



Guia de escrita de documentação para o Firebird

Paul Vinkenoog

26 Mar 2005 – Documento versão 1.1-pt_br

Tradução para o Português do Brasil: Fabrício Araújo

Índice

Introdução	3
Propósito deste documento e público-alvo	3
Pré-requisitos	3
Tópicos discutidos neste documento	3
Onde os documentadores se encontram	4
A página do subprojeto	4
A lista firebird-docs	4
Usando a interface news do servidor Atkin	5
Achar um assunto	5
Preparando-se para escrever: faça um resumo!	6
DocBook XML - uma introdução	6
Uma introdução bem genérica a XML	7
Uma introdução ao DocBook XML	10
Ferramentas para criar documentos DocBook XML	12
Escrevendo seu documento DocBook	13
Criando o documento	14
Digitando texto	15
Elementos que usamos freqüentemente	16
Aspectos do processo de escrita não relacionados com o DocBook	29
Linguagem e estilo	29
Direitos Autorais	30
Usando os documentos do PostgreSQL	31
Adicionando seu documento ao módulo de manuais	32
Pedindo direitos de envio	32
O que se pode ou não fazer assim que você tem direito de envio	32
Enviando seu trabalho	33
Appendix A: História do Documento	34
Appendix B: Aviso de Licença	35

Introdução

Propósito deste documento e público-alvo

Este documento discute os vários aspectos de escrever documentação para o Firebird. É destinado a pessoas que desejam ajudar a escrever documentação para o projeto Firebird, ou quem pelo menos considera seriamente fazer isto. Após esta leitura, você deve o conhecimento necessário para começar a escrever documentos para o Firebird em nosso formato escolhido, o DocBook XML.

Pré-requisitos

Antes de começar a ler este documento, certifique-se que você sabe:

- O que é o módulo de manuais do Firebird
- O que é o CVS, e como se usa o cliente CVS para baixar o módulo de manuais corrente
- Como gerar a documentação atual do módulo de manuais que você baixou

Este conhecimento é essencial se você irá contribuir com o nosso projeto de documentação. Se você se sente inseguro sobre algum desses pré-requisitos, então leia o primeiro o Howto [Gerando as manuais Firebird](#) e depois volte a ler este texto.

Tópicos discutidos neste documento

Começaremos com alguns capítulos curtos sobre:

- A lista de discussão firebird-docs
- Achar um assunto
- Fazer um esboço para o seu documento

Depois disso, iremos gastar algum tempo para explicar os conceitos básicos do DocBook XML, porque este é o formato em que desejamos que você nos entregue os seu(s) documento(s). Os tópicos discutidos incluem:

- O que é o DocBook XML
- Razões para a nossa forte preferência ao DocBook em detrimento de outros formatos
- Ferramentas que você pode utilizar para produzir textos em DocBook

Não se preocupe se o DocBook não significa nada para você ainda: o conhecimento requerido pode ser aprendido em menos de uma hora, e há uma boa chance de que você pode se beneficiar deste conhecimento em outros projeto, quando você tiver que escrever documentação técnica.

A próxima parte sobre escrever os documentos inclui:

- Configurando o próprio documento
- Usando elementos do DocBook
- Algumas notas sobre linguagem e estilo de escrita
- Direitos autorais e a Licença de Documentação Pública
- Usando os documentos do PostgreSQL como base

Finalmente, nós mostraremos como adicionar seu documento finalizado ao projeto Firebird. Os tópicos principais desta seção são:

- Submetendo seus documentos ao módulo de manuais
- Onde pedir para ter direitos de submissão se você não os possui
- O que se pode e não pode assim que você recebeu seus direitos de submissão

Onde os documentadores se encontram

A página do subprojeto

A página do subprojeto de documentação é aqui:

<http://www.firebirdsql.org/index.php?op=devel&sub=doc>

Ela contém novidades sobre nossas atividades, links para os documentos que já foram publicados, planos para o futuro, etc

A lista firebird-docs

Se você está falando sério sobre escrever documentação para o Firebird, a primeira coisa que você deve fazer é assinar a lista de discussão onde nós discutimos nossos planos e nosso trabalho. Esta lista é aberta a todos, e a assinatura não o compromete em nada. Envie um email para:

`<firebird-docs-request@lists.sourceforge.net>`

com a palavra "*subscribe*" ou como assunto ou como a primeira e única linha do corpo da mensagem propriamente dita. Além desta forma, existe outra alternativa: usar o formulário desta página no link abaixo

<http://lists.sourceforge.net/lists/listinfo/firebird-docs>

Independente do meio utilizado, você receberá uma mensagem de email do programa da lista em alguns minutos. Siga as instruções e você está na lista.

Usando a interface news do servidor Atkin

Existe também uma interface news para esta e outras listas de discussão relacionadas ao Firebird. É um pouco problemática no seu funcionamento - em algumas raras situações não funciona - e eu não usaria a mesma para postar mensagens. Mas a grande utilidade desta interface é ótima para os fins de arquivamento e busca. O período de retenção é um pouco longo também, de modo que se você baixar todas as mensagens no servidor terá uma bom pedaço da história da lista antes da sua chegada.

Para acessar essas newsgroup, aponte seu leitor favorito para:

news.atkin.com

E baixe a lista de grupos. Assine os grupos que desejar. Note que a lista firebird-docs é mapeada para o grupo sourceforge.firebird.doc (sem o "s") no servidor news Atkin.

Dependendo do seu navegador ou leitor de news, este links já o coloca direto no grupo:

<news://news.atkin.com/sourceforge.firebird-doc>

Você pode postar no newsgroup até mesmo se você não faz parte na lista de discussão, mas neste caso sua mensagem será retida para ser aprovada por um moderador humano. Isso significa um atraso de um dia (em algumas raras ocasiões, mais).

Achar um assunto

Estas diretrizes talvez possam ajudá-lo a achar um assunto para escrever sobre:

- Primeiro tenha certeza que sabe o que já está disponível - ninguém precisa de um terceiro guia de conversão de MSSQL-para-o-Firebird
- Então pergunte a você mesmo que está faltando, e que seria útil para os usuários do Firebird em geral - ou talvez algum grupo específico
- E também pergunte a você mesmo sobre o que você *gostaria* de escrever. A escolha mais lógica seria um tópico o qual lhe é familiar, mas você pode também pegar um assunto sobre o qual você deseja pesquisar e aprender primeiro (é claro que é bem mais trabalhoso, mas também uma grande experiência de aprendizado se você deseja investir o tempo)
- Você não necessariamente precisa escrever o livro inteiro, guia ou artigo. Talvez já haja pessoas trabalhando numa produção maior, na qual você pode ajudar. Talvez você possa escrever um ou dois capítulos para um livro. Ou talvez você possa suprir material de documentação bruta sobre um assunto que você conheça muito bem.
- Fale sobre suas idéias - ou sua busca por idéias - na lista de discussão firebird-docs. A frequência de postagem pode ser muito baixa às vezes; mas tenha certeza que, se você postar lá, sua mensagem *será* lida, e respondida.

Preparando-se para escrever: faça um resumo!

É sempre uma boa idéia fazer um resumo antes de escrever o texto propriamente dito. Preparar um resumo ajuda você a "ficar organizado"; isso reduz a chance de esquecer alguma coisa vital, e faz o próprio trabalho de escrever muito mais fácil.

Você pode seguir estes passos quando fizer um resumo:

- Defina exatamente o que você deseja que seus leitores aprendam com o seu trabalho.
- Divida a matéria do assunto em unidades lógicas - capítulos e / ou seções e / ou sub-seções.
- Tenha certeza que a ordem das unidades faça sentido, especialmente para um passo-a-passo, tutorial e / ou guia do usuário. Assim: arrume as unidades de forma que se algo tiver de ser feito (ou entendido) primeiro, esse algo apareça primeiro em sua documentação.
- Mostre seu resumo na lista firebird-docs no sourceforge.net e pergunte por comentários.

Quando estiver satisfeito com seu resumo, examine-o inteiramente e decida se você já tem toda informação básica para iniciar a escrita. Numa situação ideal, você deverá contar com toda a informação necessária antes do início da escrita, porque em alguns casos uma peça de informação anteriormente desconhecida pode fazê-lo escolher uma estrutura de documento diferente. De modo que é bem melhor ter esta informação ainda na fase de resumo.

DocBook XML - uma introdução

O formato escolhido para a documentação do módulo de manuais do Firebird é o *DocBook XML*. Para quem não é familiar com XML e / ou Docbook, seguem-se algumas introduções curtas ao XML em geral e DocBook XML em particular. Esteja avisado que estas introduções dão uma imagem grosseiramente simplificada. Mas está tudo bem: não é necessário ser um especialista em DocBook para escrever documentos para o Firebird. O necessário é algum conhecimento básico - o qual você pode depreender em meia hora dos parágrafos abaixo - e uma pequena experiência em aplicar tags do DocBook XML em seus textos (o qual você irá obter assim que começar a escrever).

[Para pular a introdução ao XML](#) se você não precisa saber nada sobre XML, tags, elementos, atributos, renderização e publicação multicanal

[Pule ambas as introduções](#) se já for um autor já experiente com o DocBook.

Nota

Embora nós fortemente insistamos que você ao menos *tente* entregar seu trabalho no formato DocBook, também sabemos que algumas pessoas não tem o tempo necessário para aprendê-lo (ou converter seus documentos existentes para DocBook). Se isso se aplica a você, por favor fale sobre isso na lista de discussão firebird-docs. Nós certamente não desejamos recusar documentação útil apenas porque não está no formato certo.

Uma introdução bem genérica a XML

XML é sigla de *Linguagem de Marcação Extensível* (a sigla está em inglês), o que na verdade significa texto puro com algumas tags de marcação. Um fragmento de texto em XML seria algo como:

```
<paragrafo>
<alto>'Nao!'/> ela gritou. <assustador>Mas a mao sangrenta
<italico>continuou rastejando</italico> em direcao a ela.</assustador>
<figura arquivo="bloody_hand.png" />
</paragrafo>
```

Atributos e tags

No exemplo dado acima, as palavras e frases entre os sinais de “<” e “>” são tags de marcação. `<italico>` é uma *tag inicial*, `</italico>` é uma *tag final*, e `<figura arquivo="bloody_hand.png" />` é uma tag isolada, cuja nomenclatura oficial é *tag de elemento vazio*. Tags em XML são sempre formatadas deste jeito:

Tabela 1. Formato das tags em XML

Tipo de Tag	Inicia com	Termina com
Tag inicial	<	>
Tag final	</	>
Tag de elemento vazio	<	/>

Ainda nos referindo ao exemplo, as palavras `paragrafo`, `alto`, `assustador`, `italico` e `figura` são *nomes de tag*. Na tag `<figura... />`, `arquivo="bloody_hand.png"` é chamado de *atributo*, com `arquivo` sendo o *nome do atributo*, e `bloody_hand.png` sendo o *valor do atributo*. Os valores dos atributos devem sempre estar entre aspas, tanto faz aspas duplas ou simples.

XML permite que você defina do jeito que você desejar, desde que você as construa corretamente. Deste modo `<estatag>`, `<aquelatag>`, e `<isto_nao_e_uma_tag />` são todas tags XML *corretamente formadas*. Por que eu disse “corretamente formadas” em vez de “válidas”? Por causa de que o termo “validar” só usado para algumas implementações especificamente definidas - como por exemplo o DocBook.

Obviamente as tags propriamente ditas não são para aparecer no documento final (isto é, aquele que é apresentado aos leitores). Em vez disso, ele contém instruções que afetam sua aparência. XML, quando usado para escrever documentação, é um típico *formato fonte*, criado para ser processado por software para produzir documentos de saída bem formatados. Este processo é chamado *renderização*.

Algumas tags são, sem sombra de dúvida, instruções de maquiagem:

```
<italico>continuou rastejando</italico>
```

significando que as palavras *continuou rastejando* sejam mostradas em itálico. Porém

```
<alto>Nao</alto>
```

é menos óbvio. Deve a palavra *Não* aparecer em negrito? Ou sublinhada? Ou mesmo em itálico? Ou talvez este texto é para ser lido em voz alta por sintetizador de discurso, e a tag `<alto>` o instrua a le-

vantar a voz? Todas essas coisas são possíveis, e tem mais: freqüentemente um documento fonte em XML é convertido para diferente formatos de saída - exemplos: um arquivo PDF, uma página HTML ou mesmo um arquivo de som. Isto é chamado de publicação multicanal. Neste contexto, `<alto>` pode ser traduzido por negrito no formato PDF; para uma fonte com cor vermelha em negrito numa página web, e no sintetizador signifique um aumento de volume de 50%.

Olhando as outras tags, `<figura.../>` é obviamente uma instrução para inserir a imagem `bloody_hand.png` no documento, e `<scary>`, bem... essa aí é bem menos clara que `<loud>`. Talvez a frase entre as tags `<scary>` tenham que pingar sangue. Ou talvez alguma música assustadora tenha que ser tocada aqui. Isso tudo depende da pessoa que definiu as tags, e do software que eles usam para renderização.

A tag `<paragrafo>`, finalmente, é uma tag estrutural. Ela nos diz algo sobre o lugar que as linhas tem dentro da hierarquia interna do documento. No documento final, parágrafos podem ou não ser separados por linhas vazias. Novamente, tudo depende no software de renderização e possivelmente em alguma opção de configuração. Algumas outras tags estruturais que se pode pensar são, por exemplo, `<capitulo>`, `<seção>`, and `<subdocumento>`.

Caracteres especiais e Entidades

Por causa o caractere '`<`' tem um significado especiais como o início de uma tag, você não pode incluí-lo diretamente, como um valor literal. Em vez disso, se você deseja que os seus leitores possam vê-lo, você digita isso:

```
&lt;
```

É um '&', seguido pelas letras l e t (para *menos que*, em inglês), seguido de um ponto-e-vírgula. Você pode usar também `>` (maior que) para o símbolo '`>`' embora não seja necessário.

XML tem montes de códigos como estes, eles são chamados *entidades*. Alguns representam caracteres, como `<` e `¨` e alguns servem para outros propósitos completamente diferentes. Mas todos começam com um '&' e terminam com um ponto-e-vírgula.

Mas espere um minuto... Se um '&' marca o início de uma entidade, como você o inclui como um literal no seu texto? Claro que existe uma entidade para isso também:

```
&amp;
```

Então esta linha de XML:

```
Kernigan &amp; Ritchie escolheram '&lt;'' como o operador menos-que para o C.
```

virá no documento final como

```
Kernigan & Ritchie escolheram '<' como o operador menos-que para o C.
```

E aqui vão algumas boas notícias: se você usar algum editor especializado em XML para criar o seu documento, você provavelmente apenas precisará digitar '`<`' e '&' normalmente onde você gostaria que eles estivessem. O editor cuida para que saia exatamente como literal e não como um início de tag ou entidade no XML que será salvo no disco. Neste documento, você encontrará (mais adiante) algumas sugestões de editores de XML/DocBook.

Elementos

Tem mais um conceito importante de XML que você precisa conhecer: o *elemento*. O elemento é a combinação de uma tag inicial, uma tag final respectiva, e tudo que estiver no meio. Esse “tudo que está no meio” é chamado de *conteúdo* do elemento, e pode incluir outros elementos. Elementos são nomeados de acordo com suas tags, de modo que podemos falar em elementos do parágrafo, elementos do itálico, etc.

Nota

Verdadeiramente, elementos são um conceito mais básico que as tags: as tags na verdade são coisas que identificam os elementos. Então é melhor dizer que as tags são nomeadas de acordo com seus elementos. Mas como as tags são mais fáceis de reconhecer que os elementos, eu acabei mostrando-as primeiro a você.

Isto é um elemento:

```
<alto>'Nao! '</alto>
```

Isto também é um elemento:

```
<paragrafo>Este e um elemento contendo <negrito>outro</negrito>
  elemento!</paragrafo>
```

Tags de elemento vazio constituem um elemento por si mesmas. Esses elementos não possuem nenhum conteúdo, porque não possuem um *par* de tags:

```
<figura arquivo="bloody_hand.png" />
```

Importante

Não confunda conteúdo com atributos. Conteúdo ficam *entre* tags e os atributos *dentro* das tags. O tag de elemento vazio do último exemplo possui um atributo, mas não possui conteúdo.

Estou insistindo no conceito de elementos por causa que grande parte da documentação tende a falar em “elementos do capítulo”, “elementos do título”, etc em vez de “tags de capítulo” ou “tags de título”. Embora estes termos possam ser intercambiáveis na maior parte do tempo, existem situações onde é importante saber a diferença.

Conclusão do XML

Bom - isso é tudo que você necessita saber sobre XML. Agora você deve ter uma idéia geral de como um texto em XML se aparenta, o que são tags e elementos, e para quem eles existem. E como eu disse antes, a idéia é supersimplificada mas é suficiente para os nossos propósitos.

Deve-se entender também que escrever puramente em um XML auto-inventado é algo sem muito sentido a não ser que você tenha software que entenda e processe *suas* tags. Senão, como você vai tornar o seu XML em um documento bem formatado e apresentável?

Felizmente, não temos que nos preocupar em desenvolver nossas definições de elementos e software de conversão. Existem alguns tipos de XML formalizados, cada um definindo um número de tags e, igualmente importante, algumas regras sobre como usá-las. DocBook XML é um desses tipos.

Uma introdução ao DocBook XML

DocBook foi projetado tendo em mente a escrita de documentos estruturados usando SGML ou XML (mas não se preocupe com SGML - nós utilizamos a variante XML). É particularmente interessante para escrever livros técnicos e artigos, especialmente em assuntos relacionados a computadores. DocBook XML é definido em uma DTD (*Document Type Definition* ou Definição de Tipo de Documento): um conjunto de definições e regras descrevendo exatamente como um documento DocBook válido é estruturado. Este formato está se tornando um padrão de fato para documentos técnicos para computação, e é suportado por um crescente número de aplicações e ferramentas.

Características do DocBook XML

Características importantes do DocBook - ao invés de um XML genericamente - são:

- A DTD do DocBook define um número limitado de tags, e dá regras exatas em como usá-las: quais atributos são válidos para a tag A, que elemento B pode ser aninhado dentro do elemento C, e por aí vai. Usando tags indefinidas, ou não seguindo as regras, seu documento não é DocBook mais (e as ferramentas que suportam o DocBook provavelmente não serão capazes de processá-lo sem erros)
- As tags do DocBook trabalham com estrutura e semântica (significado), nunca cosmética. No DocBook, você achará tags estruturais como `<book>`, `<part>`, `<chapter>`, `<section>`, `<para>`, `<table>`; e tags semânticas como `<filename>`, `<warning>`, `<emphasis>`, `<postcode>`; mas nada como ``, `<bold>`, `<center>`, `<indent>`, `<backgroundcolor>` – ou seja, nada que tenha a ver com layout ou cosmética.
- Por causa disso, uma decisão deve ser tomada em como as tags do DocBook serão transformadas em maquiagem apresentável. Esta decisão (ou melhor: as regras de renderização) poderiam ser codificadas direto nas ferramentas mas isso tornaria tudo muito inflexível. Este é o motivo pelo qual estas regras são definidas em *folhas de estilo* (*stylesheets*, no original). Uma folha de estilos é um documento, usualmente em texto puro, que diz às ferramentas coisas como:

"Imprima os títulos dos capítulos numa fonte preta de 24 pontos; inicie cada capítulo numa nova página; use itálico para ênfase; renderize avisos com uma fonte vermelha 12 pontos, em negrito; use caixa baixa para acrônimos; etc,etc"

Esta abordagem permite que o usuário altere as folhas de estilo se ele (ou ela) não gosta do resultado final na aparência do documento. Isto seria muito mais difícil - ou mesmo impossível - sem ter que alterar as próprias ferramentas

Benefícios do DocBook XML

DocBook tem uma monte vantagens para qualquer quem escreve documentação técnica. Embora nenhum desses benefícios seja único do DocBook, ele possui todos e é largamente suportado. Isto o faz a escolha perfeita para a nossa documentação do Firebird.

Cuidado

Esta lista de benefícios está ligeiramente diferente da versão original do autor.

Para nós, estas são as mais importantes:

- A primeira vantagem do DocBook é inexistência de informação de apresentação. Com isso temos apenas estrutura e *conteúdo*, tornando o documento-fonte independente dos formatos a serem acessados pelos leitores finais.
- Além disso, o fato de DocBook estar focado na estrutura do documento, torna a tarefa de transferir um rascunho de assuntos para a estrutura do documento final muito mais fácil.
- DocBook é um padrão aberto. Com isso, um documento fonte pode gerar documentos finais em qualquer sistema operacional que tenha ferramentas disponíveis para isso. E também permite que um documento fonte seja gerado, por exemplo, em Mac OS X e tenha seus documentos finais renderizados em Windows. Ao baixar o módulo de manuais do Firebird, são incluídas ferramentas de renderização em PDF e HTML (tanto para Linux e Windows) - e no futuro poderemos abranger mais plataformas conforme seja necessário.
- Integração simplificada com o CVS, uma vez que o DocBook é baseado em texto puro, com isso habilitando-nos a utilizar diversas funcionalidades do CVS que só são utilizáveis com texto puro.
- Simplifica enormemente o gerenciamento da documentação, devido ao fato que as regras de renderização são centralizadas e com isso todos ganham. Ninguém necessita ficar seguindo regras chatas de formatação e as decisões sobre a identidade visual dos documentos ficam todas armazenadas nas stylesheets (folhas de estilos). Com isso todos ganham: os escritores se preocupam com o conteúdo e os gerentes da documentação com as regras de formatação e renderização.

Documentação sobre DocBook na Internet

Aqui estão alguns links em caso você deseje mais informações sobre o DocBook (todos em inglês):

- <http://opensource.bureau-cornavin.com/crash-course/>

Writing Documentation Using DocBook - a Crash Course por David Rugge, Mark Galassi e Eric Bischoff. Um ótimo tutorial, mesmo que a maior das ferramentas usadas não são as que nós usamos.

- <http://docbook.org/tdg/en/>

DocBook - The Definitive Guide, por Normam Walsh e Leonard Mueller. Não espere seja um tutorial para o iniciante - de fato, a primeira parte é meio assustador se você é um novato em DocBook. A razão do mesmo estar listado aqui é o fato de este site ser uma excelente referência online para os elementos.

- <http://www.tldp.org/HOWTO/DocBook-Demystification-HOWTO/>

O *DocBook Demystification Howto* é interessante que caso você deseje conhecer um pouco mais sobre XML e DocBook do que é mostrado aqui. Possui também um monte de material sobre SGML, e - de novo - ferramentas que não são utilizadas no subprojeto de documentação do Firebird.

- <http://sourceforge.net/projects/docbook>

O projeto open source DocBook no SourceForge.

Se você conhece algum outra fonte de informação onlie, deixe-nos saber postando uma mensagem na lista de discussão firebird-docs

Ferramentas para criar documentos DocBook XML

Por causa que DocBook é um formato não-binário, você pode usar qualquer editor de texto puro como emacs, pico, Bloco de Notas do Windows, vi, etc para escrever sua documentação. E alguns escritores de documentação realmente preferem essa abordagem em vez de ferramentas mais sofisticadas por que isso lhes dá controle completo do texto, e as tags digitadas são sempre visíveis. Mas o preço disso é que os editores de texto puro não podem *validar* seu documento em DocBook, de modo que somente após a geração dos documentos finais falhar é que você se dará conta do problema. E as estrutura do seu documento - especialmente quando o mesmo é grande - é também difícil de visualiza em modo texto, embora um uso consistente de indentação possa melhorar muito a situação.

Se você deseja seguir por este caminho ou apenas tentar para ver como funciona, seria uma boa idéia de um editor que permita realce de sintaxe no XML. Um muito bom, e também grátis, é o ConText. Obtenha-o em <http://www.fixedsys.com/context/>.

Mas para a maioria das pessoas é desejável o uso de uma ferramenta dedicada. Ferramentas de elaboração de XML geralmente possuem interfaces gráficas que fazem as tags sobressaírem muito bem (e algumas vezes irritantemente); muitas permitirão que recolham e expandam elementos de modo que você possa ver a estrutura do seu documento e ao mesmo tempo fazer zoom no elemento que você está trabalhando; podem também você trocar diferentes visões. Algumas tem um modo DocBook especial, onde seu documento será validado não somente como XML, mas também validado como DocBook.

O autor deste documento tentou diversas dessas ferramentas (algumas grátis, outras baratas e versões de avaliação) e descobriu que o XMLMind XML Editor (XXE) como sendo o mais útil. Essa é uma opinião pessoal, claro; sua experiência pode ser diferente.

Nota

Do Tradutor: Faço coro com o autor com relação ao XXE. Excelente editor, mas tenha um outro editor de XML à mão para editar o arquivo firebirddocs-pt_br.xml.

Alguns editores que você talvez deseje avaliar:

- XMLMind XML Editor, ou XXE. A Edição Standard é grátis.

<http://www.xmlmind.com/xmleditor/>

Roda em: Linux, Windows, Mac OS X. Requer Java, mas você precisa do Java de qualquer maneira ou você não será capaz de gerar os documentos a partir dos fontes – veja o Howto [Gerando as manuais Firebird](#)

Funcionalidades: Visão em árvore (todos os elementos recolhíveis) e visão estilizada (Capítulos e seções recolhíveis). A última é aquela na qual eu trabalho: mostra o documento num simples (mas limpo) layout imitando processador de texto, definido numa stylesheet que vêm com o programa. Ambas as visões podem estar ativas simultaneamente. O modo DocBook não deixa você entrar nada inválido. Seletor de elementos. Editor de atributos. Funções de busca e edição. Barra de botões para criar elementos mais utilizados como seções, listas, tabelas, etc. Seletor de caracteres especiais. Corretor ortográfico. O que eu sinto falta é uma visão de XML em texto puro.

- epcEdit. € 89 para uso não-comercial. Versão de avaliação (60-dias) disponível.

<http://www.epcedit.com>

Rode em: Linux, Windows, Solaris. Requer Tcl/Tk 8.1 ou acima (incluído na instalação).

Funcionalidades: Pannel da Árvore de estrutura. Seletor de Elementos. Editor de Atributos. Pannel de documento pode intercambiar com modo texto puro e modo XML gráfico. Não possui nenhum modo especial para DocBook, mas pode validar qualquer documento XML baseado sobre uma DTD.

Runs on: Linux, Windows, Solaris. Requires Tcl/Tk 8.1 or above (included in package).

- Altova XMLSpy Home Edition. US\$ 49. Versão de avaliação (30 dias) disponível.

http://www.altova.com/products_ide.html

Roda em Windows. (Algumas pessoas já disseram que roda em linux usando Wine, e no Mac OS X usando o Virtual PC 6).

Funcionalidades: Visões: Texto, Grid (caixas em caixas), Navegador. Todos os elementos recolhíveis exceto na visão texto. Visão Browser é apenas para leitura. Seletor de elementos. Editor de atributos. Funções de Busca e Edição. Seletor de caracteres especiais. Corretor ortográfico.

- Altova Authentic. Grátis.

http://www.altova.com/products_doc.html

Tem as mesmas funcionalidades (exceto as visões de Texto e Grid) e roda nos mesmos ambientes do XML Spy. Possui a visão "Authentic" em vez disso e um corretor ortográfico. Este editor na verdade é uma interface, a qual usa uma versão mais restrita do XMLSpy como base.

Esta lista não tem a intensão de ser exaustiva, mas se você conhece alguma *bom* XML editor (bom sob a perspectiva de um escritor de documentos do Firebird) que você acha que deva estar aqui, deixe-nos saber via a lista de discussão.

Escrevendo seu documento DocBook

Alô - ainda conosco? Eu sei que gastei um bom tempo escrevendo sobre XML e DocBook, mas eu realmente sinto que era necessário - devido ao fato que esses conceitos são novos para muita gente. Apenas jogando alguns links e dizendo-os para se virarem por si mesmo certamente faria com que perdêssemos alguns escritores valiosos.

De qualquer modo, aqui estamos: finalmente prontos para escrever nossos documentos. Esta seção descreve como configurar seu documento DocBook, e aplicando as tags e atributos corretos nos lugares de direito.

Por favor leia a [subseção sobre elementos hierárquicos](#) mesmo que você seja um escritor DocBook experiente, pois contém algumas diretrizes específicas do nosso projeto. Depois disso, pode pular o resto da subseções sobre DocBook.

Criando o documento

Todo a nossa documentação é parte um grande elemento DocBook: o <set>. Este é o mais alto elemento na hierarquia DocBook. Nosso conjunto contém um número de livros (tag <book>), os quais contém capítulos (tag <chapter>), e por aí afora. A vantagem de pôr todos os livros em um único conjunto é que podem referenciar os outros, por exemplo, pode-se colocar links em sua documentação apontando para um determinado ponto em outro livro.

Felizmente, colocando todos os livros num único conjunto não implica em que eles tenham que estar em um único grande documento. DocBook permite que se possa configurar um documento principal como mostrado abaixo. (Não se preocupe se você não entende a seção iniciada com <!DOCTYPE - nós já temos um documento principal de modo que você não tem que escrever tal tipo de coisa horrível por você mesmo).

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

<!DOCTYPE set PUBLIC "-//OASIS//DTD DocBook XML V4.1.2//EN"
"file:docbookx.dtd" [
  <!ENTITY firebird-intro SYSTEM "file:fbintro.xml">
  <!ENTITY firebird-sql-ref SYSTEM "file:fbsqlref.xml">
  <!ENTITY ...>
  <!ENTITY ...>
]>

<set id="firebird-books">
  &firebird-intro;
  &firebird-sql-ref;
  ...
  ...
</set>
```

Com o documento principal configurado, os vários outros livros podem estar em arquivos separados: fbintro.xml, fbsqlref.xml, etc., o quais podemos editar independentemente. Tal arquivo - o seu, por exemplo - é mais ou menos estruturado deste jeito:

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

<book id="fbintro">
  <chapter id="fbintro-preface">
    ...
    ...
  </chapter>
  <chapter id="fbintro-installing-firebird">
    ...
    ...
  </chapter>
  ...
  ...
</book>
```

É claro que você configurando um novo documento ele tem que ser colocado no conjunto principal, mas isso é algo que discutiremos quando você estiver preparado começar a escrever. (Não colocamos uma regra geral aqui por causa que isso depende do que você irá escrever - um livro, artigo, um ou alguns capítulos, ... - e como seu trabalho se encaixa no resto).

Todo documento DocBook tem que iniciar com esta linha:

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

Se você escreve sua documentação "na mão"; por exemplo, em um editor de texto, você deve digitar a linha acima. Se você usa um editor específico de XML, esta será inserida automaticamente no novo documento.

Você deve sempre colocar sua documentação na pasta `src/docs/firebirddocs` da árvore dos manuais - pelo menos se este estiver em inglês. Em documentos em outras línguas deve-se colocar o documento em pastas tipo `src/docs/firebirddocs-fr`, `src/docs/firebirddocs-es`, etc.

Digitando texto

Se você digitar seu texto DocBook XML em um editor tipo o Bloco de Notas, emacs ou ConText, você pode usar quebras de linha, endentação e múltiplos espaços do jeito que desejar. Cada ocorrência de *whitespace* (uma sequência de um ou mais caracteres `space`, `tab`, `linefeed` or `formfeed`) será convertida em um único espaço em branco na saída. Então isto:

```
<section><title>Arquiteturas do Firebird</title><para>Agora vamos dar
uma olhada nas diferentes arquiteturas do Firebird.</para><itemizedlist>
<listitem><para>Primeiro, existe a chamada arquitetura<firsttterm>Classic
Server</firsttterm>.</para></listitem><listitem><para>E também existe a
arquitetura <firsttterm>Superserver</firsttterm>.</para></listitem>
<listitem><para>E finalmente, com o lançamento do Firebird 1.5 nós
também temos o <firsttterm>embedded server</firsttterm>.</para></listitem>
</itemizedlist></section>
```

resulta na mesma saída disto:

```
<section>
  <title>Arquiteturas do Firebird</title>
  <para>Vamos dar uma olhada nas diferentes arquiteturas
do Firebird.</para>
  <itemizedlist>
    <listitem>
      <para>Primeiro, temos a chamada arquitetura
      <firsttterm>Classic Server</firsttterm>.</para>
    </listitem>
    <listitem>
      <para>Entar temos a arquitetura
      <firsttterm>Superserver</firsttterm>.</para>
    </listitem>
    <listitem>
      <para>E finalmente, com o lançamento do Firebird 1.5 nós também
      temos o <firsttterm>embedded server</firsttterm>.</para>
    </listitem>
  </itemizedlist>
</section>
```

Não é necessário dizer, a segunda forma é muito mais fácil de ler e entender para um humanos. Então, se você digita seu XML na mão, formate o texto em uma forma que a estrutura seja tão clara quanto possível. Como os profetas disseram: "Indente! Indente! Indente!"

Se, na outra vertente, você usa um editor XML dedicado, pressionando **Enter** pode automaticamente fechar o `<para>` corrente e abrir um novo. Tenha certeza de como o seu editor se comporta, e use a tecla `Enter` de acordo. Verifique também o que acontece com múltiplos caracteres espaço em branco, porque certos editores XML podem user certos truques para preservá-los.

Elementos que usamos freqüentemente

Esta seção discute os elementos DocBook que nós mais usamos em nossos documentos do Firebird. Isso inclui montes de exemplos em formato DocBook XML. Se você usa uma ferramenta de elaboração XML, o que você vê na sua tela pode ser completamente diferente da maneira que aparece em nossos exemplos, mas se você abrir o arquivo XML em um editor de texto - ou escolher a visão de texto da sua ferramenta XML - você verá o XML por baixo. Você devia dar uma olhada nos fontes em XML que já estão no módulo de manuais, para ver como os outros autores criaram seus documentos e aplicam as tags.

Dica

Esta é uma longa seção, mas não deixe isso desencorajá-lo. Meu conselho é que você cuidadosamente leia a subseção em elementos hierárquicos, e folheie as outras. Não se preocupe se há coisas que você ainda não entende de uma vez, não tente decorar o material. Apenas tenha-o à quando você escrever seu documento, e revise as subseções de vez em quando (tipo "quando você precisar dele").

Elementos Hierárquicos

A hierarquia mais comum, partindo do topo: `<set>` - `<book>` - `<chapter>` - `<section>` - `<para>`. Elementos de seção são um pouco diferentes do resto, por haver dois sabores deles:

- O elemento `<section>` como colocado acima. Ele pode ser usado recursivamente; por exemplo, você pode ter uma `<section>` dentro de uma outra `<section>` e daí por diante. Este sabor tem a vantagem de que você pode mover subárvores para cima ou abaixo sem ter que mudar as tags
- E também existe a faixa de `<sect1>`, `<sect2>` ... `<sect5>`. Esses elementos devem ser devidamente aninhados, com o `<sect1>` no topo, `<sect2>` dentro `<sect1>` e assim por diante. Não se pode colocar `<sect3>` diretamente dentro de `<sect1>`. É menos flexível que `<section>`, mas raramente isso causa problemas na prática. Afinal, a mesma "rigidez" também se aplica a `<set>`, `<book>` e `<chapter>`, e nós conseguimos viver com isso sem problemas.

Nota

Em versões anteriores deste documento, a série `<sectN>`, nos recomendávamos por razões de apresentação. Devido à melhores nas folhas de estilo, tais razões não mais se aplicam. Siga o estilo que melhor lhe convier.

Livros, capítulos e seções de topo devem sempre possui um *id* e um *title* (um identificador e um título, respectivamente). O *id* permite que o elemento seja referenciado de uma outra parte do documento, e até mesmo de outros documentos no mesmo conjunto. *Id* é um atributo, o que significa que aparece dentro de uma tag; não é visível nos documentos renderizados (exceto no texto fonte do HTML; mais ainda, os identificadores da seção de alto nível são utilizados para nomear os arquivos HTML). *Title* é um elemento, geralmente o primeiro elemento-filho a ser achado no pai. O conteúdo of elemento título aparecerá nos documentos gerados.

Se você for rescrever um livro ou artigo, é também uma boa idéia incluir um elemento `<bookinfo>` ou `<articleinfo>`, onde você pode a informação de autor (e mais coisas).

Eis um exemplo de como estruturar seu documento:

```
<?xml version="1.0" encoding="UTF-8"?>
<book id="usersguide">
  <bookinfo>
    <title>Guia do Usuário do Firebird </title>
    <author>
      <firstname>William</firstname>
      <surname>Shakespeare</surname>
    </author>
  </bookinfo>
  <chapter id="usersguide-intro">
    <title>Introdução</title>
    <para>Alô! Este é texto introdutório do Guia do Usuário
      do Firebird.</para>
  </chapter>
  <chapter id="usersguide-download-install">
    <title>Baixnado e instalando o Firebird</title>
    <para>Neste capítulo iremos demonstrar como baixar e instalar
      o Firebird.</para>
    <section id="usersguide-download">
      <title>Baixando o Firebird</title>
      <para>Para baixar o Firebird, você deve ir primeiro na seguinte
        URL:etc etc etc</para>
      <!-- ...mais parágrafos, possivelmente mais subseções... -->
    </section>
    <section id="usersguide-install">
      <title>Instalando o Firebird</title>
      <para>Instalação do Firebird no seu sistema será deste jeito: etc
        etc etc.</para>
      <!-- ...mais parágrafos, possivelmente mais subseções... -->
    </section>
  </chapter>
  <!-- ... mais capítulos... -->
</book>
```

Algumas regras e comentários:

- Primeiro, note de novo que valores de atributos sempre devem estar entre aspas. (Mas se você os preencher em editor de atributos, não as coloque: o editor cuidará desse detalhe).
- O `title` (título) de um livro ou artigo deve aparecer no próprio elemento `book` (livro) ou `article` (artigo). É até legal incluir em ambos, mas nesse caso os dois `title`'s (títulos) devem possuir o mesmo conteúdo. Em outras palavras: fique com one ou outro, não ambos, se quiser evitar encrenca.
- Todos atributos `id` devem ser únicos dentro do mesmo conjunto de livros. Note que versões em diferentes linguagens estão contidas cada uma em seu próprio conjunto, é perfeitamente possível manter as `id` iguais aos originais.
- Dentro de um livro ou artigo, todos os `id`'s (identificadores) devem começar com a mesma palavra em minúsculas; por exemplo, `usersguide`, seguida de um traço, seguido por uma ou mais palavras em minúsculas. Exemplos são `usersguide-intro` e `usersguide-download-install`. Esta não é um requisito do DocBook, mas nossa própria convenção.
- Como você pode ver no exemplo, `chapter`'s e `section`'s (capítulos e seções, respectivamente) podem iniciar diretamente com um ou mais elementos `para`. Mas uma vez que você inclua seções num capítulo, ou subseções numa seção, você não pode adicionar mais elementos `para` depois deles, somente dentro deles. Bom editores com suporte a DocBook simplesmente não permitir que

you cometa tal erro, mas caso você digite seu XML do DocBook na mão isso é algo que você precisa ficar atento.

- Se você utiliza um editor de XML, é bem provável que você raramente tenha que criar os elementos explicitamente. Por exemplo, se você inserir um `chapter` ou uma `section` (um capítulo ou uma seção, respectivamente) no XMLMind XML Editor, o primeiro para parágrafo vazio é automaticamente criado. E quando você digita texto em um parágrafo e pressiona **ENTER**, esse parágrafo é automaticamente fechado e um novo é criado.

[Pule o resto das subseções sobre elementos DocBook](#) se você já conhece tudo sobre elementos do DocBook.

Listas

DocBook oferece vários elementos de listas, dos quais são usados frequentemente:

itemizedlist

Uma `itemizedlist` é usada para enumerar itens cuja ordem não é muito importante:

```
<itemizedlist spacing="compact">
  <listitem><para>Laranjas são suculentas</para></listitem>
  <listitem><para>Maçãs supostamente são saudáveis</para></listitem>
  <listitem><para>A maioria das pessoas acham o limão muito
    ácido</para></listitem>
</itemizedlist>
```

Os itens na lista são geralmente marcados com um marcador nos documentos de saída:

- Laranjas são suculentas
- Maçãs supostamente são saudáveis
- A maioria das pessoas acham o limão muito ácido

orderedlist

Use uma lista ordenada (tag `<orderedlist>`) quando se deseja forçar a ordem entradas:

```
<orderedlist spacing="compact" numeration="loweralpha">
  <listitem><para>Sumérios 3300 AC - 1900 AC</para></listitem>
  <listitem><para>Império Assírio 1350 AC - 612 AC</para></listitem>
  <listitem><para>Império Pérsio 6º século AC - 330 AC</para>
</listitem>
</orderedlist>
```

Por padrão, numerais arábicos (1, 2, 3, ...) serão colocados antes dos itens, mas você pode mudar isso com o atributo `numeration`. Saída:

- a. Sumérios 3300 AC - 1900 AC
- b. Império Assírio 1350 AC - 612 AC
- c. Império Pérsio 6º século AC - 330 AC

procedure

Um procedimento (tag `<procedure>`) é frequentemente renderizado como uma lista ordenada (tag `<orderedlist>`), mas com uma semântica diferente: um procedimento denota uma seqüência de *passos* a serem executados numa dada ordem:

```
<procedure>
  <step><para>Arrombe a fechadura</para></step>
  <step><para>Roube a casa</para></step>
  <step><para>Seja preso</para></step>
</procedure>
```

Aqui está a saída:

1. Pick the lock
2. Rob the house
3. Get arrested

Dentro de um passo você pode incluir um elemento subpasso (tag `<substep>`), o qual pode conter mais passos.

variablelist

Uma lista variável (tag `<variablelist>`) é feita de *entradas de lista variável* (tag `<varlistentry>`), cada uma das quais possuem um termo (tag `<term>`) seguido de um item de lista (tag `<listitem>`):

```
<variablelist>
  <varlistentry>
    <term>Tag</term>
    <listitem>
      <para>Uma peça de texto entre um '<' e um '>'</para>
    </listitem>
  </varlistentry>
  <varlistentry>
    <term>Element</term>
    <listitem>
      <para>Uma tag inicial, uma tag final correspondendo e tudo
        entre elas</para>
    </listitem>
  </varlistentry>
  <varlistentry>
    <term>Conteúdo de um elemento</term>
    <listitem>
      <para>Tudo dentro das tags correspondentes</para>
    </listitem>
  </varlistentry>
</variablelist>
```

A lista que você está vendo agora, enumerando os diferentes tipos de listas, é uma lista variável com os nomes dos elementos (`itemizedlist`, `orderedlist`, etc.) como termos. A próxima seção, sobre links, também consiste de uma sentença introdutória seguida de uma lista variável.

Links

Você pode criar hiperlinks para alvos no seu próprio documento, em outro documento no mesmo conjunto ou na internet.

link

`link` é o elemento genérico que aponta para qualquer local no conjunto de documentos. O atributo `linkend` sempre deve estar presente; seu valor deve ser o id de um *alvo de link* (o elemento para o qual se irá saltar).

Clique `<link linkend="docwritehowto-introduction-pt_br">aqui</link>` para saltar para a introdução.

No documento renderizado, a palavra "aqui" ficará *quente*, o que significa um link clicável apontando para a introdução.

Clique [aqui](#) para pular a introdução.

Cuidado

Embora você possa usar a tag `link` para apontar para qualquer elemento no conjunto inteiro, você só deve fazê-lo se o alvo do link estiver no mesmo documento PDF que o próprio link. A versão em HTML é pode ser totalmente hiperlinkada, mas links na versão em PDF não podem ser feitos para fora do mesmo documento. Nossos PDFs tipicamente contém um livro (tag `book`) ou artigo (tag `article`); se o alvo está fora do documento corrente, use a tag `ulink` em de `link`. (veja abaixo)

Nota

Como você já deve ter percebido, a região clicável é por vezes recuada com respeito ao texto do link. Isso é um problema conhecido no Apache FOP, não havendo muito que possamos fazer a respeito.

ulink

Use um `ulink` para linkar para um recurso na Internet. O atributo `url` é obrigatório:

Clique `<ulink url="http://docbook.org/tdg/en/">neste link</ulink>` para ler o O Guia Definitivo para DocBook.

As palavras “neste link” aparecerão como um hiperlink para `http://docbook.org/tdg/en/`, assim:

Clique [neste link](http://docbook.org/tdg/en/) para ler O Guia Definitivo para DocBook.

email

Você pode fazer um link para email com um `ulink`, porém é mais fácil usando o elemento `email`. Este irá mostrar o endereço de email como texto clicável na saída. Esta linha de XML:

```
Envie mensagem para
<email>firebird-docs-request@lists.sourceforge.net</email> para
assinar.
```

resulta na seguinte saída: "Envie mensagem para <firebird-docs-request@lists.sourceforge.net> para se assinar."

Se você deseja que o texto clicável seja diferente do próprio endereço de email, use um `ulink` com uma URL `mailto:`.

Atenção

Se você incluir links para endereços de email - seja com `email` ou com `ulink` - ou mesmo se você apenas *mencioná-los* no seu texto, e seu documento é subsequentemente publicado na Internet, esses endereços de email serão expostos a robôs de exploração usados por spammers. De modo que será previsível um aumento de spam nesses endereços. Tenha sempre certeza que os donos dos endereços concordam em tê-los publicados na internet!

anchor

Um `anchor` (âncora) é um elemento vazio marcando um local exato dentro de um documento. Ele não é mostrado no texto que seus leitores lêem, mas podem ser usados como alvos para links. Isto é útil se você deseja linkar para um local qualquer no meio de um grande parágrafo.

```
<para id="lost-at-sea">
  Blah blah blah...
  e algo mais...
  e então algo...
```

```
Agora aqui está um lugar interessante no parágrafo que eu quero ser
capaz de linkar:
<anchor id="captain-haddock"/>Aí está!
Drones de parágrafo ligados...
e ligados...
e ligados...
</para>
```

Tendo colocado a âncora, você pode criar um link para ela:

```
<link linkend="captain-haddock">Vá para o lugar interessante</link>
nesse longo, longo parágrafo.
```

Se você linkar para um elemento curto, ou para o início de um elemento, é mais fácil dar ao dito elemento uma id e usá-la como alvo do link.

Listagens de programas, telas, layout literal e exemplos

programlisting

Se você incluir fragmentos de código no seu documento, coloque-os num elemento `programlisting` (listagem de programa). Tudo que você digitar será renderizado como estiver escrito, incluindo quebras de linhas, espaços em branco, etc. E além disso, uma fonte de tamanho fixo será utilizada nos documentos renderizados. O termo "listagem de programa" pode ser interpretado de forma abrangente aqui: você pode usá-lo para fragmentos de SQL ou exemplos de DocBook XML. Este documento - especialmente na seção sobre elementos, a qual você está lendo agora - é cheia de `programlisting`'s de modo que você já viu como elas se parecem:

Programlistings são renderizadas deste jeito.

Importante

Em listagens de programas (tag `programlisting`) você deve limitar o tamanho da linha por volta de 70 caracteres, caso contrário o texto sairá das margens quando renderizado em PDF. A mesma regra vale para outros elementos que preservam layout como `screen`, `literallayout`, etc.

screen

Use um elemento `screen` (tela) para mostrar o que o usuário vê ou deverá ver numa tela de computador em modo texto, ou numa janela de terminal. Aqui também, seu layout será preservado e uma fonte de tamanho fixo será usada, mas a semântica é diferente. Ela pode ou não ter aparência diferente de uma listagem de programa na saída. Este é um exemplo curto, mostrando o que acontece se você tentar gerar um alvo não existente na árvore de manuais:

```
<screen>
D:\Firebird\manual_incl_howto\src\build>build ugh
java version "1.4.2_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_01-b06)
Java HotSpot(TM) Client VM (build 1.4.2_01-b06, mixed mode)

Buildfile: build.xml

BUILD FAILED
Target `ugh' does not exist in this project.
</screen>
```

E aqui como isso é renderizado:

```
D:\Firebird\manual_incl_howto\src\build>build ugh
java version "1.4.2_01"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_01-b06)
Java HotSpot(TM) Client VM (build 1.4.2_01-b06, mixed mode)
```

```
Buildfile: build.xml
```

```
BUILD FAILED
Target `ugh' does not exist in this project.
```

literallayout

`literallayout`, assim como `screen` e `programlisting`, mantém seu layout intacto, mas geralmente não muda a fonte - a não ser que você altere o atributo `class` para "monospaced". É também mais genérico que os outros dois anteriores no contexto de que não há nenhum significado atrelado a seu conteúdo - você pode colocar qualquer tipo de texto aqui o qual você deseje preservar o layout.

Exemplo de fonte:

```
<literallayout>
The Sick Rose

Oh Rose, thou art sick!
The invisible worm
That flies in the night,
In the howling storm,

Has found out thy bed
Of crimson joy,
And his dark secret love
Doth thy life destroy.

-- William Blake
</literallayout>
```

Saída:

```
The Sick Rose

Oh Rose, thou art sick!
The invisible worm
That flies in the night,
In the howling storm,

Has found out thy bed
Of crimson joy,
And his dark secret love
Doth thy life destroy.

-- William Blake
```

Nota

Nas versões anteriores deste documento, você era avisado para não usar a variante com fonte de tamanho não fixo de `literallayout`, porque ficava horrível a aparência na saída em PDF. Mas o problema já foi resolvido, como você pode ver nos PDFs.

example

Um `example` é um exemplo formal com um título. Geralmente se dá um `id` ao mesmo para possa ser referenciado de outros lugares no documento. Um índice de exemplos é gerado automaticamente.

mente quando o documento é renderizado. Você frequentemente encontrará `programlisting's` em um exemplo, mas este pode conter `screens`, `paras`, `listas`, etc.

Eis um exemplo de um `example`:

```
<example id="docwritehowto-sql-example">
  <title>Um exemplo SQL</title>
  <para>Com este comando você pode listar todos os registros na tabela
    COUNTRY:</para>
  <programlisting>SELECT * FROM COUNTRY;</programlisting>
</example>
```

Na saída terá o seguinte aspecto:

Exemplo 1. Um exemplo de SQL

Com este comando você poderá listar todos os registros na tabela COUNTRY:

```
SELECT * FROM COUNTRY;
```

Se você deseja um exemplo sem um título obrigatório, use um `informalexample` (exemplo informal). Exemplos informais são também deixados de fora do índice de exemplos.

Tabelas

Se você já fez alguma vez uma tabela HTML para um website, você não terá muita dificuldade em criar tabelas no DocBook. Existem diferenças porém, e as tabelas do DocBook são bem ricas.

Uma `<table>` (tabela) consiste de um `<title>` (título) e um ou mais `<tgroup>`'s - mas geralmente apenas um. O elemento `<tgroup>` é obrigatório e tem um atributo obrigatório: `cols`. Você deve setar esse atributo com o número de colunas no `<tgroup>`. Dentro de um `<tgroup>` você pode colocar os elementos `<thead>`, `<tfoot>` e `<tbody>`. Cada um desses tem um ou mais `<row>`'s, os quais podem ter tantos `<entry>`'s (células) quanto forem necessários para chegar ao número de colunas do atributo `cols`. (Você pode combinar células criando *spans*, mas eu não entrarei nesse mérito).

Tanta coisas para uma estrutura básica. Agora mostraremos um exemplo; primeiro em fonte DocBook XML, e depois a tabela resultante no documento de saída renderizado. Não se preocupe com os `<colspec>` por enquanto; estes são subelementos opcionais usados para ajustes finos:

```
<table id="docwritehowto-table-dboftheyear">
  <title>Pesquisa LinuxQuestions.org: Banco de dados do ano de
    2003</title>

  <tgroup cols="3">
    <colspec align="left" colname="col-dbname" colwidth="2*" />
    <colspec align="right" colname="col-votes" colwidth="1*" />
    <colspec align="right" colname="col-perc" colwidth="1*" />

    <thead>
      <row>
        <entry align="center">Banco de Dados</entry>
        <entry align="center">Votos</entry>
        <entry align="center">Porcentagem</entry>
      </row>
    </thead>

    <tfoot>
```

```

<row>
  <entry>Total</entry>
  <entry>1111</entry>
  <entry>99.99</entry>
</row>
</tfoot>

<tbody>
  <row>
    <entry>MySQL</entry>
    <entry>405</entry>
    <entry>36.45</entry>
  </row>
  <row>
    <entry>Firebird</entry>
    <entry>403</entry>
    <entry>36.27</entry>
  </row>
  ... e mais 5 linhas não mostradas aqui....
</tbody>
</tfoot>
</table>

```

E aqui a tabela resultante:

Tabela 2. Pesquisa LinuxQuestions.org: Banco de Dados of ano de 2003

Banco de Dados	Votos	Porcentagem
MySQL	405	36.45
Firebird	403	36.27
Postgres	269	24.21
Oracle	25	2.25
Berkeley DB	4	0.36
Sybase	3	0.27
DB2	2	0.18
Total	1111	99.99

A propósito, esses são resultados atuais de uma pesquisa real no LinuxQuestions.org. Como você pode ver, faltaram 3 votos para o Firebird vencer a pesquisa. Se você conhece essas pessoas, por favor reporte-as ao seu Inquisitor Chefe. Ele gostaria de ter uma pequena, ahhh... humm... *conversa* com eles. ;-)

Tabelas são automaticamente indexadas. Uma `<informaltable>` (tabela informal) tem a mesma estrutura de uma `<table>` mas não requer um título e não será colocada no índice. (Tal como no caso do `<example>/<informal example>`). Se você deseja aninhar tabelas, use uma `<table>/<informaltable>` dentro de uma `<entry>`, ou uma `<entrytbl>` em vez de uma `<entry>`.

Tabelas possuem muitas outras funcionalidades além das mostradas aqui, mas essas nós deixaremos para você as explorar.

Avisos

DocBook tem muitas tags para marcar um bloco de texto como uma nota, aviso, dica, etc. Na saída tais blocos tipicamente aparecem indentados, e marcados com um ícone ou palavra para denotar seu propósito. Estas tags são, em ordem alfabética:

<caution>, <important>, <note>, <tip> e <warning>

Usarei uma <tip> em um exemplo; as outras funcionam exatamente do mesmo jeito:

```
<tip>
  <para>Se você inserir um elemento caution, important, note, tip ou
    warning no seu texto, não inicie o mesmo com as palavras advertência,
    importante, nota, dica ou aviso, porque essas palavras geralmente
    são geradas automaticamente pelo programa de renderização.</para>
</tip>
```

E este é o resultado:

Dica

Se você inserir um elemento caution, important, note, tip ou warning no seu texto, não inicie o mesmo com as palavras advertência, importante, nota, dica ou aviso, porque essas palavras geralmente são geradas automaticamente pelo programa de renderização.

Você deve ter notado que as palavras important, caution etc têm aparência diferente do resto do texto da dica. Como? Bem, para dizer a verdade, eu as coloquei entre tags especiais (as primeiras com <sgmltag>'s, nas segundas com <literal>'s) para fazê-las aparecerem diferente do resto do texto. Mas aí o fonte em XML ficaria muito poluído, então eu decidi tirá-las do código exemplo que apresentei a você.

Se você destacar um bloco de texto daquilo que o circunda sem marcá-lo como uma dica ou o que seja, use a tag <blockquote>.

Vários elemento em linha

Para concluir a subseção sobre elementos do DocBook, agora eu falarei sobre um certo número de *elementos em linha*. Eles são chamados “em linha” porque não interrompem o fluxo do texto. Se você usa (por exemplo) um elemento emphasis:

```
<emphasis>Em hipótese alguma</emphasis> me chame de gordo de novo!
```

o resultado será:

Em hipótese alguma me chame de gordo de novo!

A expressão "Em hipótese alguma" está enfatizada, mas mantém-se no seu lugar na sentença. Nós já encontramos alguns elementos em linha antes: os vários tipos de links. Outros elementos - como table, warning, blockquote e programlisting - são todos mostrados como um bloco, aparte do texto em volta (mesmo que você os coloque "em linha" no seu fonte em XML). Não é muito surpreendente que sejam chamados de *elementos de bloco*. Elementos de bloco freqüentemente possui elementos em linha; o inverso não é possível.

Ok, vamos começar a experimentar esses elementos em linha. Incluirei exemplos - tanto como fonte XML e saída renderizada - para a maior parte deles:

filename - command - application - envar

Use a tag `filename` para marcar nomes de arquivos no mais abrangente sentido. Atributos podem opcionalmente indicar que o mesmo é um arquivo de cabeçalho, um diretório, etc.

```
Coloque seu documento no subdiretório<filename
class="directory">src/docs/firebirddocs</filename>.
```

Na saída lê-se:

Coloque seu documento no subdiretório `src/docs/firebirddocs`.

`command` e `application` são ambos usados para programas executáveis. `command` é geralmente escolhido para programas menores e comandos internos; seu conteúdo deve ser o comando exato de disparado em uma linha de comando; `application` é usado para programas maiores e não precisa ser o nome do arquivo executável. Ambos podem se referir ao mesmo programa.

Digite `<command>netscape&</command>` numa janela de terminal para iniciar o `<application>Netscape Navigator</application>`.

Isto é renderizado como:

Digite **netscape&** numa janela de terminal para iniciar o Netscape Navigator.

`envar` denota uma variável de ambiente.

subscript - superscript

(subscrito - sobrescrito)

Fazem o que é esperado:

Após inventar a fórmula $e = mc^2$, eu realmente tive vontade de um copo de H_2O líquido!

Saída: Após inventar a fórmula $e = mc^2$, eu realmente tive vontade de um copo de H_2O líquido!

varname - constant - database

O uso de `varname` e `constant` (nome de variável e constante - respectivamente) deve ser óbvio. A tag `<database>` não é só para bancos de dados, mas também para objetos do banco de dados:

```
A tabela <database class="table">COUNTRY</database> tem dois campos:
<database class="field">COUNTRY</database> e
<database class="field">CURRENCY</database>.
```

Saída: A tabela COUNTRY tem dois campos: COUNTRY e CURRENCY.

function - parameter - returnvalue

(função, parâmetro e valor de retorno - respectivamente)

Esses três falam por si mesmos, eu confio.

A função `<function>log</function>` tem os parâmetros `<parameter>a</parameter>` e `<parameter>b</parameter>`.

Saída: A função `log` tem os parâmetros `a` e `b`.

prompt - *userinput* - *computeroutput*

Nota

do Tradutor: *prompt* será usado no original, seu significado no caso poderia ser indicação de prontidão – tradução que realmente ajuda. *userinput* é entrada do usuário e *computeroutput* é saída do computador.

prompt é usado para uma string que indica que o usuário deve entrar algum texto; *userinput* refere-se ao texto entrado pelo usuário (não necessariamente num *prompt*); *computeroutput* é texto exibido pelo computador:

Digite `<userinput>convidado</userinput>` no *prompt* `<prompt>login:</prompt>` e o servidor irá saudá-lo com um `<computeroutput>Bem-vindo, usuário convidado</computeroutput>`.

Saída: Digite **convidade** no *prompt* `login:` e o servidor irá saudá-lo com um Bem-vindo, usuário convidade.

keycap

O texto de uma tecla do teclado, ou o seu nome comum:

Pressione a tecla `<keycap>Del</keycap>` para apagar a mensagem, ou `<keycap>ESPAÇO</keycap>` para movê-la.

Saída: Pressione a tecla **Del** para apagar a mensagem, ou **SPACE** para movê-la.

sgmltag

Este elemento foi usado extensivamente durante este documento: ele marca tags, elementos, atributos, entidades, etc de SGML e XML :

Se é um diretório, ajuste o atributo `<sgmltag class="attribute">class</sgmltag>` do elemento `<sgmltag class="element">filename</sgmltag>` para to `<sgmltag class="attvalue">directory</sgmltag>`.

Saída: Se é um diretório, ajuste o atributo `class` do elemento `filename` para `directory`.

Outros possíveis valores de para `sgmltag.class` são: `starttag`, `endtag`, `emptytag` e `gentity` (para uma entidade).

emphasis - *citetitle* - *firstterm*

(ênfase - título de citação - primeiro termo)

Use *emphasis* para enfatizar palavras em geral, *citetitle* para títulos de livro etc, e *firstterm* se você introduzir um novo conceito ou palavras para os seus leitores:

Nós usamos `<firstterm>DocBook XML</firstterm>` para nossa documentação do Firebird. Uma introdução curta segue; `<emphasis>por favor</emphasis>` leia-a cuidadosamente! Se você deseja saber mais sobre o assunto, compre `<citetitle>DocBook - The Definitive Guide</citetitle>`.

Saída: Nós usamos *DocBook XML* para nossa documentação do Firebird. Uma introdução curta segue; *por favor* leia-a cuidadosamente! Se deseja saber mais sobre o assunto, compre *DocBook - The Definitive Guide*.

`quote - literal`

(citação - literal)

Use `quote` para uma citação em linha (ao oposto de um `blockquote`). Caracteres aspas serão inseridas automaticamente. Usando `quote` ao invés de digitando as aspas por você mesmo (o que é perfeitamente legal) tem a vantagem de que nós podemos alterar o tipo de aspas através das folhas de estilo se nós quisermos.

Um `literal` é um fragmento de texto ou palavra para ser lido literalmente. É um elemento bem genérico, freqüentemente usado para fazer certas palavras se destacarem tipograficamente:

A qualquer custo evite usar a palavra `<literal>monstruoso</literal>` em sua documentação.

Saída: A qualquer custo evite usar a palavra `monstruoso` em sua documentação.

Você deve usar esses elementos em linha onde você puder? Bem, se o fizer, certamente fará o seu documento ficar mais rico; fará com que seja fácil procurar por nomes de arquivos por exemplo, ou gerar um índice de todas aplicações mencionadas em seu documento. Por outro lado, existem tantos des-se elementos semânticos (de fato nós discutimos apenas uns *poucos* deles) que se você aplicar todos em todo lugar que você possa, provavelmente ficará maluco antes de terminar o documento. Não é isso que desejamos: se você realmente tiver que ficar doido, por favor faça-o *depois* de terminar seu documento. ;-)

Então, como um conselho geral: vá devagar com esses elementos; use-os toda vez que você achar que fazem sentido, mas não exagere.

Para empacotar os elementos

Você deve ter notado que os documentos renderizados (você está lendo um agora, a não ser que você tenha aberto a versão XML) muitos diferentes elementos têm a mesma aparência: um `filename`, um `literal` e uma `application` têm exatamente a mesma tipografia; o mesmo valendo para `emphasis`, `firstterm` e `citetitle`.

Então, qual é o sentido de todas essas tags diferentes? Por que não usar umas poucas, tipo `emphasis` e `literal`, se todos aparacer da mesma maneira? Bem, há duas boas razões para não fazer isso:

- Primeiro, se nós deixarmos de usar a maior parte de nossos elementos em linha em favor de, por exemplo, `emphasis` e `literal`, a semântica seria perdida. E lembramos que o DocBook XML é todo sobre estrutura e semântica. `firstterm` e `citetitle` podem *parecer* a mesma coisa uma vez renderizado, mas eles *não são* a mesma coisa. O fonte em XML sabe disso, apesar não mostrar sempre. Esta informação é útil, e não queremos perdê-la.
- Além disso, nós podemos adaptar nossas folhas de estilos para cada de elemento individualmente. Assim que decidirmos que um `firstterm` deva ter aparência diferente de um `citetitle`, nós arranjamos isso - mas *somente* se eles forem marcados com tags diferentes, não se eles forem ambos `emphasis` no fonte em XML.

Isso conclui as seções no DocBook. Com o conhecimento apresentado acima, você deve saber o suficiente para elaborar documentos em DocBook XML para o projeto Firebird. Claro que se você usar um editor de XML dedicado - o que é altamente recomendado - você deve consultar a documentação do mesmo para aprender como usá-lo; o que é uma que este documento não cobre.

Aspectos do processo de escrita não relacionados com o DocBook

Após a enchente de informação sobre DocBook nas seções anteriores, nós discutiremos rapidamente alguns outros aspectos importantes da escrita de documentos: linguagem, estilo, e direitos autorais.

Linguagem e estilo

Linguagem

A comunidade Firebird é bem diversa, e feita de pessoas com muitas línguas-mater diferentes. Se você escrever documentação em uma linguagem outra que não a sua própria, provavelmente cometerá alguns enganos. Isso não é catastrófico, mas você deve tentar reduzir o número de erros. Algumas estratégias para ajudá-lo nisso são:

- Use um dicionário! Simples, efetivo e não high-tech
- Quando estiver indeciso entre duas escritas de uma palavra, ou entre várias possíveis versões de uma expressão, *google* (procurar no site de busca [Google](#), é um verbo criado por causa da popularidade desse site) pelas alternativas e veja suas frequências. E siga alguns dos links resultantes da pesquisa para ver como os falantes nativos usam a palavra ou expressão.
- Tenha um falante nativo para revisar seu texto e corrigí-lo onde necessário.

Estilo

Não espere um Guia de Estilo aqui - eu não saberia como escrever um mesmo... Apenas algumas diretrizes e dicas:

- Tente escrever em linguagem simples e cotidiana sempre que possível. Evite palavras difíceis se existem alternativas similares mais simples.
- Evite frases longas (acima de 25 palavras) se puder; especialmente duas ou mais frases longas uma após a outra.
- Seja cuidadoso em construções como negativos duplos ou triplos ("Eu não posso negar que não estou aborrecido") e voz passiva ("Cuidados devem ser tomados..."). Você não precisa evitá-las a todo custo, mas lembre-se que elas podem fazer a sentença mais difícil de entender. Nesse caso, use o positivo ("Estou satisfeito") e a voz ativa ("Tenha cuidado...").
- Use listas para enumerar um número de itens paralelos, por exemplo:
 - Uma coleção de sugestões e dicas.
 - Uma seqüência de exemplos (como este aqui).
 - Passos para serem seguidos num procedimento.
 - Soluções alternativas para um problema.

Mas se há apenas um pequeno número de itens curtos, use uma sentença simples: "Minha mãe ama

três homens: John, Dick e Dave."

- Não abuse dos pontos de exclamação. Nunca use múltiplas exclamações ou interrogações. Isto é irritante!!!!!! Ou você discorda?????

Bloqueio do escritor de documentos

Algumas vezes você sabe o que deseja escrever, e tem as palavras prontas, mas não consegue iniciar a frase - você não consegue fazer isso *fluir*. Isso é muito frustrante e às vezes bloqueia o avanço do seu texto por vários minutos. E mais frustrante de tudo isso é fato de que você *sabe* o que deseja passar para seus leitores, mas não parece capaz de produzir uma sentença decente. Após muitas experiências dolorosas deste tipo, eu desenvolvi a seguinte estratégia (não que eu pense que sou o primeiro):

1. Escreva o que você tem que dizer em sentenças soltas e grupinhos de palavras. Não se esquite com estilo, nem que isso pareça feio. Apenas escreva o que você deseja passar ao leitor, tenha certeza que está tudo lá e na ordem certa. Se, enquanto você estiver fazendo isto, você notar que não tem certeza de alguma coisa; inclua um comentário exatamente nesse ponto. Faça os comentários se destacarem do texto em volta, por exemplo <<deste jeito>> ou !DESTE OUTRO JEITO!

Isso pode resultar num texto assim:

CVS significa Sistema de Versões Concorrentes (<<verifique!>>). Propósito: gerenciamento de versões de software. Você pode usá-lo sozinho ou em um grupo. Você precisa de um cliente CVS para usá-lo. Um cliente CVS é um programa com o qual você pode acessar o repositório do CVS (<<explicar este termo?>>). Para verificar se um cliente CVS está instalado no seu sistema, digite "cvs" na linha de comando. Se não está lá, vá para a URL para baixá-lo(...)

2. Se você tiver incluído algum comentário, resolva-os primeiro. *Verifique* se CVS realmente significa Sistema de Versões Concorrentes (sim, significa). *Decida* se você deve explicar o termo "repositório do CVS" agora (sim, deve).
3. Agora, trabalhe sobre o parágrafo novamente e tente fazer o texto fluir mais naturalmente sempre que puder. Muito provavelmente será bem mais fácil do que você imaginava.
4. Se ainda está meio estranho, não se aborreça - melhor esquisito e claro que fluído macio e vago. Talvez você deva visitar esta passagem mais tarde e ver se você pode fazê-la ficar melhor.

Esta abordagem funciona para mim. Se você está travado dessa maneira, tente-a; espero que funcione para você.

Direitos Autorais

Respeitando os Direitos Autorais de Outros

Quando nós escrevemos manuais, podemos consultar todo tipo de outras documentações - e nós devemos, porque nós desejamos o melhor resultado possível. Qualquer informação que é publicamente disponível em manuais de terceiros, guias de usuário, tutoriais, etc. pode ser livremente usado nos nossos documentos, mas é importante não confundir *informação* com *texto literal*. Nós não podemos copiar e colar texto de outros trabalhos na nossa documentação, a não ser que o autor explicitamente permita-nos isso. Verifique o aviso de copyright da obra em questão; se não há nenhum, considere

que a obra está automaticamente sujeita à convenção de Berne e você deve entender que é *ilegal* copiá-la - mesmo parcialmente. É o mesmo caso da obra disponível grátis. Não ter que pagar por um documento não significa que você pode copiar livremente partes do mesmo em uma obra sua.

Mais especificamente, a documentação beta do Interbase 6 da Borland - mesmo grátis - não fazem parte do pacote do Interbase que foi aberto em julho de 2000. Nós perguntamos à Borland diversas vezes se nós poderíamos usar esses documentos como se estivessem sob a "Licença Pública do Interbase", mas eles sequer se deram ao trabalho de responder. Então sinta-se livre para usar essa documentação como fonte de informação, mas não copie texto diretamente dela.

Seu direito autoral e a LDP

Se você contribui para o projeto de documentação do Firebird, seu trabalho será incluído no repositório de código aberto no SourceForge. Em Janeiro de 2005, o time de documentação do Firebird decidiu que as documentação geradas por ele estariam sob a *Licença de Documentação Pública*. Licenciado seu trabalho sob a LDP significa que você manterá o direito autoral, mas garante aos outros certos direitos:

- *Uso livre*: todos podem usar e distribuir seu trabalho, grátis ou por dinheiro, desde que o aviso da licença esteja intacto.
- *Direito de modificação*: todos podem modificar e redistribuir seu trabalho, desde que as versões modificadas sejam licenciadas também pela LDP, o aviso da licença esteja intacto, e as modificações sejam documentadas.
- *Trabalhos maiores*: todos podem incorporar sua documentação (modificada ou não) em um trabalho maior. O trabalho maior como um todo não necessita estar licenciado pela LDP, mas os requisitos da licença devem ser cumpridos para as partes que estão sob a LDP.

O que é tão bom a respeito da LDP é que esta provê os mesmos direitos e restrições em termos de uso de nossos documentos quanto a IPL e IDPL (licenças do código do Firebird) para o código do Firebird. Para o texto completo da licença, veja os links no aviso de licença abaixo; o fonte DocBook está em `src/docs/firebirddocs/licenses.xml`.

Usando os documentos do PostgreSQL

PostgreSQL é outro grande banco de dados de código aberto, com (surpresa, surpresa) muitas similaridades com o Firebird, mas também muitas diferenças. Dependendo do tipo de documentação você irá escrever, pode ser benéfico baseá-la na documentação já existente do PostgreSQL. Mas lembre-se: se você usar o material do PostgreSQL, você *TEM* que incluir o aviso de licença deles no seu documento.

A página de documentação do PostgreSQL é aqui:

<http://www.postgresql.org/docs>

A licença mais recente do PostgreSQL está atualmente aqui:

<http://www.postgresql.org/about/licence>

Uma coisa bacana sobre a documentação do PostgreSQL é que ela foi elaborada em DocBook, assim como a nossa. Porém, eles usam DocBook SGML em vez de XML, o que significa que algumas ajustes serão necessários. Os fontes em DocBook SGML podem ser encontrados aqui:

<http://developer.postgresql.org/cvsweb.cgi/pgsql-server/doc/src/sgml/>

Adicionando seu documento ao módulo de manuais

Quando seu documento é finalizado, e você o verificou que o mesmo é gerado corretamente, você deseja que o mesmo seja adicionado ao módulo de manuais. Se esta é a primeira contribuição ao projeto de documentação, você deve ter concordado com os coordenadores que você os enviará para eles para revisão, ou colocar no ar a versão html num website de modo que a mesma possa ser discutida na lista. Depois disso - e após algumas correções feitas - seu documento será adicionado ao módulo. Se você já possui direito de envio você pode fazer isso você mesmo; se não possui, um dos coordenadores fará isso por você.

Pedindo direitos de envio

Para receber direitos de envio primeiro você precisa de uma conta de usuário no SourceForge. Se não uma, registre-se em <http://sourceforge.net/account/register.php> . Então poste uma mensagem na lista firebird-docs com o seu nome no SF e pedindo para ser adicionado ao projeto Firebird. O líder sub-projeto de manuais e de vários outros projetos lêem a lista; eles irão considerar sua requisição. Como regra geral você deve perguntar sobre direitos de envio *após* sua primeira contribuição, porque as pessoas que decidem sua precisam de algo para se basearem.

As seguintes frases atualmente significam a mesma coisa, a propósito:

- Se tornar um membro do projeto
- Ter direitos de envio
- Ter acesso de leitura e escrita ao repositório.

O que se pode ou não fazer assim que você tem direito de envio

Assim que você for aceito como membro do projeto, você terá acesso a todo o repositório Firebird, não apenas ao módulo de manuais. Não existe nenhuma barreira técnica que o impeça de enviar mudanças aos outros módulos - o módulo núcleo `firebird2` por exemplo, ou mesmo `CVSROOT` onde importantes informações do projeto são guardadas.

Você já deve ter adivinhado que isso *NÃO* é a idéia. Siga as seguintes regras:

- Nunca envie nada aos outros módulos a não ser que as pessoas responsáveis peçam isso explicitamente.
- Apenas envie trabalho para o módulo de manuais se isso é uma tarefa designada para você. E mesmo assim, é de bom tom que você anuncie suas mudanças e adições para lista de discussão, de modo que outros documentadores tenham uma chance de comentá-las. Afinal, isto é um esforço coletivo.
- Se você pensa que um novo documento ou pasta deve ser adicionado, não crie e envie direto, mas proponha isso na lista.

Na prática, as coisas são um pouco mais leves que o dito acima, especialmente onde isso se refere às suas tarefas. Nós não queremos que você se sinta preso e tendo a sensação de que tem de pedir permissão para cada coisinha que tiver de mudar. Mas nós queremos que você aja responsabilmente, e também que saibamos o que cada um está fazendo. Além disso, mantendo-se em contato com os outros é inspirativo. Junto nós podemos fazer isso funcionar!

Enviando seu trabalho

Mesmo que você seja um membro do projeto, você pode apenas enviar mudanças de uma cópia local se o checkout for feito com o seu login no SF (SourceForge). Se você ainda está trabalhando com uma cópia a qual você fez *check out* anonimamente você primeiro deve fazer um *SSH check out*, e então reaplicar suas mudanças e enviá-las. Reveja o Howto [Gerando as manuais Firebird](#) se você não lembra com fazer um *SSH checkout*.

Se algum tiver passado do seu último *checkout* ou atualização, então faça uma atualização antes de enviar. Isto fará que sua cópia local esteja em sincronismo com o repositório e reduzirá a possibilidade de conflitos.

Assim que você esteja pronto para enviar, vá para a pasta manual. Se você usa um CVS em linha de comando, digite:

```
cvs update -d [ apenas se você deseja fazer uma atualização antes ]
```

```
cvs commit -m "Alguma mensagem informativa aqui"
```

Depois do `-m`, e dentro de aspas, voce digita uma mensagem curto sobre este envio, algo como "Adicionadas algumas funções para a referência da API" ou "Erros no tutorial do isql corrigidos".

Dê a sua senha do SF quando perguntado, e todas as mudanças que você fez - incluindo nas subpastas - serão enviadas. Seu cliente CVS sabe qual servidor deve contatar; esta e outras informações são armazenadas nos subdiretórios CVS que foram criados no *checkout*.

Se você usa outro cliente CVS, reveja sua (do programa) documentação.

Appendix A

História do Documento

A história exata do arquivo é gravado no módulo `manual` na nossa árvore do CVS; veja http://sourceforge.net/cvs/?group_id=9028

Histórico de Revisões

0.1	17 Jan 2004	PV	Primeiro rascunho incompleto publicado sob o título <i>Writing Documentation for Firebird</i> (aka <i>Firebird Docwriting Howto</i>)
0.2	27 Jan 2004	PV	Primeira versão completa (Entrou no CVS: 31 Jan 2004)
1.0	8 Mar 2004	PV	Primeira versão original no site do Firebird.
1.1	26 Fev 2005	PV	<p><i>As seguintes mudanças foram acumuladas entre Março 2004 e Fev. 2005:</i></p> <p>O título foi modificado para <i>Guia de escrita de documentação para o Firebird</i>.</p> <p>Adicionada seção sobre a documentação do PostgreSQL. Explicado o termo XML bem-formatado.</p> <p>Lista de benefícios do DocBook sintetizada.</p> <p>Recomendação sobre elementos <code>section</code> vs <code>sectN</code> modificada.</p> <p>Retirados <code>xref</code> e alguns outros elementos raramente utilizados; adicionado <code>procedure</code>.</p> <p>Atualizada informação sobre <code>literallayout</code> não-monoespaçado.</p> <p>Adicionada seção sobre LDP e como incluir um Aviso de Licença e História do Documento.</p> <p>Numerosas melhorias menores.</p> <p>Adicionada história do documento e número de revisão.</p> <p>Este trabalho está licenciado sob a Licença de Documentação Pública.</p>
1.1-pt_br	26 Mar 2005	FA	Tradução para o Português do Brasil.

Appendix B

Aviso de Licença

Os conteúdos desta Documentação são sujeitos à Licença de Documentação Pública versão 1.0 (a Licença); você só pode usar esta documentação caso se sujeite aos termos desta Licença. Cópias da Licença estão disponíveis em <http://www.firebirdsql.org/pdfmanual/pdl.pdf> (PDF) e <http://www.firebirdsql.org/manual/pdl.html> (HTML).

A Documentação Original é intitulada *Firebird Docwriting Guide*.

O Escritor Inicial da Documentação Original é: Paul Vinkenoog.

O Tradutor para o Português do Brasil é: Fabrício Pontes de Araújo

Copyright (C) 2003-2005. Todos os direitos reservados. Contato com o Escritor Original: paulvink at users dot sourceforge dot net.

Contato com o Tradutor para o Português do Brasil: fabricio.araujo ARROBA infolink.com.br