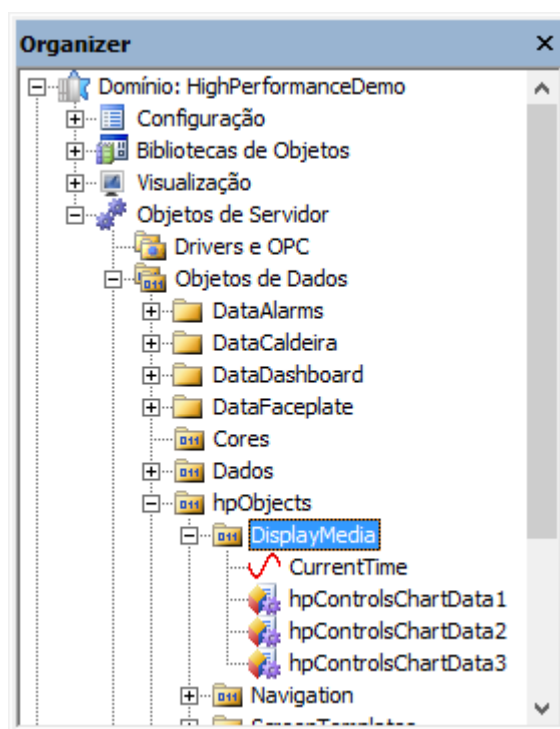
```
Sub hpComboBox1_OnEventKeyDown()  
    'Propriedade EventKeyDown é um array  
    vKeyCode = EventKeyDown(0)  
    vShift = EventKeyDown(1)  
    MsgBox "KeyCode = " & CStr(vKeyCode)  
    MsgBox "Shift = " & CStr(vShift)  
End Sub
```

O objeto **hpControlsChart** é um objeto gráfico que exibe dados em tempo real e dados históricos recentes de uma única Pena. Seu objetivo é apresentar ao usuário o comportamento mais recente de uma variável, se o valor está subindo, descendo ou mantendo-se estável.



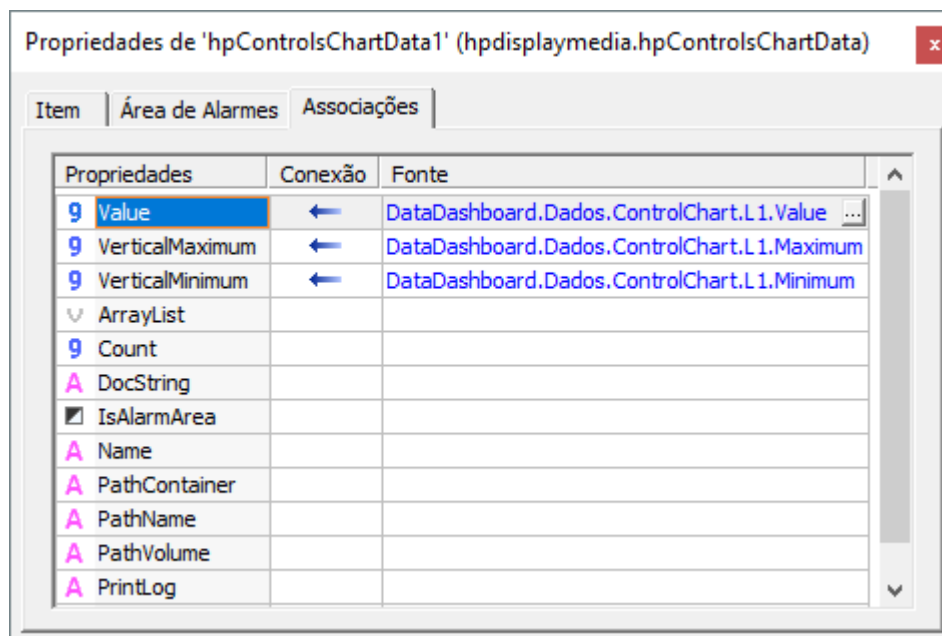
Objeto **hpControlsChart**

Este objeto trabalha em conjunto com um XObject do tipo **hpControlsChartData**, que recebe o valor da variável e armazena seu histórico recente pelo intervalo de tempo definido pelo usuário através da propriedade **TimeInterval**. Estes objetos de dados estão situados no item **Objetos de Servidor - Objetos de Dados - hpObjects - DisplayMedia**, conforme a figura a seguir.



Pasta **DisplayMedia**

O Tag a ser monitorado deve ser associado à propriedade **Value** do objeto de dados.

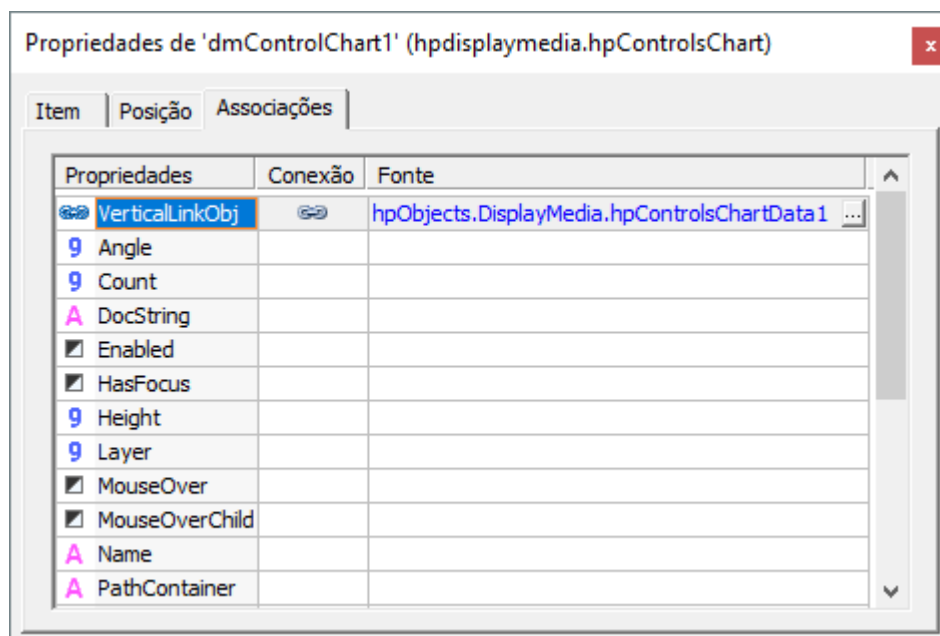


Associação na propriedade Value

As outras propriedades deste objeto são as seguintes:

- **TimeInterval**: Intervalo de tempo, em segundos, que é usado para visualização e armazenamento dos dados
- **VerticalMaximum**: Limite máximo do eixo Y do gráfico
- **VerticalMinimum**: Limite mínimo do eixo Y do gráfico

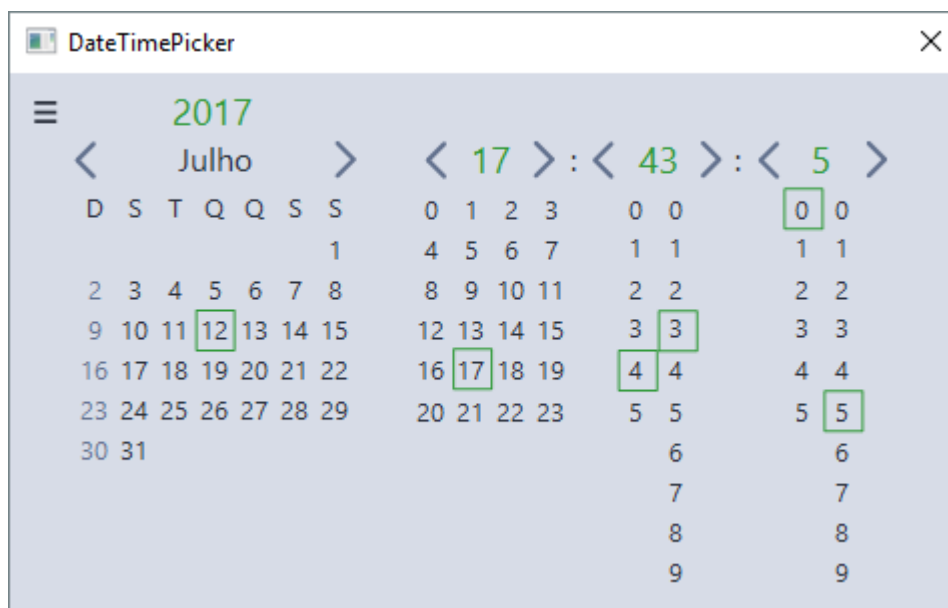
Internamente este objeto armazena o valor recebido pela sua propriedade **Value**, monta um *buffer* de dados históricos e o envia para a instância do gráfico em Tela, que processa seus valores a fim de que sejam plotados. Desta forma, o objeto **hpControlsChart** deve ser instanciado em Tela e associado ao seu respectivo XObject, conforme a figura a seguir.



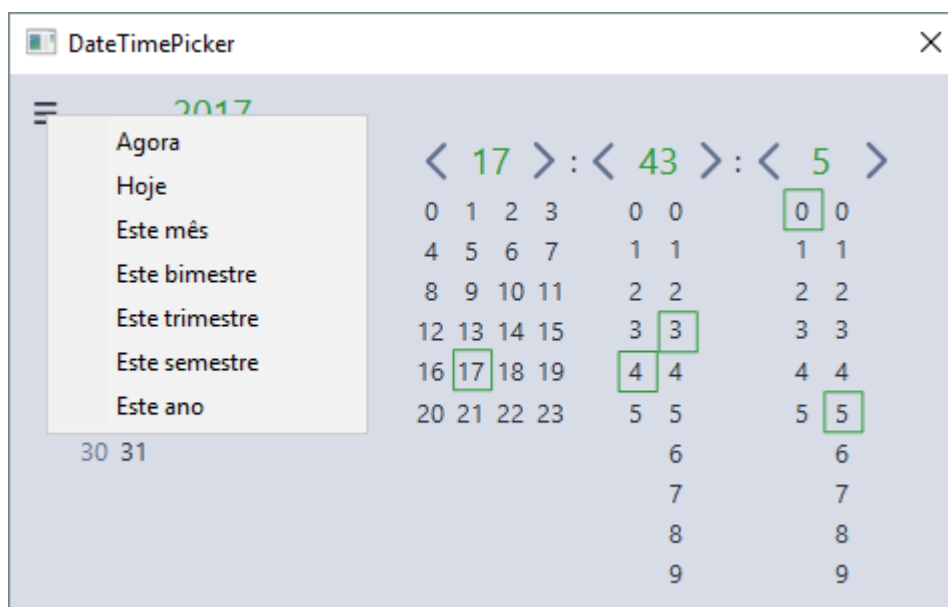
Propriedade VerticalLinkObj

O *buffer* de valores históricos é armazenado no servidor e transferido ao Viewer apenas ao iniciar a instância do gráfico, por exemplo ao abrir uma Tela. Após instanciar o gráfico, os dados são atualizados em tempo real.

O objeto **hpDateTimePicker** é o seletor de data e hora do **HighPerformance Template**, conforme a figura a seguir.

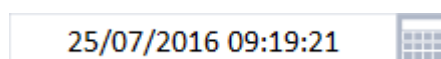


Objeto hpDateTimePicker



Seleção de um período

Para usá-lo, instancie em Tela o objeto **hpDateTimeCombo**, conforme a figura a seguir.



Objeto hpDateTimeCombo

Este objeto tem as seguintes propriedades para configuração:

- **AutoApply:** Se configurada para Verdadeiro, aplica as alterações automaticamente. Se configurada para Falso, é necessário clicar em **Aplicar** para validar as alterações
- **IconGallery:** Nome do arquivo da Biblioteca onde está contido o ícone usado no objeto
- **IconName:** Nome do ícone exibido no objeto
- **ShowIcon:** Se configurada para Verdadeiro, exibe o ícone no objeto

- **Value:** Valor de data e hora selecionado

Caso seja necessário usar uma configuração diferenciada, é possível usar o objeto **hpTextBox** alterando a sua propriedade **Behaviour** para 1 (um).

A Biblioteca **hpMenus** possui os objetos **hpLinkMain**, **hpLinkDetail** e **hpLinkIcon**, que podem ser usados individualmente ou em grupo para compor uma barra de menus.

4.1 Associação Simples

Para uso individual dos objetos de menu como Associações simples, siga estes procedimentos:

1. Instancie o objeto na Tela.
2. Desabilite o recurso de seleção, configurando a propriedade **DisableSelection** do objeto para Verdadeiro.

4.2 Menus em Grupo

Para usar dois ou mais objetos em grupo como uma barra de menu, onde um dos objetos fica sempre selecionado, siga estes procedimentos:

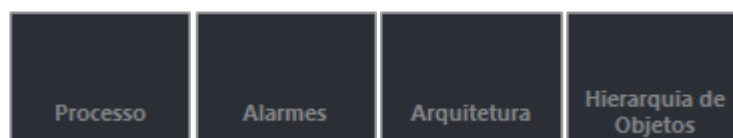
1. Instancie os objetos diretamente na Tela ou em grupos.
2. Configure a propriedade **GroupName** dos objetos com o mesmo valor, para que todos pertençam ao mesmo grupo lógico.
3. Configure a propriedade **DisableSelection** de todos os objetos do grupo para Falso, para habilitar a seleção de cada objeto.

4.3 Alinhamento Automático

Os objetos **hpLinkMain**, **hpLinkDetail**, **hpLinkIcon** e **hpScreenNav** contam com um recurso de alinhamento automático, desde que pertençam ao mesmo grupo lógico configurado em suas propriedades **GroupName**. Com este recurso é mais fácil posicionar os objetos com alinhamento e distribuição uniforme para que formem um menu.



Objetos sem alinhamento e com distribuição irregular



Objetos alinhados e distribuídos horizontalmente

Para aplicar este recurso, siga estes procedimentos:

1. Instancie cada objeto na Tela.
2. Configure a propriedade **GroupName** de cada um dos objetos para um mesmo valor.
3. Configure a propriedade **GroupOrientation** de cada um dos objetos em **1** (um) para alinhamento horizontal ou em **2** (dois) para alinhamento vertical.
4. Configure a propriedade **Margin** com o valor desejado para o espaçamento entre os objetos.

5. Configure a propriedade **GroupElementID** de cada objeto, iniciando pelo valor **1** (um) no objeto que inicia o menu e aumentando sequencialmente para os demais objetos. É importante verificar que o objeto cuja propriedade **GroupElementID** seja igual a **1** (um) deve estar na posição exata onde o menu inicia.
6. Clique com o botão direito do mouse no objeto cuja propriedade **GroupElementID** é igual a **2** (dois) e selecione a opção **Configurar**. O objeto é então posicionado de forma a estar alinhado com o objeto anterior.

4.4 Propriedade Behaviour

A propriedade **Behaviour** permite alterar o comportamento padrão do objeto, de acordo com o valor configurado pelo usuário.

Propriedade Behaviour igual a 0 (zero)

Neste caso, que é o padrão, a funcionalidade do objeto é configurada pelo usuário, incluindo a execução de scripts quando a propriedade **EventClick** alterar o seu valor. As seguintes situações podem ser configuradas:

- Se a propriedade **DisableSelection** é igual a Verdadeiro, (o objeto não pode ser selecionado), um clique do mouse apenas altera a propriedade **EventClick**
- Se a propriedade **DisableSelection** é igual a Falso, o objeto está selecionado e a propriedade **EnableClickWhenSelected** é igual a Verdadeiro, um clique do mouse apenas altera a propriedade **EventClick**
- Se a propriedade **DisableSelection** é igual a Falso, o objeto está selecionado e a propriedade **EnableClickWhenSelected** é igual a Falso, um clique do mouse não tem nenhum efeito
- Se a propriedade **DisableSelection** é igual a Falso e o objeto não está selecionado, desmarca a seleção de todos os objetos que tenham o mesmo valor na propriedade **GroupName**, seleciona o objeto clicado (para assegurar o funcionamento correto do recurso) e altera a propriedade **EventClick**

Propriedade Behaviour igual a 1 (um)

Neste caso, a funcionalidade do objeto é a de exibir um menu para a seleção do Tema.

Propriedade Behaviour igual a 2 (dois)

Neste caso, a funcionalidade do objeto é a de exibir um menu para a seleção do idioma, caso o módulo **Translator** esteja em uso.

Propriedade Behaviour igual a 3 (três)

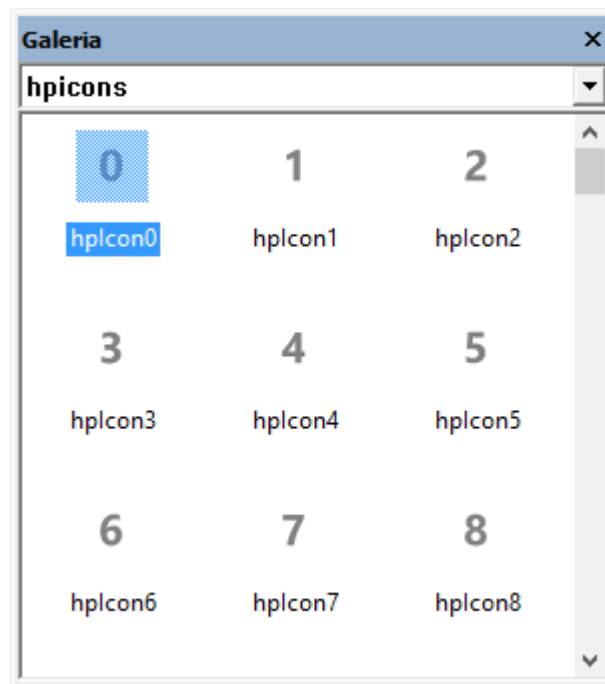
Neste caso, a funcionalidade do objeto é a de abrir as Telas da aplicação, com registro desta atividade no histórico de navegação. Para mais detalhes, consulte o tópico **Telas e Navegação**.

Propriedade Behaviour igual a 4 (quatro)

Neste caso, a funcionalidade do objeto é a de abrir Telas *pop-ups* e modais. Para mais detalhes, consulte o tópico **Abrindo Telas Pop-ups e Modais**.

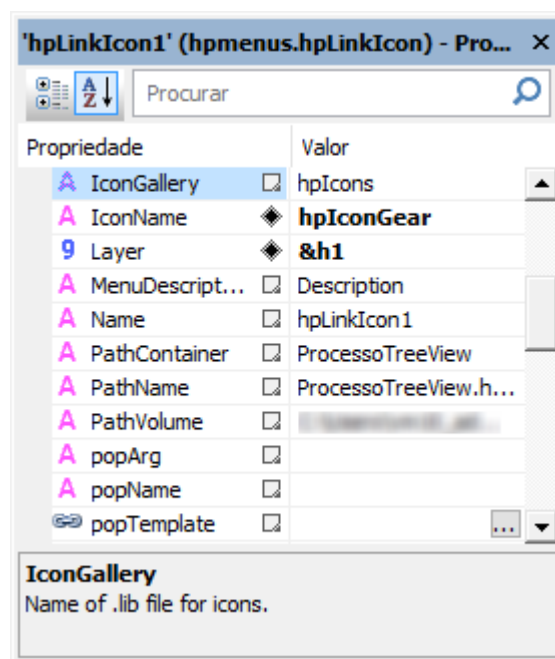
4.5 Adicionando Ícones em um Menu

A Biblioteca **hpIcons** fornece os ícones para os objetos de menus e Associações. Os ícones estão disponíveis na Galeria do Studio, conforme a figura a seguir.



Biblioteca hpIcons na Galeria

Estes ícones foram concebidos para serem usados com os menus. Para isto, selecione um objeto de menu e configure suas propriedades **IconGallery** e **IconName** com o nome da Biblioteca e o nome do ícone, conforme a figura a seguir.



Propriedades IconGallery e IconName

No ambiente do **Studio** o ícone não aparece no objeto, porém em tempo de execução o objeto carrega automaticamente o ícone configurado.

Utilize somente as classes de texto fornecidas com o **HighPerformance Template** a fim de criar uma hierarquia visual consistente em uma aplicação.

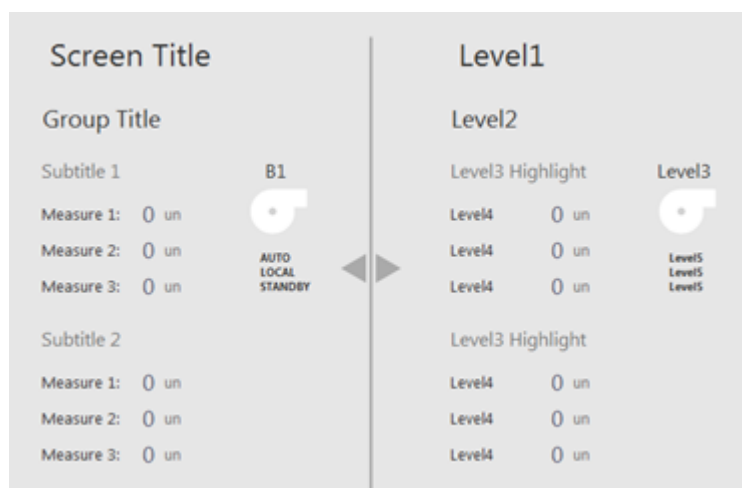
A Biblioteca **hpTexts** possui os objetos a seguir, que definem níveis fixos de categorias de texto:

- **hpLabel0**
- **hpLabel1**
- **hpLabel2**
- **hpLabel3**
- **hpLabel3Highlight**
- **hpLabel4**
- **hpLabel4Highlight**
- **hpLabel5**
- **hpLabel6**

No entanto, uma vez instanciados, estes objetos não podem ter seu tamanho ou categoria alterado. Somente um objeto permite alterar a categoria dinamicamente, o objeto **hpLabelGeneric**.

Alterando-se sua propriedade **Level** é possível mudar o nível hierárquico do texto. Recomenda-se o uso deste objeto ao invés dos demais por permitir maior flexibilidade no caso de ser realizada alguma alteração na aplicação.

A combinação ordenada de objetos de texto com diferentes níveis hierárquicos define a hierarquia visual da aplicação. A seguir são sugeridas duas configurações de uso das classes de texto, dependendo de fatores como resolução, tamanho dos monitores e sua distância do usuário. Os exemplos à esquerda ilustram a função das categorias de texto, e à direita os níveis que devem ser configurados no objeto **hpLabelGeneric**.



Configuração 1



Configuração 2

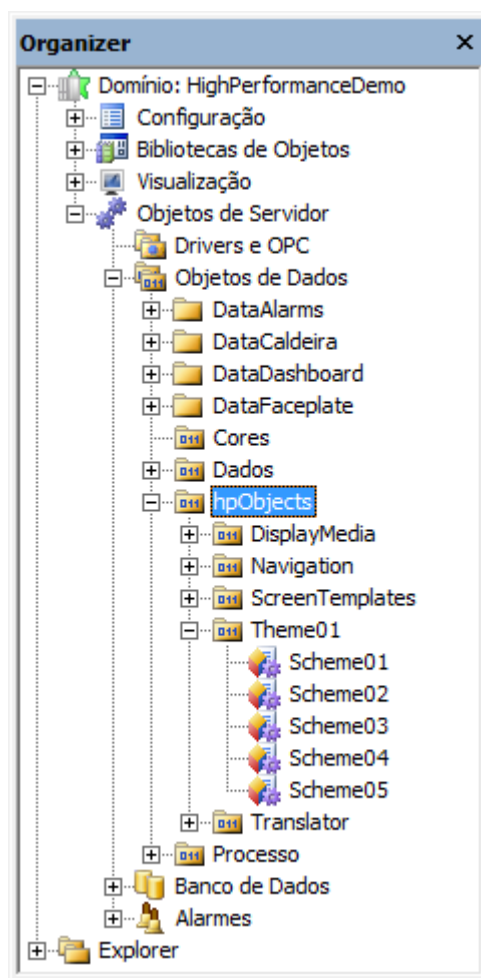
O **HighPerformance Template** possui um único Tema (*Theme*) e seis Esquemas de cores e propriedades (*Schemes*) para este Tema:

- **Acqua**
- **Rock**
- **Ocean**
- **Ground**
- **Sapphire**
- **Lead**

Cada Esquema é carregado individualmente por Viewer. Assim, se o usuário deseja mudar de Esquema, pode-se carregar um novo Esquema no Viewer corrente, sem interferir nos demais Viewers do mesmo Domínio.

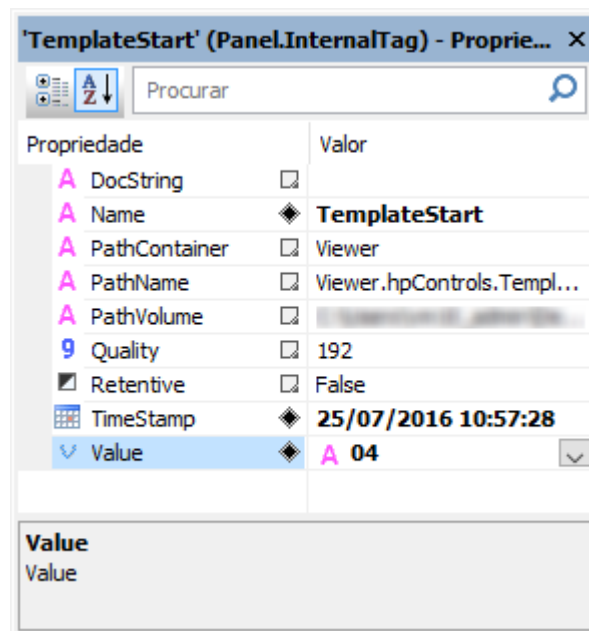
6.1 Configuração de Esquemas ao Iniciar uma Aplicação

Os Esquemas são cadastrados no item **Objetos de Servidor - Objetos de Dados - hpObjects - Theme01** e devem ser denominados segundo o padrão **SchemeNN**, onde **NN** é um sufixo numérico único.



Esquemas do Tema 01

Para que um Esquema inicie com a aplicação, forneça o número (sufixo) do nome do Esquema na propriedade **Value** do Tag **TemplateStart**, localizado em **Viewer.hpControls.Template**.



Tag TemplateStart

Portanto, para carregar o Tema **Sapphire** (*Scheme04*), deve-se digitar o valor "04" na propriedade **Value** do Tag **TemplateStart**.

6.2 Carregando um Novo Esquema

Há duas formas para trocar um Tema, usando os objetos seletores de Temas nativos ou por script.

A primeira forma é a mais simples, instancie em Tela um dos objetos a seguir e configure sua propriedade **Behaviour** para o valor 1 (um).

- **hpLinkDetail**
- **hpLinkMain**
- **hpLinkIcon**

Na segunda forma, digite uma **String** com o valor do sufixo numérico presente no nome do Esquema desejado no Tag **loadScheme**, disponível em **Visualização - Viewer e Quadros - Viewer - hpControls - Template**. Por exemplo, o comando a seguir carrega o **Scheme03** no Viewer.

```
Application.Item("hpControls").Item("Template").Item("loadScheme").Value = "03"
```

6.3 Identificando o Esquema Atual

Para saber qual Esquema está carregado no momento, leia a propriedade **Value** do Tag **TemplateID**, disponível em **Visualização - Viewer e Quadros - Viewer - hpControls - Template**.

```
Application.Item("hpControls").Item("Template").Item("TemplateID").Value
```

6.4 Incluindo um Objeto Criado pelo Usuário em um Tema

O ideal é não precisar criar novos objetos fora das Bibliotecas fornecidas com o **HighPerformance Template**. Mas se for necessário, siga estes passos:

1. Procure por objetos similares nas Bibliotecas de alto desempenho.
2. Verifique as Associações de cores existentes nestes objetos.
3. Copie as Associações equivalentes para o novo objeto, seguindo os mesmos critérios do objeto original, ou seja, cores usadas em bordas são específicas para bordas, cores usadas em preenchimentos são específicas para preenchimentos e cores usadas em textos são específicas para textos.

NOTA

A Biblioteca **hpTexts** já possui uma gama considerável de classes de textos, sendo praticamente desnecessária a criação de novas classes deste tipo.

6.5 Cadastrando Cores em um Tema

O **Tema 1** (*Theme01*) possui vinte campos para cores personalizadas, de **CustomColor1** a **CustomColor20**. No exemplo da aplicação *HighPerformanceDemo*, estas cores são usadas para alterar as cores do logotipo da Elipse Software quando o fundo muda para claro ou escuro.

Fora este caso ou similares, não é recomendável alterar as cores dos Esquemas. Lembre-se que todas as cores disponíveis no **HighPerformance Template** foram pensadas para criar uma interface harmoniosa e segura. Por exemplo, as sete cores **ExtraChartPen** foram estudadas e definidas para permitir, a quem possua daltonismo, a identificação segura das diferentes penas de um gráfico. Alterar um detalhe como este pode significar o empobrecimento da funcionalidade de um recurso.

Outra dica neste caso é de, ao invés de criar cores novas, copiar cores já existentes no Esquema de cor para usos que não foram previstos. Por exemplo, **ColorProcessLine** é a cor usada para as linhas representativas do processo, porém também pode ser usada para outro tipo de objeto ou função de uma aplicação. Neste exemplo ela pode ser copiada para o campo **CustomColor2** e os novos objetos que fizerem uso dela devem apontar para este campo, ao invés de apontarem para **ColorProcessLine**.

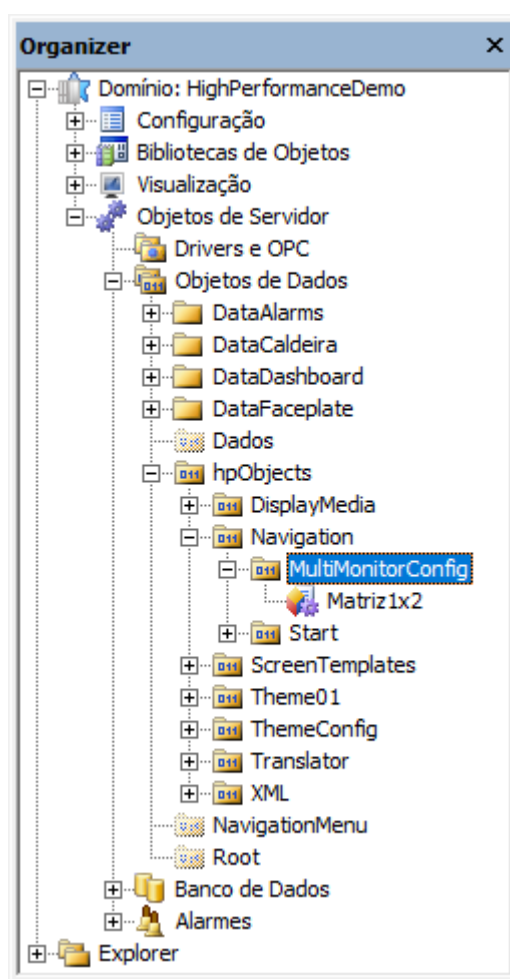
O **HighPerformance Template** contém diversas configurações de Quadros para uso imediato, dentro de uma matriz de Quadros que pode ser personalizada pelo usuário.

Primeiramente, define-se uma matriz de monitores. Depois seleciona-se um *Template* de Quadros para cada parte da aplicação. Os tópicos a seguir descrevem estes passos em detalhes.

7.1 Definindo uma Matriz de Monitores

Definir uma matriz de monitores é o primeiro passo para configurar a visualização da aplicação no **HighPerformance Template**.

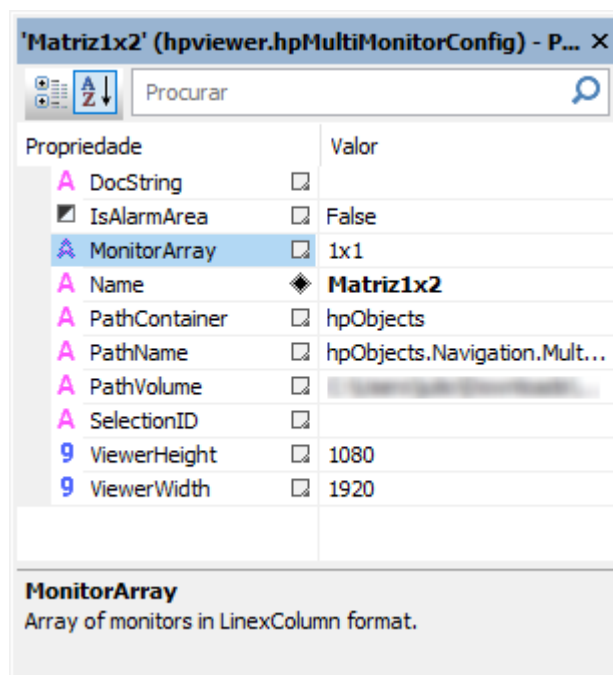
Para isto, acesse a pasta **hpObjects.Navigation.MultiMonitorConfig** e configure os objetos do tipo **hpMultiMonitorConfig** conforme desejado. Crie um objeto para cada estação onde é executado o Viewer.



**Objetos de configuração
hpMultiMonitorConfig**

Este tipo de objeto possui as seguintes propriedades:

- **MonitorArray:** Especifica o array de monitores desejados, no formato de linhas por colunas
- **SelectionID:** Informa o nome da estação
- **ViewerHeight:** Valor da propriedade **Height** do Viewer
- **ViewerWidth:** Valor da propriedade **Width** do Viewer



Exemplo de configuração para
hpMultiMonitorConfig

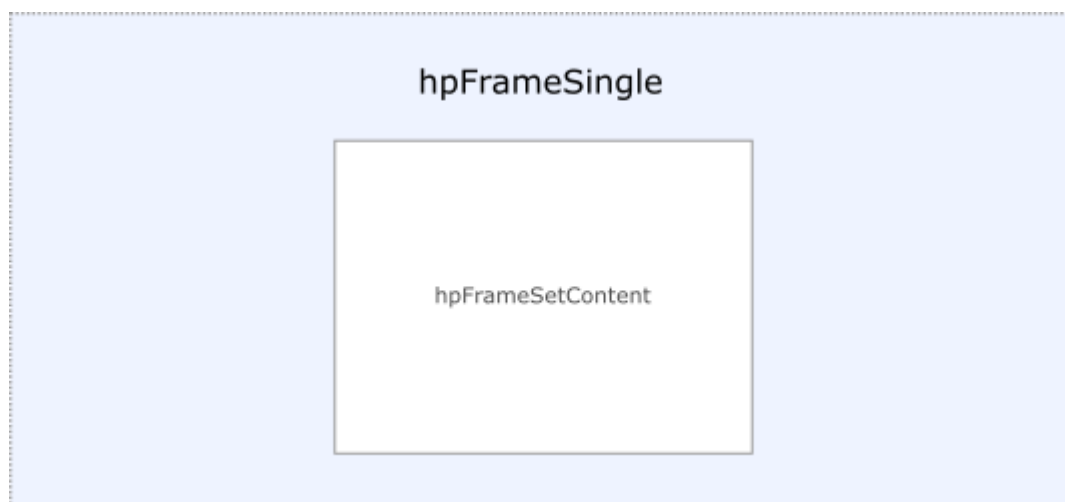
Para visualizar o Viewer em um único monitor, configure a propriedade **MonitorArray** com seu valor padrão "1x1".

O **HighPerformance Template** suporta um total de nove monitores, independente de sua distribuição.

7.2 Selecionando um Template de Quadros

Uma vez definida e configurada a matriz de monitores, deve-se selecionar os *Templates* de monitores a serem usados na aplicação.

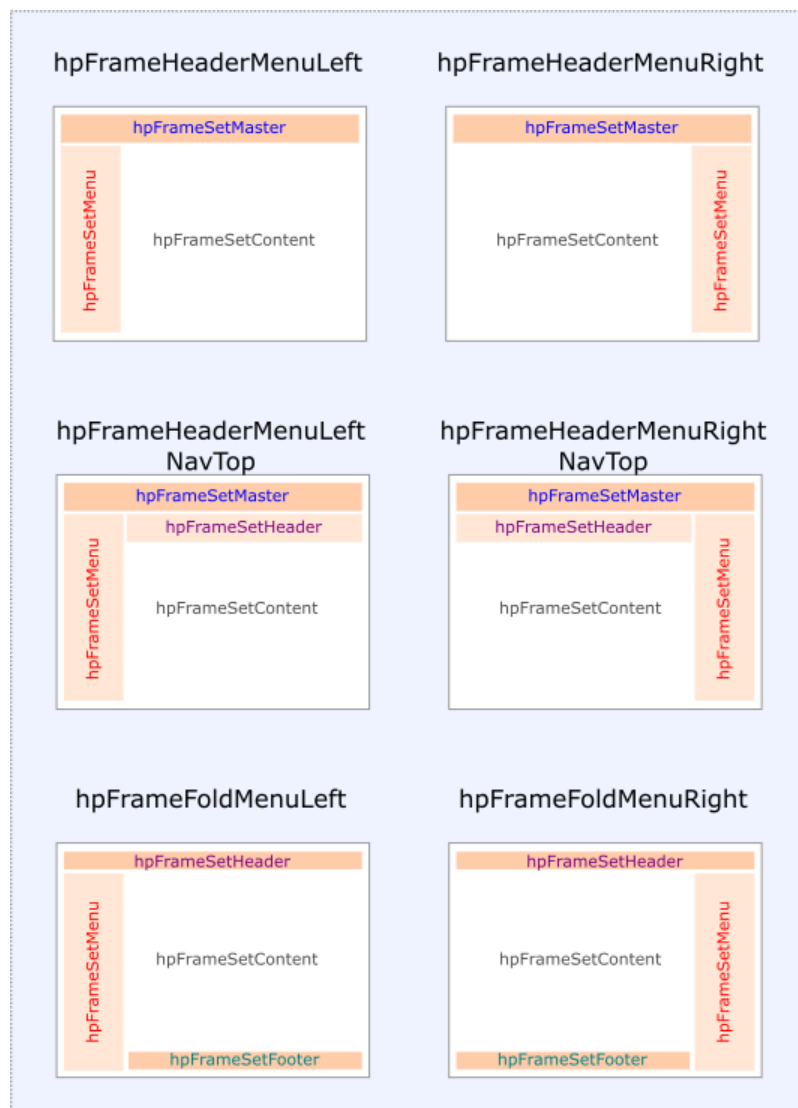
A seguir são mostrados os *Templates* de Quadros disponíveis no **HighPerformance Template**. As figuras estão organizadas por Quadro dominante, isto é, pelo Quadro que atravessa a Tela de uma extremidade a outra.



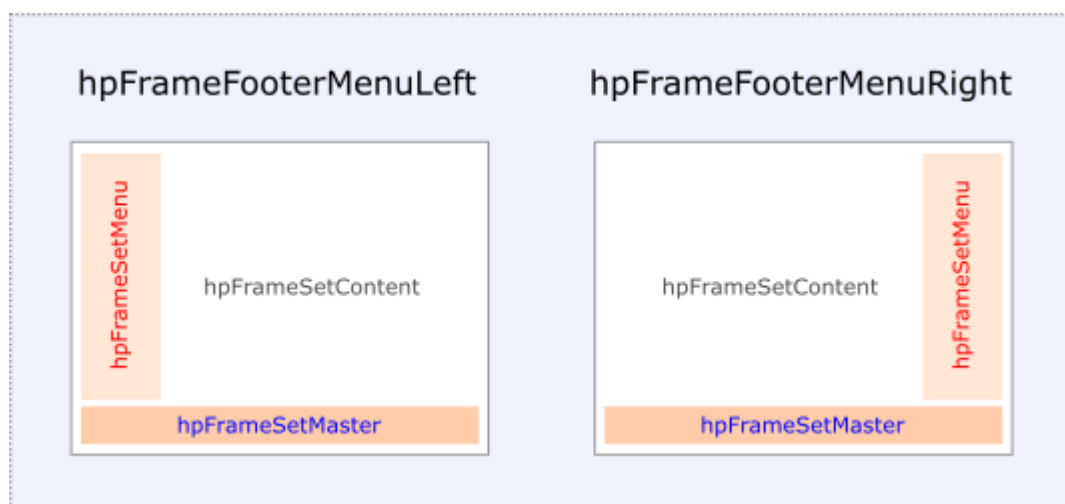
Conteúdo dominante



Quadros superiores e inferiores dominantes



Somente o Quadro superior dominante



Somente o Quadro inferior dominante



Quadro lateral dominante

Algumas considerações sobre os Quadros existentes em cada *Template*:

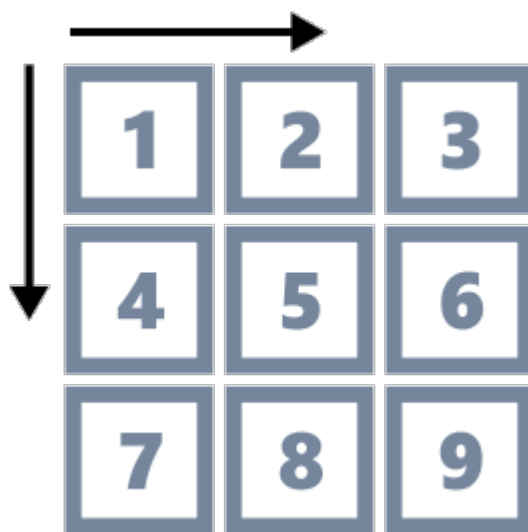
- **hpFrameSetContent:** Presente em todos os *Templates*, representa a área de conteúdo ou a área que recebe a Tela principal da aplicação. O nome da Tela a ser aberta neste Quadro deve ser configurado no Tag **scrContent**, disponível em **Visualização - Viewer e Quadros - Viewer - hpControls - Navigation**
- **hpFrameSetMaster:** Usado em diversos *Templates*, tanto como cabeçalho ou rodapé. Pode ser usado para acomodar um menu, um objeto de alarme, etc. O nome da Tela a ser aberta neste Quadro deve ser configurado no Tag **scrMaster**, disponível em **Visualização - Viewer e Quadros - Viewer - hpControls - Navigation**
- **hpFrameSetMenu:** Concebido para acomodar menus ou árvores de navegação (*TreeViews*). O nome da Tela a ser aberta neste Quadro deve ser configurado no Tag **scrMenu**, disponível em **Visualização - Viewer e Quadros - Viewer - hpControls - Navigation**
- **hpFrameSetHeader e hpFrameSetFooter:** Função idêntica ao Quadro **hpFrameSetMaster**. A diferença aqui é que ambos existem no mesmo *Template*, ao contrário de **hpFrameSetMaster**, que assume o papel de cabeçalho ou

rodapé separadamente em cada *Template*. O nome da Tela a ser aberta nestes Quadros deve ser configurado nos Tags **scrHeader** e **scrFooter**, respectivamente, disponível em **Visualização - Viewer e Quadros - Viewer - hpControls - Navigation**

- **hpFrameSetLeft e hpFrameSetRight**: Função idêntica ao Quadro **hpFrameSetMenu**. A diferença aqui é que ambos existem no mesmo *Template*, ao contrário de **hpFrameSetMenu**, que assumia o papel de menu esquerdo ou menu direito separadamente em cada *Template*. O nome da Tela a ser aberta nestes Quadros deve ser configurado nos Tags **scrLeft** e **scrRight**, respectivamente, disponível em **Visualização - Viewer e Quadros - Viewer - hpControls - Navigation**

Os *Templates* são abertos dentro da matriz de monitores, que pode carregar um ou mais *Templates* simultaneamente. A título de exemplo, uma matriz 1x2 pode carregar o *Template* **hpFrameHeaderMenuLeft** no monitor 01 e o *Template* **hpFrameMenuRight** no monitor 02. Esta configuração é realizada no item **Objetos de Servidor - Objetos de Dados - hpObjects - Navigation - Start** do Organizador. Cada Tag Interno nesta pasta representa um monitor, cujo valor é o nome do *Template* que deve ser aberto na inicialização do Viewer.

A numeração dos monitores cresce da esquerda para a direita e de cima para baixo na matriz de monitores, conforme a figura a seguir.



Ordem da numeração de monitores

7.3 Guia Rápido de Configuração de Quadros

Um roteiro rápido para que uma aplicação passe a usar os *Templates* de Quadros do **HighPerformance Template** seria o seguinte:

1. Configure o Viewer de cada estação para abrir a matriz de monitores contida em **Objetos de Servidor - Objetos de Dados - hpObjects - Navigation - MultiMonitorConfig**.
2. Configure o *Template* que deve ser aberto na inicialização do Viewer para cada monitor em **Objetos de Servidor - Objetos de Dados - hpObjects - Navigation - Start**.
3. No item **Visualização - Viewer e Quadros - Viewer - hpControls - Navigation**, configure os nomes das Telas a ser abertas na inicialização do Viewer nos Tags com os nomes dos respectivos Quadros.

Os objetos **hpLinkMain**, **hpLinkDetail** e **hpLinkIcon** contêm nativamente a funcionalidade de abertura de Telas. É possível abrir uma Tela dentro de um Quadro já carregado no Viewer ou carregar um novo Quadro.

8.1 Abrindo uma Tela em um Quadro já Existente

Para abrir uma Tela em um Quadro já carregado no Viewer, proceda da seguinte forma:

1. No objeto **hpLinkMain**, **hpLinkDetail** ou **hpLinkIcon**, configure o valor da propriedade **Behaviour** para 3 (três).
2. Selecione uma propriedade que corresponda ao Divisor onde a Tela deve ser aberta, **scrContent**, **scrFooter**, **scrHeader**, **scrLeft**, **scrMaster**, **scrMenu** ou **scrRight**. Por exemplo, para abrir uma Tela no Divisor **hpFrameSetHeader**, a propriedade selecionada deve ser **scrHeader**.
3. Configure a propriedade selecionada com o endereço da Tela. Por exemplo, "Tela1?100?0" (sem as aspas).
4. Se for necessário passar algum argumento para a Tela, configure-o na propriedade **scrArg**.
5. Certifique-se de que a propriedade **scrTemplate** esteja vazia.

Neste caso só é permitida a configuração de uma única Tela.

8.2 Abrindo um Novo Quadro

Para abrir um novo Quadro com diversas Telas, proceda da seguinte forma:

1. No objeto **hpLinkMain**, **hpLinkDetail** ou **hpLinkIcon**, configure o valor da propriedade **Behaviour** para 3 (três).
2. Digite o endereço das Telas a abrir nas propriedades que correspondem aos seus respectivos Divisores, **scrContent**, **scrFooter**, **scrHeader**, **scrLeft**, **scrMaster**, **scrMenu** ou **scrRight**.
3. Especifique o nome do *Template* de Quadros a ser aberto na propriedade **scrTemplate**. Por exemplo, "hpFrameHeaderMenuRight" (sem as aspas).
4. Se for necessário passar algum argumento para o novo Quadro, configure-o na propriedade **scrArg**.

8.3 Abrindo Telas e Quadros em Outro Monitor

A propriedade **scrAutoMonitor**, quando configurada para Verdadeiro, detecta em qual monitor ocorreu o clique do mouse, permitindo abrir a Tela naquele monitor. Caso seu valor esteja configurado para Falso, esta detecção automática é desativada, passando a valer o número do monitor configurado na propriedade **scrMonitorNumber**.

Portanto, para abrir uma Tela ou Quadro em um segundo monitor, proceda da seguinte forma:

1. Configure o objeto como indicado nos tópicos anteriores.
2. Em seguida, configure a propriedade **scrAutoMonitor** para Falso.
3. Configure a propriedade **scrMonitorNumber** com o valor do monitor desejado, como por exemplo 2 (dois).

8.4 Usando os Recursos Voltar e Avançar

Utilizando os objetos mencionados anteriormente para abertura de Telas, automaticamente a ação é registrada em um histórico de Telas visitadas. Isto torna possível usar o recurso de voltar ou avançar pelos endereços registrados neste histórico. Para isto, proceda da seguinte forma:

1. Crie uma instância do objeto **hpScreenNav** na Tela.

2. Selecione a propriedade que corresponde ao Divisor onde se deseja abrir a Tela armazenada no histórico, **scrContent**, **scrFooter**, **scrHeader**, **scrLeft**, **scrMaster**, **scrMenu** ou **scrRight**.
3. Ajuste o valor da propriedade selecionada para Verdadeiro.

Isto permite que, ao clicar em um dos dois botões do controle, seja aberta a Tela registrada no histórico do Divisor especificado.

IMPORTANTE

O histórico vale apenas para o *Template* de Quadro carregado no momento. Caso o usuário abra um novo Quadro, o histórico antigo é perdido.

8.5 Abrindo Telas e Templates por Script

Há duas formas diferentes para abrir Telas nos *Templates* do **HighPerformance Template**:

Abrindo uma nova Tela em um novo Template

Tomando o exemplo anterior, para carregar o *Template* **hpFrameHeaderMenuLeft** no Quadro **hpMainFrame01** (monitor 01) abrindo a Tela **TelaProcesso**, execute o script a seguir.

```
Set navFolder = Application.Item("hpControls").Item("Navigation")
navFolder.Item("scrMaster").Value = "TelaCabecalho?100?0"
navFolder.Item("scrContent").Value = "TelaProcesso?100?0"
navFolder.Item("scrMenu").Value = "TelaMenu?100?0"
Application.GetFrame("hpMainFrame01").OpenScreen("hpFrames.hpFrameHeaderMenuLeft01"), 0
```

No Quadro **hpMainFrame02** (monitor 02), para abrir o *Template* **hpFrameMenuRight** execute o mesmo método, definindo primeiro as Telas a serem abertas e depois carregando o *Template*.

```
Set navFolder = Application.Item("hpControls").Item("Navigation")
navFolder.Item("scrContent").Value = "TelaAlarmes?100?0"
navFolder.Item("scrMenu").Value = "TelaMenu?100?0"
Application.GetFrame("hpMainFrame02").OpenScreen("hpFrames.hpFrameHeader02"), 0
```

Abrindo uma nova Tela em um Template já carregado

Para abrir uma Tela em um *Template* já carregado, por exemplo uma Tela chamada **TelaRelatorio** no monitor 01 dentro do *Template* **hpFrameSetContent**, a partir de algum Quadro interno a este *Template*, não é necessário carregar novamente o *Template*. Para isto, abra a Tela usando o script a seguir.

```
Application.GetFrame("hpFrameSetContent01").OpenScreen("TelaRelatorio?100?0"), 0
```

8.6 Abrindo Telas Pop-ups e Modais

Os objetos **hpLinkIcon**, **hpLinkDetail** e **hpLinkMain**, através da propriedade **Behaviour** com valor igual a 4 (quatro), e o objeto **hpPopupOpen** podem abrir Telas *pop-ups* e modais nativamente.

Telas *pop-ups* permitem interagir com a Tela de fundo. Já as Telas modais não permitem interação com outras Telas da aplicação.

A configuração de objetos para abertura de Telas *pop-ups* e modais deve ser executada da seguinte forma:

1. Crie um *Template* de *pop-up* para cada tipo de janela a ser usada.

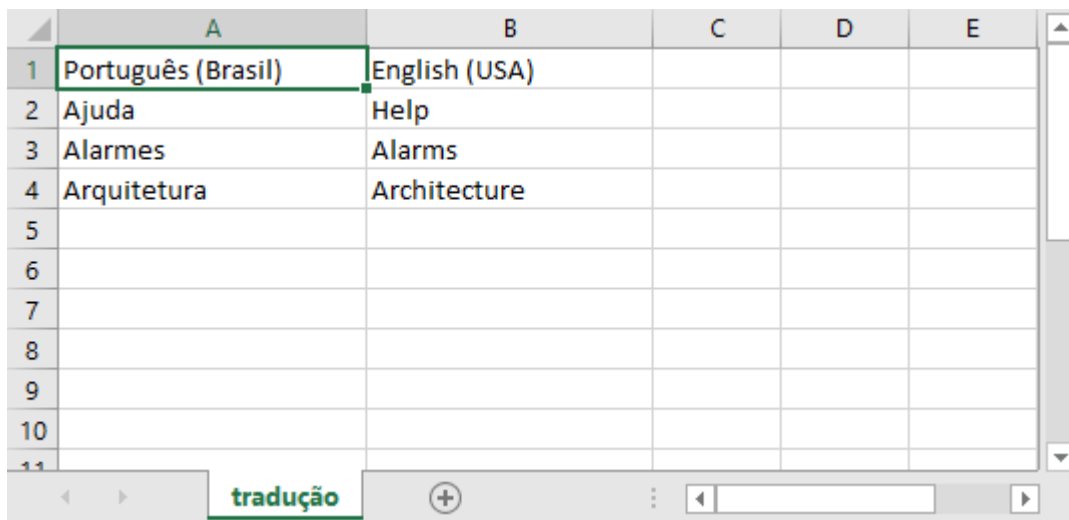
Os *Templates* são alocados no item **Objetos de Servidor - Objetos de Dados - hpObjects - ScreenTemplates**. Nestes objetos é possível configurar diversas propriedades, destacando-se as seguintes:

- **ModalScreen**: Se configurada para Verdadeiro, a janela a ser aberta é do tipo **Modal**. Se configurada para Falso, é do tipo **Pop-up**
 - **Autosize**: Se configurada para Verdadeiro, executa uma leitura da Tela a ser aberta e calcula automaticamente seu tamanho com base nos seus objetos. Neste caso, as propriedades **Width** e **Height** são ignoradas
 - **MarginMode**: Habilitada quando a propriedade **Autosize** está configurada para Verdadeiro. Determina as margens direita e inferior aplicadas ao ajuste automático de tamanho da janela. Baseia-se nas margens superior e esquerda definidas pelo usuário ao posicionar os objetos na Tela. Os valores são calculados em diferentes configurações para que o usuário selecione qual o melhor ajuste:
 - **0**: Margem direita é igual à margem esquerda e a margem inferior é igual à margem superior
 - **1**: Margem direita é igual à margem superior e a margem inferior é igual à margem esquerda
 - **2**: Margens direita e inferior são iguais à margem esquerda
 - **3**: Margens direita e inferior são iguais à margem superior
 - **4**: Margens direita e inferior são iguais à 0 (zero)
2. Instancie um dos objetos **hpLinkIcon**, **hpLinkDetail**, **hpLinkMain** (todos com a propriedade **Behaviour** configurada para 4) ou **hpPopupOpen** (objeto invisível) em Tela.
3. Configure as seguintes propriedades específicas destes objetos:
- **popArg**: Argumento a ser passado à janela
 - **popName**: Nome da Tela, como por exemplo "PopupDemo?100?0" (sem aspas)
 - **popTemplate**: Associação para o respectivo objeto **hpPopupTemplate**
 - **popTitle**: Título da janela

CAPÍTULO 9 Tradução Automática

A Biblioteca **Translator** permite importar um dicionário contendo termos usados na aplicação em vários idiomas, de forma que cada Viewer de um mesmo Domínio possa exibir um idioma distinto, podendo trocá-lo a qualquer momento com um clique do mouse.

O dicionário de termos é um arquivo CSV (*Comma-Separated Values*) localizado na pasta **Dictionaries**. Este arquivo deve conter uma linha de cabeçalho com o nome de cada idioma, e na sequência seus respectivos termos, conforme a figura a seguir.



	A	B	C	D	E
1	Português (Brasil)	English (USA)			
2	Ajuda	Help			
3	Alarmes	Alarms			
4	Arquitetura	Architecture			
5					
6					
7					
8					
9					
10					
11					

Arquivo CSV

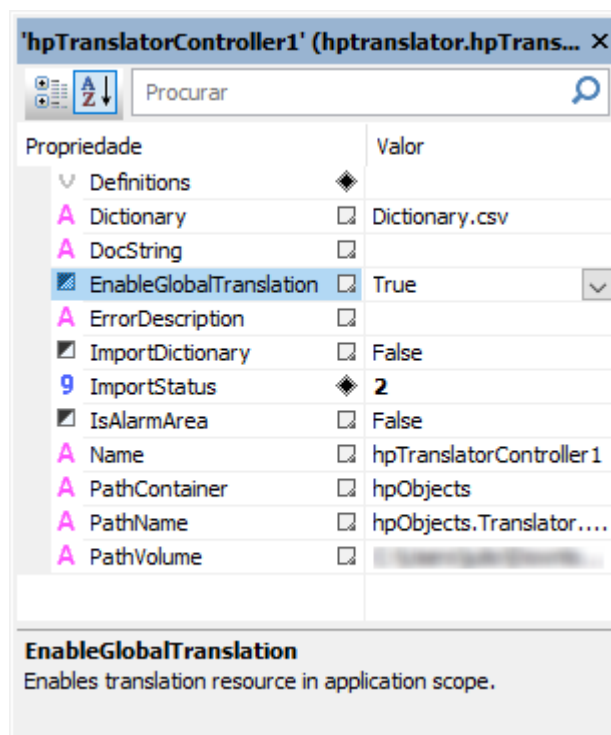
Este dicionário deve ser gerado pelo usuário com os termos a traduzir. A tradução é realizada em todo objeto do **HighPerformance Template** cuja propriedade **EnableTranslation** esteja configurada para Verdadeiro. O número de idiomas (colunas), bem como seu nome, é livre, assim como o número de termos cadastrados.

O idioma especificado na primeira coluna (A) é o idioma principal, a partir do qual todos os objetos da aplicação devem estar configurados. Desta forma, os objetos devem ser configurados no idioma principal no Studio, a fim de que o módulo de tradução possa endereçar corretamente os termos dos demais idiomas.

9.1 Importando um Dicionário

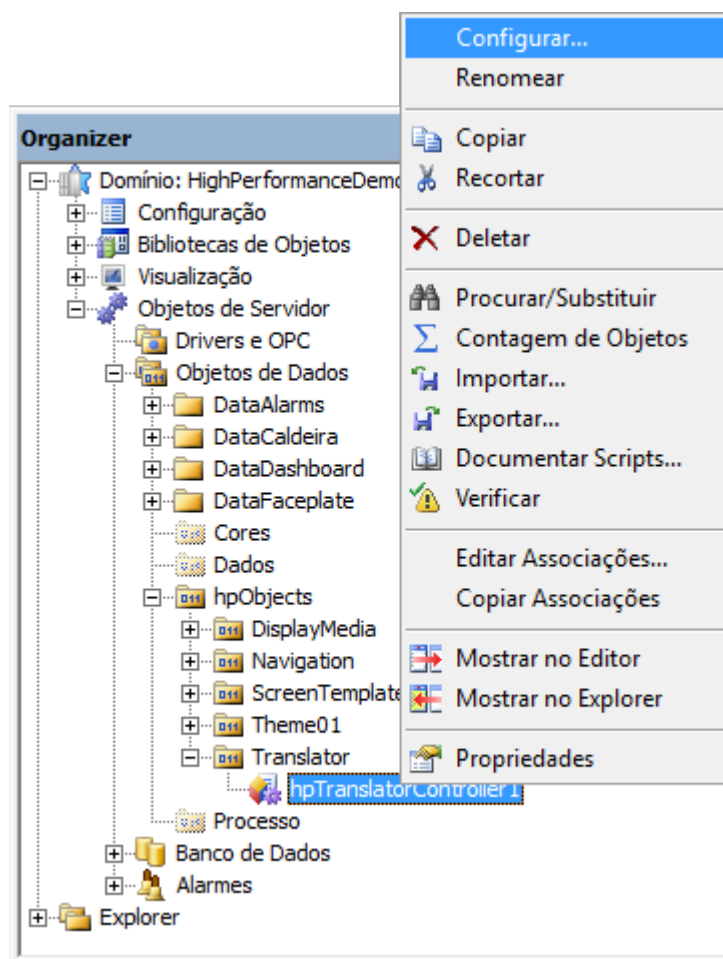
Para importar um dicionário, siga estes procedimentos:

1. Acesse a instância do objeto **hpTranslatorController** localizada no item **Objetos de Servidor - Objetos de Dados - hpObjects - Translator**.
2. Verifique se o valor da propriedade **EnableGlobalTranslation** é igual a **True**. Caso o valor esteja configurado em **False**, o recurso de tradução está desabilitado para toda a aplicação.
3. Especifique o nome do arquivo CSV na propriedade **Dictionary** deste objeto.



Propriedade Dictionary

3. Clique com o botão direito do mouse na instância e selecione a opção **Configurar** para importar o dicionário. Caso a importação seja bem-sucedida, a propriedade **ImportStatus** exibe o valor 2 (dois).



Opção Configurar

4. Salve a Pasta **hpObjects** para concluir a importação.

9.2 Alterando um Idioma em Tempo de Execução

Para alterar um idioma exibido em um Viewer, instancie em Tela um objeto **hpLinkDetail**, **hpLinkMain** ou **hpLinkIcon** e configure a propriedade **Behaviour** para o valor 1 (um).

No Studio também é possível configurar o idioma ao escrever o respectivo índice na propriedade **Value** do Tag **ActiveLanguage**, disponível em **Visualização - Viewer e Quadros - Viewer - hpControls - Translator**. O valor 0 (zero) é o índice para o idioma principal, o valor 1 (um) indexa o segundo idioma, e assim sucessivamente.

O objeto **hpXMLCatalog** lê uma ou mais pastas de objetos e gera o código XML correspondente na propriedade **xmlOutput** no formato **XML Document Object Model** (DOM).

Propriedades do objeto hpXMLCatalog

PROPRIEDADE	DESCRIÇÃO
InitialPath	Lista de pastas a serem lidas para gerar o código XML, separadas por ponto-e-vírgula
GetAllChildren	Retorna os objetos filhos em todos os níveis da árvore. Configurada em False retorna somente o primeiro nível hierárquico
HideDataType	Tipo ou tipos de objetos que se deseja excluir da consulta
HideObjectName	Nome ou nomes de objetos que se deseja excluir da consulta
ShowDataType	Tipo ou tipos de objetos que se deseja incluir na consulta. Um valor vazio retorna todos os tipos de objetos
ShowObjectName	Nome ou nomes de objetos que se deseja incluir na consulta. Um valor vazio retorna todos os nomes de objetos
KeywordProperty	Nome da propriedade, no objeto fonte, que contém a palavra-chave ou palavras-chaves
TreeviewXML	Habilita a geração de código XML com parâmetros específicos para o funcionamento do objeto hpTreeView
ExtraProperties	Parâmetros personalizados para o código XML

Os parâmetros descritos na tabela a seguir são gerados nativamente pelo objeto **hpXMLCatalog** em qualquer configuração.

Parâmetros gerados nativamente

PARÂMETRO	DESCRIÇÃO
H	Número de objetos filhos
D	Valor da propriedade DocString
K	Valor da propriedade especificada em KeywordProperty
N	Nome do objeto
P	Valor da propriedade PathName

PARÂMETRO	DESCRIÇÃO
R	Número sequencial
T	Tipo do objeto
W	-1 : Objeto do Elipse Power; 0 : Não é um objeto do Elipse Power
X	Uso reservado

Se o valor da propriedade **Treeview** é igual a True, são gerados automaticamente os parâmetros descritos na tabela a seguir no código XML.

Parâmetros no código XML

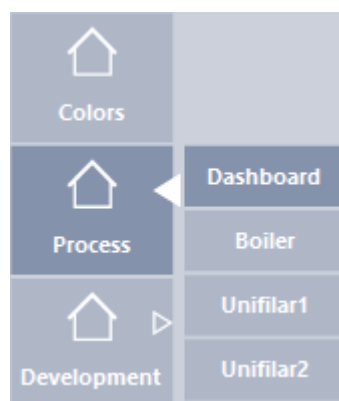
PARÂMETRO	DESCRIÇÃO
I	Valor da propriedade DocString
S	Nó selecionado
E	Nó expandido
L	Nível hierárquico do nó
C	Nó selecionado com a seleção múltipla
Y	Visibilidade do objeto filho ou objetos filhos do nó

Parâmetros personalizados podem ser informados pelo usuário através da propriedade **ExtraProperties**, usando o formato "Propriedade=Código". Podem ser informados diversos parâmetros separados por ponto-e-vírgula, como no exemplo a seguir.

```
Caption=CA;Format=FO;Value=V
```

As propriedades informadas (**Caption**, **Format**, **Value**, etc.) devem existir em todos os objetos fonte. Já os códigos informados podem ter uma ou mais letras (**CA**, **FO**, **V**, etc.) e devem ser diferentes dos demais parâmetros usados pela função. Os parâmetros **LC** e **PP** são reservados para a propriedade **ExtraProperties** e não podem ser usados pelo usuário em outras funções.

O **HighPerformance Template** contém o menu de navegação **Lyra**, com dois níveis hierárquicos (**L1** e **L2**) e orientação vertical.

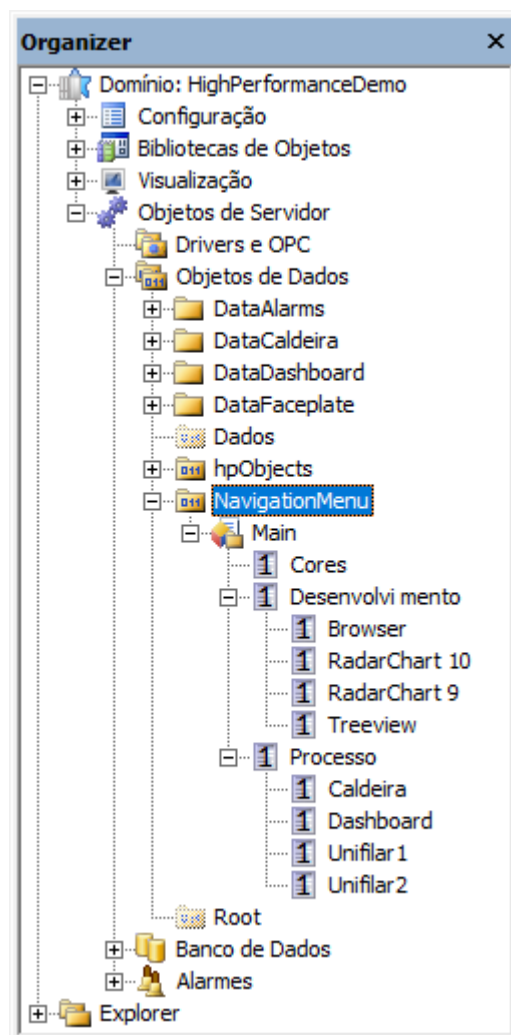


Menu Lyra

Este menu é totalmente integrado ao **HighPerformance Template**, com as seguintes características:

- Suporte a temas
- Integrado ao módulo **Translator**
- Exibe a tela aberta no momento, desde que tenha sido aberta com um objeto nativo do **HighPerformance Template**

O primeiro passo para usar este menu é construir uma estrutura de navegação utilizando objetos **hpNavFolder** como pasta raiz e objetos **hpNavData** como os elementos que devem aparecer no menu, conforme a figura a seguir.



**Objetos hpNavData contidos em
hpNavFolder (Main)**

A configuração de cada objeto **hpNavData** é similar à configuração dos demais objetos de abertura de tela, como **hpScreenOpen** ou **hpPopupOpen**, mas com as particularidades descritas na tabela a seguir.

Particularidades do objeto hpNavData

PROPRIEDADE	DESCRIÇÃO
ShiftL2	Deslocamento vertical dos objetos L2 em relação ao objeto L1 . Se é especificado um valor 0 (zero), então o primeiro objeto filho (L2) é exibido na mesma coordenada vertical do objeto pai (L1)
StartScreen	Especifica que o menu indica a Tela em questão na inicialização

Uma vez que a estrutura esteja pronta, é preciso gerar o código XML correspondente e colocá-lo no respectivo Tag Interno do Viewer, de forma similar ao controle **hpTreeview**. Para o caso específico deste menu, o objeto **hpXMLCatalog** deve ter a propriedade **ExtraProperties** configurada com o valor a seguir.

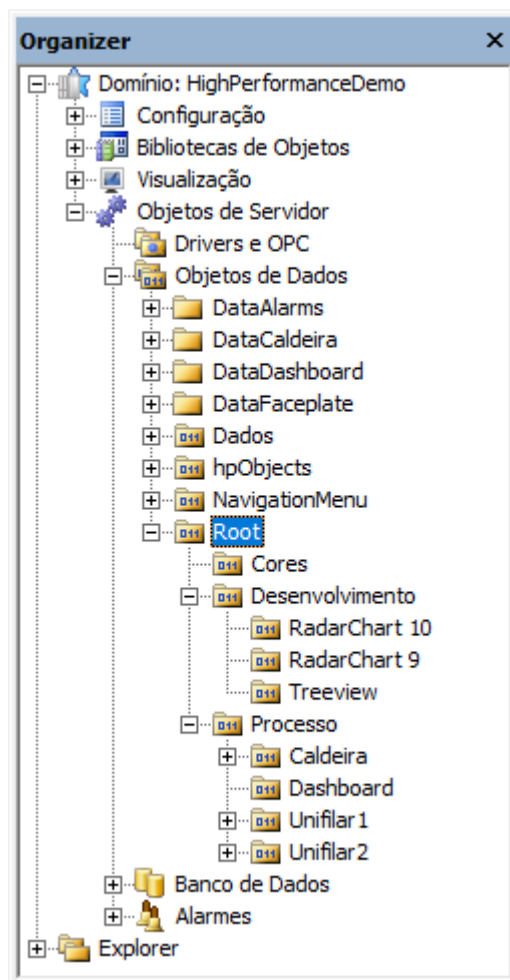
```
StartScreen=S;scrContent=C0;scrFooter=F0;scrHeader=HE;scrLeft=LE;scrMaster=MA;scrMenu=ME;scrRight=RI
```

O próximo passo é instanciar o objeto **hpLyraMenu** na Tela e configurar as propriedades descritas na tabela a seguir.

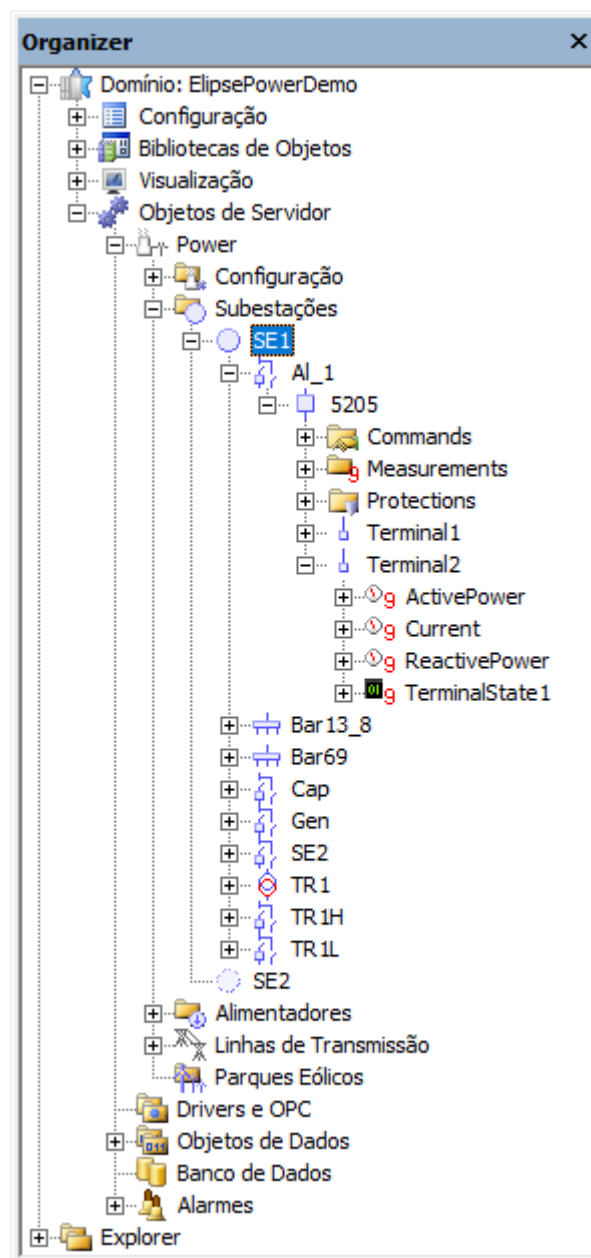
Propriedades do objeto hpLyraMenu

PROPRIEDADE	DESCRIÇÃO
DataFolder	Propriedade PathName da pasta que contém a estrutura de navegação
DocInput	Associação com o Tag Interno que contém a estrutura de navegação no formato XML DOM
ForceReset	Habilita ou desabilita o <i>reset</i> do menu ao ser iniciado
Margin	Espaço entre os controles do menu
RedrawTrigger	Uso reservado do HighPerformance Template
ShowScreen	Propriedade PathName do objeto hpNavData correspondente à Tela que se deseja ver indicada no menu

O controle **Treeview** do **HighPerformance Template** replica a estrutura de um objeto fonte, que pode ser uma Pasta de Dados (DataServer), uma Subestação (PowerSubstation), etc. As figuras a seguir mostram exemplos de estruturas de objetos fonte.

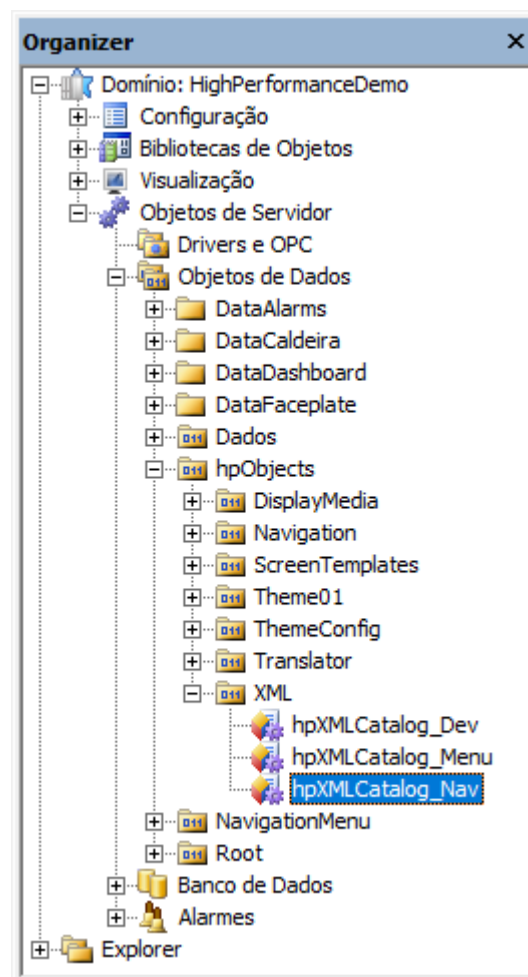


Pasta de Dados

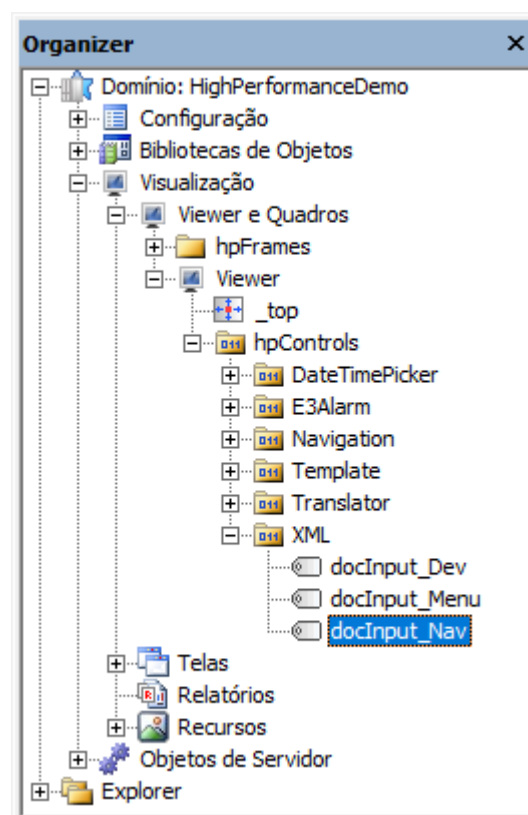


Subestação

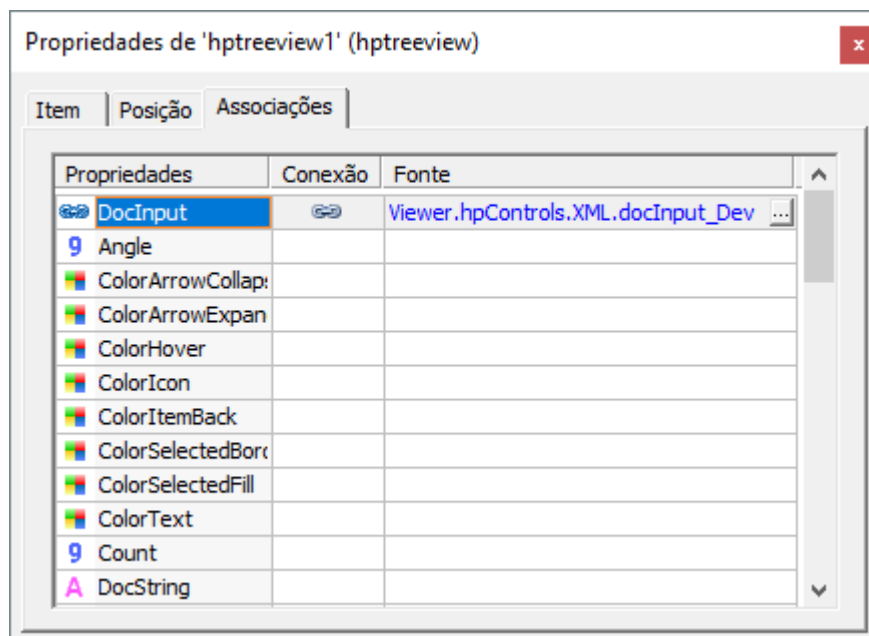
Em linhas gerais, a estrutura com objetos fontes deve ser lida por um objeto gerador de XML, o **hpXMLCatalog** da biblioteca **hpXMLIndexer**. O código XML correspondente à estrutura deve ser repassado a um Tag Interno no Viewer, que o repassa para o objeto **hpTreeview** instanciado em Tela. As figuras a seguir ilustram este fluxo de dados.



Objeto hpXMLCatalog



Tag Interno no Viewer



Associação na propriedade DocInput

Os procedimentos para usar o controle Treeview são os seguintes:

1. Instancie um objeto **hpXMLCatalog** na pasta **hpObjects.XML**.
2. Configure o objeto **hpXMLCatalog**.
3. Copie e renomeie um Tag Interno existente na pasta **Viewer.hpControls.XML**, disponível na aplicação demo do **HighPerformance Template**.
4. Altere o script do evento **OnStartRunning** deste Tag para acessar o objeto **hpXMLCatalog** correspondente.

```
Sub docInput_Nav_OnStartRunning(
    Set docI = CreateObject("MSXML2.DOMDocument.6.0")
    docI.Async = False
    docI.LoadXML(Application.GetObject("hpObjects.XML.hpXMLCatalog_Nav").xmlOutput)
    Value = docI
End Sub
```

5. Instancie em Tela o objeto **hpTreeView**, da biblioteca **hpTreeView.lib**.
6. Associe a propriedade **DocInput** do objeto **hpTreeView** ao Tag Interno correspondente.
7. Execute a aplicação.

O nó selecionado retorna seu caminho (*path*) na propriedade **SelectedNode** do objeto **hpTreeView**.

12.1 Modo HighPerformance ou Convencional

A Biblioteca **hpTreeView** é parte integrante do **HighPerformance Template**, porém pode ser usada fora do *Template*. Para isto, configure no objeto **hpTreeView** a propriedade **HighPerformanceMode** para Falso e ajuste as cores manualmente. Apesar disto, sempre é necessário incluir a Biblioteca **hpTranslator** ao Domínio, pois a Biblioteca **hpTreeView** faz uso de alguns de seus objetos. A Biblioteca **hpTranslator** é um módulo do **HighPerformance Template** que traduz automaticamente a aplicação em tempo de execução, mas que não tem efeito fora do **HighPerformance Template**.

Ao configurar a propriedade **HighPerformanceMode** para Falso, as seguintes propriedades são habilitadas:

- **ColorArrowCollapsed**: Cor do indicador de pasta quando retraído
- **ColorArrowExpanded**: Cor do indicador de pasta quando expandido
- **ColorHover**: Cor usada para indicar que o ponteiro do mouse se encontra sobre o nó

- **ColorIcon:** Cor usada nos ícones do **Eclipse Power** quando a propriedade **PowerIconsAutoShow** estiver configurada em Verdadeiro
- **ColorItemBack:** Cor de fundo de cada nó
- **ColorSelectedBorder:** Cor da borda de um nó selecionado
- **ColorSelectedFill:** Cor de fundo do nó selecionado
- **ColorText:** Cor do texto de cada nó

Ao configurar a propriedade **HighPerformanceMode** para Verdadeiro, o objeto **TreeView** é integrado ao **HighPerformance Template**, herdando todas as suas facilidades. Neste modo são habilitadas as seguintes propriedades:

- **EnableIconBackground:** Desenha um retângulo como fundo de cada ícone
- **EnableTranslation:** Habilita a tradução do objeto em tempo de execução
- **InvertColors:** Inverte as cores dos ícones. Se o ícone exibe o traço em cor clara, passa a exibir o traço em cor escura, e vice-versa

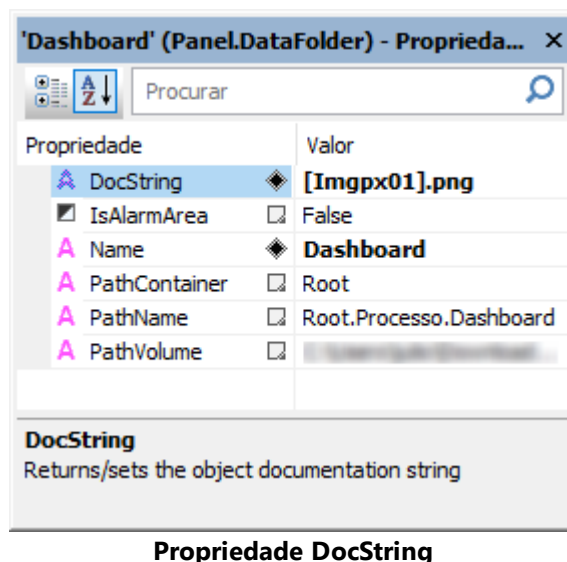
12.2 Exibindo Ícones

Para exibir ícones no objeto **TreeView**, primeiro é necessário alterar o valor da propriedade **ShowIcons** para Verdadeiro. Se o objeto **TreeView** é usado dentro do **HighPerformance Template**, é possível:

- **Exibir ícones do Eclipse Power automaticamente:** Para isto, mude o valor da propriedade **PowerIconsAutoShow** para Verdadeiro
- **Exibir ícones da biblioteca hpIcons:** Para isto, digite o nome do ícone na propriedade **DocString** dos objetos fonte, como por exemplo "hpIconHome"

Se o objeto **TreeView** é usado fora do **HighPerformance Template**, é possível:

- **Exibir ícones do Eclipse Power automaticamente:** Para isto, mude o valor da propriedade **PowerIconsAutoShow** para Verdadeiro. A cor do traço dos ícones deve ser especificada na propriedade **ColorIcon** do objeto **hpTreeView**
- **Exibir ícones gerados pelo usuário:** Para isto, digite o endereço e o nome do ícone na propriedade **DocString** dos objetos fonte. Se o caminho da imagem contém números ou caracteres especiais, deve ser colocado entre colchetes. A figura a seguir mostra um exemplo de referência a uma imagem inserida como Recurso na aplicação



Independente do valor da propriedade **HighPerformanceMode**, quando a propriedade **PowerIconsAutoShow** estiver configurada em Verdadeiro, permite a exibição automática dos ícones dos objetos do **Elipse Power**, ignorando qualquer texto configurado na propriedade **DocString** dos objetos fonte.

12.3 Exibindo a Barra de Rolagem

A barra de rolagem do controle **TreeView** é exibida automaticamente quando o número de nós visíveis excede o limite estabelecido para o objeto, configurado na propriedade **MaxVisibleElements**. Da mesma forma, a barra de rolagem é oculta automaticamente quando o número de nós visíveis é inferior ao definido na propriedade **MaxVisibleElements**. As propriedades a seguir influenciam o funcionamento da barra de rolagem:

- **MaxVisibleElements**: Número máximo de nós exibidos no objeto
- **TreeWidth**: Largura do controle

12.4 Selecionando Múltiplos Nós

Para ativar o modo de seleção múltipla de nós, configure a propriedade **MultipleSelection** para Verdadeiro. Neste modo, cada item do controle **TreeView** pode ser selecionado através de uma caixa de seleção. As seguintes propriedades retornam valores neste modo:

- **SelectionList**: Lista de todos os nós cujas respectivas caixas de seleção estão selecionadas. Ao selecionar um nó pai, todos os seus nós filhos e descendentes são marcados e aparecem nesta lista, independente de estarem expandidos ou não. Da mesma forma, ao desmarcar um nó pai, todos os seus nós filhos e descendentes são desmarcados e removidos desta lista, independente de estarem expandidos ou não
- **ItemCheckedOnClick**: Último nó cuja caixa de seleção foi selecionada através de um clique de mouse
- **ItemUncheckedOnClick**: Último nó cuja caixa de seleção foi desmarcada através de um clique de mouse

12.5 Filtrando o Conteúdo

O objeto **hpTreeView** permite filtrar itens pelo nome ou pelo tipo do objeto fonte e por palavras-chave, utilizando as seguintes propriedades:

- **EnableFiltering**: Habilita ou desabilita o recurso de filtro
- **FilterMode**:
 - **0 - ByName**: Busca somente pelo nome do objeto
 - **1 - ByType**: Busca somente pelo tipo do objeto
 - **2 - ByNameAndType**: Busca tanto pelo nome quanto pelo tipo do objeto
- **FilterCriteria**:
 - **0 - HierarchyBased**: Traz o nó que corresponde ao filtro e junto todos os seus objetos filhos, mesmo que estes não correspondam ao critério definido no filtro
 - **1 - ContentBased**: Traz o nó que corresponde ao filtro e junto apenas os objetos filhos que correspondam ao critério do filtro
- **EnableKeywords**: Habilita ou desabilita a utilização de palavras-chave no filtro

Para filtrar os nós, siga estes procedimentos:

- Certifique-se de que o filtro esteja habilitado, ou seja, que o valor da propriedade **EnableFiltering** seja igual a True
- Para incluir um nó na visualização, digite na caixa de busca localizada sobre a TreeView um texto que esteja contido no nome ou tipo do nó que se deseja exibir. Por exemplo, ao digitar "Break", a TreeView exibe todos os

objetos que contenham a expressão "Break" em seu tipo, como PowerBreaker, ou que contenham a **String** "Break" em seu nome

- Para excluir um nó da visualização, siga o mesmo critério anterior, porém adicionando o sinal "-" à frente do termo. Por exemplo, ao digitar "-Scada" a TreeView deixa de exibir os nós cujo nome ou tipo contenham a **String** "Scada"
- É possível filtrar mais de um termo por vez. Para isto, digite os termos separados por espaço. Por exemplo, ao digitar "TR1 -Operator", o filtro inclui os nós relacionados a "TR1" e exclui da visualização todos os nós relacionados a "Operator"
- É possível também alterar o conteúdo do campo de busca através da propriedade **FilterString** da TreeView
- Se o filtro estiver habilitado e o valor da propriedade **FilterString** é uma **String** vazia, o filtro exibe todos os nós possíveis
- Para cadastrar palavras-chave, consulte o tópico **Conversão de Objetos para o Formato XML**

12.6 Troca Dinâmica da Fonte de Dados

O objeto **hpTreeView** permite trocar dinamicamente sua fonte de dados. Para isto, atribua a propriedade **PathName** do Tag Interno no Viewer à propriedade **DocInput** do objeto **hpTreeview**, conforme o exemplo a seguir.

```
Screen.Item("hptreeview1").DocInput = Application.Item("hpControls")._
Item("XML").Item("docInput_Nav").PathName
```

12.7 Evento Personalizado

É possível personalizar a resposta do objeto ao receber um duplo-clique em um de seus nós com o seguinte procedimento:

1. Configure o valor da propriedade **DoubleClickMode** para **True**.
2. Crie um evento personalizado que responda à alteração da propriedade **EventDoubleClick**.
3. Escreva o script personalizado.

Desta forma, sempre que o usuário clicar duas vezes em um nó, o script do usuário é executado e o nó que recebeu o duplo-clique não muda o estado entre expandido e retraído.

Versão 2.0.16 (08/16/2018)

Pré-Requisitos

RECURSO	VERSÃO
Objeto E3Alarm sensível aos temas	Elipse Power ou E3 versão 4.8.265
Objeto hpLyraMenu	Elipse Power ou E3 versão 4.8.58
Objeto hpChartFull da biblioteca hpEPMChart	Elipse Power ou E3 versão 4.8.58
Dimensionamento automático de janelas <i>pop-ups</i> e modais (método GetScreen)	Elipse Power ou E3 versão 4.7.91

Novas Implementações

Alarmes

- Implementada a indicação de alarmes digitais e analógicos não reconhecidos em Tela. Os objetos afetados são **hpAlarmsIndicator**, **hpCombinedBarLineGraph**, **hpCombinedBarLineGraphIO** e **hpDigitalDisplay**.
- Implementada a indicação de alarme normalizado e reconhecido nos indicadores de alarmes **hpAlarmIndicatorAnalog**, **hpAlarmIndicatorDeadBand**, **hpAlarmIndicatorDigital**, **hpAlarmIndicatorDiscrete** e **hpAlarmIndicatorROC**.
- Criado um novo objeto **E3AlarmToolbarMultipleView** para controle de até três objetos simultâneos do tipo **E3Alarm**.

Visualização de Dados

- Implementado o objeto Rosa dos Ventos (**hpWindRose**) e o objeto de dados associado (**hpWindRoseData**).
- Criado um novo objeto **TreeView**, **hpTreeview**. Devido a esta implementação, a **TreeView** antiga, **xcTVDraw**, é marcada como **DEPRECATED** e vai ser excluída das próximas atualizações.
- O objeto **hpPageReference** ganhou comportamento nativo de abertura de Telas através da propriedade **Behaviour**.
- Implementados os novos objetos **hpRadarChart9Var** (*single zone*) e **hpRadarChart10Var** (*three zones*).
- Criada a propriedade **EnableGlobalTranslation** para habilitar ou desabilitar o recurso de tradução no escopo geral da aplicação.

Ícones

- Criado um ícone representativo para "fogo" na biblioteca de ícones.
- Implementados novos ícones para indicar tipos de arquivos, *links* ou objetos: TAG, HTTP, TIFF, HTML, PDF, TXT, EXCEL, PNG, JPG e RTF.
- Implementado o ícone **hpIconPadLockOpen2**.

- Implementado um ícone para indicar moeda ou valor, **hpIconCurrency**.

Miscelânea

- Criado o ícone **hpArcServerDesktopOutline2D** na biblioteca **hpArchitecture2D**.
- Implementado um menu vertical de dois níveis (Lyra).
- Implementado um objeto gerador de código XML, **hpXMLCatalog**.
- Implementada a biblioteca **hpEPMChart.lib** para visualização de dados provenientes do **EPM**.
- Implementado o recurso de distribuição e alinhamento automáticos em menus para os objetos **hpLinkMain**, **hpLinkDetail**, **hpLinkIcon** e **hpScreenNav**.

Múltiplos Monitores

- Implementado o posicionamento automático da janela aberta pelo objeto **hpPopupOpen**, levando-se em conta a área de clique e os limites da Tela.
- Implementados os *templates* de matrizes de monitores.
- Implementada a propriedade **Behaviour**, que exibe a propriedade **Caption** de Telas nos objetos **hpLabel0**, **hpLabel1**, **hpLabel2** e **hpLabelGeneric**.

Temas

- Implementado um recurso que possibilita inverter automaticamente a convenção de cores que relaciona os status On-Off com os tons de cores Claro-Escuro.
- Implementada uma configuração no objeto **E3Browser** para adequação ao tema.
- Implementado o esquema de cores **Acqua**.
- Implementada uma configuração no objeto **E3Alarm** para adequação ao tema.

Compatibilidade com Versões Anteriores

Devido às melhorias e correções de erros implementadas na atual versão, é possível que surjam algumas ocorrências indesejadas de incompatibilidade com versões anteriores do **HighPerformance Template**. A seguir estão listadas as ocorrências conhecidas.

Alarmes

- O objeto **hpAlarmsIndicator** foi extinto e substituído pelos novos objetos **hpAlarmIndicatorAnalog**, **hpAlarmIndicatorDeadBand**, **hpAlarmIndicatorDigital**, **hpAlarmIndicatorDiscrete** e **hpAlarmIndicatorROC**.
- Os objetos **hpCombinedBarLineGraph**, **hpCombinedBarLineGraphIO** e **hpDigitalDisplay** necessitam de nova configuração de alarmes.

Visualização de Dados

- O objeto **xcTVDraw** foi descontinuado e substituído pelo novo objeto **hpTreeView**.
- O objeto **hpPageReference** pode apresentar diferenças na exibição do texto.
- A cor dos eixos do objeto **hpScatterGraph** pode apresentar diferenças de cor.

Miscelânea

- A estrutura de navegação foi refeita, portanto a navegação pode apresentar incompatibilidades com as versões anteriores.
- O objeto **hpLabelGeneric** não indica mais o tamanho real do texto no **Studio**, devido a questões de performance.
- Pode ocorrer um desalinhamento em objetos do tipo **hpLinkMain** que estejam alinhados.
- Pode ocorrer um desalinhamento em objetos do tipo **hpScreenNav** que estejam alinhados.

Múltiplos Monitores

- As Telas *pop-ups* que abrem nas coordenadas (0, 0) e que tenham a propriedade **flagCenterWindow** igual a **False** passam a abrir na posição determinada pelo clique do mouse.

**Matriz**

Rua 24 de Outubro, 353 - 10º andar
90510-002 Porto Alegre
Fone: (+55 51) 3346-4699
Fax: (+55 51) 3222-6226
E-mail: elipse-rs@elipse.com.br

Filial PR

Av. Sete de Setembro, 4698/1705
80240-000 Curitiba - PR
Fone: (+55 41) 4062-5824
E-mail: elipse-pr@elipse.com.br

Filial RJ

Praia de Botafogo, 300/525
22250-044 Rio de Janeiro - RJ
Fone: (+55 21) 2158-1015
Fax: (+55 21) 2158-1099
E-mail: elipse-rj@elipse.com.br

Filial SP

Rua dos Pinheiros, 870 - Conj. 141/142
05422-001 São Paulo - SP
Fone: (+55 11) 3061-2828
Fax: (+55 11) 3086-2338
E-mail: elipse-sp@elipse.com.br

Filial MG

Rua Antônio de Albuquerque, 156
7º andar Sala 705
30112-010 Belo Horizonte - MG
Fone: (+55 31) 4062-5824
E-mail: elipse-mg@elipse.com.br

Taiwan

9F., No.12, Beiping 2nd St., Sanmin
Dist.
807 Kaohsiung City - Taiwan
Fone: (+886 7) 323-8468
Fax: (+886 7) 323-9656
E-mail: evan@elipse.com.br

Consulte nosso website para informações sobre o representante do seu estado.

www.elipse.com.br

kb.elipse.com.br

forum.elipse.com.br

www.youtube.com/elipsesoftware

elipse@elipse.com.br



Gartner, Cool Vendors in Brazil 2014, April 2014.

Gartner does not endorse any vendor, product or service depicted in its research publications, and does not advise technology users to select only those vendors with the highest ratings. Gartner research publications consist of the opinions of Gartner's research organization and should not be construed as statements of fact. Gartner disclaims all warranties, expressed or implied, with respect to this research, including any warranties of merchantability or fitness for a particular purpose.

Microsoft Partner

Gold Independent Software Vendor (ISV)