

Dominando o Progress

Escrito por Márcio Brener Costa

Revisão Técnica

Fernanda Magalhães Favilla

Paulo Afonso Israel Freire

Patrícia Alves de Freitas

Daniela Martino dos Santos

Agradeço a todos os meus amigos da CASSI pelo incentivo e companheirismo que sempre recebi para realização desta obra.

Obrigado em especial aos meus Amigos da juventude, Giuler Alberto, Júlio Pacheco e Roberto Catanhede e a minha família pela ajuda nos meus primeiros passos na descoberta do Mundo da Informática.

Um grande abraço para todos amigos de Brasília!.

Dominando o Progress

“Faça o que é certo ainda que seja o mais difícil.”

Todos os direitos reservados para o autor desta obra.

Toda ou qualquer parte desta publicação poderá ser reproduzida ou transmitida por qualquer meio, seja este eletrônico, mecânico, de fotocópia, de gravação ou outros sem prévia autorização, desde que citado o Autor (Márcio Brener Costa – brener@zipmail.com.br).

Progress, Report Builder, AppServer e WebSpeed são Marcas Registradas da Progress^a Software Corporation EUA.

IBM/DB2 é Marca Registra da International Bussines Machine - IBM^a .

Oracle é Marca Registrada da Oracle^a Inc.

Windows 3.11, Windows 95, Windows 98, Windows NT, Windows 2000 são Marcas Registradas da Microsoft^a .

NetTerm é Marca Registrada da InterSoft International Inc.

KeaX! é Marca Registrada da Attachmate^a

Vipper e PCase são Marcas Registradas da 4Tool Software^a

OpenLink é Marca Registrada da OpenLink^a

Índice

| | |
|-------------------------------------------|----|
| Capítulo 1 | 20 |
| Apresentando o Progress..... | 20 |
| O que é o Progress? | 20 |
| Como funciona?..... | 21 |
| Instalando o software..... | 22 |
| Apresentando os aplicativos..... | 26 |
| Capítulo 2 | 33 |
| Configuração do Produto | 33 |
| Configuração do Progress no Windows | 33 |
| O arquivo progress.ini | 35 |
| Listagem do Arquivo progress.ini | 36 |
| Configurando o Progress para UNIX..... | 42 |
| Variáveis de ambiente | 42 |
| Arquivos | 47 |
| Arquivos comuns | 47 |
| Arquivos temporários | 47 |
| Banco de Dados..... | 48 |
| Sessão cliente | 49 |
| Protocolos disponíveis..... | 49 |
| Capítulo 3 | 50 |
| Criando & Definindo os Dados..... | 50 |
| Criando um banco de dados | 50 |

Dominando o Progress

| | |
|---------------------------------------------|----|
| Como funciona o Banco de Dados..... | 52 |
| Criando tabelas, campos, índices, etc..... | 53 |
| Criando uma tabela..... | 55 |
| Criando os campos | 59 |
| Validando os Campos..... | 61 |
| A propriedade Validation | 62 |
| Triggers..... | 63 |
| O que um Trigger? | 63 |
| Criando índices | 65 |
| Aparência do Campo (View-As) | 69 |
| Sequences | 70 |
| O que são Sequences?..... | 70 |
| Como usar um Sequence?..... | 70 |
| Metaschema | 71 |
| O que é o Metaschema?..... | 71 |
| Capítulo 4 | 73 |
| Aprendendo a linguagem..... | 73 |
| Desenvolvendo o código | 73 |
| O primeiro programa | 73 |
| Como fazer a interface com o usuário? | 73 |
| Conectando o banco de dados | 74 |
| Aprendendo a sintaxe | 77 |
| Definindo variáveis | 77 |
| View-As | 83 |
| Definindo Frames..... | 86 |

Dominando o Progress

| | |
|------------------------------------------------------|-----|
| Definindo Queries | 95 |
| O quê é uma Query? | 95 |
| BROWSES..... | 97 |
| Utilizando a Query no Browse | 103 |
| OPEN QUERY | 103 |
| ASSIGN | 105 |
| Comandos de Impressão e Manipulação de Arquivos | 106 |
| OUTPUT | 106 |
| PARÂMETROS..... | 106 |
| INPUT | 110 |
| Eventos | 114 |
| Lista de Eventos | 114 |
| Preprocessadores | 143 |
| O que são preprocessadores?..... | 143 |
| Utilização | 143 |
| &SCOPED-DEFINE..... | 143 |
| &GLOBAL-DEFINE | 144 |
| &UNDEFINE | 145 |
| DIRETIVAS PREPROCESSADAS | 145 |
| Operadores..... | 146 |
| &IF, &THEN, &ELSEIF, &ELSE, &ENDIF | 146 |
| MESSAGE | 147 |
| PARÂMETROS..... | 147 |
| Criando MENUS | 150 |
| MENU | 150 |

Dominando o Progress

| | |
|----------------------------------------|-----|
| BOTÕES..... | 155 |
| PARÂMETROS | 159 |
| Importação e Exportação de Dados | 162 |
| Limites do Progress..... | 163 |
| Banco de Dados..... | 163 |
| VARIÁVEIS | 164 |
| NOMENCLATURAS..... | 165 |
| Funções..... | 166 |
| ABSOLUTE | 166 |
| ACCUM | 166 |
| ALIAS | 167 |
| AMBIGUOUS..... | 167 |
| ASC | 168 |
| AVAILABLE..... | 168 |
| CAN-DO | 169 |
| CAN-FIND | 169 |
| CAN-QUERY | 170 |
| CAN-SET | 171 |
| ENCODE..... | 171 |
| CHR..... | 172 |
| CODEPAGE-CONVERT | 172 |
| CONNECTED..... | 173 |
| COUNT-OF | 173 |
| CURRENT-CHANGED | 174 |
| CURRENT-LANGUAGE..... | 175 |

Dominando o Progress

| | |
|----------------------------|-----|
| CURRENT-RESULT-ROW | 175 |
| CURRENT-ROW-MODIFIED | 176 |
| CURRENT-VALUE | 176 |
| DATASERVERS | 176 |
| DATE | 176 |
| DAY | 177 |
| DBCODEPAGE..... | 177 |
| DBCOLLATION..... | 178 |
| DBNAME | 178 |
| DBPARAM | 178 |
| DBRESTRICTIONS..... | 179 |
| DBTASKID..... | 179 |
| DBTYPE | 180 |
| DBVERSION | 180 |
| DECIMAL | 181 |
| DEFINED | 181 |
| DYNAMIC-FUNCTION | 182 |
| ENTERED..... | 183 |
| ENTRY..... | 183 |
| ETIME | 184 |
| EXP | 184 |
| EXTENT | 184 |
| FILL | 185 |
| FIRST | 185 |
| FIRST-OF..... | 186 |
| FRAME-COL..... | 187 |

Dominando o Progress

| | |
|--------------------------|-----|
| FRAME-DB..... | 187 |
| FRAME-DOWN | 187 |
| FRAME-FIELD | 188 |
| FRAME-FILE | 188 |
| FRAME-INDEX | 188 |
| FRAME-LINE..... | 189 |
| FRAME-NAME | 189 |
| FRAME-ROW | 189 |
| FRAME-VALUE | 190 |
| GATEWAYS | 190 |
| GET-BYTE | 190 |
| GET-CODEPAGES..... | 190 |
| GET-COLLATIONS..... | 191 |
| GET-DOUBLE..... | 191 |
| GET-FLOAT | 192 |
| GET-LONG..... | 192 |
| GET-POINTER-VALUE | 192 |
| GET-SHORT..... | 192 |
| GET-SIZE | 192 |
| GET-STRING..... | 193 |
| GET-UNSIGNED-SHORT | 193 |
| INDEX | 193 |
| INTEGER..... | 193 |
| IS-ATTR-SPACE | 194 |
| KBLABEL | 194 |
| KEYCODE | 195 |

Dominando o Progress

| | |
|------------------------|-----|
| KEYFUNCTION..... | 195 |
| KEYLABEL | 196 |
| KEYWORD..... | 197 |
| KEYWORD-ALL | 197 |
| LAST | 197 |
| LASTKEY..... | 198 |
| LAST-OF | 198 |
| LC..... | 198 |
| LDBNAME | 198 |
| LEFT-TRIM..... | 199 |
| LENGTH..... | 199 |
| LINE-COUNTER | 199 |
| LIST-EVENTS..... | 199 |
| LIST-QUERY-ATTRS | 200 |
| LIST-SET-ATTRS..... | 200 |
| LIST-WIDGETS..... | 200 |
| LOCKED..... | 201 |
| LOG | 201 |
| LOOKUP..... | 201 |
| MAXIMUM | 202 |
| MESSAGE-LINES | 202 |
| MINIMUM | 202 |
| MONTH | 203 |
| NEW..... | 203 |
| NEXT-VALUE | 203 |
| NOT ENTERED..... | 203 |

Dominando o Progress

| | |
|--------------------|-----|
| NUM-ALIASES..... | 204 |
| NUM-DBS..... | 204 |
| NUM-ENTRIES | 204 |
| NUM-RESULTS..... | 205 |
| OPSYS | 205 |
| OS-DRIVES..... | 206 |
| OS-ERROR | 206 |
| OS-GETENV..... | 206 |
| PAGE-NUMBER | 207 |
| PAGE-SIZE | 207 |
| PDBNAME | 207 |
| PROGRAM-NAME..... | 208 |
| PROGRESS | 208 |
| PROMSGS | 208 |
| PROPATH..... | 209 |
| PROVERSION..... | 209 |
| QUERY-OFF-END..... | 209 |
| RANDOM..... | 210 |
| RECID..... | 211 |
| RECORD-LENGTH..... | 212 |
| REPLACE..... | 212 |
| RETRY | 213 |
| RETURN-VALUE | 214 |
| RIGHT-TRIM..... | 214 |
| R-INDEX | 214 |
| ROUND | 215 |

Dominando o Progress

| | |
|-----------------------------------------------------|-----|
| ROWID | 215 |
| SCREEN-LINES | 215 |
| SDBNAME | 215 |
| SEARCH..... | 216 |
| SEEK | 216 |
| SETUSERID | 216 |
| SQRT | 216 |
| STRING | 217 |
| SUBSTITUTE | 217 |
| SUBSTRING..... | 218 |
| TERMINAL | 219 |
| TIME | 219 |
| TODAY | 220 |
| TO-ROWID..... | 220 |
| TRANSACTION | 220 |
| TRIM..... | 221 |
| TRUNCATE..... | 221 |
| USERID | 221 |
| VALID-EVENT | 222 |
| VALID-HANDLE..... | 222 |
| WEEKDAY | 222 |
| WIDGET-HANDLE | 223 |
| YEAR..... | 223 |
| Capítulo n | 224 |
| Utilitários & Parâmetros do Banco de Dados Progress | 224 |

Dominando o Progress

| | |
|----------------------------------------------|-----|
| Utilitários do Banco de Dados Progress | 224 |
| PARÂMETROS | 226 |
| Parâmetro | 230 |
| Descrição | 230 |
| -1 | 230 |
| -a arquivo | 230 |
| -aibufs n | 230 |
| -aistall | 231 |
| -b | 231 |
| -basekey texto | 231 |
| -B n | 232 |
| -bibufs n | 232 |
| -brl | 233 |
| -Bt n | 233 |
| -C qualificador | 233 |
| -c n | 233 |
| -cache arquivo | 234 |
| -charset Código de Página | 234 |
| -checkdbe | 234 |
| -convmap arquivo | 234 |
| -cp parâmetros ou arquivo | 235 |
| -cpcase tabela | 235 |
| -cpcoll coleção | 235 |
| -cpdb Código Página | 236 |
| -cpinternal Código Página | 236 |

Dominando o Progress

| | |
|---------------------------------|-----|
| -cpprint Código Página | 236 |
| -cprcodein Código Página..... | 236 |
| -cprcodeout Código Página | 237 |
| -cpstream | 237 |
| -cpterm | 237 |
| -cs n, i | 237 |
| -ct n | 238 |
| -d mdy..... | 238 |
| -D n | 238 |
| -db banco de dados | 239 |
| -debug | 239 |
| -decdtm..... | 239 |
| -dictexps..... | 239 |
| -directio..... | 240 |
| -dt | 240 |
| -E | 240 |
| -esqllog | 241 |
| -esqlnopad..... | 241 |
| -evtleve parâmetro | 241 |
| -F..... | 241 |
| -fc n | 241 |
| -fldisable..... | 242 |
| -g arquivo | 242 |
| -G n | 242 |
| -h n..... | 242 |
| -H servidor | 243 |

Dominando o Progress

| | |
|------------------------|-----|
| -H servidor | 243 |
| -hardlimit | 243 |
| -hs s..... | 243 |
| -i..... | 244 |
| -iname arquivo | 244 |
| -inp n | 245 |
| -is | 245 |
| -k..... | 245 |
| -L n..... | 246 |
| -ld nome | 246 |
| -lng linguagem..... | 247 |
| -logfile arquivo | 247 |
| -logname arquivo | 247 |
| -m1 | 247 |
| -m2 | 247 |
| -m3 | 248 |
| -Ma n..... | 248 |
| -maxport n..... | 248 |
| -Mf n | 248 |
| -Mi n..... | 249 |
| -minport n..... | 249 |
| -Mm n..... | 249 |
| -mmax n..... | 249 |
| -Mn | 249 |
| -Mp n | 250 |
| -Mpb n..... | 250 |

Dominando o Progress

| | |
|---------------------|-----|
| -Mr n..... | 250 |
| -Mv n | 250 |
| -Mxs n..... | 250 |
| -n n..... | 251 |
| -N protocolo..... | 251 |
| -nb n..... | 251 |
| -Nd device | 251 |
| -NL..... | 252 |
| -Nn Cliente..... | 252 |
| -Nv n..... | 252 |
| -o impressora | 252 |
| -p procedure | 253 |
| -P senha | 253 |
| -param texto | 253 |
| -pf arquivo | 253 |
| -plm..... | 253 |
| -pls | 254 |
| -populate..... | 254 |
| -pp diretório..... | 254 |
| -pwqdelay..... | 254 |
| -pwqmin n | 254 |
| -pwscan n | 255 |
| -pwsdelay n | 255 |
| -pwwmax n..... | 255 |
| -q | 255 |
| -Q..... | 256 |

Dominando o Progress

| | |
|-----------------------------|-----|
| -Q2 | 256 |
| -r | 256 |
| -rand n | 256 |
| -rg | 257 |
| -RO | 257 |
| -rptint n | 257 |
| -rq | 257 |
| -rr | 257 |
| -s n | 258 |
| -S serviço | 258 |
| -Sn serviço | 259 |
| -spin n | 259 |
| -stream Código Página | 259 |
| -stsh n | 259 |
| -SYBc n | 259 |
| -SYBt n | 260 |
| -t | 260 |
| -T diretório | 260 |
| -TB n | 260 |
| -TM n | 260 |
| -tok n | 261 |
| -trig diretório | 261 |
| -tstamp | 261 |
| -usrcount n | 261 |
| -U usuário | 261 |
| -vw | 262 |

Dominando o Progress

| | |
|--------------------------------------------------|-----|
| -VO versão | 262 |
| -v6colon | 262 |
| -v6q | 262 |
| -xc coleção..... | 262 |
| -y..... | 263 |
| -yc | 263 |
| -yd..... | 263 |
| -yx | 264 |
| -yy ano | 265 |
| Banco de Dados Multi-Volumes | 266 |
| O que é o banco multi-volumes?..... | 266 |
| Por que usar Banco Multi-Volumes? | 267 |
| Criando a estrutura multi-volumes | 268 |
| Formação do arquivo de estrutura | 268 |
| Criando o Banco | 269 |
| MAGNUS | 272 |
| Apresentação..... | 272 |
| Como o MAGNUS funciona? | 272 |
| Lista de Diretórios do MAGNUS (principais) | 273 |
| Funcionamento do MENU..... | 274 |
| Validando o acesso ao Programa | 276 |
| Bancos Históricos | 277 |
| Desenvolver em Windows para usar no UNIX... .. | 279 |
| Criar Bancos para Testes/Desenvolvimento..... | 284 |
| Criando os bancos..... | 284 |

Dominando o Progress

| | |
|--------------------------------------------------|-----|
| UNIX AIX | 286 |
| Windows NT | 287 |
| Utilizando zoom de cadastro Datasul | 289 |
| Capítulo n | 293 |
| Softwares, Utilitários & Sites diversos | 293 |
| NetTerm | 293 |
| Configure as teclas de acordo com o MAGNUS | 294 |
| Kea!X | 295 |
| PCase | 296 |
| Viper | 297 |
| ERWin | 297 |
| Sites sobre Progress | 299 |

Capítulo 1

Apresentando o Progress

O que é o Progress?

A linguagem Progress foi desenvolvida pela empresa Progress Software Corporation em 1984, inicialmente para sistema operacional UNIX com uso em mainframes para processamento de grande volume de dados, como alternativa para outras linguagens da época como Cobol, Adabas, Natural, Clipper, etc, que exigiam do programador escrever um código muito extenso para qualquer aplicação. Também uma alternativa como Banco de Dados Relacional de alta performance e segurança, embutido em um único produto.

A Progress Corporation tem sua sede em Bedford, Massachusetts USA, com filiais em diversos países. No Brasil sua representante é a Progress do Brasil/SP.

Uma das maiores atrativos do Progress é sua portabilidade e independência de plataforma. Ele funciona em praticamente todos os sistemas operacionais existentes como DOS, Windows 3x,95,NT, UNIX, OS/2, Novell, VMS, Motif, Xenix, CTOS entre diversos outros, isso utilizando o mesmo código fonte.

Seu Banco de Dados Relacional é de altíssima performance e extrema segurança, igual ou até mesmo

superior ao bancos mais populares do mercado como Oracle, DB/2 IBM, Sybase, MS-SQL Server, entre outros.

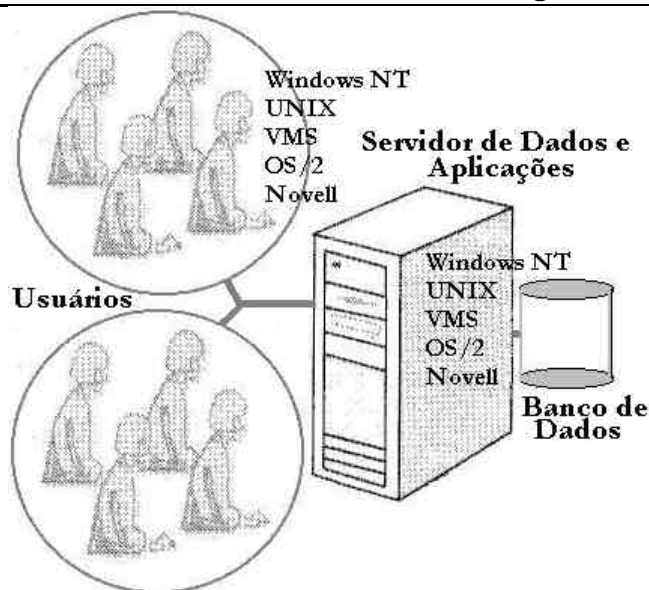
A perfeita integração entre linguagem e banco de dados fazem do Progress uma excelente ferramenta para construção de qualquer aplicação comercial. Isso porque as camadas de desenvolvimento - regras de negócios, dados e interface estão totalmente interligadas, o que evita qualquer redundância ou retrabalho em qualquer camada da aplicação.

A atualização e distribuição dos aplicativos em Progress também é bastante simplificada. Um sistema pode ser totalmente centralizado para acesso de vários clientes simultâneos e atualizado a qualquer tempo, parcialmente ou completamente sem gerar qualquer erro ou necessitar de uma parada momentânea do sistema.

Como funciona o Progress?

O funcionamento do Progress é bastante simples. A configuração típica e mais comum é a instalação e armazenamento centralizado do Progress servidor e cliente, bancos de dados e aplicações em um servidor disponível para acesso de diversos clientes em rede.

Há também diversas outras configurações possíveis para uma aplicação Progress: múltiplos servidores, bancos de dados distribuídos entre servidores e/ou plataformas distintas, como será visto no capítulo Arquiteturas de aplicação. A figura a seguir exemplifica melhor uma configuração mais simples.



Arquitetura de aplicação

Servidor de rede com instalação do Progress server e client;
Sessão multi-usuário dos bancos de dados;
Aplicativos em Progress

Instalando o software

Será apresentada a instalação do Provision para Windows

Os pré-requisitos para instalação são apenas mídia(s) e licença(s) para começar a usar.

**WELCOME TO THE PROGRESS
SOFTWARE
SETUP UTILITY**

**Release 8.2B
Copyright (c) 1992-1997
Progress Software Corp.
All Rights Reserved.**



Tela inicial de instalação

Este é a primeira tela para instalação do Progress versão 8.x para sistema operacional Microsoft Windows 95/98/NT

Será preciso especificar na tela de instalação o Nome da Empresa, o Número de Série e Códigos de Controle para validar o produto e prosseguir a instalação.

Você pode também acrescentar outros produtos à mesma instalação, especificando os números de séries e validadores.

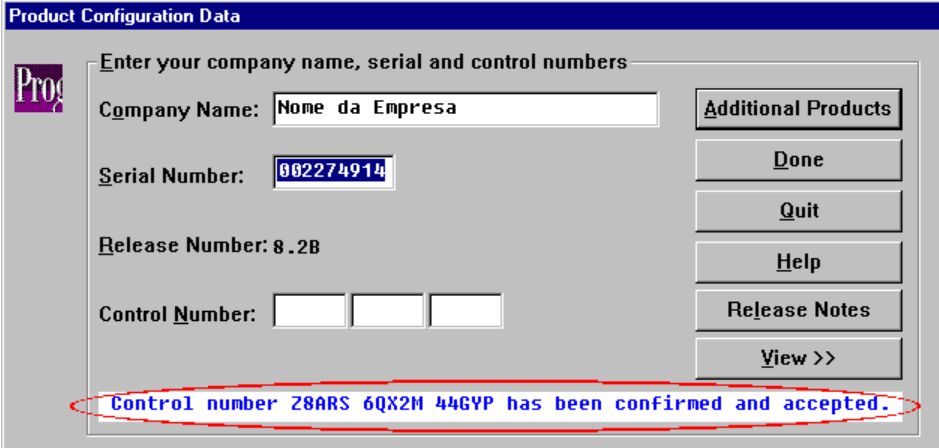
Informe os dados e clique no botão **Additional Products** para acrescentar à lista de programas, conforme mostrado na figura acima destacado em azul

A próxima tela solicita a localização para instalação do Progress, você pode especificar o drive o diretório aonde serão copiados os arquivos. Por padrão o diretório sugerido é "DLC", mas você pode alterá-lo para outro de sua escolha. Todos os dados informados na instalação serão incluídos na configuração do Progress contida no arquivo DLC/progress.cfg, o qual validará a execução de todos os aplicativos Progress.



Como é padrão do Progress o diretório DLC toda sua literatura o cita por padrão também. Então saiba que a

nomenclatura DLC sempre se refere à localização de instalação do Progress.



The 'Product Configuration Data' dialog box contains fields for 'Company Name' (None da Empresa), 'Serial Number' (002274914), 'Release Number' (8.2B), and 'Control Number' (three empty boxes). On the right are buttons for 'Additional Products', 'Done', 'Quit', 'Help', 'Release Notes', and 'View >>'. A red oval highlights a status message at the bottom: 'Control number 28ARS 6QX2M 44GYP has been confirmed and accepted.'

Product Configuration Data

Enter your company name, serial and control numbers

Company Name: None da Empresa

Serial Number: 002274914

Release Number: 8.2B

Control Number:

Additional Products

Done

Quit

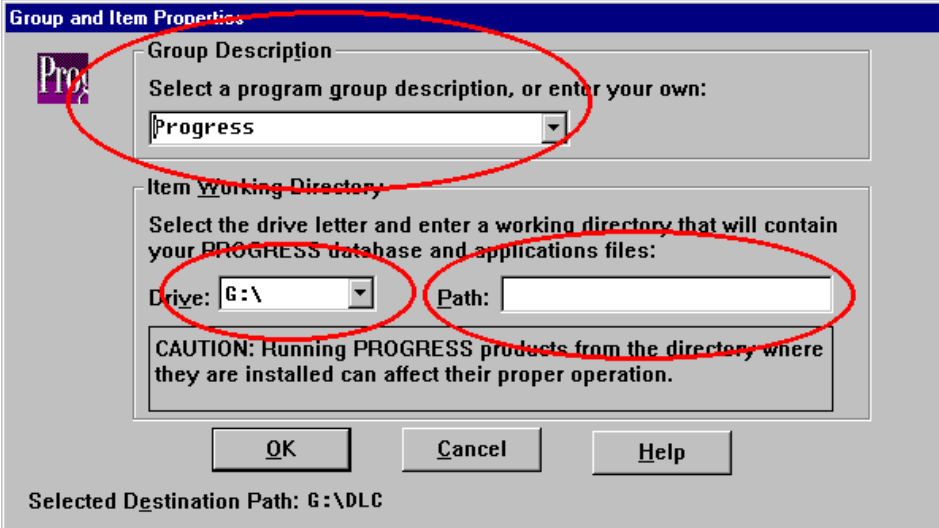
Help

Release Notes

View >>

Control number 28ARS 6QX2M 44GYP has been confirmed and accepted.

A próxima tela solicita o grupo de programas a ser criado no servidor e o diretório padrão aonde o Progress salvará arquivos de programas, temporários entre outros.



The 'Group and Item Properties' dialog box has two sections. The 'Group Description' section has a dropdown menu with 'Progress' selected. The 'Item Working Directory' section has a 'Drive' dropdown with 'G:\' and an empty 'Path' field. A red oval circles the 'Group Description' section. Another red oval circles the 'Drive' and 'Path' fields. A third red oval circles a 'CAUTION' message at the bottom. At the bottom are 'OK', 'Cancel', and 'Help' buttons. Below the dialog, it says 'Selected Destination Path: G:\DLC'.

Group and Item Properties

Group Description

Select a program group description, or enter your own:

Progress

Item Working Directory

Select the drive letter and enter a working directory that will contain your PROGRESS database and applications files:

Drive: G:\

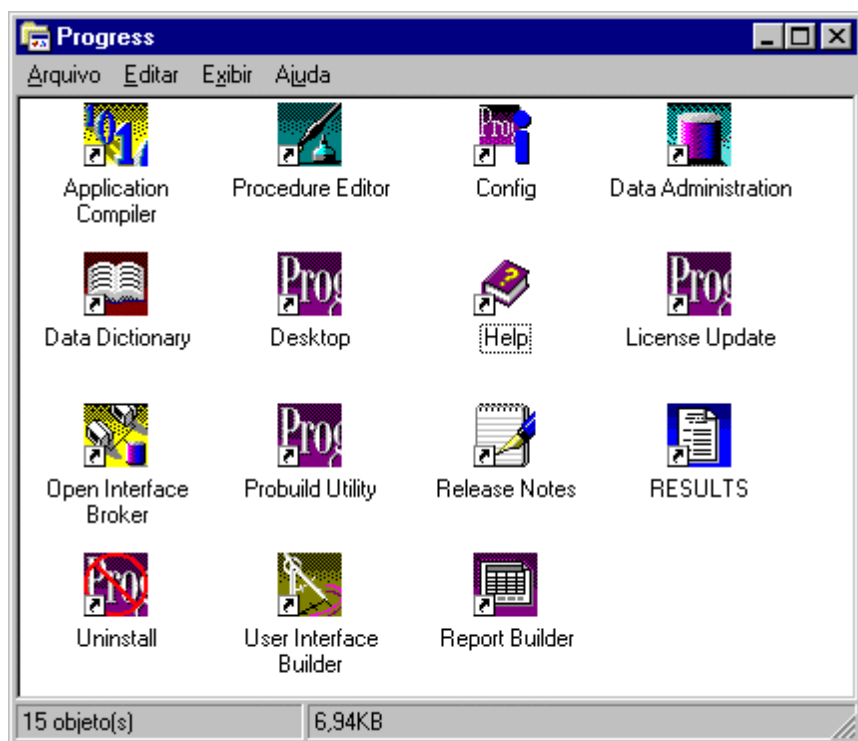
Path:

CAUTION: Running PROGRESS products from the directory where they are installed can affect their proper operation.

OK Cancel Help

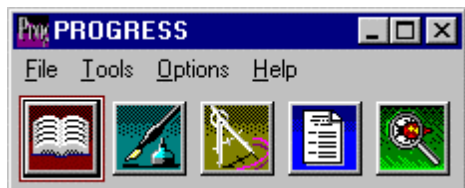
Selected Destination Path: G:\DLC

Após o término da instalação serão criados esses ícones no grupo de programas informado na instalação, como a figura abaixo:



A figura acima demonstra os utilitários do Progress Provision, um tipo de licença full da Progress similar por exemplo ao Microsoft Visual Basic Professional ou Delphi Client/Server.

Apresentando os aplicativos

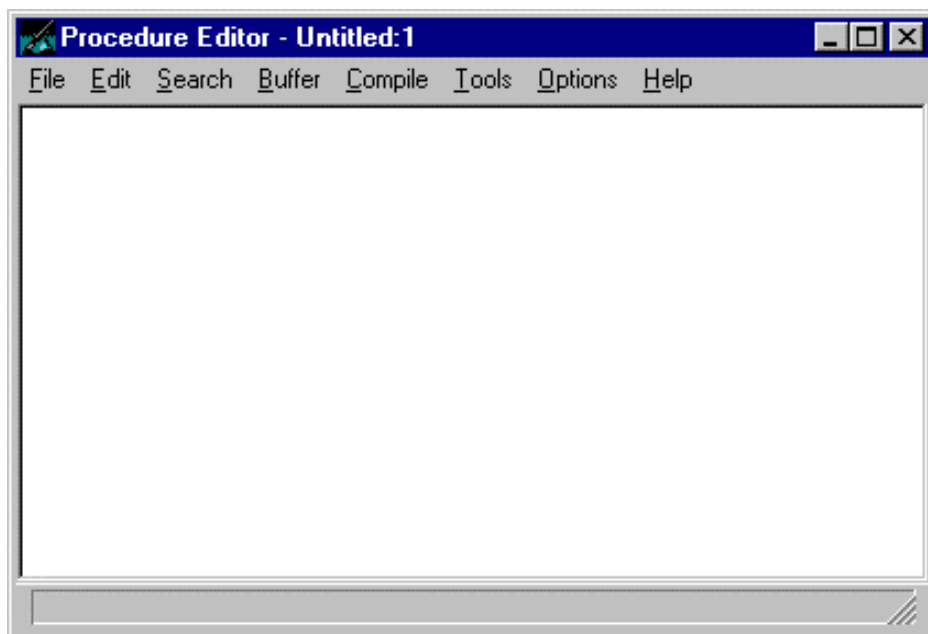


Desktop

É um aplicativo desenvolvido em Progress 4GL modo gráfico que apresenta os botões dos aplicativos Progress disponíveis.

Arquivo: **`_desk.p`**

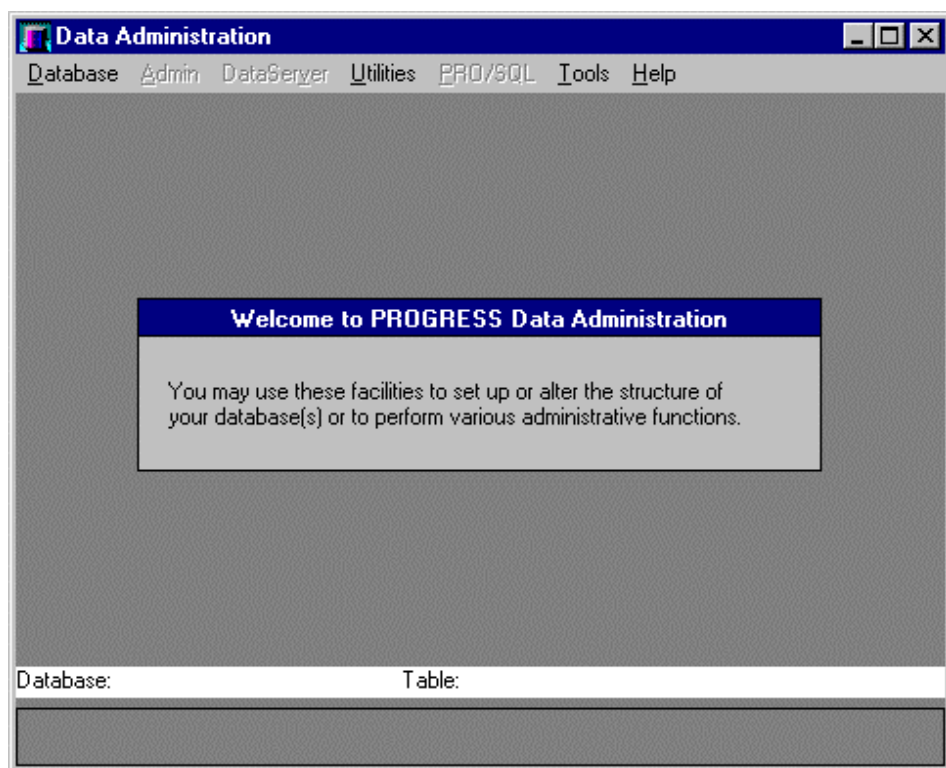
Procedure Editor



Aplicativo em Progress 4GL para escrita do código fonte, execução de procedures, debugger, compilação e qualquer procedimento por meio de código.

Arquivo **`_edit.p`**

Data Administration



Aplicativo desenvolvido em Progress 4GL para administração e manutenção dos bancos de dados.

Com ele você pode:

Criar, conectar e alterar definições de bancos de dados;

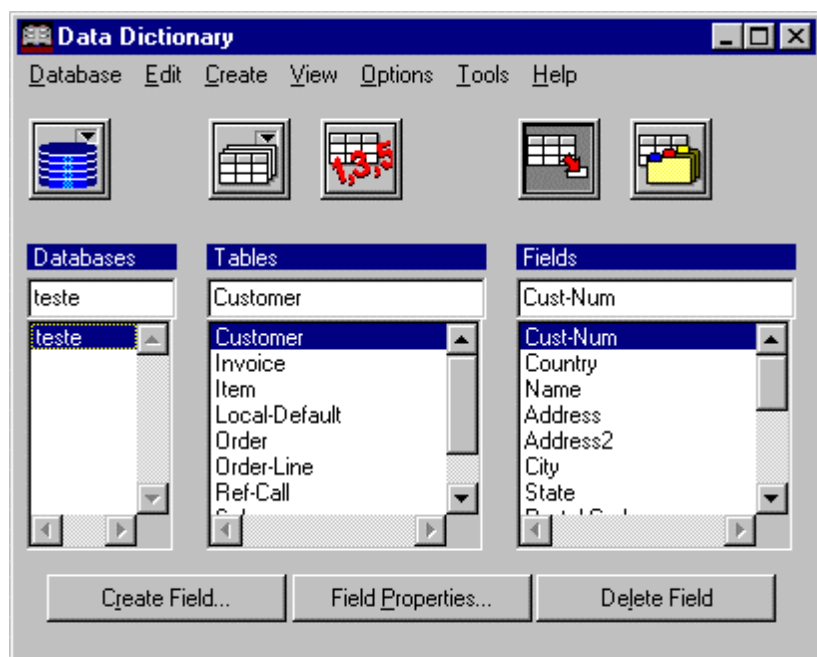
Visualizar relatórios de estruturas de tabelas, campos, índices;

Definir segurança dos dados, salvar definições e conteúdo dos dados, editar parâmetros, etc.

Arquivo: **_admin.p**

Escrito por Márcio Brener - <http://www.geocities.com/marcio-brener>

Data Dictionary

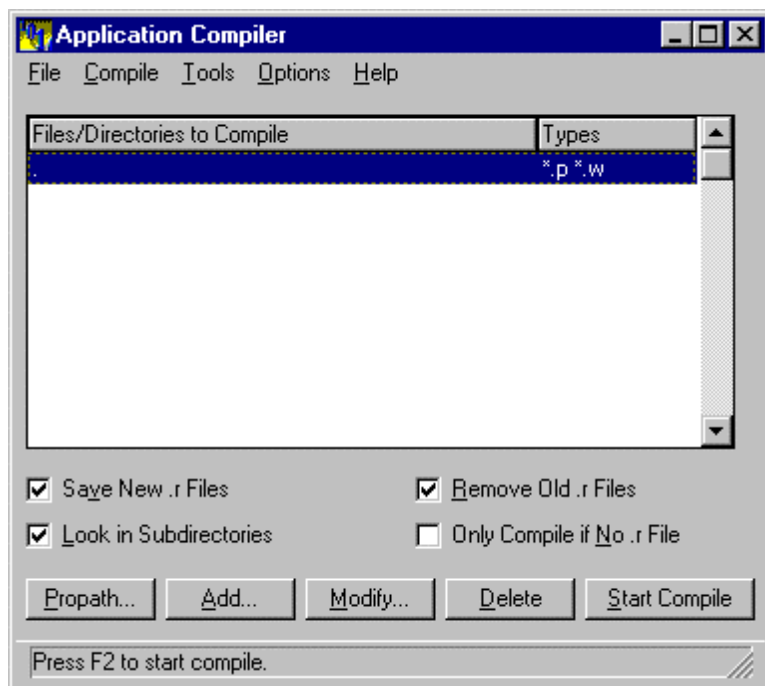


Com Dicionário de Dados você pode modelar a estrutura de tabelas, campos, índices, seqüências e triggers.

Também um aplicativo totalmente em Progress 4GL

Arquivo: **`_dict.p`**

Application Compiler



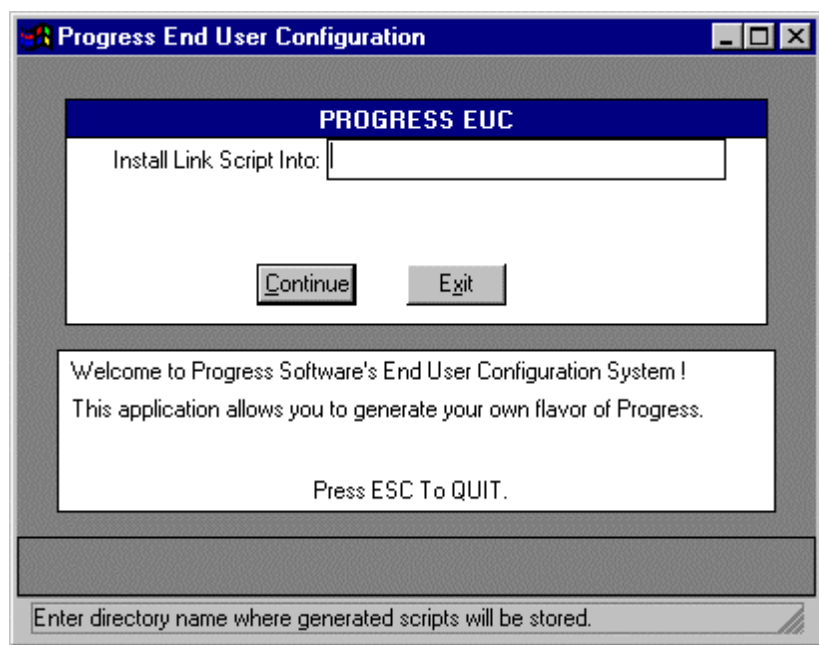
Como o próprio nome diz é um aplicativo para compilação de procedures (programas).

Ele possibilita compilar um diretório ou uma árvore de diretórios, especificar arquivos iniciados por uma subpalavra ou apenas extensão, confirmar sobreposição de procedures já existentes, etc.

Também desenvolvido em Progress 4GL

Arquivo: **`_comp.p`**

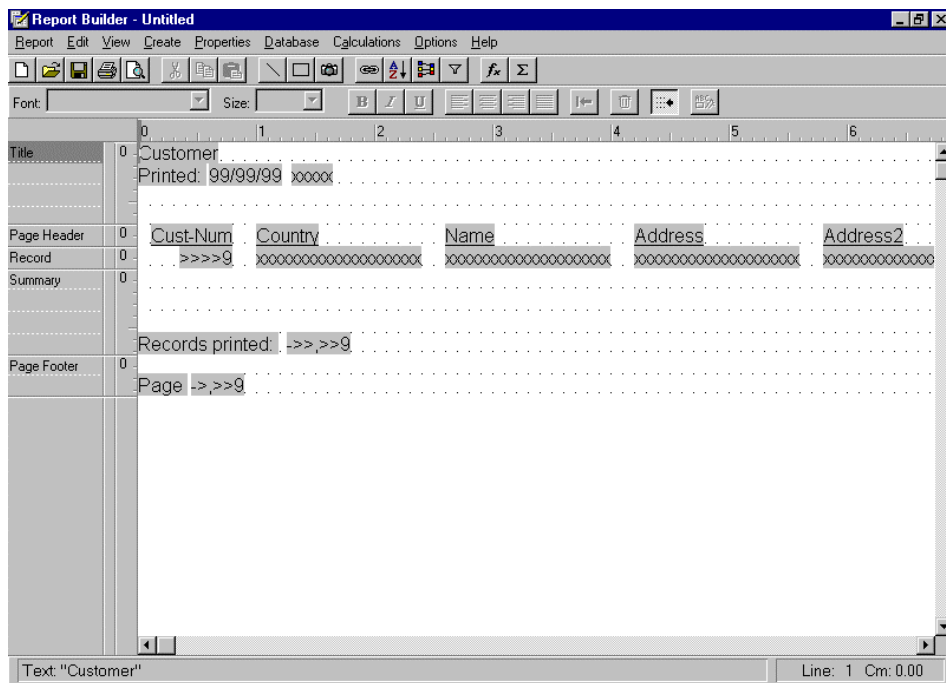
User Configuration



Utilitário desenvolvido em Progress 4GL para criação e edição de scripts de conexão e/ou de inicialização do Progress.

Arquivo: **`_script.p`**

Report Builder



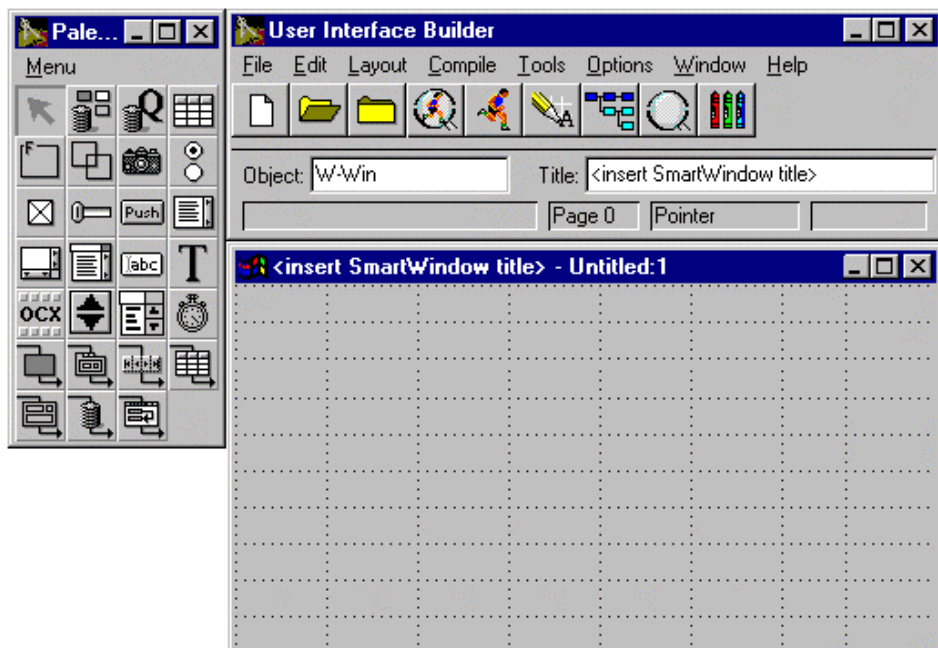
Utilitário do Progress para criação de relatório totalmente gráficos em MS-Windows.

Com ele você pode usar tabelas, inserir gráficos, usar e criar funções definidas pelo usuário.



Este aplicativo Progress é disponível apenas para ambiente MS-Windows 3.x/95 ou NT.

UIB (User Interface Builder)



O User Interface Builder é um utilitário desenvolvido em Progress 4GL para edição rápida e gráfica de programas baseados no Windows.

Ele foi introduzido a partir da versão 7 para Windows trazendo consigo diversas inovações e conceitos quanto à orientação a objetos, encapsulamento, polimorfismo e hereditriedade, como um início para entrada da linguagem Progress na Quarta Geração, daí Progress 4GL.

Arquivo: **_uib.p**



Este utilitário não fará parte da abordagem deste livro.

Capítulo 2

Configuração do Software

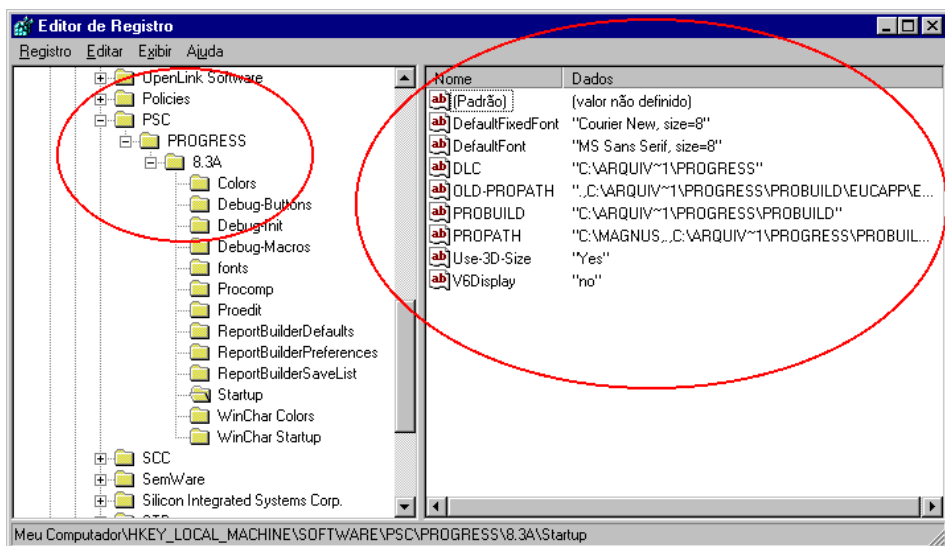
Configuração do Progress no Windows

O Progress por abranger diversos sistemas operacionais tem algumas diferenças quanto à configuração do ambiente para funcionar adequadamente. Por exemplo, em UNIX, MS-DOS e Novell as parametrizações são estabelecidas por meio de linha de comando e definições de variáveis de ambiente, no Windows 95 ou NT os parâmetros são registrados por meio de arquivos INIs, ou ainda na Registry – como é o novo padrão introduzido pela versão 8.2 e superiores.

A instalação do Progress 8.x para Windows grava as informações relativas às propriedades do aplicativo na Registry do Windows, e também em arquivo INI chamado **progress.ini** como na versão 7.x, localizado no diretório DLC/bin.

As entradas na Registry estarão nos diretórios HKEY_CURRENT_USER ou HKEY_LOCAL_MACHINE nesta sequência, na pasta **Software**, em uma sub-pasta chamada **PSC** (Progress Software Corporation), conforme mostra a figura abaixo:

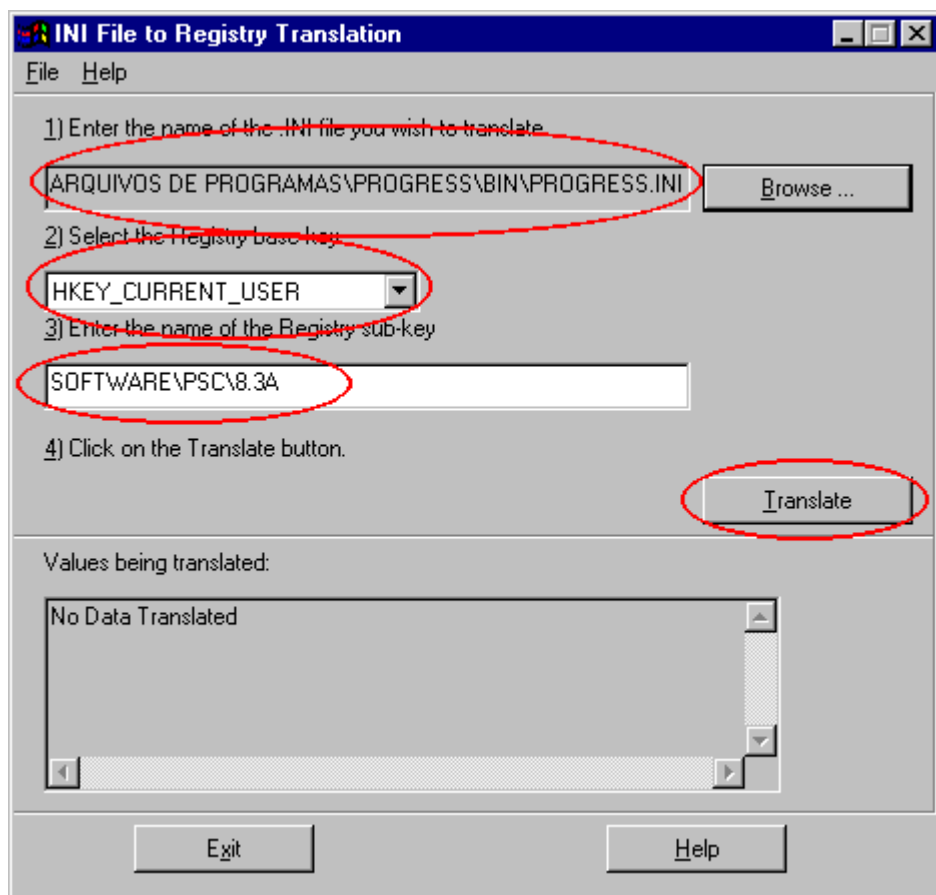
Configuração na Registry



As configurações contidas na Registry também estão no arquivo INI e podem ser usada para definir as propriedades para o funcionamento e aparência das aplicações.

Você também pode registrar as configurações contidas em um arquivo INI para a Registry do Windows de uma máquina cliente, através de um aplicativo em **DLC\bin\ini2reg.exe**, veja a figura a seguir:

Inclui um arquivo INI na Registry do Windows



O arquivo progress.ini

Os parâmetros **V6Display**, **V6Colon**, **V6Frame**, **V6FKeys**, **V6Keys**, **DefaultFont** **FixedSys**, **DefaultFixedFont** **FixedSys** definem uma aparência para a aplicação igual a utilizada na versão Progress 6, com tela na dimensão de 80 linhas por 21 colunas, área de mensagens e área de status e fonte proporcional, de

acordo com a configuração para funcionamento do **MAGNUS versão I**.



Caso você não conheça, o **MAGNUS** é um Sistema de Gestão Empresarial, o mais popular no Brasil. Desenvolvido totalmente em Progress pela Datasul, uma empresa brasileira sediada em Joinvile Santa Catarina.

Listagem do Arquivo progress.ini

```
[Startup]
V6Display=no
V6Colon=no
V6Frame=no
V6FKeys=no
V6Keys=no
ImmediateDisplay=yes
MultitaskingInterval=100
DefaultFont=FixedSys
DefaultFixedFont=FixedSys
DLC=C:\DLC
Use-3D-Size=Yes
PROBUILD=C:\DLC\PROBUILD
PROPATH=.,D:\MAGNUS,C:\DLC\PROBUILD\EUCAPP\EUC.PL,C:\DLC\PROBUILD\EUCAPP,C:\DLC\gui,C:\DLC,C:\DLC\bin
OLD-PROPATH=.,C:\DLC\PROBUILD\EUCAPP\EUC.PL,C:\DLC\PROBUILD\EUCAPP
PROCFG=C:\DLC\PROGRESS.CFG
PROMSGS=C:\DLC\PROMSGS
PROMON=C:\TEMP\ESTAT.LST

[Colors]
;*****
; THE DEFINITION OF COLOR 0 THROUGH 15 IS PRIVATE TO THE PROGRESS ADE.
; MODIFYING COLORS 0 THROUGH 15 MAY PREVENT THE PROGRESS ADE FROM RUNNING.
; The following color definitions correspond to the ADE standards.
; 0 to 15 - reserved
color0=0,0,0
color1=0,0,128
color2=0,128,0
color3=0,128,128
color4=128,0,0
color5=128,0,128
```

```
color6=128,128,0
color7=128,128,128
color8=192,192,192
color9=0,0,255
color10=0,255,0
color11=0,255,255
color12=255,0,0
color13=255,0,255
color14=255,255,0
color15=255,255,255
color16=255,255,0
```

```
NORMAL=0,15
INPUT=15,0
MESSAGES=15,1
```

[Default Window]

```
;x=
;y=
;rows=
;columns=
```

[fonts]

```
;*****
; THE DEFINITION OF FONT 0 THROUGH 7 IS PRIVATE TO THE PROGRESS ADE.
; MODIFYING FONTS 0 THROUGH 7 MAY PREVENT THE PROGRESS ADE FROM RUNNING.
; The following fonts definitions correspond to the ADE standards.
;      ? - DefaultFont from Startup Section
;      0 - DefaultFixedFont from Startup Section (1 char per PPU)
;      1 - Proportional System Font
;      2 - Editor Font for 4GL program entry
;      3 - TTY Simulator (should be fixed)
;      4 - Dynamically-sized widgets, eg status-line, selection-list
;      5 - Static widgets, eg. combo-boxes
;      6 - Dynamic, bold (TranMan2)
;      7 - Reserved
font0=Courier New, size=8
font1=MS Sans Serif, size=8
font2=Courier New, size=8
font3=Courier New, size=8
font4=MS Sans Serif, size=8
font5=MS Sans Serif, size=10
font6=MS Sans Serif, size=8, bold
font7=MS Sans Serif, size=8
font8=Courier New, size=12
font9=MS Sans Serif, size=8
font10=Garamond, size=20
font11=MS Sans Serif, size=8
```

[WinChar Startup]

```

DLC=C:\DLC
PROBUILD=C:\DLC\PROBUILD
PROPATH=.,C:\DLC\PROBUILD\EUCAPP\EUC.PL,C:\DLC\PROBUILD\EUCAPP
;DLC=
;PROCFG=
;PROMSGS=
;PROPATH=

```

[WinChar Colors]

```

color0=WHITE/BLUE           "NORMAL"
color1=BLACK/GRAY           "INPUT, UNDERLINE"
color2=BLACK/GRAY           "MESSAGES, REVERSE"
color3=BLUE/WHITE           "HIGHLIGHT, HELP"
color4=BLINK-RED/WHITE      "URGENT"

```

```

;NORMAL=WHITE/BLUE
;INPUT=BLACK/GRAY
;MESSAGES=BLACK/GRAY

```

[WinChar Default Window]

```

;rows=25
;rows=50

```

[WinChar Keys]

```

;GO=F1,CTRL-X

```

[Debug-Init]

```

;*****
; THE FOLLOWING INFORMATION IS PRIVATE TO THE DEBUGGER. IT SHOULD NEVER BE
; MODIFIED EXCEPT BY THE DEBUGGER PROGRAM.
Location=1,146,15,460,464
Pane0=1,46," "
Panel=1,14," "
Pane2=1,16,"Commands in Queue: 0"
Pane3=1,25," "
DebuggerName=PRODEBUG.EXE

```

[Debug-Macros]

```

Macro0=r, run &file
Macro1=c, continue
Macro2=s, step
Macro3=n, next
Macro4=b, break &file &line
Macro5=cb, cancel break &file &line
Macro6=sb, show breaks
Macro7=ss, show stack

```

```
Macro8=u, up
Macro9=d, down
Macro10=di, display &text
Macro11=sm, show macros
```

[Debug-Buttons]

```
Button0=Run,run &file
Button1=Cont,continue
Button2=Step,step
Button3=Next,next
Button4=Break,break &file &line
Button5=Stack,show stack
Button6=Up,up
Button7=Down,down
Button8=Disp,display &text
Button9=Exit,exit
```

[ProADE]

```
;DividerFgColor=15
;DividerBgColor=1
;OKBoxFgColor=1
;OKBoxBgColor=8
;FillinFgColor=0
;FillinBgColor=8
;Editor4GLFgColor=DEFAULT
;Editor4GLBgColor=DEFAULT
;Editor4GLFgSmallColor=0
;Editor4GLBgSmallColor=8
;FixedFont=0
;StandardFont=1
;Editor4GLFont=2
;EditorTabStop=4
```

[Proedit]

```
SaveSettings=yes
BufList=
ExitWarning=yes
SaveBufList=no
MinimizeBeforeRun=no
RestoreAfterRun=yes
PauseAfterRun=yes
AutoCleanup=yes
EditorFont=8
New=SHIFT-F3
Open=F3
Close=F8
```

```
NewProcedureWindow=CTRL-F3
Save=F6
SaveAs=SHIFT-F6
Undo=CTRL-Z
Cut=CTRL-X
Copy=CTRL-C
Paste=CTRL-V
Find=CTRL-F
FindNext=F9
FindPrevious=SHIFT-F9
Replace=CTRL-R
GotoLine=CTRL-G
List=CTRL-L
NextBuffer=F7
PreviousBuffer=SHIFT-F7
Run=F1
CheckSyntax=SHIFT-F2
Debug=SHIFT-F4
CompilerMessages=CTRL-E
```

[RBStartup]

```
;RBSTARTUP=
```

[ReportBuilderSaveList]

```
ActionBar=1
FormatBar=1
FieldNames=0
SortFieldNames=0
ShowHiddenTables=0
PasteArgs=1
ShHzRul=1
ShVtRul=1
ShGrd=1
Maximize=1
NotFirstTime=1
Library=
Report=Teste
```

[ReportBuilderPreferences]

```
InstantRep=2
PromptFields=1
ScrBarHz=1
ScrBarVt=1
PromptConnection=0
```

[ReportBuilderDefaults]

```
DefMeas=1
PgLM=1
BotM=50
LeftM=50
RightM=50
TopM=50
RulPit=10
VRulPit=10
Pts=120
SnapTo=1
Color=0
WidthRep=1
WidthColor=8
FaceName=Arial
LogF=No
LogT=Yes
LibDir=
DatabaseDir=
ImgDir=
UDFDirectory=
MemExt=TXT
ImgExt=BMP
ImageQuality=0
```

[uib]**[Procomp]**

```
SaveSettings=yes
FileSpec01=.,*.p *.w
DefFileSpec=*.p *.w
LogFile=compile.log
ShowStatus=yes
RemoveOldRs=yes
IfNoR=no
SubDirs=yes
SaveNewRs=yes
XrefAppend=no
ListAppend=no
PageLength=60
PageWidth=80
V6Frame=yes
StreamIO=no
```

A listagem acima é a utilizada para o funcionamento do MAGNUS versão I em Windows.

Configurando o Progress para UNIX

Após a instalação por meio de fita ou CD, apenas edite a(s) profile(s) setando as variáveis de ambiente.

Exemplo de script

```
TERM=vt220 ; export TERM
DLC=/sistemas/dlc ; export DLC
PATH=/usr/bin:$DLC:$DLC/bin:/sistemas/magnus ; export PATH
PROCFG=$DLC/progress.cfg ; export PROCFG
PROMSGS=$DLC/promsgs ; export PROMSGS
PROPATH=$DLC,$DLC/bin,$DLC/tty,/sistemas/magnus
export PROPATH
$DLC/bin/mpro -pf magnus.pf
```

Variáveis de ambiente

| Variável | Definição |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| ACCESS | Especifica quem terá permissão no ProControl em ambiente Windows NT <u>Exemplo</u> ACCESS=Administrator |
| DLC | Informa a localização da instalação do Progress <u>Exemplo</u> SET DLC=C:\DLC (DOS/Windows) DLC=/app/dlc ; export DLC (UNIX) |
| DLCRES | Informa a localização do aplicativo Progress Results |

Exemplo

DLC=/usr/myres ; export DLC

EVTLEVEL

Especifica o nível de informações a serem gravadas no Event Log em Windows NT

Parâmetros

NONE - nenhuma informação;

BRIEF - apenas erros e alertas;

NORMAL - erros, alertas e os mesmos gravados no arquivo .lg;

FULL - todas mensagens

Exemplo

EVTLEVEL=BRIEF

PATH

Lista de diretórios para uso do Sistema Operacional

Exemplo

(UNIX)
PATH=\$PATH:\$DLC:\$DLC/bin:/magnus:/rhplus
export PATH

(DOS/Windows)
SET PATH=%PATH%;%DLC%;%DLC%/bin;M:\MAGNUS;M:\RHPLUS

PROAPSV

Localização do executável do Progress Application Server

Exemplo

PROAPSV=\$DLC/bin/_proapsv (UNIX)

PROAPSV=%\$DLC%\bin_proapsv.exe (Windows NT)

PROCFG Localização do arquivo de configuração do Progress

Exemplo

PROCFG=C:\DLC\PROGRESS.CFG

PROCONV Localizaçãdo arquivo de Códigos de Páginas a ser utilizado na sessão cliente

Exemplo

PROCONV=C:\DLC\CONVMAP.CP

PROEXE Localização do arquivo executável do Progress

Exemplo

SET PROEXE=c:\dlc\prowin32.exe

PROLOAD Localização do diretório de instalação do aplicativo ProBuild

Exemplo

SET PROLOAD=%DLC%\probuild

PROMSGS Localização do arquivo de mensagens Progress

Exemplo

PROMSGS=\$DLC/prolang/promsgs.por

(arquivo de mensagens em português)

PROPATH Lista de diretórios a ser utilizado pela sessão Progress

Exemplo

(DOS/Windows)
SET PROPATH=%PATH%;%DLC%;%DLC%\BIN;M:\MAGNUS

(UNIX)
PROPATH=/usr/bin:\$PATH:\$DLC:\$DLC/bin:/magnus
export PROPATH

PROSRV Localização do arquivo executável do servidor de dados

Exemplo

PROSVR=%DLC%\bin_mprosrv.exe
(DOS/Windows)

PROSTARTUP Define o arquivo de parâmetros (.pf) a ser usado como padrão pela sessão Progress, caso omitido o padrão assumido será o arquivo startup.pf dentro do diretório do Progress.

Exemplo

(DOS/Windows)
PROSTARTUP=M:\magnus\magnus.pf

(UNIX)
PROSTARTUP=/magnus/magnus.pf
export PROSTARTUP

PROTERMCAP Define o arquivo de configuração de terminal para sessão em ambiente character. Caso omitido será assumido como padrão o arquivo proterm dentro do diretório do Progress

Exemplo

```
PROTERMCAP=$DLC/protermcap ; export PROTERMCAP
```

TERM

Define o tipo de terminal de uma sessão cliente

Exemplo

```
(UNIX)  
TERM=vt200 ; export TERM
```

PROMON

Define o arquivo para gravação de estatísticas de programas



Lembre-se que em UNIX as variáveis são case-sensitive, isto é, há distinção entre maiúsculas e minúscula, como também para o valores das variáveis!

Arquivos

Arquivos comuns

| | |
|--------------|------------------------------------------|
| progress.cfg | Configuração do Progress |
| PROMSGS | Mensagens do Progress |
| .pf | Parâmetros sessão servidor/cliente |
| .cp | Definição de Códigos de Página |
| .df | Definição de estrutura de bancos/tabelas |
| .d | Dados de tabelas |
| .fd | Descrições do Bulk loader |
| .csh | Conteúdo ou cache do Metaschema |
| .rpt | Licença de uso de relatórios |
| .v | Validação |
| .pl | Biblioteca de procedures |

Arquivos temporários

| | |
|------|-----------------------------------|
| .lbi | Before-Image local de transações |
| .dbi | Definições de tabelas temporárias |
| .ped | Procedures temporárias |



Quando você edita uma procedure,

seja no Procedure Editor ou UIB as alterações são salvas em um arquivo temporário com extensão .PED no diretório temporário definido na sessão a cada execução da mesma. Então se houver algum problema e a sessão seja encerrada sem que antes você salve suas alterações, procure no diretório temporário os arquivos .PED, um deles contém a procedure até o momento da última execução.

| | |
|------|----------------------------------------------------------------------------------------------------|
| .srt | Organização de registros. Geralmente criado quando ordenado campos que não sejam indexados. |
| .trp | Definições do Dicionário de Dados temporárias, antes de serem confirmadas (Commit) no banco. |

Banco de Dados

| | |
|-----|-------------------------------------------------|
| .db | Banco de Dados |
| .dn | Arquivo estendido de banco multi-volumes |
| .bi | Before-Image |
| .bn | Arquivo estendido do Before-Image multi-volumes |
| .ai | After-Image |
| .na | Arquivo estendido do After-Image |

multi-volumes

| | |
|-----|--------------------------|
| .lg | Log de eventos do Banco |
| .lk | Lock de conexão do Banco |
| .tl | Log de transações |
| .ds | Descrição de estruturas |

Sessão cliente

| | |
|----|-------------------------|
| .p | Procedure |
| .w | Procedure UIB (Windows) |
| .i | Arquivo include |

Protocolos disponíveis

| | |
|------------------------------------|--------|
| NETBIOS | TCP/IP |
| SNA | DECnet |
| SPX | TLI |
| WIPC (Servidor/Cliente Windows) | DDE |
| CTOS | |

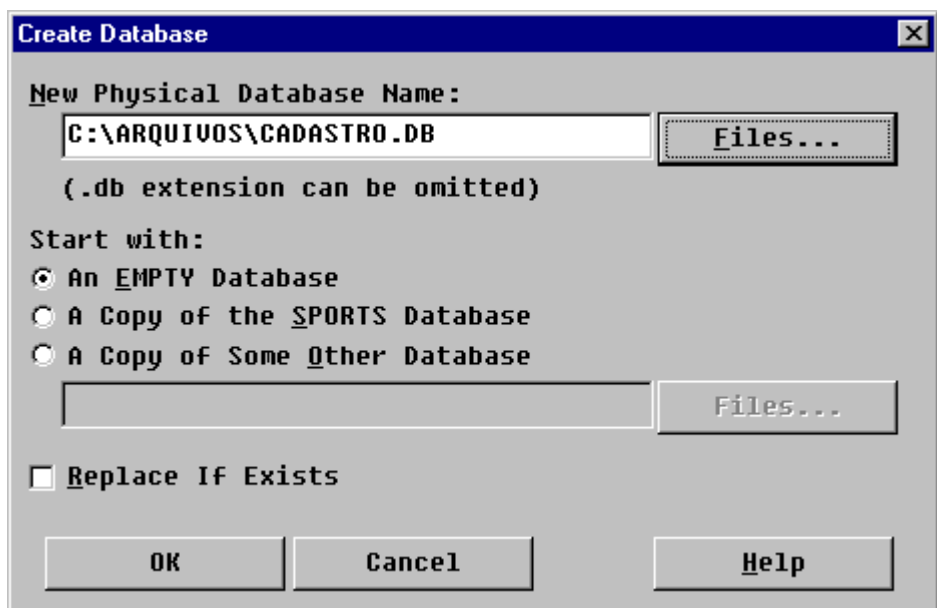
Capítulo 3

Criando & Definindo os Dados

Criando um banco de dados

Existem três maneiras de criar um banco de dados em Progress, como será mostrado abaixo:

- 1º No Data Administrator menu Database opção Create, então informe a localização e o nome do novo banco de dados a ser criado.



Você tem a opção criar um banco de dados vazio, uma cópia do sports (um banco de exemplo do Progress) ou a partir de um outra já existente.

- 2ª** Através de utilitário de banco de dados ProDB, em linha de comando.

Exemplo:

C:\DLC\bin\proddb.exe cadastro

Então será solicitado a partir de qual banco de dados será criado, com as mesma opções do Data Administration, conforme abaixo:

```
C:\ARQUIVOS>C:\DLC\BIN\PRODB cadastro
demo      to get the system demonstration database, or
sports   to get the sports demonstration database, or
isports  to get the international sports database, or
empty    to get the system empty database, or
anyname  to get a copy of that database.           :
```

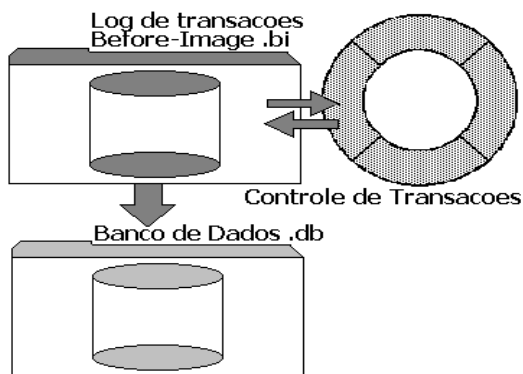
Lembre-se que a variável de ambiente DLC deve estar configurada adequadamente para pode criar o novo banco de dados.

- 3ª** Por código fonte no Procedure Editor, conforme o código abaixo:

```
create database "c:\arquivos\cadastro" replace no-error.
```

Como funciona o Banco de Dados

Em Progress o banco de dados é constituído de, pelo menos, dois arquivos básicos, o próprio Banco de Dados com extensão DB e o Controle de Transações com extensão BI (Before-Image).



Estes dois arquivos funcionam em conjunto para gerenciar todas as solicitações dos aplicativos a eles ligados.

O arquivo de "Image Anterior" (Before Image) tem um papel importantíssimo de assegurar a total e completa integridade física e referencial dos dados, como por exemplo índices, validações, tipos e formatos através do "ENGINE DATABASE" e confirmar estas transações para gravação e/ou deleção no banco de dados após uma completa consistência dos dados existentes.



Os métodos para criação de um banco de dados mostrados se aplicam apenas para Bancos de Volume Único ou Simples, existem técnicas em Progress mais sofisticadas para criação de bancos de dados em Múltiplos Volumes, conforme será mostrado no capítulo Performance.

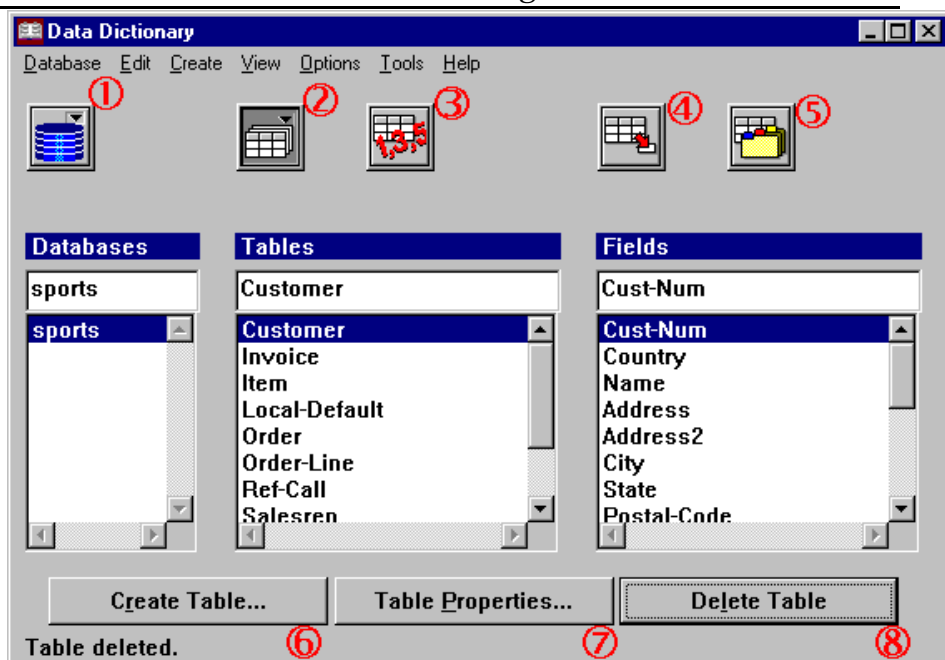
Criando tabelas, campos, índices, etc

Para modelagem de dados é utilizado o aplicativo Data Dictionary que pode ser executado de diversos modos, por meio de linha de comando ou menu.

Linha de comando: `DLC/bin/prowin32.exe -p _dict.p`

- ☒ Sempre que informado uma localização de aplicativo para o Progress esta será apresentada com barra invertida, como padrão do UNIX e também padrão Progress, ainda que em Windows.
- ☒ ProWin32 refere-se apenas ao Progress em ambiente gráfico(MS-Windows, OSF/Motif e outros).

No Procedure Editor escolha o menu Tools opção Data Dictionary. A figura abaixo descreve cada detalhe do utilitário.



- ① Habilita a manipulação dos bancos de dados conectados, conforme a listagem Databases;
- ② Habilita a manipulação de tabelas, conforme o banco de dados selecionado na listagem Databases;
- ③ Habilita a manipulação de **SEQUENCES** para o banco de dados selecionado. Será explicado o Sequence neste capítulo;
- ④ Habilita a edição de tabelas conforme o banco de dados selecionado;
- ⑤ Habilita a edição de índices conforme a tabela selecionada na listagem Tables;
- ⑥ ⑦ ⑧ Disponibiliza as opções conforme os itens acima escolhidos.

Criando uma tabela

A figura abaixo mostra a janela de propriedades da tabela no Data Dictionary.

Create Table

Table Name: clientes

Optional

Dump File: clientes ☐ Hidden ☐ Frozen

Table Type: PROGRESS

Label: ?

Description: Cadastro de Clientes

Replication: ?

DB Link: n/a Record Size:


DataServer Name: n/a

Owner: n/a

Triggers... Validation... String Attrs...

OK Create Cancel Help

As propriedades abaixo definem cada opção da figura anterior.

| Propriedade | Descrição |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Table Name | <p>Define o nome da tabela a ser criada.</p> <p>Este nome deve ser iniciado por uma letra de A a Z, com no máximo 32 caracteres e não conter nenhum caracter inválido, tipo asterisco, vírgula, parênteses, entre outros</p> <p> No Progress você pode inserir "-" (hifen) para nomear tabelas, campos, variáveis e outros.</p> |

Dump File Informa o nome do arquivo de DUMP para exportação das definições e dos dados da tabela.

Este nome poder ter até 8 caracteres alfanuméricos e também não deve apresentar sinais inválidos não reconhecidos para o nome de arquivo, de acordo com os princípios dos Sistemas Operacionais em geral, exemplo (*,?,%,&,etc).



Independente da plataforma o arquivo para DUMP será sempre tamanho 8 e por padrão o Progress ao exportar o arquivo definirá a extensão D(arquivo de dados = dump.d) e DF(arquivo de definições = dump.df). Isso para manter

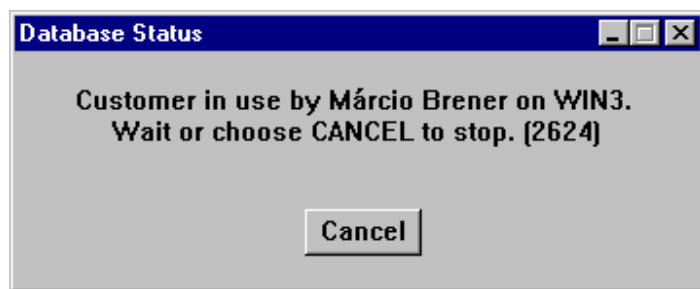
compatibilidade do banco de dados caso seja mudado para outro sistema operacional.

| | |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hidden | Ocultar a tabela da listagem. Esta opção não é disponível no momento de criação, mas pode ser selecionada após a confirmação de gravação da tabela no banco. |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|---------------|-------------------------------------------------------------------------------------------|
| Frozen | Você pode congelar as definições da tabela após a conclusão e gravação no banco de dados. |
|---------------|-------------------------------------------------------------------------------------------|

| | |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Label | Define um nome de até 30 caracteres quaisquer para exibição de informações de avisos ou erros para a tabela em tempo de execução do aplicativo, como por exemplo a informação de travamento de uma tabela por outro usuário, exemplo: |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Tabela customer tem o label Customer, o qual será mostrado no aviso, conforme a figura abaixo:



| | |
|--------------------|----------------------------------------------------------------------|
| Description | Texto descritivo para informações ou anotações a respeito da tabela. |
|--------------------|----------------------------------------------------------------------|

| | |
|--------------------|-----------------------------------------------------------------|
| Replication | Informa uma outra tabela destino que será espelho desta tabela. |
|--------------------|-----------------------------------------------------------------|

Criando os campos

A figura abaixo mostra as propriedades para a criação de um campo.

Field Properties

Field Name: Member of an Index: no
Member of a View: no

Data Type: integer

Optional

Format: Examples...

Label:

Column Label:

Initial Value:

Order #: Decimals: ?

Description:

Help Text:

☒ Mandatory ☐ Case-Sensitive ☐ Extent

Triggers... Validation... View-As... String Attrs... DataServer...

OK Save Cancel < > Help

| Propriedade | Descrição |
|-------------------|----------------------------------------------------------------------------------------------|
| Field Name | Nome do campo com até 32 caracteres alfanuméricos, sem acentos ou sinais inválidos. |
| Data Type | Define o tipo de dado do campo em: INTEGER, DECIMAL, CHARACTER, LOGICAL, DATE, RAW ou RECID. |



Após o campo ser criado o Tipo de Dado não poderá ser alterado, como também não poderá ser removido caso exista algum índice da tabela a que pertença.

Format Define um formato ou máscara de acordo com o tipo de dado para entrada e visualização do campo.

Exemplos:

->>>, >>9.99 (DECIMAL)

999 (INTEGER)

99/99/9999 (DATE)

Masculino/Feminino (LOGICAL yes/no)

Label Define um texto informativo a ser exibido com o campo, tanto para apresentação em coluna, lateral e também em informações como regras de índices e outros.


Column Label Define um texto informativo para apresentação quando em coluna.

Initial Value Atribui um valor inicial para o campo após o registro ser criado.

Order Define a ordem do campo dentro da tabela.

Por padrão esta ordem é incrementada de 10 em 10 e não pode repetir, mas pode ser alterada e conter qualquer intervalo de número inteiro.

Decimals Define o número de casas decimais num

| | |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | limite de 10 e mínimo de 0 para dados do tipo DECIMAL |
| Description | Texto descritivo ou informativo para o campo. |
| Help Text | Informa um texto de Ajuda ou Informação sobre o campo para ser exibido na área de status da aplicação quando este estiver sendo editado. |
| Mandatory | Define o preenchimento de um campo como obrigatório. |
| Case-Sensitive | Informa que um campo tipo CHARACTER fará distinção entre letras maiúsculas e minúsculas se utilizado em um índice ou para uma comparação. |
| Extent | Define o campo como um array de n elementos, nomendo-os como: CAMPO[n elemento]  Esta propriedade do campo não poderá ser alterada após o campo criado! |

Validando os Campos

Você pode validar os dados e eventos para um campo através da propriedade Validation ou por meio de Triggers.

A propriedade Validation

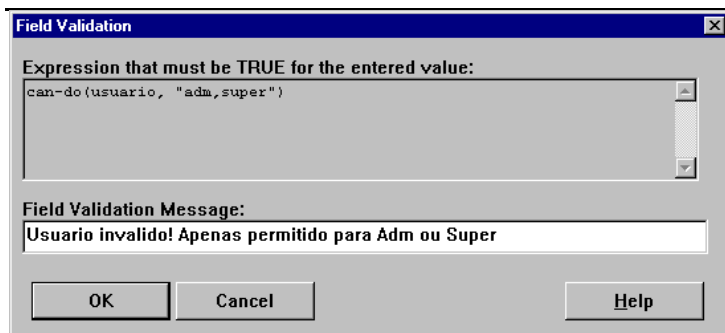
O meio mais fácil e rápido para validação é a propriedade Validation do campo.

Você especifica as cláusulas de validação para valores permitidos e uma mensagem de retorno a ser informada para o usuário caso as cláusulas não sejam satisfeitas, conforme a próxima figura demonstra.

As verificações são executadas durante a atualização do campo com o "feedback" de mensagens e help deste para o usuário, mostrados na barra de status na janela da aplicação.



Você pode, também, incluir uma Include na validação do campo. Exemplo: {inc/empresa.inc}, esta Include deve retornar um erro para a validação do campo, RETURN ERROR, caso as condições não forem satisfeitas!



Validação do Campo

A validação permite apenas a gravação se o usuário for Adm ou Super, caso contrário informa ao usuário "Usuario invalido! Apenas permitido para Adm ou Super".

Triggers

O que um Trigger?

Um Trigger, como na própria tradução da palavra, é um "gatilho" que é disparado para eventos da tabela ou campo como CREATE, UPDATE e DELETE, e a partir da versão 8 do Progress estes mesmos eventos para Replicação de dados.

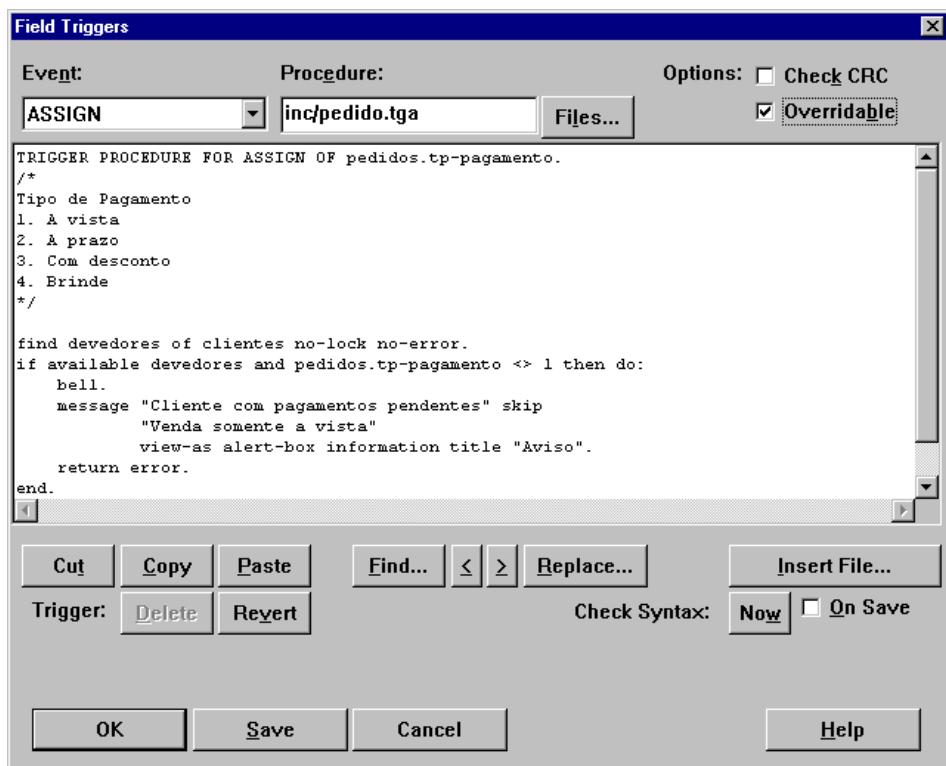
O Trigger é um mecanismo mais sofisticado e completo para validação de CAMPOS ou TABELAS. Ele é uma procedure em Progress, um arquivo a parte do banco de dados que é disparado para os eventos de CRIAÇÃO, DELEÇÃO de uma tabela ou ATUALIZAÇÃO de um campo.



Os Triggers em Progress são arquivos não armazenados no banco de dados. As informações armazenadas são apenas localização e verificação do conteúdo ao contrário de outros bancos como MS-SQL Server, DB/2 ou Oracle. Então sempre mantenha os Trigger bem armazenados e seguros!

A próxima tela exemplifica a sintaxe de um Trigger.

Propriedades do Trigger



| Evento/Propriedade | Descrição |
|--------------------|--------------------------------------------------------------------|
| Event | Define qual o evento executara o trigger: ASSIGN, DELETE ou UPDATE |
| Procedure | Informa o arquivo do Trigger |



Use sempre a localização definida pelo PROPATH com barra do tipo UNIX,

exemplo: inc/cliente.p

Check CRC

Verifica se o Trigger foi alterado depois de definido no banco de dados.

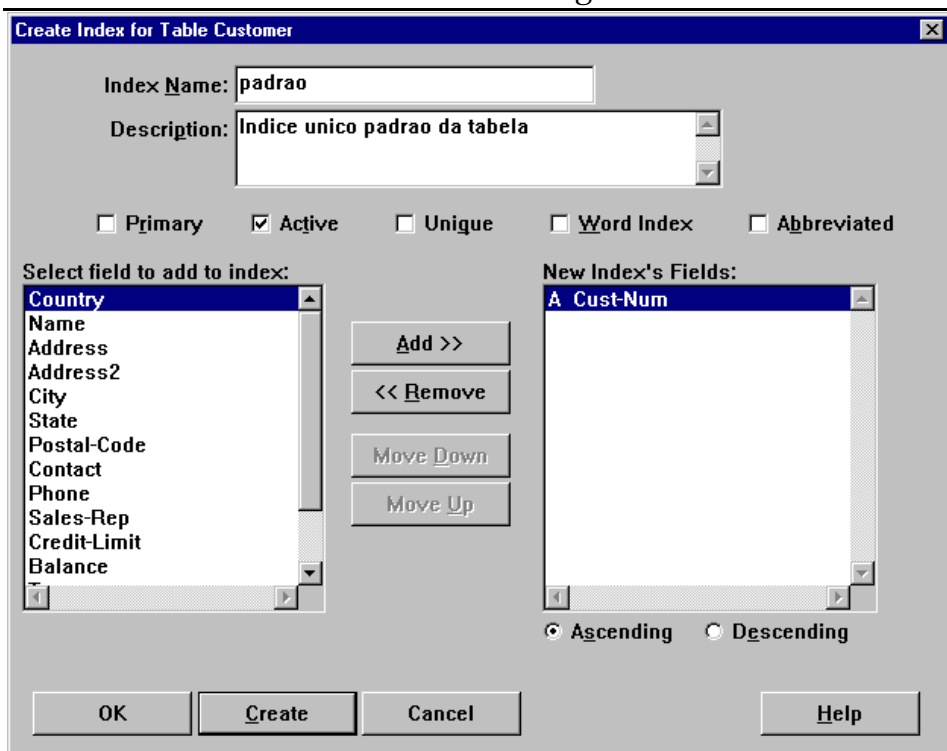
Overridable

Informe se o Trigger pode ser editado ou modificado mesmo durante utilização.





A ordem de execução dos procedimentos no banco de dados Progress é: Validação do Campo (Validation), Trigger do Campo e Trigger da Tabela do banco de dados, respectivamente nesta ordem.

Criando índices



A criação, bem como a manutenção de índices em Progress é totalmente simplificada. Ao contrário de outros bancos de dados como MS-SQL Server, por exemplo, você não precisa especificar diversos parâmetros e atributos como taxa de crescimento, organização física, e outros apenas atributos básicos.

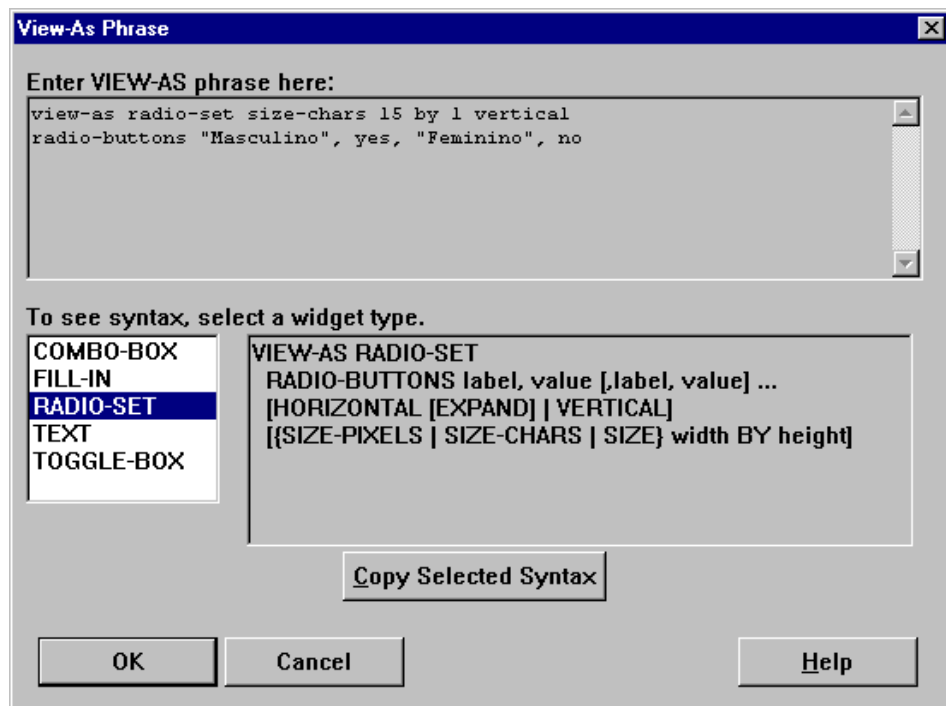
| Propriedade | Descrição |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Index Name | Nome do índice para identificação com até 32 caracteres alfanuméricos. Deve iniciar por uma letra, não deve conter letras inválidas como ç, ó, ê, como também caracteres reservados. |

| | |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Description | Texto descritivo ou informativo para o índice |
| Primary | <p>Define que o índice será o padrão utilizado para procuras ou ordenação da tabela.</p> <p>Pode haver apenas um índice primário para a tabela, e esse índice também não pode ser excluído.</p> <p>Para excluir um índice primário você deve definir qual outro índice será primário antes de efetuar a exclusão.</p> |
| Active | <p>Define o índice como Ativo ou Inativo.</p> <p>Caso o índice esteja inativo ele não será atualizado automaticamente, como também não poderá ser utilizado para pesquisa ou ordenação.</p> |
| Unique | <p>Informa que não será aceita duplicação de valores no índice.</p> <p> Caso você defina um campo tipo CHARACTER como case-sensitive e crie um índice único para este campo, ele aceitará valores duplicados, ainda que não idênticos, exemplo: Maria/maria, Joao/joao,etc</p> <p> Lembre-se que o índice único é sempre a Chave Primária (Primary Key) para o relacionamento entre</p> |

| | tabelas! |
|-------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Word Index | <p>Define um índice para campo do tipo CHARACTER organizado por palavras contidas em cada registro.</p> <p><input checked="" type="checkbox"/> Este índice consome bastante espaços em disco, como também onera bastante a performance do banco, sendo aconselhado apenas se realmente não houver uma outra solução.</p> |
| Abbreviated | <p>Cria um índice abreviado para campo tipo CHARACTER utilizando apenas algumas palavras ou caracteres iniciais do conteúdo do campo.</p> |
| Ascending ou Desceding | <p>Define a ordem para um campo dentro do índice.</p> <p>Como padrão inicial todos os campos são definidos como ascendentes.</p> |

Aparência do Campo (View-As)

Em Progress os campos são objetos intrínsecos, com características como tipo de visualização, ou em outras palavras tipo de objeto.



Você pode definir por exemplo que o campo lógico sexo seja uma TOGGLE-BOX para exibição e atualização, sem necessidade de nenhuma linha de código para tal.

Depois de definido o tipo de visualização para o campo faça o teste com o código abaixo:

Exemplo:

```
create cliente.  
update cliente.sexo.
```

Tipos disponíveis: COMBO-BOX, EDITOR, SELECTION-LIST, FILL-IN, TEXT e RADIO-SET

Sequences

O que são Sequences?

Um Sequence nada mais é que um contador que pode ser atribuído a um campo de uma tabela, igual a campo tipo IDENTITY SQL ANSI.

Como usar um Sequence?

Utilize um Sequence em uma Trigger CREATE utilizando uma as funções NEXT-VALUE ou CURRENT-VALUE.

Exemplo:

```
TRIGGER PROCEDURE FOR Create OF Customer.
```

```
/* Automaticamente define um novo código para o cliente*/
```

```
ASSIGN Customer.Cust-Num = NEXT-VALUE(Next-Cust-Num).
```

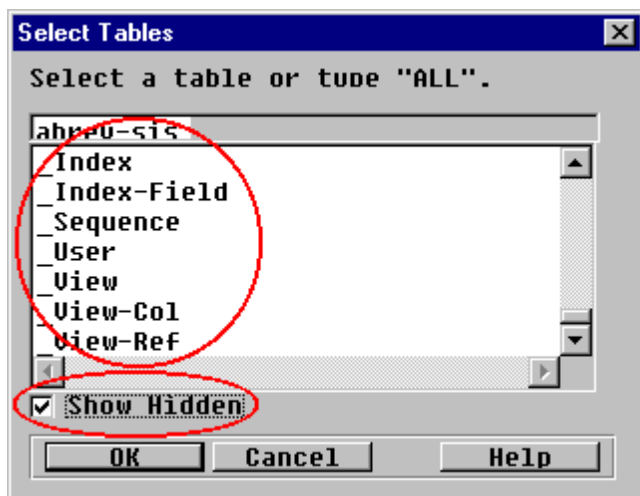


A modelagem dos dados é uma das principais etapas para o sucesso de qualquer aplicativo. Então faça seu modelo da melhor maneira possível afim de evitar problemas ou retrabalho posterior!

Metaschema

O que é o Metaschema?

Metaschema ou Metadados são tabelas de tabelas, isto é, são tabelas intrínsecas do próprio banco de dados que contêm as definições de tabelas, índices, campos, views, etc, como também para manipulação de dados do próprio funcionamento do banco de dados.



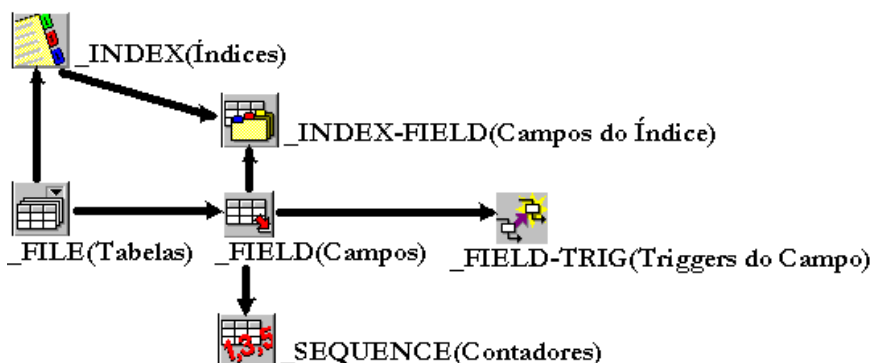
Estas tabelas estão ocultas, mas podem ser consultadas facilmente através do Data Administration, opção Reports, Detailed Tables e marcando a opção Show Hidden.

O funcionamento e relacionamento destas tabelas são facilmente manipulados para qualquer tipo de consulta ou alteração de algumas informações.

Tabelas do Metaschema

| | |
|---------------------|-------------------------------------------|
| _Db | Informações do Banco de Dados |
| _Field | Informações sobre campos das tabelas |
| _Field-Trig | Informações sobre triggers dos campos |
| _File | Informações sobre as tabelas do banco |
| _File-Trig | Informações sobre os triggers das tabelas |
| _Index | Informações sobre os índices das tabelas |
| _Index-Field | Informações sobre os campos dos índices |
| _Sequence | Informações sobre os sequences |
| _User | Informações sobre os usuários do banco |
| _View | Informações sobre Views |
| _View-Col | Informações sobre os campos das Views |
| _View-Ref | Informações sobre as tabelas das Views |

Fluxograma



Capítulo 4

Aprendendo a linguagem

Desenvolvendo o código

O primeiro programa

Como é tradição entre os programadores o nosso primeiro programa será o "HELLO WORLD!", escreva o texto de código mostrado abaixo, escolha o menu Compile e opção Run.

```
display "Hello World!".
```

Para salvar o programa – em linguagem Progress chamado procedure – escolha o menu File opção Save As, coloque em um diretório de trabalho e defina um nome com a extensão P(procedure) como padrão do Progress, isso salvará seu código.

Como fazer a interface com o usuário?

Em Progress a interface com usuário é bastante interativa, ele automaticamente lhe disponibiliza mensagens de tela, formatos, validações e visualização que já foram previamente estabelecidas no banco de dados, o que lhe pouca um grande trabalho como também torna mais fácil a padronização de um sistema.

Você pode fazer um programa de inserção apenas com uma linha de código, exemplo: insert customer (pronto esta criado um programa de inserção).

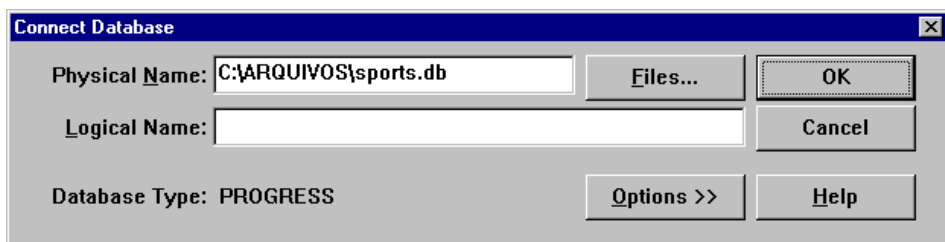
Conectando o banco de dados

Após você já ter criado o banco de dados a conexão é bastante fácil.

Você têm dois modos de conexão em um banco, Progress ou Não Progress: **Mono-Usuário** ou **Multi-Usuário**.

Iniciaremos uma conexão mono-usuário, que é mais simples e ideal para testes em geral.

No Data Administration escolha o menu Database opção Connect, então será apresentada a tela abaixo:



Especifique o banco de dados e clique em OK.

Caso não seja especificado o nome lógico do banco (Logical Name) será assumido o mesmo nome do banco físico.

Pode-se também conectar um banco por código no Procedure Editor, exemplo:

Sintaxe: CONNECT -db [Localização do banco] -1

```
connect -db c:\arquivos\sports.db -1
```

criando um programa para inclusão

O código abaixo exemplifica um programa bastante simples para inclusão de dados em uma tabela.

Exemplo

```
repeat:  
    insert customer with 1 columns.  
end.
```

Procedure Editor - Run

| | |
|---------------|-------|
| Cust-Num: | 1040 |
| Country: | USA |
| Name: | |
| Address: | |
| Address2: | |
| City: | |
| State: | |
| Postal-Code: | |
| Contact: | |
| Phone: | |
| Sales-Rep: | |
| Credit-Limit: | 1.500 |
| Balance: | 0.00 |
| Terms: | Net30 |
| Discount: | 0% |
| Comments: | |

Enter data or press ESC to end.

O código acima cria um loop habilitando a inserção, INSERT, de dados na tabela, o complemento do comando WITH SIDE-LABELS define que o texto dos campos estão alinhados ao lado esquerdo da área de atualização dos mesmo, utilizando também o formato, informação ou ajuda no rodapé da janela e validação já definidos.

Note que a aparência da tela é a devida à configuração com aspecto do Progress versão 6, parâmetros V6Frame, V6Colon, ...

Aprendendo a sintaxe

Definindo variáveis

SINTAXE

DEFINE

```
[[NEW [GLOBAL]] SHARED] VARIABLE variável  
{ {AS Tipo} | {LIKE Campo} } { [NO-UNDO]  
[BGCOLOR n] [COLUMN-LABEL Texto] [DCOLOR n]  
[DECIMALS n] [EXTENT n] [FONT n] [FGCOLOR n]  
[FORMAT Texto]  
[INITIAL constante | [constante[,constante]...]] } }  
[LABEL Texto[, Texto]...]  
[MOUSE-POINTER expressão]  
[[NOT] CASE-SENSITIVE] [PFCOLOR n]  
[VIEW-AS Tipo]} [TRIGGERS]
```

A definição de uma variável no Progress vai além da especificação do tipo de dado. Uma variável para o Progress é uma entidade como atributos e eventos, ou seja, um Objeto(**Widget**).

Na variável é possível configurar todos os atributos de um Objeto como por exemplo LABEL, COR, FORMATO, VISUALIZAÇÃO, EVENTOS, etc.

PARÂMETROS

NEW SHARED VARIABLE Variável

Define uma variável a ser compartilhada entre procedures, diretamente ou indiretamente a partir da declaração atual. A procedure chamada tem que nomear a mesma variável em informando que está já está previamente instanciada.

NEW GLOBAL VARIABLE Variável

Define uma variável a ser compartilhada entre todas as procedures previamente executadas

SHARED VARIABLE Variável

Define um variável que foi previamente criada por um procedure anterior com as cláusulas DEFINE NEW SHARED VARIABLE ou DEFINE NEW GLOBAL SHARED VARIABLE [Declarações].

VARIABLE Variável

Define uma variável cujo valor só está disponível dentro da procedure atual.

AS Tipo

Indica o Tipo de Variável a ser definida. Os tipos são CHARACTER, DATE, DECIMAL, HANDLE, INTEGER, LOGICAL, MEMPTR, RAW, RECID, ROWID e WIDGET-HANDLE.

LIKE Campo de Dado

Indica que a variável a ser definida herdará todas as características do campo original escolhido. Os atributos/valores herdados são FORMAT, LABEL, COLUMN-LABEL, INITIAL, DECIMALS, EXTENT e VIEW-AS Opção.

NO-UNDO

Quando um valor de uma variável é mudada durante uma transação e está é desfeita, o Progress restaura o seu valor anterior. Caso você a variável não seja definida com esta atribuição o valor não será reiniciado caso seja desfeita a transação.

Para melhor performance use esta opção apenas quando necessário, porque no NO-UNDO adiciona um novo elemento a ser manipulado na transação o que onera mais a aplicação.

BGCOLOR n

Especifica a cor de fundo da variável para um ambiente gráfico. Esta opção é ignorada em ambiente character.

[NOT] CASE-SENSITIVE

Indica que a variável faz distinção entre Maiúsculas e Minúsculas em uma comparação que a utilize. Por padrão a variável não distingue a caixa da letra.

Para trocar o atributo de sensibilidade à caixa do texto use NOT CASE-SENSITIVE.

COLUMN-LABEL Texto

Texto de identificação a ser exibido com a variável. Caso você queira usar mais de uma linha na exibição deste use uma exclamação (!) antes do texto a ser exibido na linha inferior.

```
define variable nome as character  
    column-label "Nome do!Cliente".  
update nome.
```

DCOLOR n

Especifica a cor dos caracteres da variável para um ambiente gráfico. Esta opção é ignorada em ambiente caracter.

DECIMALS n

Especifica o número de casas decimais de 0 a 10 para uma variável do tipo decimal. Como padrão o Progress atribui automaticamente 10 casas decimais se esta declaração for omitida.

EXTENT n

Declara uma variável com um Array e define o seu tamanho no limite de 28000 elementos. Para remover o atributo de Array de uma variável defina seu tamanho como 0.

FGCOLOR n

Especifica a cor de fundo da variável para um ambiente gráfico. Esta opção é ignorada em ambiente caracter.

FONT n

Especifica a fonte de apresentação da variável.

FORMAT Expressão

Especifica o formato para a apresentação e uso da variável.

INITIAL Expressão

Especifica o valor inicial de uma variável. Não pode ser atribuído o valor de uma função ou constante Progress.

LABEL Texto

Define o texto de exibição da variável. Use & para sublinhar a letra usada na tecla de aceleração utilizada pelo MS-Windows(ALT Letra). Se necessário utilizar um & literalmente no texto de exibição use && para inclui-lo.

MOUSE-POINTER Expressão

Especifica o ponteiro do mouse padrão usado na atualização ou exibição da variável.

PFCOLOR n

Especifica a cor da variável na atualização por declaração PROMPT-FOR. Esta opção é ignorada em interfaces gráficas.

VIEW-AS Expressão

Define o tipo de visualização ou objeto da variável.

TRIGGER

Define os eventos utilizados na atualização da variável.

```
define variable nome as character
  triggers:
    on return message 'Fim da atualizacao'
    view-as alert-box.
  end.
update nome.
```

Exemplo

```
/* ex0031.p */

define variable dia as integer format "9" initial 1
  triggers:
    on entry message 'Escolha as datas'.
  end.

define variable ano as integer format "9999" initial 2000.
define variable ndias as character format "x(12)"
  extent 7 initial ['Domingo','Segunda','Terca',
    'Quarta','Quinta','Sexta',
    'Sabado'].

define variable mes as character format "x(12)"
  view-as combo-box list-items
    'Janeiro', 'Fevereiro', 'Marco',
    'Abril', 'Maio', 'Junho',
    'Julho', 'Agosto', 'Setembro',
    'Outubro', 'Novembro', 'Dezembro'
  mouse-pointer 'glove' no-undo initial 'Novembro'.

update dia
  ano
  mes.
```

View-As

Define o tipo do objeto da variável mostrada na tela.

SINTAXE

VIEW-AS

```
{Combo-Box|Editor|  
{FILL-IN [NATIVE] [Tamanho]}|Radio-Set|  
Selection-List|Slider|  
{TEXT [Tamanho]}|  
{TOGGLE-BOX [Tamanho]}} [Tooltip Texto]
```

PARÂMETROS

COMBO-BOX

Especifica que um campo ou variável será apresentada como um Combo-Box. Pode conter valores tipo Caracter, Inteiro, Decimal, Lógico e Data.

EDITOR

Visualização tipo Editor de Texto para um campo ou variável. Suporta funções de Copia, Colar, Deletar, Quebra de Palavras, Auto-Identificação, Barras de Rolagem, Sub-Menu, etc.

FILL

Caixa de Texto para valores de qualquer tipo. É o tipo de visualização/objeto padrão para atualização de um campo ou variável no Progress.

RADIO-SET

Mostrado com uma série de opções para escolha de apenas uma. Pode receber valor tipo Caracter, Inteiro, Decimal, Lógico e Data.

SELECTION-LIST

Visualização como uma lista para seleção. Apenas aceita valores caracter e é apta a múltiplas seleções.

TEXT

Indica que um campo ou variável será apenas um texto exibido na tela, sem nenhuma manipulação ou atualização do conteúdo.

TOGGLE-BOX

Exibe com uma Caixa de Opção com valor lógico YES/NO. Aceita apenas valor lógico.

Exemplo

```
/* ex0032.p */

define variable texto as character
    view-as editor
    inner-lines 20
    inner-chars 60
    scrollbar-vertical
    scrollbar-horizontal
    column-label 'Editor de Texto'
    no-undo mouse-pointer "glove".

define variable meses as character format "x(12)"
    view-as combo-box
    list-items 'Janeiro', 'Fevereiro', 'Marco',
               'Abril', 'Maio', 'Junho', 'Julho',
               'Agosto', 'Setembro', 'Outubro',
               'Novembro', 'Dezembro'
```

```
initial 'Maio'.

define variable salvar as logical label 'Salva Informacao'
  view-as toggle-box initial yes format "Sim/Nao".

define variable dia as character
  view-as selection-list
  inner-lines 7 inner-chars 12
  list-items 'Domingo','Segunda','Terca','Quarta',
             'Quinta','Sexta','Sabado'
  initial 'Terca'.

define variable ano as integer
  view-as radio-set
  radio-buttons "1998", 1998, "1999", 1999.

form texto
  meses
  dia
  ano
  salvar
  with frame f-editor
    title "Edicao de Texto"
    row 2 centered overlay
    view-as dialog-box
    side-labels.

assign input frame f-editor texto:tooltip = 'Edite seu texto'
input frame f-editor meses:tooltip = 'Escolha um mes'
input frame f-editor dia:tooltip = 'Escolha um dia'
input frame f-editor ano:tooltip = 'Escolha um ano'
input frame f-editor salvar:tooltip = 'Salva informacao'.

update texto
  meses
  dia
  ano
  salvar
  with frame f-editor.

message "Voce escolheu..." skip(1)
"Dia:" dia:screen-value skip
"Mes:" meses:screen-value skip
"Ano:" ano skip
"Salvar:" salvar skip
"Observacao:" texto
view-as alert-box.
```

Definindo Frames

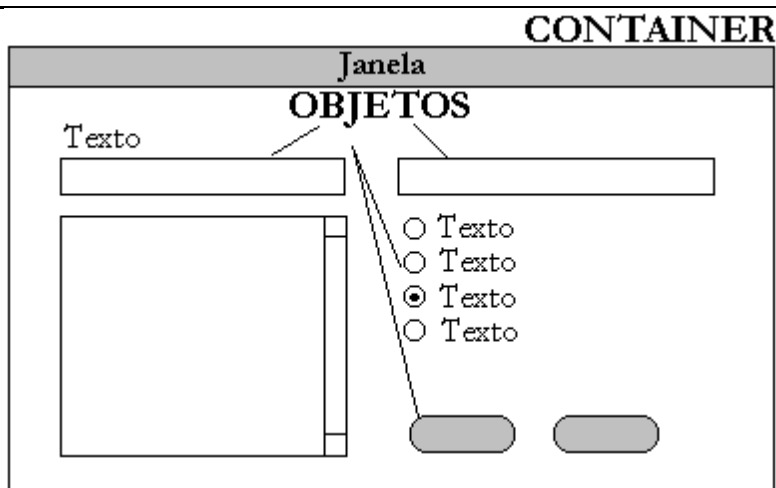
SINTAXE

```

DEFINE [[NEW] SHARED] FRAME Nome do Frame
    [Item1, Item2, Item n, ...]
    [{HEADER|BACKGROUND} Item1, Item2, Item n, ...]
    WITH [ACCUM [Tamanho Máximo] [AT Posição]
    [ATTR-SPACE|NO-ATTR-SPACE]
    [CANCEL-BUTTON Botão] [CENTERED]
    [COLOR] [COLUMN n] [n COLUMNS]
    [DEFAULT-BUTTON Botão]
    [[n] DOWN] [EXPORT] [FONT n]
    [FRAME Nome do Frame] [KEEP-TAB-ORDER] [NO-
    BOX]
    [NO-HIDE] [NO-LABELS] [USE-DICT-EXPS]
    [NO-VALIDATE] [NO-HELP] [NO-UNDERLINE]
    [OVERLAY]
    [PAGE-BOTTOM|PAGE-TOP] [RETAIN n] [ROW n]
    [SCREEN-IO|STREAM-IO] [SCROLL n] [SCROLLABLE]
    [SIDE-LABELS] [SIZE n BY n]
    [STREAM Stream] [THREE-D] [TITLE Texto] [TOP-
    ONLY]
    [USE-TEXT] [V6FRAME [USE-REVVIDEO|USE-
    UNDERLINE]]
    [VIEW-AS DIALOG-BOX] [WIDTH n] [IN WINDOW
    Janela]
  
```

Um Frame na linguagem Progress é um formulário ou simplesmente Form. Ele é um objeto tipo container, onde outros objetos estão acoplados ou sobrepostos, igual por exemplo uma janela do MS-Windows com diversas caixas de textos, botões, labels, e outros.

A janela acima equivale a um frame que possui diversos objetos em seu interior, inclusive outros frames.



PARÂMETROS

NEW SHARED FRAME Nome do Frame

Declara que o frame será compartilhado diretamente ou indiretamente pela procedure atual e outras executadas posteriormente. Para as novas procedures usarem a definição deve-se então declarar da mesma forma com **DEFINE SHARED FRAME Nome do Frame**.

SHARED FRAME Nome do Frame

Obtem a definição de um frame previamente criado por uma procedure anterior. Quando utilizado **DEFINE SHARED FRAME** você não deve acrescentar ou alterar qualquer característica previamente estipulada, caso você queira que o frame tenha as mesmas condições pré-definidas, porque as declarações atribuídas se tornarão parte do frame ao retornar à procedure anterior ou ao executar novas procedures.

ITEM

É o objeto a ser adicionado ao frame. Este Objeto(Widget) pode ser um desses tipos: FILL-IN, COMBO-BOX, SELECTION-LIST, SLIDER, FRAME, RECTANGLE, TOGGLE-BOX, RADIO-SET, EDITOR, OBJETO-COM ou TEXT, com as seguintes atribuições:

ATRIBUTOS DOS ITENS

AT ROW n COLUMN n [LEFT|COLON|RIGHT-ALIGED]

Define a linha e coluna aonde estará posicionado o objeto dentro do frame com as opções de alinhamento à Esquerda, pelo Texto informativo ou à Direita.

TO n

Especifica que o posicionamento do objeto será de uma posição n para trás.

BGCOLOR n

Cor de background do objeto em interface gráfica.

DCOLOR n

Cor do objeto em interface gráfica.

FGCOLOR n

Cor do primeiro plano do objeto em ambiente gráfico.

FONT n

Fonte do objeto em ambiente gráfico.

PFCOLOR

Cor de prompt-for para o objeto em ambiente gráfico.

VIEW-AS Tipo

Tipo de objeto para exibição/atualização na tela.

SPACE(n)

n espaços em branco.

SKIP(n)

n saltos de linha.

LABEL Texto

Texto informativo a ser exibido com o objeto.

COLUMN-LABEL Texto

Texto informativo a ser exibido com objeto mostrado em colunas.

TABELA/REGISTRO

Representa uma tabela ou um registro individual a ser acrescentado ao frame.

EXCEPT Campo n, Campo n

Não acrescenta n campos de uma tabela em um frame - os coloca como exceção.

HEADER [PAGE-TOP| PAGE-BOTTOM| BACKGROUND]

Define o frame como um cabeçalho, rodapé ou objeto de fundo.

PAGE-TOP/PAGE-BOTTOM

Apresenta o frame na parte superior/inferior da saída de impressão Tela, Arquivo ou Impressora.

BACKGROUND

Especifica que o frame será mostrado como objeto de fundo. Tipicamente usado para apresentação de imagens ou retângulos.

ACCUM

Acumula valores do frame usando as funções MAX, MIN, TOTAL e SUBTOTAL

ATTR-SPACE/NO-ATTR-SPACE

Reserva/Não Reserva espaço(s) para o elemento/texto dentro do frame

CANCEL-BUTTON

Nome do botão padrão de cancelamento dentro do frame

CENTERED

Centraliza o frame na tela

COLOR

Especificação da cor.

WHITE, BLUE, GRAY, YELLOW, RED, GREEN, etc

ESPECIFICAÇÃO FONTE E FUNDO**AMBIENTE GRÁFICO(GUI INTERFACE)**

BGCOLOR – background

FGCOLOR – foreground

AMBIENTE CHARACTER(TTY INTERFACE)

DCOLOR – background

PFCOLOR – foreground

ROW n

Linha inicial do frame

COLUMN n

Coluna inicial do frame

n COLUMNS

Número de colunas do frame

DEFAULT-BUTTON

Botão padrão do frame ao pressionar Return/Enter

[n] DOWN

Número de vezes que os elementos serão mostrados/rolados no frame

EXPORT

Esta expressão só é válida para instrução SQL

FONT n

Número da fonte a ser utilizada no frame, apenas para interfaces gráficas(Windows, Motif, OS/2, Linux, etc)

KEEP-TAB-ORDER

Mantém a ordem(zorder) dos elementos do frame ao uso do Enter ou TAB

NO-BOX

Desabilita a borda do frame. Esta opção não pode ser declarada com a especificação

TITLE Texto

Título do frame

NO-HIDE

Não apaga o frame para a próxima apresentação do mesmo

NO-LABELS

Desabilita os labels dos elementos contidos no frame

NO-UNDERLINE

Inibe o sublinhado sob os labels quando mostrados em coluna.

NO-VALIDATE

Desabilita as validações contidas nos elementos do frame.

OVERLAY

Mantêm todos os frames de fundo, este parâmetro deve ser usado sempre na declaração do frame

RETAIN n

Número de vezes para visualização/rolagem dos elementos dentro do frame.

SCREEN IO/STREAM IO

Define a saída do frame TELA ou STREAM

SCROLL n

Número de vezes que os elementos do frame serão exibidos dentro do mesmo.

SCROLLABLE

Define auto scroll para o frame, ou seja, cria automaticamente as barras de rolagem no frame

SIDE-LABELS

Mostra os labels dos elementos ao lado dos mesmos

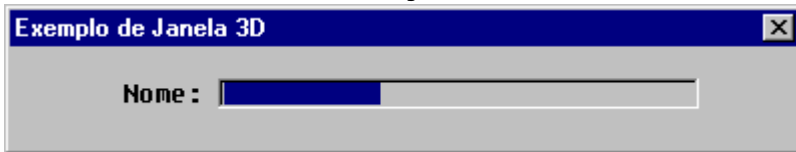
THREE-D

Mostra o frame em aparência 3D

Exemplo:

```
define variable c-nome as character format "x(30)" label "Nome".  
form c-nome at row 2 column 12 colon-aligned  
  with frame f-janela3d  
  title "Exemplo de Janela 3D"  
  three-d side-labels view-as dialog-box  
  size 50 by 5.
```

```
update c-nome  
  with frame f-janela3d.
```



TITLE Texto

Título do frame a ser exibido

V6FRAME/USE-REVVIDEO/USE-UNDERLINE

Tipo do frame. Aparência utilizada na versão Progress 6, Cor reversa ou Habilita sublinhado

VIEW-AS DIALOG-BOX

Mostra o frame como uma janela tipo pop-up

WIDTH n

Largura do frame no máximo de 640 caracteres

IN WINDOWN Janela

Mostra o frame em uma janela específica. A janela padrão é a janela atual(CURRENT-WINDOW)

Definindo Queries**O quê é uma Query?**

Query é uma pesquisa e/ou seleção de dados únicos, relacionados ou agrupados a serem utilizados em um Browse de registros.

DEFINE QUERY

Define uma query que pode ser aberta com (OPEN QUERY) ou (GET) e associado a um BROWSE.

SINTAXE

```
DEFINE [[NEW] SHARED] QUERY Nome da Query
      FOR Buffer [Campos][, Buffer [Campos]]...
      [CACHE n] [SCROLLING]
```

PARÂMETROS**NEW SHARED QUERY Nome da Query**

Define a query como compartilhada entre procedures

SHARED QUERY Nome da Query

Redefine uma query já definida em uma procedure anterior

QUERY Nome da Query

Atribui o nome da query

FOR Buffer [Campos] [, Buffer [Campos]] ...

Define o Buffer e/ou campo(s) da(s) tabela(s) a serem utilizadas pela query

Você pode, também, atribuir uma tabela e citar os campos com `FIELDS(Campos ...)` ou exceções com `EXCEPT(Campos ...)`

Exemplo

Usando FIELDS()

```
define query q-customer for customer fields(cust-num name).
```

Usando EXCEPT()

```
define query q-customer for customer except(balance).
```

CACHE n

Define o número de registros a serem colocados em buffer de memória quando os campos estiverem locados (NO-LOCK). Os registros em buffer propiciam melhor performance para a manipulação da query e diminui o acesso ao banco de dados, com menor tráfego de I/O para a rede.

Por padrão o Progress coloca 50 registros em buffer para queries como apenas uma tabela, e queries com duas ou mais tabelas contidas é definido 30 registros em buffer. Caso você não queira colocar registros em buffer define `CACHE 0`.

SCROLLING

Especifica que a query será apta a delocar entre registros em conjunto com a cláusula `REPOSITION`. Caso

você não especifique a query como SCROLLING apenas poderá ser utilizado os comando FIRST, NEXT, PREV, LAST com GET para movimentar entre os registros da query.

BROWSES

Browse é um quadro, como um planilha, onde você navega por dados, geralmente registros de uma tabela de uma banco de dados.

SINTAXE

```
DEFINE [[NEW] SHARED] BROWSE Nome do Browse
    QUERY Nome da Query
    [SHARE-LOCK|EXCLUSIVE-LOCK|NO-LOCK]
    [NO-WAIT]
    DISPLAY {Colunas | {Registros [EXCEPT Campo ...]}}
    [OPÇÕES do CAMPO] WITH n [TOOLTIP Texto]
    WITH [OPÇÕES]
```

PARÂMETROS

NEW SHARED BROWSE Nome do Browse

Define um browse compartilhado entre procedures

SHARED BROWSE Nome do Browse

Obtem uma definição já declarada em uma procedure anterior

BROWSE Nome do Browse

Define o nome do browse

QUERY Nome da Query

Define o nome da query a ser associado com o browse

SHARE-LOCK|EXCLUSIVE-LOCK|NO-LOCK

Especifica o estatus dos registros da query manipulados pelo browse

NO-WAIT

Informa que um registro locado não informará, e não aguardará liberação para atualização

DISPLAY {Colunas|{Registros [EXCEPT Campo...]} }

Informa os registros da query a serem exibidos pelo browse.

OPÇÕES DE DISPLAY

SINTAXE

Nome do Campo, Variável ou Expressão
[FORMAT Formato] [{LABEL Texto}|NO-LABELS]
[COLUMN-LABEL Texto] [COLUMN-BGCOLOR Cor]
[COLUMN-DCOLOR Cor] [COLUMN-FGCOLOR Cor]
[COLUMN-FONT Fonte] [COLUMN-PFCOLOR Cor]

PARÂMETROS

FORMAT Nome do Campo

Determina o formato de visualização do campo, variável ou expressão no browse

LABEL Texto | COLUMN-LABEL Texto

Determina o label da coluna no browse

NO-LABELS

Informa que o campo não terá label no browse

COLUMN-BGCOLOR Cor

Determina a cor de background da coluna no browse

COLUMN-DCOLOR Cor

Determina a cor de background da coluna no browse em ambiente caracter (tty)

COLUMN-FGCOLOR Cor

Determina a cor de foreground da coluna no browse

COLUMN-FONT Fonte

Determina a fonte de exibição da coluna no browse

COLUMN-PFCOLOR Cor

Determina a cor de foreground da coluna no browse em ambiente caracter (tty)

☒ Algumas convenções do Progress são baseados no UNIX, devido esta ser a primeira plataforma da linguagem. TTY significa "ambiente caracter e/ou ANSI", e GUI "gráfico".

☒ Exemplo: if session:display-type = "gui ou tty" then...

☒ Como também outras definições como a barra de diretórios utilizada pelo UNIX e no Progress também do mesmo modo "/" (barra simples).



Lembre-se que a fonte é um número contido na configuração do Progress

HABILITANDO A COLUNA PARA ATUALIZAÇÃO

SINTAXE

ENABLE Campo

[HELP Texto] [VALIDATE(Condição, Texto)]

PARÂMETROS

PARÂMETROS

ENABLE Campo

Habilita a coluna para atualização

HELP Texto

Atribui a mensagem de ajuda a ser exibida na barra de status quando da atualização do campo.

VALIDATE(Condição, Texto)

Determina a validação do campo e o texto informativo desta validação

WIDTH n

Especifica a largura da coluna do browse

TOOLTIP Texto (MS-Windows)

Cria um texto de ajuda e/ou informativo quando o mouse apontado para o browse

OPÇÕES DO BROWSE

SINTAXE

```
WITH {[n] DOWN [WIDTH n] | [SIZE x BY y]}  
    [FGCOLOR Cor] [BGCOLOR Cor]  
    [DCOLOR Cor] [PFCOLOR Cor]  
    [LABEL-FONT Fonte] [LABEL-DCOLOR Cor]  
    [LABEL-FGCOLOR Cor] [LABEL-BGCOLOR Cor]  
    [MULTIPLE|SINGLE]  
    [SEPARATORS|NO-SEPARATORS]  
    [NO-ASSIGN]  
    [NO-ROW-MARKERS] [NO-LABELS] [NO-BOX]  
    [FONT Fonte] [TITLE Texto]
```

PARÂMETROS

n DOWN

Determina o número de linhas a exibir no browse

WIDTH n

Determina a largura do browse

SIZE x by y

Determina as dimensões do browse

MULTIPLE|SINGLE

MULTIPLE – Habilita seleção de múltiplos registros no browse. Caso exista alguma coluna habilitada para atualização (ENABLE Coluna) não será possível fazer seleção múltipla, mesmo se declarado MULTIPLE no browse.

SINGLE – Desabilita seleção de múltiplos registros no browse

SEPARATORS|NO-SEPARATORS

Inclui/Exclui separadores entre os registros exibidos no browse

NO-ROW-MARKERS

Não mostra marcadores laterais do browse quando algum campo estiver habilitado para atualização

NO-LABELS

Não mostra os labels das colunas do browse

NO-BOX

Remove o contorno do browse

FONT Fonte

Define a fonte do browse

TITLE [BGCOLOR Cor] [DCOLOR Cor] [FGCOLOR Cor] [FONT Fonte] Texto

Define o título de exibição browse

Exemplo

```
/* ex0036 */

define query q-customer for customer scrolling.
define browse b-customer query q-customer
  display customer.cust-num
    customer.name column-bgcolumn 15 column-fgcolor 2
    customer.address
    enable name address
    with 15 down separators
    title 'Customers'.

open query q-customer for each customer share-lock.

update b-customer.
```

Utilizando a Query no Browse

OPEN QUERY

SINTAXE

```
OPEN QUERY Nome da Query {FOR|PRESELECT}  
      EACH Registro [, {EACH|FIRST} Registro]  
      [[BY Registro ou Expressão DESCENDING]]  
      [INDEXED-REPOSITION]  
      [MAX-ROWS n Colunas]
```

PARÂMETROS

OPEN QUERY Nome da Query

Informa qual query será aberta

FOR | PRESELECT EACH Registro

Determina o escopo da query, ou seja, a seleção dos registros a serem utilizados pela mesma

EACH | FIRST

Determina a abrangência dos registros a serem abertos ou FIRST para apenas os primeiros registros de uma seleção.

BY Registro ou Expressão [DESCENDING]

Informa a query qual o registro de ordem ou uma expressão de determinará a ordem da query

INDEXED-REPOSITION

Otimiza outros processos de manuseio da query como REPOSITION TO ROWID

MAX-ROWS n Colunas

Limita a quantidade de registros retornados por uma query.

Esta opção é válida somente quando a query tiver a declaração SCROLLING explícita.

Exemplo:

```
/* ex0038 */

define variable cidade like customer.city no-undo.

define query q-customer for customer scrolling.
define browse b-customer query q-customer
    display customer.cust-num
           customer.name
           customer.address
           customer.city
           with 10 down scrollable title "Pressione F5
para filtro".

open query q-customer for each customer where
                                customer.city begins
'Boston' no-lock.

form b-customer with frame f-customer row 3 centered.

on "get":u, "F5":u of b-customer in frame f-customer do:
    message "Informe a cidade" update cidade.
    open query q-customer for each customer where
                                customer.city matches '*'
+ cidade + '*' no-lock.

end.

update b-customer with frame f-customer.
```


ASSIGN

Atualiza os dados para frames, variáveis ou registros.

SINTAXE

ASSIGN {[FRAME Nome do Frame] | [BROWSE] Registro
[= Expressão][WHEN Condição]}.[NO-ERROR]

ASSIGN {Registro [EXCEPT Registro...]} [NO-ERROR]

PARÂMETROS

**ASSIGN [FRAME Nome do Frame] [BROWSE]
Registro**

Atualiza os valores em buffer para campos e variáveis num frame ou browse

WHEN Expressão

Determina uma condição para a atualização pelo comando ASSIGN

NO-ERROR

Não causa e/ou mostra nenhum erro quando ocorrido no ASSIGN

Registro EXCEPT

Informa os registro que não serão afetado pelo comando ASSIGN

Comandos de Impressão e Manipulação de Arquivos

OUTPUT

Habilita a saída de dados para impressão, arquivo ou área de transferência quando em ambiente MS-Windows

SINTAXE

```
OUTPUT [STREAM Nome da Stream] TO
    {PRINTER|Arquivo|Dispositivo|
    TERMINAL|VALUE(expression)|
    "CLIPBOARD"}
[APPEND] [ECHO|NO-ECHO] [KEEP-MESSAGES]
[NO-MAP|{MAP Entrada do ProtermCap}]
[PAGED]
[PAGE-SIZE {Constante|VALUE(Expressão)}]
[UNBUFFERED]
[NO-CONVERT|
{CONVERT [TARGET Código de Página]
[SOURCE Código de Página]}]
```

PARÂMETROS

OUTPUT [STREAM Nome da Stream] TO

Habilita a saída de dados para impressão em impressora, arquivo ou área de transferência do MS-Windows(Clipboard), com a escolha de saída em uma Stream (Canal de Input/Output no Progress).

PRINTER

Ecoa a saída de dados para a impressora.

Quando em MS-Windows a impressora de saída será a impressora padrão pré-definida.

Em MS-DOS o padrão será o mesmo que a saída PRN

No UNIX será utilizado o spooler do comando (lp ou lpr).

Arquivo

Informa ao comando OUTPUT para saída em um arquivo. Exemplo: OUTPUT TO /tmp/cadastro.lst UNBUFFERED.

Dispositivo

Especifica o dispositivo de saída para a impressão. Exemplo: OUTPUT TO COM2 UNBUFFERED, OUTPUT TO \\SERVIDOR1\HP

Terminal

Define a saída de impressão para a tela

Value

Informa um valor de saída para a impressão. Exemplo: OUTPUT TO VALUE(SEARCH(Arquivo)) UNBUFFERED APPEND.

"Clipboard"

Ecoa os dados de saída para a Área de Transferência dos Windows

APPEND

Continua a saída de impressão ao fim de um arquivo

ECHO/NO-ECHO

Suprime o eco dos dados

KEEP-MESSAGES

Desabilita mensagens durante a saída de impressão, por exemplo erros ocorridos durante o processo

PAGED

Informa que a saída será paginada

PAGE-SIZE

Informa o número de linhas para cada página na saída de impressão

UNBUFFERED

Não coloca os dados de saída em buffer, cada caracter é enviado para a saída destino imediatamente, como em um arquivo.

NO-CONVERT

Não faz nenhum tipo de conversão quanto aos Códigos de Página definidos

CONVERT TARGET Código de Página SOURCE Código de Página

Converte os caracteres de saída de acordo com os Códigos de Página Origem e Códigos de Página Destino.

Exemplo:

```
/* ex0039.p */
```

```
output to d:\temp\teste.txt paged page-size 62 unbuffered.
```

```
form header
```

```
fill("-", 130) format "x(130)" skip
today format "99/99/9999"
"Relatorio de Clientes" at 30
"Pagina:" at 120 page-number format "999" to 130 skip
fill("-", 130) format "x(130)"
with frame f-cabecalho no-box width 130 page-top.
```

```
for each customer no-lock transaction:
```

```
    view frame f-cabecalho.
```

```
    display customer.cust-num
           customer.name
           customer.address
           customer.city
           customer.phone
           customer.sales-rep
           with column 15 width 130.
```

```
end.
```

```
output close.
```

INPUT

Habilita a entrada de dados a partir de um arquivo

```
PUT [STREAM stream] [UNFORMATTED]
  [{expression [FORMAT string] [{AT|TO} expression]} |
  {SKIP[(expression)]} | {SPACE[(expression)]} ...
```

```
-----
PUT [STREAM stream] CONTROL expression ...
-----
```

SEEK

Retorna um ponteiro indicando o fim do arquivo ou a posição de um arquivo aberto.

SEEK (INPUT|OUTPUT)

SEEK(INPUT)

Retorna a posição no arquivo aberto

SEEK(OUTPUT)

Retorna a posição do fim do arquivo

Exemplo:

```
define variable posicao as integer no-undo.
input from value(search('clientes.txt')) unbuffered.
assign posicao = seek(input).
```

```
input close.
```

```
display posicao.
```

SEEK

Posiciona um ponteiro para escrita ou leitura em uma posicao do arquivo aberto.

SEEK {INPUT|OUTPUT|{STREAM Nome da Stream}} TO {n|END}

SEEK INPUT

Posiciona em um ponto do arquivo aberto para leitura

SEEK OUTPUT

Posiciona em um ponto do arquivo aberto para escrita

SEEK STREAM Nome da Stream

Informa ao comando SEEK a Stream a utilizar

TO n|END

Indica uma posição ou o fim do arquivo

Exemplo:

define variable posicao as integer.

define variable texto as character.

output to teste.out append no-echo.

assign posicao = seek(output).

put unformatted 'teste de programa'.

output close.

input from teste.out unbuffered no-echo.

seek input to posicao.

set texto.

display texto format 'x(30)'.

input close.

Eventos

O Progress 4GL é uma linguagem orientada a objetos e eventos. Apesar de uma orientação a objetos não completa como em Delphi, C++ e/ou Java, mas a princípio baseada, como exemplo, também, Visual Basic.

Lista de Eventos

ANY-KEY

Evento genérico ou de abrangência total, atingido por qualquer tecla, combinação de teclas ou evento do mouse.

Aplica-se para:

| | |
|-----------|----------------|
| BROWSE | RADIO-SET |
| BUTTON | SELECTION-LIST |
| COMBO-BOX | SLIDER |
| EDITOR | TOGGLE-BOX |
| FILL-IN | |

Exemplo:

```
define variable nome as character.  
  
on "any-key":u anywhere do:  
    hide message.  
    message string(time, "hh:mm:ss").  
  
end.  
  
update nome.
```

ANY-PRINTABLE

Ocorre quando solicitada uma impressão de um objeto e não há nenhum trigger definido para o evento específico do objeto

Aplica-se para:

| | |
|-----------|----------------|
| BROWSE | RADIO-SET |
| BUTTON | SELECTION-LIST |
| COMBO-BOX | SLIDER |
| EDITOR | TOGGLE-BOX |
| FILL-IN | |

BACKSPACE

Quando pressionada um tecla de backspace no objeto

Aplica-se para:

FILL-IN

BACK-TAB

Ocorre quando solicitado um retrocesso e/ou passagem para o objeto anterior, em Windows como padrão Shift + Tab.

Aplica-se para:

| | |
|-----------|----------------|
| BROWSE | RADIO-SET |
| BUTTON | SELECTION-LIST |
| COMBO-BOX | SLIDER |
| EDITOR | TOGGLE-BOX |
| FILL-IN | |

BELL

Ocorre quando pressionada uma tecla mapeada para a função BELL no objeto

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | IMAGE |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | TEXT |
| FILL-IN | TOGGLE-BOX |
| FRAME | WINDOW |

CHOOSE

Aplica-se a botões e menus quando escolhido

Aplica-se para:

BUTTON MENU-ITEM (não toggle-box)

Exemplo:

```
define button b-ok label '&OK' auto-go.  
  
form b-ok with frame f-escolha.  
  
on "choose":u of b-ok in frame f-escolha  
    message 'Botao Ok pressionado' view-as alert-box.  
  
update b-ok with frame f-escolha.
```

CLEAR

Aplica-se a uma caixa de texto, FILL-IN, quando pressionada uma tecla, ou combinação de teclas para apagar o conteúdo, em Windows o padrão é Control Z.

Aplica-se para:

FILL-IN

DEFAULT-ACTION

Aplica-se a browses e listas de seleção quando ocorre duplo clique ou ENTER no objeto.

Aplica-se para:

BROWSE

SELECTION-LIST

DELETE-CHARACTER

Ocorre quando pressionado Delete no objeto

Aplica-se para:

FILL-IN

DESELECTION

Quando valores são desmarcados em um browse ou lista de seleção múltipla

Aplica-se para:

BROWSE

RADIO-SET

BUTTON

RECTANGLE

COMBO-BOX

SELECTION-LIST

EDITOR

SLIDER

FILL-IN

TEXT

FRAME

TOGGLE-BOX

IMAGE

EMPTY-SELECTION

Quando um objeto de conteúdo vazio é clicado

Aplica-se para:

FRAME

DIALOG-BOX

END-BOX-SELECTION

Ocorre quando o botão do mouse é liberado após a seleção de uma área.

Aplica-se para:

FRAME

DIALOG-BOX

END-ERROR

Aplica-se a objetos ou bloco de interação quando ocorrido algum tipo de erro ou solicitada a finalização, geralmente ESC ou F4.

Quando isto acontece a um bloco de interação, os valores das variáveis sem a declaração NO-UNDO e/ou valores atribuídos em banco de dados são desfeitos, caso estes valores ainda não estejam confirmados na tabela.

END-ERROR EVENT

Ocorre quando o usuário pressiona uma tecla que equivale à função END-ERROR.

Applies To:

BROWSE

IMAGE

BUTTON

RADIO-SET

COMBO-BOX

RECTANGLE

| | |
|-------------|----------------|
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | TEXT |
| FILL-IN | TOGGLE-BOX |
| FRAME | WINDOW |

END-MOVE

Ocorre ao finalizar uma ação de arrastar e soltar um objeto.

Aplica-se para:

| | |
|-----------|----------------|
| BROWSE | RADIO-SET |
| BUTTON | RECTANGLE |
| COMBO-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | |

END-RESIZE

Ocorre ao finalizar um redimensionamento da janela.

Aplica-se para:

| | |
|-----------|----------------|
| BROWSE | RADIO-SET |
| BUTTON | RECTANGLE |
| COMBO-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | |

ENDKEY

Quando pressionada uma tecla de finalização de tarefa, por padrão F4 e/ou ESC.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | IMAGE |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | TEXT |
| FILL-IN | TOGGLE-BOX |
| FRAME | WINDOW |

ENTRY

Quando o objeto recebe foco.

Aplica-se para:

| | |
|------------|----------------|
| BROWSE | FRAME |
| BUTTON | RADIO-SET |
| COMBO-BOX | SELECTION-LIST |
| DIALOG-BOX | SLIDER |
| EDITOR | TOGGLE-BOX |
| FILL-IN | WINDOW |

ERROR

Evocado quando qualquer erro não especificado é causado, tanto para objeto, como para blocos em transação.

Aplica-se para:

| | |
|--------|-----------|
| BROWSE | IMAGE |
| BUTTON | RADIO-SET |

| | |
|-------------|----------------|
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | TEXT |
| FILL-IN | TOGGLE-BOX |
| FRAME | WINDOW |

Exemplo:

```
for each _user on error undo, retry transaction:
    update _user.
end.
```

GO

Este evento é uma confirmação simultânea de um ou vários objetos em estado de inserção, funcionando como ENTER contínuo. O exemplo abaixo demonstra uma combinação de vários eventos reunidos para um mesmo propósito.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | IMAGE |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | TEXT |
| FILL-IN | TOGGLE-BOX |
| FRAME | WINDOW |

Exemplo:

```
form customer.cust-num
    customer.name
    with frame f-customer side-labels
    row 3 centered title "Alteracao".

on "return":u, "go":u, "tab":u, "entry":u, "leave":u of
```

```
customer.cust-num do:
    find customer where
        customer.cust-num =
            input frame f-customer customer.cust-num
        no-lock no-error.
    if available customer then display
        customer.name with frame f-customer.
    else clear frame f-customer no-pause.
end.

repeat on error undo, retry transaction:
    prompt-for customer.cust-num with frame f-customer.
    update customer.name with frame f-customer.
end.
```

HELP

Quando solicitada a tecla configurada para Help, como padrão a tecla em Windows é F1, mas caso o Progress esteja configurado para V6 a tecla padrão será F2, tanto em Windows como em UNIX, DOS, Novell, etc.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | IMAGE |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | TEXT |
| FILL-IN | TOGGLE-BOX |
| FRAME | WINDOW |

ITERATION-CHANGED

Ocorre quando qualquer interação/ação com o mouse é feita em um browse.

Aplica-se para:

BROWSE

LEAVE

Ocorre quando o objeto perde o foco.

Aplica-se para:

| | |
|------------|----------------|
| BROWSE | FRAME |
| BUTTON | RADIO-SET |
| COMBO-BOX | SELECTION-LIST |
| DIALOG-BOX | SLIDER |
| EDITOR | TOGGLE-BOX |
| FILL-IN | WINDOW |

LEFT-MOUSE-CLICK

Quando o botão esquerdo do mouse é pressionado sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

LEFT-MOUSE-DBLCLICK

Quando o botão esquerdo do mouse é pressionado duplamente sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

LEFT-MOUSE-DOWN

Quando o botão esquerdo do mouse é mantido pressionado sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

LEFT-MOUSE-UP

Quando o botão esquerdo do mouse é liberado do objeto outrora escolhido.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MENU-DROP

É iniciado quando com o mouse e feita uma ação de drag and drop, arrastar e soltar.

Aplica-se para:

| | |
|------|----------|
| MENU | SUB-MENU |
|------|----------|

MIDDLE-MOUSE-CLICK

Quando é pressionado o botão do meio do mouse, caso este esteja habilitado pelo Windows.

Aplica-se para:

| | |
|------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |

| | |
|-------------|------------|
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MIDDLE-MOUSE-DBLCLICK

Quando ocorre duplo-clique com o botão do meio do mouse sobre o objeto escolhido.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MIDDLE-MOUSE-DOWN

Quando pressionado o botão do meio do mouse sobre o objeto.

Aplica-se para:

| | |
|------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |

| | |
|-------------|------------|
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MIDDLE-MOUSE-UP

Quando liberado o botão do meio do mouse sobre o objeto outrora escolhido.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-EXTEND-CLICK

Quando clicado e liberado o botão do mouse sobre o objeto. O mesmo que MOUSE-MOVE-CLICK.

Aplica-se para:

| | |
|------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |

| | |
|-------------|------------|
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-EXTEND-DBLCLICK

Quando clicado duplamente o botão do mouse sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-EXTEND-DOWN

Quando mantido pressionado o botão do mouse sobre o objeto.

Aplica-se para:

| | |
|------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |

| | |
|-------------|------------|
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-EXTEND-UP

Quando liberado o botão do mouse sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-MENU-CLICK

Quando escolhido um item de menu.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |

| | |
|---------|------------|
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-MENU-DBLCLICK

Quando escolhido com duplo clique um item de menu.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-MENU-DOWN

Quando mantido pressionado o botão do mouse sobre o item de menu.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |

| | |
|-------|------------|
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-MENU-UP

Quando liberado o botão do mouse sobre um item de menu.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-MOVE-CLICK

Quando escolhido um objeto com um clique do mouse. O mesmo que MOUSE-EXTENT-CLICK.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |

IMAGE
MENU

WINDOW

MOUSE-MOVE-DBLCLICK

Quando clicado duplamente sobre um objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-MOVE-DOWN

Quando mantido pressionado o botão do mouse sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |

| | |
|-------|--------|
| IMAGE | WINDOW |
| MENU | |

MOUSE-MOVE-UP

Quando liberado o botão do mouse sobre o objeto outrora escolhido.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-SELECT-CLICK

Ocorre quando se clica e libera o botão do mouse.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |

MENU

MOUSE-SELECT-DBLCLICK

Ocorre quando se clica duplamente o mouse sobre um objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

MOUSE-SELECT-DOWN

Ocorre quando se seleciona um objeto com o botão do mouse pressionado.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |

MENU

MOUSE-SELECT-UP

Ocorre quando se seleciona um objeto com o botão do mouse e este é liberado.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

NEXT-FRAME

Quando pressionado uma tecla ocorrendo a passagem para outro frame.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | FILL-IN |
| BROWSE CELL | RADIO-SET |
| BUTTON | SELECTION-LIST |
| COMBO-BOX | SLIDER |
| EDITOR | TOGGLE-BOX |

OFF-END

Quando solicitado o último elemento de uma lista em um combo-box, exemplo: utilizando as teclas Control PageDown.

Aplica-se para:

BROWSE

OFF-HOME

Quando solicitado o primeiro elemento de uma lista em um combo-box, exemplo: utilizando as teclas Control PageUp.

Aplica-se para:

BROWSE

PREV-FRAME

Quando solicitado uma tecla para voltar ao frame anterior.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | FILL-IN |
| BROWSE CELL | RADIO-SET |
| BUTTON | SELECTION-LIST |
| COMBO-BOX | SLIDER |
| EDITOR | TOGGLE-BOX |

RECALL

Restaura o valor inicial contido em uma caixa de edição.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | FILL-IN |
| BROWSE CELL | RADIO-SET |
| BUTTON | SELECTION-LIST |
| COMBO-BOX | SLIDER |
| EDITOR | TOGGLE-BOX |

RETURN

Quando pressionado ENTER em um objeto.

Aplica-se para:

| | |
|---------|-------------|
| FILL-IN | BROWSE CELL |
|---------|-------------|

RIGHT-MOUSE-CLICK

Quando clicado sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

RIGHT-MOUSE-DBLCLICK

Quando pressionado duplo clique sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

RIGHT-MOUSE-DOWN

Quando pressionado o botão direito do mouse sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

RIGHT-MOUSE-UP

Quando liberado o botão direito do mouse sobre o objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | MENU-ITEM |
| BUTTON | RADIO-SET |
| COMBO-BOX | RECTANGLE |
| DIALOG-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FIELD-GROUP | SUB-MENU |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | WINDOW |
| MENU | |

SELECTION

Quando selecionado algum texto ou item em um objeto.

Aplica-se para:

| | |
|-----------|----------------|
| BROWSE | RADIO-SET |
| BUTTON | RECTANGLE |
| COMBO-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | |

START-BOX-SELECTION

Quando iniciado evento de marcação de área na janela.

Aplica-se para:

FRAME ou DIALOG BOX com a propriedade BOX-SELECTABLE igual a TRUE.

START-MOVE

Quando iniciado o evento move da janela.

Aplica-se para:

| | |
|-----------|----------------|
| BROWSE | RADIO-SET |
| BUTTON | RECTANGLE |
| COMBO-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | |

START-RESIZE

Quando iniciado o evento de redimensionamento da janela.

Aplica-se para:

| | |
|-----------|----------------|
| BROWSE | RADIO-SET |
| BUTTON | RECTANGLE |
| COMBO-BOX | SELECTION-LIST |
| EDITOR | SLIDER |
| FILL-IN | TEXT |
| FRAME | TOGGLE-BOX |
| IMAGE | |

TAB

Quando pressionado a tecla de TAB no objeto.

Aplica-se para:

| | |
|-------------|----------------|
| BROWSE | FILL-IN |
| BROWSE CELL | RADIO-SET |
| BUTTON | SELECTION-LIST |
| COMBO-BOX | SLIDER |

CONTROL-FRAME TOGGLE-BOX
EDITOR

VALUE-CHANGED

Quando é modificado o valor para browses, combo-box, fill-in, radio-set, selection-list, slider, toggle-box e menu-item.

Aplica-se para:

| | |
|----------------|------------------------|
| BROWSE | SLIDER |
| COMBO-BOX | TOGGLE-BOX |
| RADIO-SET | MENU-ITEM (toggle-box) |
| SELECTION-LIST | |

WINDOW-CLOSE

Evocado quando é fechada a janela via mouse ou tecla de finalização.

Aplica-se para:

| | |
|------------|--------|
| DIALOG-BOX | WINDOW |
|------------|--------|

WINDOW-MAXIMIZED

Quando a janela é maximizada.

Aplica-se para:

WINDOW

WINDOW-MINIMIZED

Quando a janela é minimizada.

Aplica-se para:

WINDOW

WINDOW-RESIZED

Quando a janela é redimensionada.

Aplica-se para:

WINDOW

WINDOW-RESTORED

Quando é restaurado o tamanho inicial da janela.

Aplica-se para:

WINDOW

Preprocessadores

O que são preprocessadores?

Os preprocessadores no Progress funcionam como comandos e/ou instruções literais a serem pre-utilizados para o funcionamento de um programa. Simplesmente como uma palavra-chave que equivale a própria linguagem ou lista de instruções e/ou valores para utilização.

Exemplo

```
&scoped-define p-exemplo display today.  
  
{&p-exemplo}
```

No exemplo acima foi definido um preprocessador local, disponível apenas para a procedure atual, definido como *p-exemplo* que equivale a instrução "display today".

Utilização

Após definido o preprocessador use CHAVES, E-COMERCIAL, PREPROCESSADOR.

&SCOPED-DEFINE

Define um preprocessador visível apenas para a procedure onde está declarado.

SINTAXE

&SCOPED-DEFINE Preprocessador Expressão

Preprocessador

Escrito por Márcio Brener - <http://www.geocities.com/marcio-brener>

O Nome do processador a ser instanciado na procedure.

Expressão

Uma lista de comandos ou instruções.

Exemplo

```
&scoped-define p-campos customer.cust-num~
                        customer.name~
                        customer.address

for each customer no-lock:
    display {&p-campos}.
end.
```

&GLOBAL-DEFINE

Define um preprocessador global para compartilhamento entre procedures.

SINTAXE

&GLOBAL-DEFINE Preprocessador Expressão

Preprocessador

O Nome do processador a ser instanciado na procedure.

Expressão

Uma lista de comandos ou instrução

Exemplo

```
&global-define p-exemplo cust-num,~
                        name,~
                        address,~
                        phone

select
```



```
{&p-exemplo}  
from  
    customer nlock
```

&UNDEFINE

Exclui a definição de um preprocessador.

SINTAXE

&UNDEFINE Nome do Preprocessador

Exemplo

```
&scoped-define p-exemplo 'Exemplo de Preprocessador'  
  
&undefine p-exemplo  
  
/* nada será apresentado porque o preprocessador não mais  
existe */  
&undefine p-exemplo  
  
display {&p-exemplo}.
```

DIRETIVAS PREPROCESSADAS

Diretivas preprocessadas são avaliadas apenas em tempo de execução, isso é, sua sintaxe e conteúdo são validados apenas quando solicitados sob uma condição válida.

Foi implementada a partir da versão 7 do Progress.

Exemplo

```
&scoped-define campos cust-num name  
  
&if substring(proversion, 1, 1) = '8' &then  
    for each customer fields({&campos}) no-lock:  
        display {&campos}  
        with title 'Progress versao >= 8'.  
    end.  
&else
```

```
for each customer no-lock:
    display cust-num name
        with title 'Progress versao <= 7'.
end.
&endif
```

No exemplo acima é verificada a versão do Run-Time Progress para execução de uma sintaxe condicional. Caso a versão seja 8 são informadas cláusulas válidas para esta versão, caso contrário somente instruções válidas para as versões inferiores à 8.

Operadores

&IF, &THEN, &ELSEIF, &ELSE, &ENDIF

OBSERVAÇÃO: apenas algumas funções e comandos são válidos para utilização com diretivas preprocessadas.

Instruções válidas

| | | |
|----------|------------|-------------|
| ABSOLUTE | LEFT-TRIM | R-INDEX |
| ASC | LENGTH | RANDOM |
| DATE | LIBRARY | REPLACE |
| DAY | LOG | RIGHT-TRIM |
| DECIMAL | LOOKUP | ROUND |
| ENCODE | SQRT | ENTRY |
| MAXIMUM | STRING | ETIME |
| MEMBER | SUBSTITUTE | EXP |
| MINIMUM | SUBSTRING | FILL |
| TIME | INDEX | MONTH |
| TODAY | INTEGER | NUM-ENTRIES |
| TRIM | KEYWORD | OPSYS |
| TRUNCATE | KEYWORDALL | PROPATH |
| WEEKDAY | LC | PROVERSION |
| YEAR | | |

MESSAGE

Mostra uma mensagem informativa para o usuário.

SINTAXE

```
MESSAGE [COLOR Cor] {Texto |{SKIP [(n)]}}...  
    [VIEW-AS ALERT-BOX [Tipo] [BUTTONS Botões]  
    [TITLE Texto]]  
    [{SET|UPDATE} field {{AS Tipo}}|{LIKE Campo}}  
    [FORMAT Texto] [AUTO-RETURN]]  
    [IN WINDOW Janela]
```

PARÂMETROS

COLOR Cor

Nome ou número da cor do texto da mensagem.

Texto

Texto da mensagem.

SKIP (n)

Quantidade de linhas a serem saltadas entre textos da mensagem.

VIEW-AS Tipo (Windows/X-Windows)

Define o tipo da mensagem (MESSAGE, QUESTION, INFORMATION, ERROR, WARNING).

BUTTONS Botões

Os botões de opção da mensagem, caso esta mensagem retorne um valor lógico (YES-NO, YES-NO-CANCEL, OK, OK-CANCEL, RETRY-CANCEL).

TITLE Texto

Titulo da janela do contexto da mensagem.

SET/UPDATE Campo

Campo a ser atualizado na mensagem.

AS/LIKE Tipo FORMAT

Definem o tipo e o formato da variável ou campo a ser a atualizado na mensagem.

AUTO-RETURN

Aceita <ENTER> para finalizar a atualização da mensagem.

IN WINDOW Janela

Define a janela de contexto onde será exibida a mensagem.

Exemplos

```
for each customer share-lock:
    display customer.cust-num
        customer.name
        with frame f-customer.

    message 'Confirma exclusao?'
        update l-confirma as logical format 'Sim/Nao'.

    if l-confirma then do:
        delete customer.
    end.
end.
```

```
for each customer share-lock:
    display customer.cust-num
        customer.name
    with frame f-customer.

    message 'Confirma exclusao?' view-as alert-box question
        buttons yes-no
        update l-confirma as logical format 'Sim/Nao'.

    if l-confirma then do:
        delete customer.
    end.
end.
```

Criando MENUS

MENU

Cria um menu de opções.

SINTAXE

```
DEFINE [[NEW] SHARED] MENU Nome do Menu  
      [FGCOLOR Cor] [BGCOLOR Cor]  
      [DCOLOR Cor] [PFCOLOR Cor]  
      [FONT Fonte] [{TITLE Texto} |MENUBAR]  
      [{LIKE Menu}] Item do Menu
```

PARÂMETROS

NEW SHARED

Define um novo menu a ser compartilhado entre procedures.

SHARED

Define um menu já compartilhado por uma procedure anterior.

MENU Nome do Menu

Informa o nome do menu a ser criado.

BGCOLOR Cor

Cor de fundo do menu.

DCOLOR Cor

Cor de fundo do menu para interface não gráfica.

FGCOLOR

Cor de segundo plano.

PFCOLOR

Cor de segundo plano para interface não gráfica.

FONT Fonte

Fonte a ser exibida no menu.

MENUBAR

Informa que o menu será mostrado como uma barra de menu.

TITLE Texto

Define o Título do menu a ser exibido.

LIKE Menu

Obtém/Herda as propriedades e valores de um menu já definido.

Item do Menu

Especifica qual o tipo de menu será criado:

SUB-MENU Sub-Menu [DISABLE] [LABEL Texto]

Define um sub-menu a ser adicionado em um menu já existente.

RULE

Inserir uma separação em um sub-menu.

SKIP

Inserir um intervalo entre opções de um menu.

Definições do Item do Menu

MENU-ITEM Item do Menu

[ACCELERATOR Tecla(s)] [BGCOLOR Cor]
[DCOLOR Cor] [DISABLED] [FGCOLOR Cor]
[FONT Fonte] [LABEL Texto] [PFCOLOR Cor]
[READ-ONLY] [TOGGLE-BOX] [Trigger]

Item do Menu

Define o nome do item a ser adicionado ao menu.

ACCELERATOR Tecla(s)

Define teclas de aceleração para o acesso a opção do menu, combinações do teclado como SHIF+Tecla, CTRL+Tecla, ALT, etc.

BGCOLOR Cor

Cor de fundo do menu.

DCOLOR Cor

Cor de fundo do menu para interface não gráfica.

DISABLED

Desabilita a opção do menu.

FGCOLOR

Cor de segundo plano.

PFCOLOR

Cor de segundo plano para interface não gráfica.

LABEL

Define o texto de apresentação da opção do menu.

FONT Fonte

Fonte a ser exibida no menu.

READ-ONLY

Desabilita a opção do menu para escolha do usuário.

TOGGLE-BOX

Define que a opção do menu será uma opção de marcação.

Trigger

Define triggers para associação de eventos a opção do menu.

Exemplos

```
define sub-menu tabela
    menu-item tabela-customer    label '&Customer'
    menu-item tabela-salesrep    label '&Sales'
    menu-item tabela-sair        label 'Sai&r'.

define menu principal menubar
    sub-menu tabela label 'Tabela'.

form customer
    with frame f-customer 1 columns
```

```
        title 'Customer' centered overlay.  
view frame f-customer.  
  
on choose of menu-item tabela-customer do:  
    for each customer with frame f-customer:  
        process events.  
        display customer.  
    end.  
end.  
  
assign current-window:menubar = menu principal:handle.  
  
wait-for choose of menu-item tabela-sair.
```

BOTÕES

SINTAXE

```
DEFINE BUTTON Botão [AUTO-GO|AUTO-ENDKEY]
    [DEFAULT] [BGCOLOR Cor] [DCOLOR Cor]
    [FGCOLOR Cor] [FONT Fonte]
    [IMAGE-DOWN Imagem] [{ IMAGE|IMAGE-UP}
    Imagem]
    [IMAGE-INSENSITIVE Imagem]
    [MOUSE-POINTER Ponteiro] [LABEL Texto]
    [LIKE Botão] [PFCOLOR Cor]
    [SIZE Largura BY ALTURA] [NO-FOCUS]
    [NO-CONVERT-3D-COLORS]
    [TOOLTIP Texto] [TRIGGER]
```

PARÂMETROS

DEFINE BUTTON Botão

Declara um novo botão. Caso o ambiente seja estilo caracter (UNIX/DOS/Novell,etc) a aparência deste botão será igual a uma opção de menu.

AUTO-GO

Declara que os eventos deste botão serão executados assim que escolhidos.

AUTO-ENDKEY

Define automaticamente que o botão é um botão de finalização, declarando o evento de escolha com o evento de saída (ENDKEY).

DEFAULT

Define o botão como padrão, ou seja, foco inicial do frame.

BGCOLOR Cor

Cor de background do botão.

FGCOLOR Cor

Cor de foreground do botão.

DCOLOR Cor

Cor de background para ambiente caracter.

PFCOLOR Cor

Cor de foreground para ambiente caracter.

FONT Fonte

Define a fonte para exibição do texto do botão.

IMAGE|IMAGE-UP File arquivo.bmp

Windows|arquivo.xpm Motif

Define a imagem do botão quando em estado normal.

IMAGE-DOWN File arquivo.bmp

Windows|arquivo.xpm Motif

Define a imagem do botão quando pressionado.

IMAGE-INSENSITIVE File arquivo.bmp

Windows|arquivo.xpm Motif

Define a imagem do botão quando desabilitado.

MOUSE-POINTER Ponteiro

Define o estilo do ponteiro do mouse quando apontado para o botão.

Para obter os estilos de ponteiros consulte o tópico `LOAD-MOUSE-POINTER()`

LABEL Texto

Informa o texto a ser exibido no botão.

LIKE Botão

Herda as propriedades e valores de um botão já existente.

SIZE Largura BY Altura

Define as dimensões do botão.

NO-FOCUS

Desabilita o foco para o botão.

NO-CONVERT-3D-COLORS (MS-Windows)

Desabilita a conversão de cores da imagem do botão.

Por padrão o Progress converte as cores para uma faixa RGB pré-definida, às vezes distorcendo a aparência da imagem, então usa-se esta opção para evitar este tipo de problema.

TOOLTIP Texto (MS-Windows)

Define um texto de exibição quando o ponteiro do mouse estiver sobre o botão

TRIGGER

Declara as triggers a serem executadas para os eventos do botão.

Exemplo

```
define button b-ok auto-go default
    label '&Ok' tooltip 'Teste'
    image-up    file 'igp/sum.bmp'
    image-down  file 'igp/delete.bmp'
    image-insensitive file 'igp/cut.bmp'
    mouse-pointer 'glove'
    auto-go.

form b-ok with frame f-botoes row 5 centered.

on choose of b-ok in frame f-botoes
    message 'Botao OK pressionado' view-as alert-box.

update b-ok with frame f-botoes.
```

UPDATE

Habilita a digitação de valores para um campo ou variável

SINTAXE

UPDATE [UNLESS-HIDDEN]

```
[{Tabela|Campo|Variável [Formato] [WHEN Expressão]}|  
  {VIEW-AS Tipo}}|  
  {Campo|Variável = Expressão n }|{[AT Coluna|TO Coluna  
COLON-ALIGNED|LEFT-ALIGNED|RIGHT-ALIGNED]}|  
  {^}|{SPACE [(n)]}|{SKIP [(n)]}}|...  
[GO-ON(Key-Label...)] [Frame] [EDITING:]  
[NO-ERROR]
```

PARÂMETROS

[UNLESS-HIDDEN]

Não habilita a digitação para campos de uma tabela cujo atributo HIDDEN seja verdadeiro.

UPDATE Tabela|Campo|Variável

Informa a tabela, campo e/ou variável a ser habilitada para digitação ou atualização.

FORMAT Formato

Define o formato de entrada dos dados para atualização.

WHEN Expressão

Condiciona a digitação de um campo e/ou variável apenas se satisfeita alguma condição estabelecida.

VIEW-AS Tipo

Define o tipo de visualização, objeto do campo ou variável. Tipos disponíveis (TEXT, FILL-IN, EDITOR, TOGGLE-BOX, RADIO-SET, SELECTION-LIST, COMBO-BOX).

Campo|Variável = Expressão n

Atribui valores a um campo ou variável.

Exemplo: update customer.name = "Marcio Brener".

AT Coluna | TO Coluna

AT - Define a coluna de apresentação do campo para atualização à esquerda.

TO – Define a coluna de apresentação do campo para atualização à direita.

COLON-ALIGNED|LEFT-ALIGNED|RIGHT-ALIGNED

Informa o alinhamento do campo em relação a coluna definida em AT Coluna ou TO Coluna.

SPACE(n)

Demonstra n espaços em um frame para atualização.

SKIP n

Salta n linhas entre campos.

GO-ON(Key-Label)

Informa qual será a tecla de atualização dos campos e/ou variáveis.

Frame

Informa o frame associado à atualização dos campos e/ou variáveis.

EDITING

Habilita a edição de eventos a serem relacionados.



Esta forma de EDIÇÃO DE EVENTOS procedural não é indicada. Este tipo de manipulação de exemplos era utilizado até a versão 6 do Progress. É mantida por questões de compatibilidade. Use ON Evento OF Objeto IN FRAME Frame.

NO-ERROR

Não apresenta erros do frame para a atualização.



Não é uma boa forma fazer as configurações em uma cláusula de UPDATE. A forma mais adequada é definir todos os parâmetros em um FRAME e associa-lo a uma cláusula de UPDATE.

Exemplo

Importação e Exportação de Dados

A importação e exportação de dados no Progress é bastante simplificada, com poucas linhas de código é possível fazer um programa completo de importação e exportação para arquivo texto ou banco de dados, como demonstrado a seguir:

Comando IMPORT

```
IMPORT [STREAM Nome da Stream] [DELIMITER Character  
Delimitador] {Registro [EXCEPT Campo [Campo]...] ou  
^} [UNFORMATTED] [NO-ERROR]
```

Parâmetros

Stre

Limites do Progress

Estes valores se referem ao Progress versão 8.x.

Banco de Dados

TABELAS

O máximo do tamanho do banco de dados.

ÍNDICES

Sem limites por tabela, com número máximo de índices por tabela de acordo com o tamanho dos blocos do Sistema Operacional.

| <u>Blocos</u> | <u>Máximo</u> |
|---------------|---------------|
| 512 | 2048 |
| 1024 | 4096 |
| 2048 | 8192 |
| 4096 | 16384 |

REGISTROS

De 1 a 32.000 bytes.

CAMPOS

Até o tamanho máximo da tabela.

WORKFILES

Tamanho mínimo de 64 bytes, com o máximo limitado pela quantidade de memória da máquina cliente.

MULTI-VOLUMES

Até 256 arquivos estendidos, com o tamanho máximo de acordo com o tamanho dos blocos do Sistema Operacional.

| | <u>BLOCOS</u> | <u>MÁXIMO</u> |
|-----------------------------------|---------------------------------------------------------------------------------|----------------------|
| | 512 bytes | 32GB |
| | 1024 bytes (1K) | 64GB |
| | 2048 bytes (2K) | 128GB |
| | 4096 bytes (4K) | 256GB |
| NÚMERO DE USUÁRIOS | Até 2048 sessões simultâneas multi-usuários em UNIX ou Windows NT. | |
| TRANSAÇÕES SIMULTÂNEAS | 1 transação por usuário com o máximo de 2048 transações. | |
| ORDENAÇÃO DE REGISTROS | De 1 a 16 campos ou expressões. De 1 a 197 caracteres iniciais do texto. | |

VARIÁVEIS

| | |
|------------------------|-----------------------------------------|
| CHARACTER | 3.000 caracteres. |
| DATE | 01/01/32768 A/C até 12/31/32767 D/C. |
| DECIMAL | 50 dígitos para inteiros mais decimais. |
| INTEGER | -2,147,483,648 até +2,147,483,647. |
| LOGICAL | Yes/No, True/False. |
| ENTRADA E SAÍDA | 1 a 3.000 caracteres por campo. |

DE DADOS

| | |
|----------------------------------------|------------------------------------------|
| NÚMERO DE COLUNAS NA TELA | 80 a 132 colunas para apresentação. |
| NÚMERO DE COLUNAS NA IMPRESSORA | 1 a 255 colunas para saída de impressão. |
| STREAMS | 1 a 5 streams por procedure. |

NOMENCLATURAS

| | |
|-----------------------------------------------|---------------------------------------------------------------------------------------------|
| PATH | UNIX: 1 - 60 Caracteres WINDOWS: 1 - 255 WINDOWS NT usando NTFS: sem limites. |
| BANCO DE DADOS | De 1 a 11 caracteres. |
| VARIÁVEIS, STREAMS, PROCEDURES | De 1 a 12 caracteres. |

Funções

ABSOLUTE

SINTAXE - ABSOLUTE(Decimal/Inteiro) Decimal/Inteiro

Retorna um valor inteiro ou decimal o mesmo formato.

Exemplo

```
message absolute(172.34 - 12.32) view-as alert-box.  
/* Retorno igual a 160,02 */
```

ACCUM

SINTAXE - ACCUM Valor de Agregação

Acumula valores pelo total da soma de um agrupamento. Este valor de retorno não pode ser recebido por uma variável, apenas mostrado.

Exemplo

```
/* ex0001.p */  
  
for each order no-lock:  
    display order-num  
           cust-num  
           order-date  
           promise-date  
           ship-date  
           with centered.  
  
    for each order-line of order no-lock:  
        display line-num  
               item-num  
               qty  
               price  
               (qty * price) label "Valor Total".  
        accumulate qty * price (total).  
        display (accum total qty * price) label "Valor Acumulado"
```

```
with centered title "Detalhes".
end.
display (accum total qty * order-line.price) label
"Total"
with centered title "Ordem".
end.
display (accum total qty * order-line.price) format "-
>>>, >>>, >>9.99"
with view-as dialog-box title "Total Geral".
pause.
```

ALIAS

SINTAXE - ALIAS(Inteiro) Caracter

Retorna o Nome Lógico do(s) banco(s) de dados conectado(s).

Exemplo

```
message "Alias:" alias(1) skip
"Nome Lógico:" ldbname(alias(1)) view-as alert-box.
```

AMBIGUOUS

SINTAXE - AMBIGUOUS Nome da Tabela (retorno Lógico)

Retorna um valor lógico para o resultado de uma procura. Caso uma procura única de um valor seja encontrado em duplicidade, isso retornará como verdadeiro em uma condição.

Exemplo:

```
/* ex0002.p */
/* O resultado do find retornará mais de uma ocorrência */
find customer where customer.name begins "a" no-error.
if available customer then
    message "Apenas um valor encontrado" view-as alert-box.
else if ambiguous customer then
    message "Mais de um valor encontrado" view-as alert-box.
else
```

```
message "Nenhum valor encontrado" view-as alert-box.
```

ASC

SINTAXE

ASC(Character, [Código de Página Destino], (Código de Página Origem) Inteiro

Retorna o valor ASC de um caracter. Os Códigos de Página de Origem e Destino por padrão são obtidos do arquivo de configuração inicial do Progress "startup.pf", ou do arquivo de parametros do aplicativo, caso esse tenha uma configuração do Código de Página.

Exemplo

```
/* ex0003.p */
/* Retorna 231 */
message asc("ç") view-as alert-box.
/* Retorna 254 */
message asc("ç", "iso8859-1", "ibm850") view-as alert-box.
/* Retorna 135 */
message asc("ç", "ibm850", "iso8859-1") view-as alert-box.
```

AVAILABLE

SINTAXE - AVAILABLE Nome da Tabela (retorno Lógico)

Retorna o sucesso/insucesso de uma procura ou registro disponível.

Exemplo

```
find first customer where customer.name begins "Al" no-
error.
if available customer then
    message "Procura com Sucesso" view-as alert-box.
else
    message "Nenhum nome encontrado" view-as alert-box.
for each customer:
```



```
if available customer then message "Procura OK!".  
end.
```

CAN-DO

SINTAXE

CAN-DO(Cadeia Caracter, Valor Caracter) Lógico

Testa se um valor está contido em uma cadeia delimitada por vírgula.

Exemplo

```
/* Retorna com VERDADEIRO */  
message can-do("Brasil,Alemanha,Japao", "Japao") view-as alert-box.  
  
/* Retorna com FALSO */  
message can-do("Brasil,Alemanha,Japao", "EUA") view-as alert-box.
```

CAN-FIND

SINTAXE

[NOT] CAN-FIND([FIRST | LAST] Registro

[OF Nome da Tabela] ou [WHERE Expressão] [USE-INDEX Índice]

ou

[USING [FRAME Nome do Frame] Campo [AND [FRAME Nome do Frame] Campo]]

[SHARE-LOCK|NO-LOCK] [NO-WAIT] [NO-PREFETCH])

Valida uma pesquisa segundo os critérios informados.

Exemplo

```
/* ex0004.p */
```

```

form customer.cust-num at row 1 column 10 colon-aligned
    validate(can-find(customer where
        customer.cust-num = input frame f-customer
        customer.cust-num),
        "Cliente nao cadastrado")

    customer.name at row 2 column 10 colon-aligned
    with frame f-customer row 5 centered
    title "Alteracao" side-labels.

repeat:
    prompt-for customer.cust-num
        with frame f-customer.
    find customer using input frame f-customer
customer.cust-num.

    update customer.name
        with frame f-customer.
end.

```

CAN-QUERY

SINTAXE

CAN-QUERY(Handle do Objeto, Propriedade [Caracter]) Lógico

Verifica se um Objeto(Widget) contém um determinada .propriedade.

Exemplo

```

/* ex0005.p */

define variable attribute as character format "x(24)" label "Propriedade".
define variable queryable as logical view-as toggle-box label "Query".
define variable setable as logical view-as toggle-box label "Set".
define variable temp-handle as widget-handle.
define variable widget-type as character format "x(24)" label "Objeto".

form widget-type attribute setable queryable
    with frame f-frame
        title "Exemplo da Funcao CAN-QUERY"
        centered row 2.

```

```
repeat:
    update widget-type attribute with frame f-frame.

    create value(widget-type) temp-handle.
    assign queryable = can-query(temp-handle, attribute)
        setable = can-set(temp-handle, attribute).

    display queryable setable.
    delete widget temp-handle.
end.
```

CAN-SET

SINTAXE - CAN-SET(Handle do Objeto, Nome d Propriedade [Character])

Verifica se um Objeto pode receber uma Propriedade específica.

Exemplo

```
/* Retorna Falso */
message can-set(session:handle, "list-items") view-as alert-box.

/* Retorna Verdadeiro */
message can-set(session:handle, "data-entry-return") view-as alert-box.
```

ENCODE

SINTAXE - ENCODE(Texto) Caracter

Recebe um texto e o retorna criptografado.

Exemplo

```
/* Retorno igual a dVXxjWfn */
message ENCODE("Progress") view-as alert-box
```

CHR

SINTAXE

**CHR(Inteiro, [Código de Página Origem],
[Código de Página Destino])** Caracter

Converte um inteiro para o seu caracter correspondente.

O Integer entre 1 e 255 corresponde a um caracter simples, passível a visualização e impressão em diversos terminais. Para valores superiores a 255 e inferiores a 65535 a função tentará verificar o byte equivalente, caso não encontrado um byte nulo será retornado.

Código de Página Origem/Destino é mapa de caracteres utilizado para a entrada/saída do caracter, que podem ser omitidos, então serão utilizados os Códigos de Páginas previamente utilizados na configuração do Progress, conforme o arquivo em DLC/convmap.cp e o parametros -cpstream e -cpinternal.

Exemplo

```
display chr(232) skip  
chr(232, "ibm850", "iso8859-1") skip  
chr(232, "iso8859-1", "ibm850").
```

CODEPAGE-CONVERT

SINTAXE

**CODEPAGE-CONVERT(Caracter ,
[Código de Página Destino],
[Codigo de Página Origem])**

Converte um texto de acordo com os Códigos Páginas especificados.

Maiores detalhes no Tópico **Códigos de Página**.

CONNECTED

SINTAXE

CONNECTED(Nome Lógico do Banco de Dados) Lógico

Informa se o Banco de Dados está conectado na aplicação.

Exemplo

```
if not connected("sports") then
    connect -db value(search("sports.db")) -ld sports -1.
```

COUNT-OF

SINTAXE - COUNT-OF(Campo do BREAK-BY) Inteiro

Retorna o número de ocorrências de um registro dentro de seleção agrupada.

Exemplo

```
/* ex0010.p */
for each customer no-lock break by state:
    display cust-num name sales-rep state.
    accumulate state (sub-count by state).
    if last-of(state) then
        display 100 * (accum sub-count by state state) /
            count-of(state) format ">>9.9999%"
            (total) column-label "% Total de!Consumidores"
            with title "Vendas por Estados" centered.
end.
pause.
```

CURRENT-CHANGED

SINTAXE

CURRENT-CHANGED Nome da Tabela [retorno Lógico]

Informa se o registro corrente está alterado.

Exemplo

```
/* ex0011.p */
assign session:data-entry-return = yes.

/*
    Para observar este exemplo adequadamente
    abra duas janelas do Procedure Editor, então
    execute este código nas duas janelas.
    Visualize o valor na primeira, e na segunda
    altere o valor e tente alterá-lo novamente na
    primeira. Será retornada uma advertência
    informando que o valor visualizado na primeira
    janela não mais condiz com o valor do registro
    atual.
*/

form customer.name
    customer.balance
    with frame f-update row 5 centered title "Mudanca de Valores".

on "go":u of frame f-update do:
    do transaction:
        find current customer exclusive-lock.
        if current-changed customer then do:
            message "Este valor foi alterado por outro usuario"
            skip
            "Por favor insira novamente a modificacao."
            view-as alert-box.
            display customer.name customer.balance with frame f-update.
            return no-apply.
        end.
        assign customer.name customer.balance.
    end.
    find current customer no-lock.
end.

find first customer no-lock.
```

```
display customer.name customer.balance with frame f-update.  
do on endkey undo, leave:  
    enable customer.name customer.balance with frame f-update.  
    wait-for "go":u of frame f-update.  
end.
```

CURRENT-LANGUAGE

SINTAXE - CURRENT-LANGUAGE [retorno Caracter]

Informa a linguagem atual de acordo com a configuração do parametro **-lng [Idioma]**.

Exemplo

```
message "O parametro -lng atual é" current-language  
view-as alert-box.
```

CURRENT-RESULT-ROW

SINTAXE

CURRENT-RESULT-ROW(Nome da Query [Caracter]) Inteiro

Retorna o número da linha na Query.

Exemplo

```
/* ex0012.p */  
  
define query q-customer for customer scrolling.  
define browse b-customer query q-customer  
    display current-result-row("q-customer") column-label "Nº"  
        country  
        cust-num  
        name  
        with title "Lista de Clientes por Pais"  
        centered 10 down separators.  
open query q-customer for each customer no-lock by country.  
  
update b-customer with centered row 5.
```

CURRENT-ROW-MODIFIED

SINTAXE

CURRENT-ROW-MODIFIED Nome da Tabela [Lógico]

Informa se o valor de uma coluna foi alterado.

CURRENT-VALUE

SINTAXE

**CURRENT-VALUE(Sequence, [Nome Lógico do Banco])
Inteiro**

Retorno o valor corrente de um Sequence do banco de dados.

Exemplo

```
message "O ultimo Cliente é" current-value(next-cust-num)  
view-as alert-box.
```

DATASERVERS

SINTAXE - DATASERVERS [retorno Lista Caracter]

Informa uma lista de Tipos de Bancos de Dados suportados na configuração/versão Progress - Oracle, ODBC, Sybase, MS-SQL, etc.

DATE

SINTAXE - DATE(Texto ou Mês,Dia,Ano) retorno Data

Converte um texto em data ou a seqüência Mês, Dia, Ano como inteiros para data.

Se informado o parâmetro como texto este deve estar no formato pré-definido pela configuração do Progress, **parâmetro -d**.

Exemplo

```
/* 31 de Janeiro de 2019 */  
display date("31/01/2019") format "99/99/9999".  
display date(01, 31, 2019) format "99/99/9999".  
pause.
```

DAY

SINTAXE - DAY(Data) Inteiro

Retorna o dia do mês de uma data entre 1 e 31.

Exemplo

```
message "Hoje é o dia" day(today) "do mes" view-as alert-box.
```

DBCODEPAGE

SINTAXE

DBCODEPAGE(Inteiro ou Nome Lógico do Banco de Dados) Caracter

Informa o Código de Página atribuído a um Banco de Dados conectado.

Exemplo

```
define variable i as integer no-undo.  
do i = 1 to num-dbs:  
    display ldbname(i) dbcodepage(i) format "x(20)".  
    pause.  
end.
```

DBCOLLATION

SINTAXE

DBCOLLATION(Inteiro ou Nome Lógico do Banco de Dados) Caracter

Retorna o Código de Página, obtido do arquivo convmap.cp, de um Banco de Dados conectado. Equivalente ao parâmetro -cpcoll especificado no "start" do Banco.

Exemplo

```
define variable i as integer no-undo.  
do i = 1 to num-dbs:  
    display ldbname(i) dbcollation(i) format "x(20)".  
    pause.  
end.
```

DBNAME

SINTAXE - DBNAME retorno Caracter

Retorna o Nome Físico do Banco de Dados, incluindo a localização (diretório).

Exemplo

```
display "Database:" to 10 dbname format "x(40)" skip  
    "Usuario:" to 10 userid.  
pause.
```

DBPARAM

SINTAXE

DBPARAM(Inteiro ou Nome Lógico do Banco de Dados) Lista Caracter

Retorna uma Lista Delimitada informado os parametros utilizados na conexão do Banco.

Exemplo

define variable i as integer no-undo.

```
do i = 1 to num-dbs:
  display "Banco:"          to 12 pdbname(i) format "x(60)" skip
    "Logico:"              to 12 ldbname(i) format "x(60)" skip
    "Parametros:"         to 12 dbparam(i) format "x(60)".
  pause.
end.
```

DBRESTRICTIONS

SINTAXE

DBRESTRICTIONS(Inteiro ou Nome Lógico do Banco de Dados) Lista Caracter

Retorna uma Lista Delimitada informado as Restrições na conexão do Banco de Dados.

Exemplo

define variable i as integer no-undo.

```
do i = 1 to num-dbs:
  display "Banco:"          to 12 pdbname(i)          format "x(60)" skip
    "Logico:"              to 12 ldbname(i)          format "x(60)" skip
    "Parametros:"         to 12 dbparam(i)          format "x(60)"
    "Restricoes:"         to 12 dbrestrictions(i) format "x(60)".
  pause.
end.
```

DBTASKID

SINTAXE

DBTASKID(Inteiro ou Nome Lógico do Banco de Dados) Inteiro

Retorna o Identificado da Transação Atual no Banco de Dados.

Exemplo

```
display "Nenhuma Transacao atual" format "x(50)"
      dbtaskid(1).

/* Inicio de uma Transacao */
create customer.

display "Uma Transacao ativa (CREATE), ID" format "x(50)"
      dbtaskid(1).
pause.
```

DBTYPE

SINTAXE

DBTYPE(Inteiro ou Nome Lógico do Banco de Dados)
Caracter

Retorna o Tipo de Banco de Dados, Progress ou outros.

Exemplo

```
define variable i as integer no-undo.

do i = 1 to num-dbs:
  display "Banco:"          to 12 pdbname(i)          format "x(60)" skip
        "Logico:"          to 12 ldbname(i)          format "x(60)" skip
        "Parametros:"      to 12 dbparam(i)          format "x(60)" skip
        "Restricoes:"      to 12 dbrestrictions(i)    format "x(60)" skip
        "Tipo:"            to 12 dbtype(i)            format "x(60)".

  pause.
end.
```

DBVERSION

SINTAXE

DBVERSION(Inteiro ou Nome Lógico do Banco de Dados)
Caracter

Retorna a Versão do Banco de Dados.

Exemplo

```
/* ex0013.p */
define variable i as integer no-undo.

do i = 1 to num-dbs:
    display "Banco:"          to 12 pdbname(i)          format "x(60)" skip
    display "Logico:"         to 12 ldbname(i)          format "x(60)" skip
    display "Parametros:"     to 12 dbparam(i)          format "x(60)" skip
    display "Restricoes:"     to 12 dbrestrictions(i)   format "x(60)" skip
    display "Tipo:"           to 12 dbtype(i)           format "x(60)" skip
    display "Versao:"         to 12 dbversion(i)        format "x(60)" with
    frame f-informacao centered row 5
    title "Informacoes dos Bancos de Dados".
    pause.
end.
```

DECIMAL

SINTAXE - DECIMAL(Inteiro/Caracter/Data) Decimal

Transforma uma expressão em Decimal.

Exemplo

```
define variable i as integer initial 192.
define variable d as decimal initial
569789215745369445.9283832.

display decimal(today)      format ">>>,>>>,>>9" skip
display decimal("9283.12") format ">>>,>>>,>>9" skip
display decimal(i)          format ">>>,>>>,>>9" skip.
message d view-as alert-box.
pause.
```

DEFINED

SINTAXE - DEFINED(Preprocessador) Lógico

Retorna se um Preprocessador foi instanciado. Somente utilizado para condições preprocessadas (&if).

Exemplo

Escrito por Márcio Brener - <http://www.geocities.com/marcio-brener>

```
&scoped-define frase 'Powered by Progress'

&if defined(frase) &then
    message 'Preprocessador teste igual a' {&frase}
    view-as alert-box.
&else
    message 'Processador nao definido' view-as alert-box.
&endif
```

DYNAMIC-FUNCTION

SINTAXE

DYNAMIC-FUNCTION(Função [In Handle] [Parâmetro 1, Parâmetro 2, ...])

Executa uma Função do Usuário em tempo de execução(Run Time).

Exemplo

```
/* ex0014.p */

define variable i          as integer    no-undo.
define variable nfuncao as character no-undo
    extent 2 initial ["Data", "Hora"].

do i = 1 to extent(nfuncao):
    message dynamic-function(nfuncao[i]) view-as alert-box.
end.

function Data returns character:
    return "A data é" + string(today, "99/99/9999").
end function.

function Hora returns character:
    return "Agora são" + string(time, "hh:mm:ss").
end function.
```

ENTERED

SINTAXE - [FRAME Nome do Frame] Campo ENTERED

Informa se um campo de um frame foi modificado desde a última atualização.

Exemplo

```
define variable nome as character format "x(15)".

repeat:
    form nome
        with frame f-update.

    update nome with frame f-update.

    if frame f-update nome entered then
        message "Novo nome" input frame f-update nome.
    else
        message "Nenhuma modificacao".
end.
```

ENTRY

SINTAXE

ENTRY(n Elemento, Lista, [Delimitador]) Caracter

Retorno o Texto de um elemento contido em uma Lista.

O Delimitador da lista pode ser omitido, então será assumido como delimitação padrão o caracter "," (vírgula).

Exemplo

```
define variable dia as character no-undo.

assign dia = "Domingo;Segunda;Terca;Quarta;Quinta;Sexta;Sabado".

message "Hoje é" entry(weekday(today), dia, ";") view-as alert-box.
```

ETIME

SINTAXE - ETIME(Lógico) Inteiro

Retorna o tempo em milissegundos a partir do início da sessão Progress. Se passado um valor Lógico Positivo (YES ou TRUE) à função o seu timer é zerado.

Exemplo

```
pause 0 before-hide.  
  
/* Zera o Timer */  
etime(yes).  
  
for each customer no-lock:  
    display customer with 1 columns.  
end.  
  
message "Este codigo foi executado em" etime "milissegundos"  
    view-as alert-box.
```

EXP

SINTAXE - EXP(Base, Expoente) Decimal

Exponenciação de um Valor Base X *n* Expoente.

Exemplo

```
/* 2 elevado ao cubo */  
display exp(2, 3).  
pause.
```

EXTENT

SINTAXE - EXTENT(Variável) Inteiro

Retorna o tamanho do array de uma variável.

Exemplo

Escrito por Márcio Brener - <http://www.geocities.com/marcio-brener>


```
define variable i as integer.
define variable dia as character extent 7
    initial ['Domingo',
            'Segunda',
            'Terca',
            'Quarta',
            'Quinta',
            'Sexta',
            'Sabado'].

do i = 1 to extent(dia):
    message dia[i] view-as alert-box.
end.
```

FILL

SINTAXE - FILL(Texto, Repetição) Caracter

Replica um Texto *n* vezes informado.

Exemplo

```
define variable nome as character format "x(30)".

repeat:
    update nome.
    display nome skip
        fill('*', length(nome)) format "x(30)".
end.
```

FIRST

SINTAXE - FIRST(Campo de Agrupação) Lógico

Retorna um valor lógico verdadeiro da primeira interação de agrupamento, então se é a primeira ocorrência em uma cláusula DO, FOR EACH, REPEAT com BREAK BY anterior é retornado como verdadeiro.

Exemplo

```

/* ex0015.p */
define variable order-value as decimal.

for each order:
  display order-num.
  for each order-line of order break by qty * price:
    /*
      Quando for o primeiro Item da Ordem o
      valor da Ordem sera zerado para agrupa-lo
      ate o fim dos Itens
    */

    if first(qty * price) then order-value = 0.
    order-value = order-value + qty * price.
    display line-num
      item-num qty * price
      column-label "Preco!Global".
  end.
  display order-value.
end.

```

FIRST-OF

SINTAXE - FIRST-OF(Campo de Agrupação) Lógico

Retorna um valor lógico verdadeiro se é o primeira ocorrência do agrupamento em uma cláusula DO, FOR EACH, REPEAT com BREAK BY.

Exemplo

```

/* ex0016.p */
for each order no-lock
break by order.order-num:
  if first-of(order.order-num) then do:
    find customer of order no-lock.
    display customer.cust-num
      customer.name
      order.order-num
      with centered title "Ordem".

  for each order-line of order no-lock:
    display order-line except order-line.order-num
      with centered title "Itens".

```

```
end.  
end.  
end.
```

FRAME-COL

SINTAXE - FRAME-COL(Nome do Frame) Inteiro

Retorna a posição da coluna de uma frame.

Exemplo

```
form "Teste"  
    with frame f-teste row 5 centered.  
view frame f-teste.  
  
message "O frame teste esta na coluna" frame-col(f-teste)  
    view-as alert-box.
```

FRAME-DB

SINTAXE - FRAME-DB [retorno Caracter]

Retorna o Nome o Banco de Dados no Frame atual.

FRAME-DOWN

SINTAXE - FRAME-DOWN [retorno Inteiro]

Retorna o número atual da linha dentro do frame atual.

Exemplo

```
/* ex0018.p */  
define variable resposta as logical format "Sim/Nao".  
assign resposta = yes.  
  
repeat:  
    find next customer no-error.  
    display customer.cust-num  
        customer.name  
        with 15 down centered row 3.
```

```
if frame-line = frame-down then
    message "Continua?" update resposta.

if not resposta or not available customer then leave.
end.
```

FRAME-FIELD

SINTAXE - FRAME-FIELD [retorno Caracter]

Retorna o Campo ou Variavel atual de um Frame.

FRAME-FILE

SINTAXE - FRAME-FILE [retorno Caracter]

Retorna a Tabela Atual usada em um Frame.

FRAME-INDEX

SINTAXE - FRAME-INDEX [retorno Inteiro]

Retorna o Número do Índice(Array) do frame atual.

Exemplo

```
define variable opcao as character
    extent 4 initial ['Primeiro',
                     'Segundo',
                     'Terceiro',
                     'Sair'].

repeat while frame-index NE 4:
    display opcao
        with frame f-opcao
        no-labels 1 columns centered
        row 5.
    choose field opcao with frame f-opcao.
    message "Opcao" frame-index.
end.
```

FRAME-LINE

SINTAXE - FRAME-LINE [retorno Inteiro]

Retorna Linha atual do frame atual.

FRAME-NAME

SINTAXE - FRAME-NAME [retorno Caracter]

Retorna o Nome do frame atual.

FRAME-ROW

SINTAXE - FRAME-ROW(Nome do Frame) Inteiro

Retorna a posição da linha superior de um frame.

Exemplo

```
/* ex0020.p */
display "^" with frame f-teste.

repeat while keyfunction(lastkey) <> "end-error":
    view frame f-teste.

    if keyfunction(lastkey) = "cursor-down" and
        frame-row(f-teste) < screen-lines - 1 then
        assign frame f-teste:row = frame-row(f-teste) + 1.

    if keyfunction(lastkey) = "cursor-up" and
        frame-row(f-teste) > 1 then
        assign frame f-teste:row = frame-row(f-teste) - 1.

    if keyfunction(lastkey) = "cursor-right" and
        frame-col(f-teste) < 78 then
        assign frame f-teste:col = frame-col(f-teste) + 1.

    if keyfunction(lastkey) = "cursor-left" and
        frame-col(f-teste) > 1 then
        assign frame f-teste:col = frame-col(f-teste) - 1.
```

```
hide message no-pause.  
message 'Linha' frame-row(f-teste)  
        'Coluna' frame-col(f-teste).  
pause.  
end.
```

FRAME-VALUE

SINTAXE - FRAME-VALUE (retorno Caracter)

Retorna o valor do frame atual.

Exemplo na função KEYFUNCTION

GATEWAYS

SINTAXE - GATEWAYS [retorno Caracter]

Mostra a lista de Bancos de Dados passíveis à conexão, exatamente igual à função DATASERVERS.

GET-BYTE

SINTAXE - GET-BYTE(Campo, Posição) Caracter

Retorna um subtexto de um campo de uma tabela não Progress

GET-CODEPAGES

SINTAXE - GET-CODEPAGES [Lista Caracter]

Retorna uma lista dos Códigos de Páginas disponíveis.

Exemplo

```
define variable i as integer.
```

```
do i = 1 to num-entries(get-codepages):  
    display entry(i, get-codepages) format "x(20)"  
        with 20 down title "Codigos de Pagina"  
        centered.  
    down.  
end.
```

GET-COLLATIONS

SINTAXE

GET-COLLATIONS(Código de Página) [Lista Caracter]

Retorna uma lista dos Idiomas disponíveis de acordo com o seu Código de Página.

Exemplo

```
/* ex0022.p */  
define variable i as integer.  
define variable j as integer.  
  
do i = 1 to num-entries(get-codepages):  
    display entry(i, get-codepages) format "x(20)".  
  
    do j = 1 to num-entries(get-collations(entry(i, get-  
codepages))):  
        display entry(j, get-collations(entry(i, get-  
codepages)))  
            with 20 down centered.  
    down.  
end.  
end.
```

GET-DOUBLE

SINTAXE - GET-DOUBLE(Campo, Posição) Float

Retorna um valor tipo Float 8 bytes de um campo de uma tabela não Progress.

GET-FLOAT

SINTAXE - GET-DOUBLE(Campo, Posição) Float

Retorna um valor tipo Float 4 bytes de um campo de uma tabela não Progress.

GET-LONG

SINTAXE - GET-DOUBLE(Campo, Posição) Longo

Retorna um valor tipo longo de 32 bit de um campo de uma tabela não Progress

GET-POINTER-VALUE

SINTAXE

GET-POINTER-VALUE(Variável MEMPTR) Integer

Retorna um valor inteiro de uma variável tipo MEMPTR

GET-SHORT

SINTAXE - GET-SORT(Campo, Posição) Longo

Retorna um valor longo 16 bit de um campo de uma tabela não Progress

GET-SIZE

SINTAXE - GET-SIZE(Variável MEMPTR)

Retorna um valor inteiro em bytes de uma região da memória de tipo MEMPTR

GET-STRING

SINTAXE - GET-STRING(Campo, Posição)

Retorna uma string de terminação nula de uma variável tipo MEMPTR

GET-UNSIGNED-SHORT

SINTAXE - GET-BYTE(Campo, Posição) Caracter

Retorna um subtexto de um campo de uma tabela não Progress

INDEX

SINTAXE - INDEX(Texto, Sub-Texto, [Início]) Inteiro

Retorna a Posição de um Sub-Texto dentro de um Texto.

Exemplo

```
define variable alfabeto as character.  
define variable letra    as character.  
  
assign alfabeto = "abcdefghijklmnopqrstuvwxyz".  
  
update letra.  
  
message "A Letra" letra "é a" index(alfabeto, letra)  
      "do Alfabeto"  
      view-as alert-box.
```

INTEGER

SINTAXE

INTEGER(Expressão Caracter/Data/Lógico) Inteiro

Converte uma expressão para um valor inteiro.

Se um texto este deverá ter um formato válido contendo de 0 a 9 como dígitos, "," milhar e "." decimal;

Se um valor lógico será atribuído 1 para Yes/True, 0 para No/False;

Se uma data este retornará o número de dias a partir da data 01/Jan/4713 A/C.

Exemplo

```
display integer("838.483") skip
integer("999,999") skip
integer(today) skip
integer(yes).
```

IS-ATTR-SPACE

SINTAXE - IS-ATTR-SPACE [retorno Caracter]

Informa se o terminal corrente faz espaçamento automático de caracteres.

KBLABEL

SINTAXE - KBLABEL(Função do Teclado) Caracter

Retorna o nome(s)/combinação(ões) da(s) tecla(s) da função.

Exemplo

```
repeat:
  display
  keyfunction(lastkey)          format "x(20)" label
"Funcao"    at 01
  kblabel(keyfunction(lastkey)) format "x(20)" label
"Tecla(s)"  at 25.
```

```
    pause.  
end.
```

KEYCODE

SINTAXE - KEYCODE(Função do Teclado) Inteiro

Retorna o número correspondente à tecla ou combinação da função.

Exemplo

```
repeat:  
    display  
    keyfunction(lastkey)          format "x(20)" label "Funcao"    at 01  
    kblabel(keyfunction(lastkey)) format "x(20)" label "Tecla(s)" at 25  
    keycode(keyfunction(lastkey)) format "->>>9" label "Codigo"    at 70.  
    pause.  
end.
```

KEYFUNCTION

SINTAXE - KEYFUNCTION(Códiga da Tecla)

Retorno o dia da semana em inteiro.

Exemplo

```
/* ex0021.p */  
find first customer.  
message "Frente e Traz muda o registro".  
repeat:  
    find current customer.  
    display cust-num  
        name  
        address  
        city  
        state  
        postal-code  
    with centered row 3 overlay  
        frame f-customer 1 columns.  
  
    set address  
        city
```

```

state
postal-code
with frame f-customer
editing:
    readkey.
    apply lastkey.
    display
    "atualizando:" frame-field format "x(20)" skip
    " da tabela:" frame-file skip
    " com valor:" frame-value format "x(20)" skip(2)
    with centered row 15 overlay.

    if keyfunction(lastkey) = "cursor-right" or
       keyfunction(lastkey) = "cursor-left" then

do:

        case keyfunction(lastkey):
            when "cursor-right" then find next customer.
            when "cursor-left" then find prev customer.
        end.

        display cust-num
            name
            address
            city
            state
            postal-code
            with frame f-customer.

    end.
end.
end.

```

KEYLABEL

SINTAXE - KEYLABEL(Código da(s) Tecla(s)) Caracter

Informa as teclas ou a combinação de acordo com o código informado.

Exemplo

```

/* ex0023.p */
/* Tecla F3 para ver melhor o resultado */
repeat:
    display
    keyfunction(lastkey)                format "x(20)" label "Funcao"    at 01

```

```
kblabel(keyfunction(lastkey)) format "x(20)" label "Tecla(s)" at 21
keylabel(lastkey) format "x(20)" label "Teclas" at 51.
pause.
end.
```

KEYWORD

SINTAXE - KEYWORD(Texto) Caracter

Informa se um Texto equivale a uma expressão reservada Progress.

Exemplo

```
define variable nome as character format "x(20)".

repeat on error undo, retry:
    update nome.
    message keyword(nome) view-as alert-box.

    if keyword(nome) <> ? then do:
        message nome + " é uma palavra reservada Progress".
        undo, retry.
    end.
end.
```

KEYWORD-ALL

SINTAXE - KEYWORD-ALL(Texto) Caracter

Similar a função KEYWORD. Verifica se um Texto equivale alguma expressão reservada Progress considerando todas as abreviações possíveis dos comandos e funções.

LAST

SINTAXE - LAST(Campo de Agrupação) Lógico

Similar a função FIRST. Retorna um valor lógico verdadeiro da primeira interação de agrupamento, então

se é a última ocorrência em uma cláusula DO, FOR EACH, REPEAT com BREAK BY anterior é retornado como verdadeiro.

LASTKEY

SINTAXE - LASTKEY [retorno Inteiro]

Retorna um inteiro que representa o valor da última tecla ou combinação pressionada.

LAST-OF

SINTAXE - LAST-OF(Campo de Agrupação) Lógico

Retorna um valor lógico verdadeiro se é o última ocorrência do agrupamento em uma cláusula DO, FOR EACH, REPEAT com BREAK BY.

LC

SINTAXE - LC(Texto) Character

Converte um Texto para minúsculo.

LDBNAME

SINTAXE

LDBNAME(Inteiro ou Nome Lógico do Banco de Dados) Character

Retorna o Nome Lógico do Banco de Dados na sessão corrente.

LEFT-TRIM

SINTAXE - LEFT-TRIM(Texto) Caracter

Remove espaços em branco a esquerda de um Texto.

LENGTH

SINTAXE - LENGTH(Texto) Inteiro

Retorna o tamanho de um texto.

LINE-COUNTER

SINTAXE - LINE-COUNTER [retorna Inteiro]

Retorna o número da linha corrente para uma saída paginada.

Exemplo

```
output to value(session:temp-directory + 'teste.txt')
paged.
for each customer:
    display line-counter label "Nro"
        cust-num
        name
        address
        city
        state
        with width 620.
end.
```

LIST-EVENTS

SINTAXE

LIST-EVENTS(Handle do Objeto, [Plataforma]) Lista

Retorna uma Lista dos Eventos disponíveis de um Objeto. O parâmetro Plataforma pode ser setado com "GUI" para ambiente gráfico ou "TTY" para ambiente caracter.

Exemplo

```
display list-events(current-window) format "x(70)".
```

LIST-QUERY-ATTRS

SINTAXE - LIST-QUERY-ATTRS(Handle do Objeto) Lista

Retorna uma Lista dos Atributos disponíveis de um Objeto.

Exemplo

```
display list-query-attrs(current-window) format "x(70)".
```

LIST-SET-ATTRS

SINTAXE - LIST-SET-ATTRS(Handle do Objeto) Lista

Retorna uma Lista dos Atributos passíveis à atribuição de um Objeto.

Exemplo

```
display list-set-attrs(current-window) format "x(70)".
```

LIST-WIDGETS

SINTAXE**LIST-WIDGETS(Handle do Objeto, [Plataforma]) Lista**

Retorna uma Lista dos Objetos que contêm um determinado Evento. O parâmetro Plataforma pode ser

setado com "GUI" para ambiente gráfico ou "TTY" para ambiente caracter.

Exemplo

```
display list-widgets("return") format "x(70)".
```

LOCKED

SINTAXE - LOCKED Registro [retorno Lógico]

Retorna o estado de *lock* um registro.

Exemplo

```
If locked Nome da Tabela then...
```

LOG

SINTAXE - LOG(Valor, [Base]) Decimal

Obtém o Log de um Valor.

LOOKUP

SINTAXE**LOOKUP(Sub-texto, Lista, [Delimitador]) Integer**

Retorna a posição de um Sub-texto dentro de uma Lista. Caso o Delimitador não seja informado este será por padrão "," (vírgula).

Exemplo

```
define variable dia as character.
```

```
assign dia =
```

```
'Domingo,Segunda,Terca,Quarta,Quinta,Sexta,Sabado'.
```

```
message 'Terça é o dia' lookup('Terça', dia) 'da semana'  
view-as alert-box.
```

MAXIMUM

SINTAXE

MAXIMUM(Valor 1, Valor 2, Valor n, ...) Decimal

Retorna o Maior valor dos parâmetros passados.

Exemplo

```
/* Valor retorno 9188 */  
display maximum(293.4, 38, 1, 32, 9384, 9188).
```

MESSAGE-LINES

SINTAXE - MESSAGE-LINES [retorna Inteiro]

Retorna o número de Linhas disponíveis na área de mensagens.

Exemplo

```
define variable i as integer.  
  
do i = 1 to message-lines:  
    message "Esta é a linha" i.  
end.
```

MINIMUM

SINTAXE

MINIMUM(Valor 1, Valor 2, Valor n, ...) Decimal

Retorna o Menor valor dos parâmetros passados.

Exemplo

```
/* Valor retorno 1 */  
display minimum(293.4, 38, 1, 32, 9384, 9188).
```

MONTH

SINTAXE - MONTH(Data) Inteiro

Retorna o mês de uma data.

NEW

SINTAXE - NEW(Registro) Lógico

Informa se um registro ainda não foi escrito no Banco de Dados, apenas criado.

Exemplo

```
create customer.  
if new customer then  
    message "Novo registro" view-as alert-box.  
find first customer.  
if new customer then  
    message "Novo registro" view-as alert-box.
```

NEXT-VALUE

SINTAXE - NEXT-VALUE(Sequence) Decimal/Inteiro

Retorna o Próximo valor de uma sequência.

Exemplo

```
display next-value(next-cust-num).
```

NOT ENTERED

SINTAXE

[FRAME Nome do Frame] Campo NOT ENTERED

Informa se um campo de um frame foi modificado desde a última atualização.

Igual a função ENTERED

NUM-ALIASES**SINTAXE - NUM-ALIASES [retorno Inteiro]**

Retorna o número de Aliases dos Bancos de Dados estão definidos na sessão atual.

NUM-DBS**SINTAXE - NUM-DBS [retorno Inteiro]**

Retorna o número de Bancos de Dados conectados na sessão atual.

Exemplo

```
define variable i as integer.  
repeat i = 1 to num-dbs:  
    display ldbname(i)          format "x(20)"  
        pdbname(i)            format "x(20)"  
        dbrestrictions(i)    format "x(20)".  
end.
```

NUM-ENTRIES**SINTAXE - NUM-ENTRIES(Lista, [Delimitador]) Inteiro**

Retorna o número de elementos contidos em uma lista. Caso o delimitador não seja especificado o padrão assumido será ","(vírgula).

Exemplo

Escrito por Márcio Brener - <http://www.geocities.com/marcio-brener>

```
define variable i as integer.
define variable mes as character.

assign mes = 'Janeiro,Fevereiro,Marco,Abril,Maio,Junho,' +
'Julho,Agosto,Setembro,Outubro,Novembro,Dezembro'.

do i = 1 to num-entries(mes):
    display entry(i, mes) format "x(20)".
    pause.
end.
```

NUM-RESULTS

SINTAXE

NUM-RESULTS(Nome da Query [Caracter]) Inteiro

Retorna o número de registros de uma seleção.

Exemplo

```
define query q-customer for customer.
open query q-customer preselect each customer no-lock.

message 'Total da Selecao' num-results('q-customer')
view-as alert-box.
```

OPSYS

SINTAXE - OPSYS [retorno Caracter]

Retorna o nome do Sistema Operacional corrente.

Exemplo

```
display opsys.
```

OS-DRIVES

SINTAXE - OS-DRIVES [retorno Caracter]

Retorna uma Lista dos drives disponíveis na sessão atual.

Exemplo

```
define variable drives as character view-as combo-box.  
  
form drives with frame f-drives.  
assign drives:list-items = os-drives.  
update drives with frame f-drives.
```

OS-ERROR

SINTAXE - OS-ERROR [retorno Caracter]

Retorna o último erro de uma operação no Sistema Operacional. Retorna o status das funções os-append, os-copy, os-create-dir, os-delete, os-rename ou save cache.

Exemplo

```
os-create-dir value("c:\temp\teste.txt").  
  
if os-error NE 0 then  
    message 'Erro nº' os-error view-as alert-box.
```

OS-GETENV

SINTAXE - GETENV(Variável de Ambiente) Caracter

Retorna o valor de uma Variável de Ambiente.

Exemplo

```
display 'DLC=' to 10 os-getenv('DLC') format 'x(60)'  
          'PATH=' to 10 os-getenv('PATH') format 'x(60)'.
```

PAGE-NUMBER

SINTAXE - PAGE-NUMBER [retorno Inteiro]

Retorna o número da página em uma saída paginada.

Exemplo

```
/* ex0024.p */
output to 'teste.txt' paged page-size 40.
form header 'Pagina nº' page-number format '>>9'
    with frame f-cabecalho page-top.

for each customer no-lock:
    view frame f-cabecalho.
    display customer.cust-num
        customer.name
        customer.address.
end.
```

PAGE-SIZE

SINTAXE - PAGE-SIZE [retorno Inteiro]

Retorna o Tamanho da Página do scopo de impressão atual.

PDBNAME

SINTAXE

**PDBNAME(Inteiro ou Nome Lógico do Banco de Dados)
Caracter**

Retorna o Nome Físico do Banco de Dados na sessão corrente. Similar às funções LDBNAME, DBTYPE, DBRESTRICTIONS, etc

PROGRAM-NAME

SINTAXE - PROGRAM-NAME(Inteiro) Caracter

Retorna o Nome do Programa em execução segundo o número informado. O programa atual equivale a 1.

Exemplo

```
/* ex0025.p */
define variable i as integer initial 1.

repeat while program-name(i) <> ?.
    display i label 'Nro'
        program-name(i) label 'Programa' format "x(30)"
        with title "Ultimos Programas executados"
        centered row 2.
    assign i = i + 1.
end.
```

PROGRESS

SINTAXE - PROGRESS [retorno Caracter]

Retorno a configuração do Progress Full, Query ou Run-Time.

Exemplo

```
message 'Este Progress é' progress
view-as alert-box.
```

PROMSGS

SINTAXE - PROMSGS

Retorna o valor do PROMSGS atual.

PROPATH

SINTAXE - PROPATH [retorno Lista]

Retorna uma Lista dos diretórios contidos no PROPATH.

Exemplo

```
/* ex0026.p */

define variable diretorios as character
  view-as selection-list
  inner-chars 50 inner-lines 10 sort.

define frame f-diretorios diretorios
  with title 'Diretorios do Propath'
  centered row 5.

diretorios:list-items = propath.

prompt diretorios with frame f-diretorios.
```

PROVERSION

SINTAXE - PROVERSION [retorno Caracter]

Retorna a versão corrente do Progress.

QUERY-OFF-END

SINTAXE - QUERY-OFF-END(Nome da Query) Lógico

Retorna se é o fim da seleção da Query.

Exemplo

```
/* ex0027.p */
open query q-customer preselect each customer no-lock.

repeat:
  get next q-customer.
```

```

    if query-off-end("q-customer") then leave.

    display cust-num name.
end.

```

RANDOM

SINTAXE

RANDOM(Valor Mínimo, Valor Máximo) Inteiro

Retorna um valor aleatório entre um intervalo. Você pode configurar a variação com o parâmetro `-rand n` na sessão clientes.

Exemplo

```

/* ex0028.p */

/*
Este exemplo simula o preenchimento
de 100 jogos da Sena, baseados na
geracao de numeros aleatorios
*/

define variable i          as integer.
define variable j          as integer.

define variable njogos     as integer.
define variable combinacao as integer extent 6.

define temp-table jogos
    field n as integer label 'Jogo nº'
    field n1 as integer format '99'
    field n2 as integer format '99'
    field n3 as integer format '99'
    field n4 as integer format '99'
    field n5 as integer format '99'
    field n6 as integer format '99'
    index jogo is primary unique n1 n2 n3 n4 n5 n6.

combinacao:
do while njogos < 100:

```

```
do i = 1 to 6:
    assign combinacao[i] = random(0, 60).
end.

do i = 1 to 6:
    if (combinacao[i] = combinacao[1] and i <> 1) or
       (combinacao[i] = combinacao[2] and i <> 2) or
       (combinacao[i] = combinacao[3] and i <> 3) or
       (combinacao[i] = combinacao[4] and i <> 4) or
       (combinacao[i] = combinacao[5] and i <> 5) or
       (combinacao[i] = combinacao[6] and i <> 6) then
        next combinacao.
    end.

find jogos where
    jogos.n1 = combinacao[1] and
    jogos.n2 = combinacao[2] and
    jogos.n3 = combinacao[3] and
    jogos.n4 = combinacao[4] and
    jogos.n5 = combinacao[5] and
    jogos.n6 = combinacao[6] no-error.

if available jogos then next combinacao.

assign njogos = njogos + 1.

create jogos.
assign jogos.n = njogos
    jogos.n1 = combinacao[1]
    jogos.n2 = combinacao[2]
    jogos.n3 = combinacao[3]
    jogos.n4 = combinacao[4]
    jogos.n5 = combinacao[5]
    jogos.n6 = combinacao[6].

end.

for each jogos by jogos.n:
    display jogos.
end.
```

RECID

SINTAXE - RECID(Registro) Recid

Retorna o Recid do Registro.

Exemplo

```
/* ex0029.p */

define buffer b-customer for customer.
for each customer no-lock
    by customer.cust-num descending:

    find b-customer where
        recid(b-customer) = recid(customer) exclusive-
lock.

    update b-customer.cust-num.

    display customer.cust-num
        b-customer.cust-num.
end.
```

RECORD-LENGTH

SINTAXE - RECORD-LENGTH(Registro) Inteiro

Retorna a quantidade de bytes em um registro.

Exemplo

```
for each customer no-lock:
    display record-length(customer) 'Bytes'.
end.
```

REPLACE

SINTAXE

**REPLACE(Texto, Sub-Texto Origem, Sub-Texto Destino)
Caracter**

Substitue um sub-texto de um Texto por outro especificado.

Exemplo

```
/* ex0030.p */

/* Remover caracteres invalidos de um texto */

define variable i          as integer.
define variable procura    as character.
define variable substituto as character.

assign procura    = "á,Á,é,É,í,Í,ó,Ó,ú,Ú,ç,Ç,ê,Ê,ü,Ü"
        substituto = "a,A,e,E,i,I,o,O,u,U,c,C,e,E,u,U".

for each customer:
    update customer.name label 'Coloque um caracter
invalido'.
    display customer.name.
    do i = 1 to num-entries(procura):
        if index(customer.name, entry(i, procura)) > 0 then
            customer.name = replace(customer.name,
                                    entry(i, procura),
                                    entry(i, substituto)).
        display customer.name.
    end.
end.
```

RETRY**SINTAXE - RETRY [retorno Lógico]**

Retorna um Lógico para um bloco de transação se este foi concluído com êxito.

Exemplo

```
repeat:
    prompt-for customer.cust-num.
    find customer using cust-num.

    if not retry then
        display name
        address
        city
        state
```

```
        country.  
    else  
        clear all no-pause.  
    set name address city state country.  
    if country = "" then undo, retry.  
end.
```

RETURN-VALUE

SINTAXE - RETURN-VALUE [retorno Caracter]

Informa o valor de retorno de uma Procedure ou Função executada.

Exemplo

```
run teste.  
display return-value.  
pause.  
procedure teste:  
    return 'Exemplo de Retorno'.  
end.
```

RIGHT-TRIM

SINTAXE - RIGHT-TRIM(Texto) Caracter

Remove espaços em banco de um Texto à direita.

R-INDEX

SINTAXE - R-INDEX(Texto, Sub-Texto) Inteiro

Retorna a posição de um sub-texto em um texto à direita.

Exemplo

```
/*  
    Retorno igual a 7, ou seja,  
    a terceira letra A  
*/
```

```
display r-index("Amazonas", "a").
```

ROUND

SINTAXE - ROUND(Valor, Precisão) Decimal

Arredonda um valor em uma precisão decimal informada.

Exemplo

```
/* Retorno igual a 18379.99 */  
display round(18379.987, 2).
```

ROWID

SINTAXE - ROWID(Registro) Rowid

Retorna o valor do Rowid do Registro, implementada a partir da versão 8.1 do Progress. Semelhante a função RECID porém mais sofisticada por usar formação hexadecimal ao contrário de RECID que é decimal, ampliando o valor de abrangência e precisão.

SCREEN-LINES

SINTAXE - SCREEN-LINES [retorno Inteiro]

Informa o número de linhas suportado na sessão atual.

SDBNAME

SDBNAME(Inteiro ou Nome Lógico do Banco de Dados) Caracter

Retorna o Schema do Dataserver do Banco de Dados conectado na sessão corrente. Similar às funções LDBNAME, PDBNAME, DBRESTRICTIONS, etc.

SEARCH

SINTAXE - SEARCH(Texto) Caracter

Retorna o caminho completo de um arquivo utilizando como base os diretórios contidos no PROPATH.

Exemplo

```
/* Editor Progress em... */  
display search("_edit.r") format "x(60)".
```

SEEK

SINTAXE - SEEK(INPUT/OUTPUT) Inteiro

Retorna a posição do texto/cursor de uma saída ou entrada de dados.

SETUSERID

SINTAXE

SETUSERID(Usuário, Senha, Nome Lógico do Banco) Lógico

Atribui/Conecta um usuário no Banco de Dados. A senha deve ser passada literalmente igual ao seu valor original, considerando também maiúsculas e minúsculas.

Exemplo

```
display setuserid("adm", "adm", "sports").
```

SQRT

SINTAXE - SQRT(Valor) Decimal

Retorna a raiz de um valor.

Exemplo

```
define variable valor as decimal.
```

```
update valor.
```

```
message 'A raiz de' valor 'é' sqrt(valor)  
view-as alert-box.
```

STRING

SINTAXE - STRING(Valor, Formato) Caracter

Transforma um valor em caracter com formato indicado.

O valor atribuído deve ser de tipo Inteiro, Decimal, Data ou Lógico

Exemplo

```
display string(today, "99/99/9999")    format "x(20)" skip  
      string(1293.21, "999,999.99") format "x(20)" skip  
      string(yes, "Sim/Nao")          format "x(20)".
```

SUBSTITUTE

SINTAXE

SUBSTITUTE(Texto, n Argumentos [Caracter]) Caracter

Substitue/Insere em um caracter um subtexto em uma posição de argumento definida na forma "&n", onde n é inteiro de 1 a 9, e os argumentos restantes são caracteres.

Exemplo

```
/* ex0009.p */  
message substitute("&1 Amor, &2 Beijcos, &3 Coracao, &4 Desejos",
```

```
"A", "B", "C", "D") view-as alert-box.
```

```
define buffer b-customer for customer.

for each customer no-lock:
    find first b-customer where
        rowid(b-customer) > rowid(customer) no-lock no-
error.
    clear all.
    display SUBSTITUTE("&1 é Cliente Atual. &2 é Cliente
Posterior",
                                customer.name, b-customer.name)
    format "x(70)" when available b-customer.

end.
```

SUBSTRING

SINTAXE

SUBSTRING(Texto, Início, Tamanho [Character], Tipo [Character]) Character

Extrai uma subcadeia de um texto.

O comando SUBSTRING também pode indicar uma substituição, conforme o exemplo mostrado abaixo.

O parâmetro **Tipo** aceita três indicações "character", "fixed" ou "raw", onde:

Character é o valor padrão se omitido este quarto parâmetro;

Fixed indica que a posição início será um inteiro e o tamanho será em bytes;

Raw indica que a posição início e o tamanho são bytes.

Exemplo

```
/* ex0008.p */
define variable texto as character.
assign texto = "Powered by Progress".
```

```
display substring(texto, 1, 50) format "x(50)".  
pause.
```

```
assign substring(texto, 12) = "Aprendendo Progress 4GL".
```

```
display substring(texto, 1, 50) format "x(50)".  
pause.
```

TERMINAL

SINTAXE - TERMINAL retorno Caracter

Retorna um caracter que representa o Terminal/Ambiente corrente.

Ambiente X-Windows interface caracter o TERMINAL retorno será BW80, CO80, ou MONO, dependendo do monitor. MS-Windows o TERMINAL retorno é WIN3. UNIX o TERMINAL retorno será a variável de ambiente \$TERM, em modo batch o TERMINAL retorna um valor nulo (?).

Exemplo

```
message "Voce esta usando um terminal" terminal view-as  
alert-box.
```

TIME

SINTAXE - TIME retorno Inteiro

Retorna o número de segundos à partir de meia-noite.

Exemplo

```
/* número de segundos a partir de 00:00h */  
message time view-as alert-box.  
/* horas a partir de 00:00h */  
message string(time, "hh:mm:ss") view-as alert-box.
```

TODAY

SINTAXE - TODAY retorno Data

Retorna a data atual.

Exemplo

```
message today view-as alert-box.
```

TO-ROWID

SINTAXE - ROWID(Valor tipo Character) Rowid

Converte um número equivalente à posição de uma tabela em formato caracter hexadecimal e o converte em tipo Rowid.

O texto de representação do Rowid deve estar em forma hexadecimal "0xhex-dígitos", exemplo 0x000000, onde os caracteres hexadecimais estão de 0 até 9 ou A até F.

Exemplo

```
/* ex0007.p */
define buffer b-customer for customer.
for each customer no-lock:
    find b-customer where
        rowid(b-customer) = to-rowid(string(rowid(customer))).

    display string(rowid(b-customer)) label "Hexadecimal"
        b-customer.cust-num.
end.
```

TRANSACTION

SINTAXE - TRANSACTION retorno Lógico

Indica se um bloco de código é uma transação

Exemplo

Escrito por Márcio Brener - <http://www.geocities.com/marcio-brener>

```
/*  
A primeira vez e nao sera uma transacao  
e as restantes serão  
*/  
do:  
    message transaction view-as alert-box.  
    pause.  
    repeat transaction:  
        message transaction view-as alert-box.  
        pause.  
    end.  
end.
```

TRIM

SINTAXE - TRIM(Texto) Caracter

Remove espaços em branco à esquerda e direita de um texto.

TRUNCATE

SINTAXE

TRUNCATE(Valor Decimal, n Casas Decimais) Inteiro

Retorna o valor inteiro de uma valor decimal.

Exemplo

```
/* Retorno igual a 128,2 */  
message truncate(128.23, 1) view-as alert-box.  
  
/* Retorno igual a 1539,29834 */  
message truncate(1539.29834699891, 5) view-as alert-box.
```

USERID

SINTAXE

USERID(Nome Lógico do Banco de Dados) Caracter

Informa o usuário atual logado no Banco de Dados.

Exemplo

```
message "seu login é" userid("sports") view-as alert-box.
```

VALID-EVENT**SINTAXE**

VALID-EVENT(Handle do Objeto, Nome do Evento, [Plataforma]) Lógico

Verifica se um objeto contém um evento válido.

Exemplo

```
display valid-event(current-window, "close") skip.  
display valid-event(current-window, "window-close").
```

VALID-HANDLE

SINTAXE - VALID-HANDLE(Handle do Objeto) Lógico

Verifica se um valor é um handle válido.

WEEKDAY

SINTAXE - WEEKDAY(Data) Inteiro

Retorno o dia da semana em inteiro.

Exemplo

```
/* ex0006.p */  
define variable dia as character extent 7  
    initial["Domingo", "Segunda", "Terca", "Quarta",  
           "Quinta", "Sexta", "Sabado"].
```

```
message dia[weekday(today)] view-as alert-box.
```

WIDGET-HANDLE

SINTAXE - WIDGET-HANDLE(Character) Handle

Cria um handle a partir de um número em formato caracter.

Exemplo

```
define variable whand as widget-handle.  
assign whand = widget-handle("3000245").
```

YEAR

SINTAXE - YEAR(Data) Inteiro

Retorna o ano de uma data

Exemplo

```
/* Retorna o ano atual */  
message year(today) view-as alert-box.
```

Capítulo n

Utilitários & Parâmetros do Banco de Dados Progress

Utilitários do Banco de Dados Progress

Os utilitários do banco de dados Progress têm finalidades para a manutenção do banco como status, estrutura física e lógica, backup, log, monitoração, entre outros como mostrado a seguir.

PRO

Inicia uma sessão Progress mono-usuário

SINTAXE

PRO [Database] [Parâmetros]

Muitos programas/utilitários Progress são apenas arquivos batchs que pré-configuram o ambiente e/ou passam parâmetros para o Progress de acordo com o utilitário. Como por exemplo o comando PRO mencionado acima.

Exemplo:

```
Arquivo: PRO.BAT
@echo off
```

```
if "%DLC%"==" " set DLC=C:\ARQUIV~1\PROGRESS
```



```
if exist %DLC%\promsgs goto BIN
    echo DLC environment variable not set correctly - Please
set DLC variable
goto END

:BIN
if not "%PROEXE%"==" " goto START
set PROEXE=_progres

:START
if "%DISPBANNER%"=="no" goto NOBANNER
type %DLC%\hello

:NOBANNER
type %DLC%\version
%PROEXE% -l -p _prostar.p %1 %2 %3 %4 %5 %6 %7 %8 %9

set DISPBANNER=
:END
```

PROAIW

Inicia o After-Image-Write (AIW)

SINTAXE

PROAIW [Database]

Inicia o processo do After-Image Writer (AIW) em background. O AIW melhora a performance para a memória compartilhada do banco de dados multi-usuário no arquivo After Image.

Nota: apenas para bancos de dados com After Image.

PROBIW

Inicia o Before-Image-Write (BIW)

SINTAXE

PROBIW [Database]

Inicia o processo do Before-Image Writer (AIW) em background. O BIW melhora a performance para a memória compartilhada do banco de dados multi-usuário no arquivo Before Image.

PROBKUP

Cria um arquivo de backup do banco de dados Progress

SINTAXE

```
probkup [online] Database [incremental]  
        Dispositivo/Diretório  
        [-vs n] [-bf n] [-verbose] [-scan] [-io n] [-com]  
        [-red n] [-estimate] [-g Before-Image]
```

PARÂMETROS

online

Indica que o backup é um backup on-line.

Database

Informa o Banco de Dados a ser salvo.

incremental

Informa que o backup será incremental, ou seja, apenas os dados ainda não gravados no arquivo de backup.

Dispositivo

Informa o dispositivo destino de backup, como um disquete, diretório no disco rígido, fita DAT ou DLT.

-estimate

Indica que o backup obterá uma média estimada apenas. Use o parâmetro scan quando usar o parâmetro incremental e/ou compression.

Aplicável apenas para backup off-line.

-vs n

Indica o tamanho em blocos do dispositivo destino do backup. Quando alcançado o número de blocos especificado será solicitado o dispositivo ou arquivo seguinte do próximo volume.

Caso este parâmetro não seja especificado o PROBACKUP assumirá que a mídia comporta todo o conteúdo do backup.

-bf n

Indica a blocagem para o arquivo destino do backup. O valor padrão é 34.

-verbose

Habilita a visualização do progresso do backup durante o backup.

-scan

Informa ao PROBKUP para obter o número de blocos, avaliar o espaço necessário e visualizar estas informações. Apenas para backup off-line.

-io i

Especifica um incremento de overlap. O padrão é 0.

-com

Informa que o backup será compactado.

-red i

Informa a redundância do backup. Isso evita problemas quando restaurado o backup e o dispositivo apresentava mau setores na mídia.

-g Before-Image

Especifica a localização do(s) arquivo(s) de Before-Image.

PROCOPY

Copia um banco de dados para outro banco de dados.

SINTAXE

procopy Database origem Database destino -g Localização do Before-Image -s

Exemplo:

Banco de dados em /bases/magnus/i00adm.db

procopy /bases/magnus/i00adm /bases/teste/i00adm

PRODB

Cria um novo banco de dados vazio ou a partir de um já existente.

SINTAXE

prodb [Database] {empty|sports|Database
Origem|demo}

PRODEL

Apaga um banco de dados Progress

SINTAXE

prodel Database

Parâmetros do Progress

A configuração do Progress, sessão cliente e carga dos bancos de dados são ajustadas através de combinações de diversos parâmetros.

| Parâmetro | Descrição |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -1 | Single-user Mode (Cliente) Conecta um banco de dados em modo mono-monousuário Exemplo (No arquivo de parametros .pf) -db sports -1 (Codigo) connect -db sports -1. |
| -a arquivo | After-image Filename (Servidor/Cliente) Especifica o arquivo de After-Image a ser utilizado na conexão |
| -aibufs n | After-Image Buffers (Servidor) Especifica o número de blocos em buffer de memória para o arquivo After-Image quando executado o utilitário AIW que |

coordena as transação de imagem posterior

Mínimo: 1

Padrão multi-usuário: 1

-aistall

After-image Stall (Servidor)

Suspende a atividade de um banco de dados multi-volumes usando After-Image para o evento de shutdown e grava as informações no arquivo de log

-b

Batch (Cliente)

Indica que uma sessão será executada em batch sem interação. Para executar uma aplicação em batch esta não deve apresentar nenhuma informação em tela, pois ocasionara um erro

**-basekey
texto**

Registry Basekey (Cliente MS-Windows)

Indica ao Progress para usar a configuração contida em uma arquivo INI, com o parâmetro "INI". Isso porque o Progress versão 8.x utilização configuração contida na registry

Os parâmetros disponíveis são:

**HKEY_CURRENT_USER
HKEY_CLASSES_ROOT
HKEY_LOCAL_MACHINE
HKEY_USERS
HKEY_CURRENT_CONFIG**

**HKEY_DYN_DATA
INI****Exemplo**

C:\DLC\BIN\PROWIN32.EXE –basekey
"INI" –ininame ARQUIVO.INI

-B n**Blocks in Database Buffers
(Servidor/Mono-usuário)**

Define o número de blocos a serem colocados em buffer de memória para cada usuário

Mínimo: 10**Máximo: 500.000****Padrão mono-usuário: 20****Padrão multi-usuário: (8 x n
Usuários)**

Caso o número de buffers ultrapasse a quantidade de memória física da máquina esta pode travar ou reiniciar!

-bibufs n**Before-image Buffers (Servidor)**

Especifica o número de blocos em buffer de memória para o arquivo After-Image Quando executado o utilitário BIW que coordena as transação de imagem anterior

Mínimo: 3**Padrão multi-usuário: 5**

| | |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -brl | Bleeding Record Lock (Cliente) Deixa a critério do Progress o tratamento de registros em lock compartilhado (share lock), utilizando mais de um buffer para registros em lock, mesmo que estes já não estejam mais em uso compartilhado |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -Bt n | Buffer Size for Temporary Tables (Cliente) Especifica o número de blocos em buffer para tabelas temporárias durante a sessão cliente Mínimo: 10 Máximo: 50.000 |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -C qualificador | Option on PROUTIL and RFUTIL Commands (ProUtil/RFUtil) Usado com os utilitários Progress ProUtil e RFUtil. Consulte o capítulo Utilitários de banco de dados |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -c n | Index Cursors (Cliente) Especifica no número de cursores de índices a serem utilizados na sessão para uso de declarações FIND, FOR EACH, QUERY e SELECT. Em sintaxe Progress FIND e FOR EACH cada loop consome apenas um cursor de |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

64 bytes. Geralmente para uso de programas utilizando linguagem SQL é necessário fazer ajuste desse parâmetro devido ao result set criado pela especificação SQL.

**-cache
arquivo**

Schema Cache File (Cliente)

Força o Progress a usar um arquivo binário que contém a estrutura do metaschema do banco de dados conectado.

Exemplo

(Arquivo de parâmetros .pf)

-db sports

-1

- cache sports.csh

**-charset
Código de
Página**

Character Set (Servidor/Cliente)

Define o Código de Página a ser utilizado na sessão para manipulação de dados.

Padrão: iso8859-1

-checkdbe

**Check Double-byte Enabled
(Servidor/Cliente)**

Habilita o suporte a caracteres double-byte para a versão Progress 7, utilizado nas funções **LENGTH**, **SUBSTRING**, **SUBSTRING**.

-convmap

Conversion Map (Servidor/Cliente)

arquivo

Define o arquivo de Código de Páginas a ser utilizado. Caso seja omitido este parâmetro o Progress usará por padrão o arquivo **convmap.cp** localizado no diretório do Progress.

**-cp
parâmetros
ou arquivo****Communications File
(Servidor/Cliente)**

Define um arquivo de Auto-Conexão que altera as conexões atuais.

As configurações especificadas pelo parâmetro -cp pode ser substituída pela configuração do metaschema na tabela _Db campo _Db-comm.

**-cpcase
tabela****Case Code Page (Servidor/Cliente)**

Especifica uma tabela utilizada para conversão entre caracteres maiúsculos e minúsculos e vice-versa, usado nas funções CAPS, LC ou character "!" da cláusula FORMAT.

A tabela está especificada dentro do arquivo de Códigos de Páginas **convmap.cp**.

**-cpcoll
coleção****Collation Code Page
(Servidor/Cliente)**

Especifica uma coleção de tabelas de Códigos de Páginas a ser utilizado para procedimento em memória.

Este parâmetro funciona em conjunto com o parâmetro `-xc` que define a utilização de alfabetos.

Para maiores detalhes consulte o manual PROGRESS Internationalization Guide.

-cpdb Código
Página

**Database Code Page
(Servidor/Cliente)**

Especifica ao banco de dados o Código de Página a ser utilizado para gravar as informações

Padrão: ibm850

-cpinternal
Código
Página

**Internal Code Page
(Servidor/Cliente)**

Especifica o Código de Página a ser utilizado na manipulação dos dados em memória pelo Progress

Padrão: iso8859-1

-cpprint
Código
Página

Print Code Page (Servidor/Cliente)

Especifica o Código de Pagina a ser usada em um escopo de impressão (OUTPUT)

**Padrão: o mesmo valor definido em
-cpstream**

-cprcodein
Código

**R-code In Code Page
(Servidor/Cliente)**

| | | |
|------------------------------------------------------|----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Página | | Especifica o Código de Página a ser usado na nos arquivos compilados .r |
| -cprcodeout Código Página | R-code Out Code Page (Servidor/Cliente) | Especifica o Código de Página a ser usado no scopo de impressão pelas procedures compiladas. |
| -cpstream | Stream Code Page (Cliente/Servidor) | Especifica o Código de Página a ser usado para o escopo de entrada/saída de dados (stream i/o), comandos INPUT FROM e OUPUT TO. |
| -cpterm | Terminal Code Page (Cliente/Servidor) | Especifica o Código de Página do terminal modo caracter (UNIX,DOS,Novell). |
| -cs n, i | Cursor Size (Cliente) | <p>Define o tamanho de um cursor para um índice específico, onde:</p> <p>n bytes e i número do índice</p> <p>Em índices demasiadamente grandes às vezes pode ocorrer um erro de cursor no índice, principalmente para índices tipo WORD-INEX, como abaixo:</p> |

Index index-number has at least n levels.
Increase -cs parameter

-ct n

AS/400 Connection Time (Cliente)

Define um **timeout**, em segundos, para resposta de uma query em um acesso ao AS/400 IBM.

-d mdy

Date Format (Cliente)

Define o formato para entrada e apresentação de datas, onde:

d dia, **m** mês, **y** ano

Caso não seja especificado o padrão será o formato americano Mês/Dia/Ano.

-D n

Directory Size (Cliente)

Define o número de procedures por diretório aonde serão compiladas e sobrepostas as procedures executáveis.

Quando o Progress compila uma procedure e a grava no disco ele mantém a versão anterior, caso não seja possível gravá-la no disco, por questões de espaço ou acesso. Então ele mantém a procedure compilada anterior e gera um erro de acesso.

Você pode monitorar a utilização de entradas de diretório através das estatísticas geradas pelo parâmetro (**-y**).

| | |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | Mínimo: 5 Máximo: 500 Padrão mono-usuário: 100 Padrão multi-usuário: 100 |
| -db banco de dados | Physical Database Name (Cliente) Define o nome físico do banco de dados a ser conectado. |
| -debug | Debugger (Cliente) Executa o Progress em mode Debug. |
| -decdtm | Distributed Transaction Manager (Cliente) Usa serviço de transações distribuídas DETdtm em ambiente VMS para performance de transações em filas de processos. É necessário está instalado o DETdtm para funcionar este serviço. |
| -dictexps | Dictionary Expressions (Cliente) Força uma procedure compilada a usar a validação do dicionário de dados, mesmo quando há alguma declaração nó código fonte ignorando às validações ou as desabilitando. No código fonte pode-se desabilitar esta especificação por frame utilizando o termo NO-VALIDATE na declaração do |

-directio**Direct I/O (Servidor)**

Força um conexão ao banco de dados com acesso de leitura e gravação direto do disco (i/o), ignorando o controle do sistema operacional.

Este parâmetro aumenta significativamente a performance, devido ao acesso direto e eliminação de competição de outros aplicativos, porém põe em risco a integridade dos dados.



Esse procedimento pode danificar a base de dados caso alguma falha de hardware ou software aconteça.

-dt**Database Type (Cliente)**

Informa o tipo de banco de dados a ser conectado.

Tipos disponíveis: PROGRESS, ODBC, ORACLE, RMS, Rdb, CISAM, DB2 e Sybase, sem distinção entre maiúsculas e minúsculas.

Padrão: Progress

-E**European Numeric Format (Cliente)**

Define o formato para entrada e apresentação de valores numéricos. Ponto separa milhares e vírgula separa decimais.

| | |
|-----------------|--------------------------------------------------------------------------|
| -esqllog | ESQL Log (Cliente) |
| | Habilita log de transações para bancos de dados não Progress conectados. |

| | |
|-------------------|---------------------------------------------------------|
| -esqlnopad | ESQL No Padding (Cliente) |
| | Define SQL dinâmico em conjunto com préprocessamento C. |

| | |
|--------------------------------|------------------------------------------------------------------|
| -evtlevel parâmetro | Event Level (Cliente) Windows NT |
| | Define os eventos a serem gravados pelo Event Log do Windows NT. |
| | None não grva nenhum evento. |
| | Brief apenas alertas e erros. |
| | Normal alertas, erros e qualquer outro evento normal. |
| | Full todas mensagens. |

| | |
|-----------|------------------------------------------------------------------------------------------------------|
| -F | Force Access (Servidor) |
| | Força o acesso ao banco de dados em estado de recovery (recuperação) para um shutdown de emergência. |

| | |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| -fc n | Schema Field Cache Size (Cliente) |
| | O esquema de campo em cache reduz a compilação de instruções SQL por uma pré-alocação dos campos em memória. |
| | Use sempre o máximo de campos em cache para otimizar instruções. Por padrão o Progress consome 150 bytes adicionais da memória por esquema de |

campo usado em cache, e novas solicitações/uso de campos serão obtidas deste cache. Valor 0 (zero) desabilita campos em cache.

Mínimo: 0

Padrão: 128

-fldisable

Field List Disable (Cliente)

Ignora a lista de campos contidos em uma procedure compilada e obtém uma nova lista de campos da tabela a partir do metaschema.

-g arquivo

Before-image Filename (Servidor)

Especifica o arquivo de Before-Image a ser utilizado na conexão do banco de dados.

-G n

Before-Image Cluster Age (Servidor)

Define os segundo para reutilização de cluster do Before-Image.

Mínimo: 60

-h n

Number of Databases

Limita o número de banco de dados que podem ser conectados a sessão.

Mínimo: 1

Máximo: 240

Padrão: 5

-H servidor**Host Name (Servidor)**

Define o nome/alias do servidor aonde está o banco de dados compartilhado para acesso multi-usuários.

O arquivo **hosts** define a lista de IPs e aliases disponíveis para conexão.

Em Windows 95/98 a localização do arquivo hosts é **C:\WINDOWS\HOSTS**, em Windows NT **C:\WINNT\SYSTEM32\DRIVERS\ETC\HOSTS**, em Unix **\etc\hosts** (Segundo os diretórios padrões de instalação de cada SO mencionado).

-H servidor**AS/400 Host Name (Servidor)**

Define o nome do servidor para uma sessão multi-usuário em AS/400.

-hardlimit**Hardlimit (Servidor)**

Habilita o excesso dos limites para parâmetros que têm, por padrão, reduzido consumo de recursos, incrementando os valores de parâmetros de Directory Entries (**-D**), Local Buffer Size (**-l**), Maximum Memory (**-mmax**) e Nested Blocks (**-nb**).

-hs s**Heap Size (Servidor)**

Incrementa o espaço da memória compartilhada pelo banco de dados.

Este parâmetro é relevante apenas em Sistema Operacional V UNIX.

Mínimo: 1

Máximo: 2000

Padrão de acordo com o Sistema Operacional

-i

No Crash Protection

Grava e deleta informações diretamente do banco de dados sem controle de transações aumentado consideravelmente a performance, ou seja, habilita apenas a gravação de alguns dados no Before-Image para controle, porém é altamente desaconselhável.

É possível desfazer algumas transações desde que estas estejam em bloco definido por TRANSACTION e seus dados estejam ainda em cache do cliente.

Ao iniciar uma sessão sem proteção contra crash é tornada uma mensagem de alerta sobre a integridade do banco de dados.



O banco de dados pode perder sua integridade não sendo mais possível reestabelecê-la para o conteúdo inicial.

-iname

Initialization File (Cliente)

arquivo

Informa ao Progress o arquivo INI a ser utilizado na configuração da sessão.

Use com o parâmetro **-basekey "ini"** para ignorar os dados gravados na registry do Windows.

-inp n**Input Characters (Cliente)**

O número de caracteres permitidos em uma única declaração.

Padrão: 4096 caracteres

-is**AS/400 Ignore Stamp (Cliente)**

Ignora uma conexão a um AS/400 se o arquivo cache contém informações diferentes da estrutura do metaschema atual.

-k**Keyword Forget List**

Força o Progress a aceitar declarações cujo nome seja uma palavra reservada da linguagem. Ele retornará uma mensagem de alerta, mas continuará a execução.

Exemplo

Um programa desenvolvido em versão 6 com uma variável cujo nome é uma palavra reservada na versão 8 ocasionará um erro de sintaxe. Porém você pode remediar este problema com o parâmetro **-k** que ignorará esta

redundância.

-L n**Lock Table Entries (Servidor)**

Define o número de registros que podem ser travados.

Cada registro acessado com a cláusula SHARE-LOCK ou EXCLUSIVE-LOCK requisita um travamento individual, caso seja necessário o lock além do valor especificado é retornada uma mensagem de erro à sessão cliente, porém a sessão multi-usuário desfaz todas as transações e mantém o processamento.

Mensagem de erro

SYSTEM ERROR: Record lock table too small. Increase -L parameter

Cada registro em lock na tabela consome 14 bytes da memória não compartilhada e cada requisição requer 18 bytes da memória compartilhada da sessão multi-usuário.

Caso você especifique um valor que não seja múltiplo de 32 este valor informado será arredondado para o valor múltiplo mais próximo do indicado.

-ld nome**Logical Database Name (Cliente)**

Informa à sessão o nome lógico a ser atribuído para o banco de dados

conectado.

**-lng
linguagem**

Language (Cliente)

Define a linguagem a ser utilizada pela sessão.

As linguagens disponíveis depende da configuração do Progress instalado. Tipicamente as linguagens instaladas são inglês internacional, inglês americano, espanhol, alemão e francês.

**-logfile
arquivo**

LogFile (Cliente)

Especifica o arquivo de log a ser criado/usado pelo AppServer.

**-logname
arquivo**

Logname (Cliente)

Especifica o arquivo de log a ser criado/usado pelo AppServer

-m1

Auto Server (Servidor)

Inicia a sessão multi-usuário definindo um auto-servidor. O serviço multi-usuário usa o servidor para iniciá-lo automaticamente. Porém, este serviço já é utilizado por padrão, dispensando a passagem deste parâmetro para a sessão.

-m2

Manual Server (Servidor)

Inicia a sessão multi-usuário definindo manualmente um serviço remoto após o

| | |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 248 | Dominando o Progress |
| | início da sessão multi-usuário. |
| -m3 | Secondary Login Broker (Servidor) Especifica um terceiro serviço a ser usado para a sessão multi-usuário. |
| -Ma n | Maximum Clients Per Server (Servidor) Define o número máximo de usuários remotos por sessão multi-usuário. O padrão utilizado será o parâmetro -n se especificado, dividido pelo número máximo de servidores, parâmetro -Mn . |
| -maxport n | Maximum Dynamic Server (Servidor/Cliente) Define o intervalo máximo de portas disponíveis para uma sessão cliente. Os parâmetros -maxport e -minport definem uma faixa específica de portas disponíveis para proteção de um firewall. Geralmente o intervalo disponível é 32768 à 65535. Mais informações no capítulo Segurança de dados e aplicação. |
| -Mf n | Delayed BI File Write (Servidor) Define um intervalo em segundo para a gravação das informações em buffer (Before-Image) para o banco de dados. |

Este intervalo não põe em risco a integridade dos dados, porém em caso de erro os dados não armazenados no intervalo definido poderão ser perdidos.

Mínimo: 0

Máximo: 32768

Padrão mono-usuário: 0

Padrão multi-usuário: 0

-Mi n

**Minimum Clients Per Server
(Servidor)**

Define o número mínimo de usuários remotos por sessão multi-usuário.

-minport n

Minimum Dynamic Server (Servidor)

Define o intervalo mínimo de portas disponíveis para uma sessão cliente. Veja também o parâmetro **-maxport**.

-Mm n

**Message Buffer Size
(Servidor/Cliente)**

Define a quantidade em bytes para o buffer de mensagens padrão.

-mmax n

Maximum Memory (Cliente)

Define um valor inicial em kilobytes para alocar procedures compiladas.

-Mn

Maximum Servers (Servidor)

Mínimo: 1

Máximo: 512

Padrão multi-usuário: 4

-Mp n**Servers Per Protocol (Servidor)**

Define o número de servidores que podem ser iniciados por usuários remotos com qualquer protocolo.

-Mpb n**Maximum Servers Per Protocol (Servidor)**

Define o número máximo de servidores por protocolo.

-Mr n**Record Buffer Size (Cliente)**

Define o tamanho em bytes para o buffer de registros.

Mínimo: 96**Máximo: 32.000****Padrão multi-usuário: 1012**

-Mv n**Maximum Open Files (Servidor/Cliente)**

Define o número máximo de arquivos abertos para o Sistema PTX.

-Mxs n**Shared Memory Overflow Size (Servidor)**

Configura o tamanho da memória compartilhada em overflow em kilobytes

Este parâmetro substitue os valores padrões atuais e realoca a memória compartilhada.

Mínimo: 1
Máximo: memória física disponível
Padrão multi-usuário: 15 kilobytes
+ (n * 300)

-n n

Number of Users (Servidor)

Define o número máximo de usuário para conexão ao banco de dados.

-N protocolo

Network Type (Cliente)

Define o tipo de protocolo/rede para conexão ao banco de dados.

Os protocolos disponíveis pelo Progress são: TCP, DECnet, NETBIOS, TLI, DDE (Windows e Releases 7.3A ou superiores), WIPC (Windows e Release 7.3B ou superiores), SNA, LFP, e CTOSCLUSTER. EmVMS apenas protocolos TCP e DECnet.

-nb n

Nested Blocks (Cliente)

Define o número de blocos solicitados para a execução de procedures.

-Nd device

Network Device (Servidor/Cliente)

Especifica o caminho/localização de um device a ser utilizado para comunicação entre processos.

Aplicável apenas para UNIX System V.3.

-NL**No Lock (Cliente)**

Define NO-LOCK para ao comando FIND e FOR EACH como padrão, caso estes não tenham nenhuma declaração de LOCK já especificada.

É aplicável apenas por procedures compiladas por uma sessão iniciada com o parâmetro -NL, para procedure não compiladas ou pré-compiladas é assumido o padrão NO-LOCK.

-Nn Cliente**PC Client Name (Cliente)**

Identifica o nome do computador cliente para a conexão do banco de dados.

-Nv n**Network Version (Cliente)**

Especifica o número da versão do protocolo de conexão. Este parâmetro é passado do cliente para o servidor para assegurar compatibilidade.

**-o
impressora****Printer (Cliente)**

Define o nome/device da impressora a ser utilizado em um scopo de impressão.

Exemplo:

- o "COM2" (Windows)
- o "ls -s" (UNIX)

(código)

output to printer.

-p procedure Startup Procedure (Cliente)

Especifica a procedure a ser executada.

-P senha Password (Cliente)

Define a senha do usuário no banco de dados.

-param texto Parameter (Cliente)

Declara um parâmetro informativo para identificar a sessão.

Por exemplo, em sua empresa você mantém bancos de dados produção e bancos de desenvolvimento, e em tempo de execução você quer saber qual o banco que está sendo utilizado, então você especifica uma palavra para o início da sessão, -p "desenv", e o obtém por código pelo comando
SESSION:PARAMETER.

-pf arquivo Parameter File (Servidor/Cliente)

Especifica um arquivo de parâmetros a ser utilizado para iniciar uma sessão servidor ou cliente.

-plm PROLIB Memory (Cliente)

Aloca 512 bytes para o cache de bibliotecas.

-pls**PROLIB Swap (Cliente)**

Define o uso de arquivos temporários .SRT para procedures compiladas.

-populate**Fast Schema Change (Cliente)**

Define para o Progress um mecanismo dinâmico para obtenção e manipulação dos dados do metaschema.

-pp diretório**ESQL PROPATH**

Adiciona um novo diretório ao Propath em uma conexão ODBC.

-pwqdelay**Page Writer Queue Delay (Servidor)**

Especifica um número em milisegundos entre scan de páginas e escrita no banco de dados.

Mínimo: 0**Padrão multi-usuário: 100**

-pwqmin n**Page Writer Queue Minimum (Servidor)**

Define um número mínimo de buffers a serem requeridos para gravação no banco de dados antes que o APW os grave no disco.

Mínimo: 0**Padrão multi-usuários: 1**

-pwscan n Page Writer Scan (Servidor)

O número de buffers que o APW deva contar antes de um ciclo.

Mínimo: 0

Padrão multi-usuário: 4% de buffers (-B)

-pwsdelay n Page Writer Scan Delay (Servidor)

O número de segundos de intervalo entre cada procura de buffers para gravação no banco de dados pelo APW.

Mínimo: 0

Padrão multi-usuário: 1

-pwwmax n Page Writer Maximum Buffers (Servidor)

Define um número máximo de buffers a serem requeridos para gravação no banco de dados antes que o APW os grave no disco.

Mínimo: 0

Padrão multi-usuários: 25

-q Quick Request (Cliente)

Mantém uma procedure residente em memória para uso/acesso dinâmico para outras execuções.

Mais informações no capítulo Performance.

-Q**ANSI SQL (Cliente)**

Força o uso do padrão ANSI SQL para declarações SQL afim de manter compatibilidade absoluta.

-Q2**ANSI SQL Client (Cliente)**

Mantém compatibilidade com o padrão ANSI SQL89 para transações.

-r**Buffered I/O (Servidor)**

Habilita a gravação em buffer de I/O para o Before-Image.

Muita atenção para este parâmetro porque ele coloca a integridade do banco de dados em risco. Geralmente quando conectado a um banco localizado em um drive remoto ou removível o Progress automaticamente informa sobre o perigo de crash quanto ao dispositivo, que no caso de falha o banco de dados só poderá ser restaurado por meio de backup.



Este modo de conexão é extremamente perigoso para a integridade dos dados. Em caso de falha apenas por meio de backup pode-se recuperar os dados.

-rand n**Alternate Random Number Generator (Cliente)**

Define um número base a ser utilizado para geração de valores randômicos.

Mínimo: 1

Máximo: 2

Padrão: 1

-rg

Run 4GL Client (Cliente)

Define que o Progress utilizado é 4GL Full, caso você tenha mais de um tipo de configuração instalada no seu Progress atual.

-RO

Read-Only (Cliente)

Define que o banco de dados conectado será apenas para leitura.

-rptint n

License Usage Reporting Interval (Servidor)

Define um intervalo, em minutos, para a sessão multi-usuários informar o uso das licenças.

-rq

Run Query Client (Cliente)

Define que o Progress utilizado será apenas Client para limitar o acesso dos clientes.

-rr

Run Run-time Client (Cliente)

Define que o Progress utilizado será apenas Client para limitar o acesso dos clientes.

-s n**Stack Size (Cliente)**

Define o número para o tamanho de stack da área da memória compartilhada pela sessão multi-usuários em 1K(1024 bytes). Apenas utilize este parâmetro caso ocorra algum erro solicitando o incremento deste parâmetro por uma tabela que ultrapasse os valores padrões, como os erros abaixo:

SYSTEM ERROR: stkpush: stack overflow.
Increase -s parameter.

SYSTEM ERROR: stkditem: stack overflow.
Increase -s parameter.

Mínimo: 3**Máximo: 63****Padrão: 40**

-S serviço**Service Name (Servidor/Cliente)**

Especifica o nome/alias do serviço multi-usuário definido na sessão multi-usuário.

Este serviço deve estar definido no arquivo services com o nome/alias, porta e protocolo.

A localização em Windows 95/98 é o diretório do Windows, se em Windows NT diretório do NT\system32\drivers\etc e UNIX /etc.

| | |
|--------------------|------------------------------------------------------|
| -Sn serviço | AS/400 Server Program Name (Servidor/Cliente) |
| | Igual ao parâmetro -S |

| | |
|----------------|---------------------------------------------------------------------------------------|
| -spin n | Spin Lock Retries (Servidor) |
| | Define o número de vezes que um processo tentará ser concluído antes de um intervalo. |

| | |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -stream Código Página | Stream (Cliente) |
| | Especifica um Código de Página a ser utilizado para scopo de streams (saída para impressora e tela, arquivos de definições, etc). Você pode especificar duas configurações iso8859-1 para códigos latinos ou ibm850 para códigos indo-arábicos. |
| | Padrão: ibm850 |

| | |
|----------------|--------------------------------------------------------------|
| -stsh n | Stash Area (Cliente) |
| | Define o número de blocos a serem alocados na área de Stash. |
| | Mínimo: 1 |
| | Máximo: 31 |
| | Padrão: 2 |

| | |
|----------------|----------------------------------------------------------|
| -SYBc n | SYBASE Connections (Cliente) |
| | Número máximo de conexões para um banco de dados Sybase. |

Padrão: 25

-SYBt n SYBASE Time (Cliente)

Define um timeout em segundos para uma conexão ao banco de dados Sybase.

Padrão: 3

-t Save Temp Files (Cliente)

Mantem arquivos temporários após o encerramento de uma sessão.

-T diretório Temporary Directory (Cliente)

Define um diretório como repositório de arquivos temporários.

Caso não seja especificado o parâmetro o Progress assumirá como padrão o diretório inicial da aplicação.

-TB n Speed Sort (Cliente)

Define um tamanho em kilobytes a ser alocado para processos de organização.

Mínimo: 3

Máximo: 31

Padrao: 12

-TM n Merge Number (Cliente)

Define o número de blocos ou streams a serem mesclados simultaneamente para

processos de organização.

Mínimo: 1

Máximo: 32

Padrão: 5

-tok n

Token (Cliente)

Define o número máximo de "tokens" permitidos em uma declaração Progress 4GL.

Máximo: 1.600

Padrão: 1024

-trig diretório

Trigger Location (Cliente)

Especifica o diretório aonde estão localizados os arquivos dos triggers do bancos de dados.

-tstamp

Time Stamp (Cliente)

Define um timer que verifica o conteúdo do código executável .R se equivale ao mesmo conteúdo do Metaschema atual.

-usrcount n

User Count (Servidor)

Define o número de usuários permitidos para conexão ao banco de dados após o excesso das licenças.

-U usuário

User ID (Cliente)

Indica o nome do usuário definido no banco de dados.

-vw**Video Write (Cliente)**

Habilita o scopo direto para o vídeo por compatibilidade com um console PC. Use em Sistema Operacional SCO para controle do stream de I/O.

-VO versão**ORACLE Version (Cliente)**

Define a versão do Oracle a ser conectada (Versão 6 ou 7).

-v6colon**Version 6 Colon (Cliente)**

Especifica compatibilidade com a versão Progress 6 para alinhamento de campos. Caso você não especifique este parâmetro o Progress desloca o campo dois caracteres a direita.

-v6q**Version 6 Query (Cliente)**

Define o uso de apenas um índice para uma cláusula FOR EACH. Caso não seja especificado o Progress utiliza o melhor número de índice para qualquer condição.

-xc coleção**Extended Alphabet Support**

Específica a tabela de coleção de Códigos de Página a ser utilizada pela sessão.

Este parâmetro é similar ao parâmetro – **cpcoll**.

Maiores informações veja o manual
PROGRESS Internationalization Guide.

-y**Statistics (Cliente)**

Ativa o log de estatísticas de uso e consumo dos programas.

Quando uma sessão termina o Progress grava as informações de utilização da sessão em um arquivo log.

Você pode especificar o arquivo de saída das estatísticas definindo a variável de ambiente **CLIENTMON**, como abaixo:

```
CLIENTMON=/tmp/estat.log ;  
export CLIENTMON
```

Caso você não defina a saída o arquivo padrão é **client.mon** localizado no diretório da aplicação.

-yc**Statistics with CTRL-C (Cliente)**

Define a criação de estatística e desabilita o uso das teclas CTRL-C para abortar uma aplicação.

Também substitue o comando **SHOW-STATS** para criação de estatísticas.

Mais informações consulte o capítulo Performance

-yd**Segment Statistics (Cliente)**

Define a criação de estatísticas geradas pela execução das procedures.

Armazena informações sobre execução de programas compilados e seus respectivos tamanhos em bytes, e criação/acessos a arquivos SRT e o consumo em bytes.

Por padrão estas estatísticas são armazenadas no arquivo **client.mon** localizado no diretório da aplicação.

Mais informações consulte o capítulo Performance.

-yx

Statistics with Cross-reference (Cliente)

Define a criação de estatísticas geradas pela execução das procedures, e armazena todas as informações de uso em um arquivo.

Este arquivo por padrão é criado no diretório da aplicação chamado proc.mon, caso você queira especificar outra localização para o arquivo deve-se configurar a variável de ambiente.

PROCMON=Localização do Arquivo

Exemplo

SET PROCMON=M:\TEMP\MONITOR.LG
(Windows)

PROCMON=/tmp/monitor.lg ; export PROCMON (UNIX)

Mais informações consulte o capítulo Performance.

-yy ano

Century (Cliente)

Especifica um ano base para o século. Por exemplo, 1950 declara que anos iguais e inferiores a 50 serão do século 20, e anos superiores a 50 serão século 21.

Mínimo: 1100

Máximo: 9900

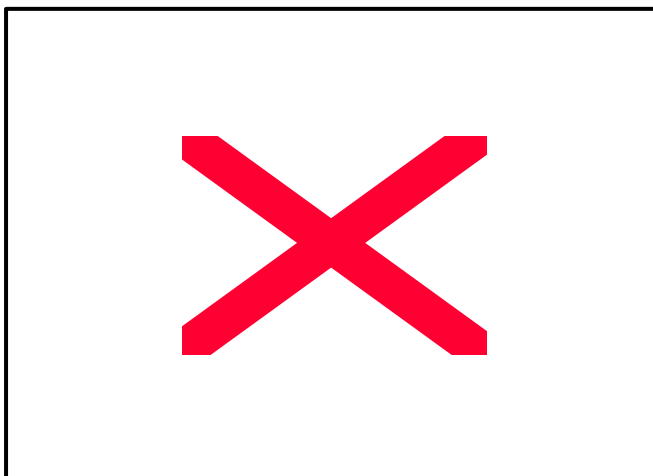
Padrão: 1950

Banco de Dados Multi-Volumes

O que é o banco multi-volumes?

O banco de dados multi-volumes no Progress nada mais é que arquivos de dados, de transações, de imagens que fazem parte, como também pertencem a um único banco de dados.

O figura abaixo exemplifica a constituição de um banco de dados dividido em 3 arquivos de dados (.DBn), 2 arquivos para o Before-Image (.Bn) e 2 arquivos para o After-Image (.An).



Os arquivos padrão como Database (.DB), Before-Image (.DB) e After-Image (.AI) quando em multi-volumes eles apenas armazenem dados informativos sobre a

própria constituição e estrutura do banco de dados. Por exemplo o arquivo .DB não mais contém dados das tabelas, apenas as estruturas das mesmas, índices, validações, etc.

Os arquivos estendidos (D_n , B_n , A_n) têm um tamanho fixo e inalterado como definido na estrutura de criação dos mesmos, inicialmente vazios, porém os tamanhos definidos já alocados.

Os arquivos **Voláteis (D_n , B_n , N_n)** não têm um tamanho definido, porque eles serão utilizados e incrementados depois de esgotado os tamanhos definidos dos arquivos anteriores. Por exemplo, um banco multi-volumes com dois arquivos de dados em 50 megas e um terceiro volátil será esgotado o tamanho pré-definido quando este ultrapassar 100 megas em quantidade de informações para tabelas e índices, então utilizando o arquivo estendido.

Por que usar Banco Multi-Volumes?

A utilização de banco de dados multi-volumes além de uma solução para o problema do tamanho máximo de um arquivo imposto pelo Sistema Operacional (2 Gigabytes UNIX, Novell e FAT16 Windows), também é recomendado como o melhor, ou um dos melhores, meios para melhorar a performance de manipulações de dados.

Arquivos menores e individuais permitem acessos mais rápidos para abertura, leitura e gravação, tanto para controle intrínseco do Sistema Operacional, como também para o SGBD do banco.

Criando a estrutura multi-volumes

Você pode particionar três partes constituintes do banco: os arquivos de Dados (.DB), o Before-Image (.BI) e o After-Image(.AI). Podendo-se definir n arquivos constituintes em n localizações iguais ou distintas, sendo o último arquivo de cada parte constituinte um arquivo de tamanho volátil.

Formação do arquivo de estrutura

| Tipo do Arquivo | Localização e Nome | Tamanho (múltiplo de 32) |
|-------------------------|-------------------------------------------------------|-----------------------------|
| D (Dados) | [Diretório/] Arquivo. Xn f | $n * 32$ |
| B (Before-Image) | Onde X é o tipo do arquivo e n é a seqüência | |
| A (After-Image) | do mesmo e f é uma constante | |

Estas definições devem estar descritas em um arquivo tipo texto com o cujo nome deve ser o mesmo do banco a ser criado e extensão **st**.



A Progress declara que você pode definir um nome para o banco e informar o arquivo de estrutura, mas em muitos casos não funciona. (**prostrct create banco [arquivo de descrição] ...** Então crie um arquivo de estrutura com o mesmo nome do banco desejado!

Arquivo **info.st**

```
d info.d1 f 2048
d info.d2 f 2048
d info.d3 f 2048
d info.d4
b info.b1 f 1024
b info.b2
a info.a1 f 1024
a info.a2
```

O exemplo acima cria uma definição com particionamento em 3 arquivos para dados com 2 megas e o quarto estendido, e dois para Before-Image e After-Image com 1 mega.



Lembre-se sempre que o último arquivo não pode informar o tamanho porque este será um arquivo volátil.



É sempre aconselhável a localização dos arquivos After-Image em outro disco, para possibilitar a recuperação caso aconteça algum problema nos bancos de dados por motivo de falha no(s) disco(s) de produção.

Criando o Banco

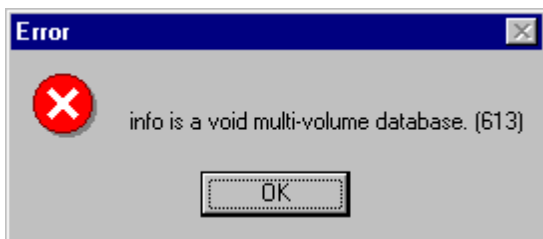
Após concluídas as definições do arquivo apenas defina a variável de ambiente DLC e o PATH, caso não definidos execute o utilitário PROSTRCT, exemplo:

prostrct create info

Pronto! Criado o banco de dados multi-volumes.

Depois de criado o banco ainda não é o suficiente para começar a utilizar. O Progress não permite conexão a um banco de dados, Progress pelo mesmos, caso este não contenha nenhuma definição de estrutura lógica. O banco de dados recém criado está vazio, as definições feitas foram somente quanto à estrutura física de multi-volumes ou multi-arquivos, não há no banco tabelas, usuários para obter permissões, etc.

Caso você tente conectar um banco multi-volumes sem nenhuma definição de estrutura, Metaschema, será apresentado este erro.



Quando você cria um banco de dados simples e vazio (Empty), por exemplo, o Progress o faz a partir de um banco já existente chamado **Empty.db**, localizado em DLC. Liste todos os arquivos .DB e você verá os bancos origem para o PRODB, Data Administration, CREATE TABLE e qualquer outro modo de criação de banco de dados Progress.

Para colocar o banco em funcionamento você deve copiar a estrutura de um banco já pronto para o banco multi-volumes criado utilizando o utilitário **PROCOPY**, exemplo:

Sintaxe: PROCOPY Banco de Origem Banco Destino

Digite este comando, lembrando-se que a variável de ambiente DLC deve estar referenciada adequadamente.

procopy %DLC%/empty info (DOS/WINDOWS)

procopy \$DLC/empty info (UNIX)

Pronto! O banco multi-volumes esta criado.

A conexão é da mesma forma que um banco de dados simples. O gerenciamento dos arquivos é todo por conta do Progress.

MAGNUS

Apresentação

Será discutido neste capítulo diversos assuntos relativos ao sistema de gestão empresarial MAGNUS Datasul, baseado em maior parte no MAGNUS versão I modo Gráfico para ambiente Windows, será discutido e mostrado telas e detalhes para ambiente UNIX, porém em menor quantidade.

Como o MAGNUS funciona?

O sistema MAGNUS contempla em seu pacote diversos módulos, alguns obrigatórios e primários (Cadastros, Utilitários, Ajuda), e outros de acordo com a necessidade da empresa (Faturamento, Exportação, Chão de Fábrica, etc).

A sua configuração quanto ao Progress é baseada, ainda, na versão Progress 6 - conforme discutido no capítulo Configuração do Produto. Ele é instalado em um diretório principal, compartilhado entre todos os clientes que o utilizarão, com diversos sub-diretórios que representam um módulo ou grupo de utilidades, por exemplo:

/MAGNUS (Diretório principal)

/MAGNUS/cdp (Cadastros)

/MAGNUS/utp (Utilitários)

/MAGNUS/ctp (Contabilidade)

Lista de Diretórios do MAGNUS (principais)

| | |
|-----|--------------------|
| CDP | Cadastros |
| UTP | Utilitários |
| IGP | Gráficos |
| INC | Trigges e Includes |



A nomenclatura de TRIGGERS utilizada pela Datasul é a tradução literal, GATILHOS

| | |
|-----|--------------------------------------------------|
| AJP | Ajuda |
| CTP | Contabilidade |
| CEP | Estoque |
| CCP | Compras |
| REP | Recebimento |
| ESP | Programas específicos desenvolvidos pelo cliente |
| PTP | Patrimônio |

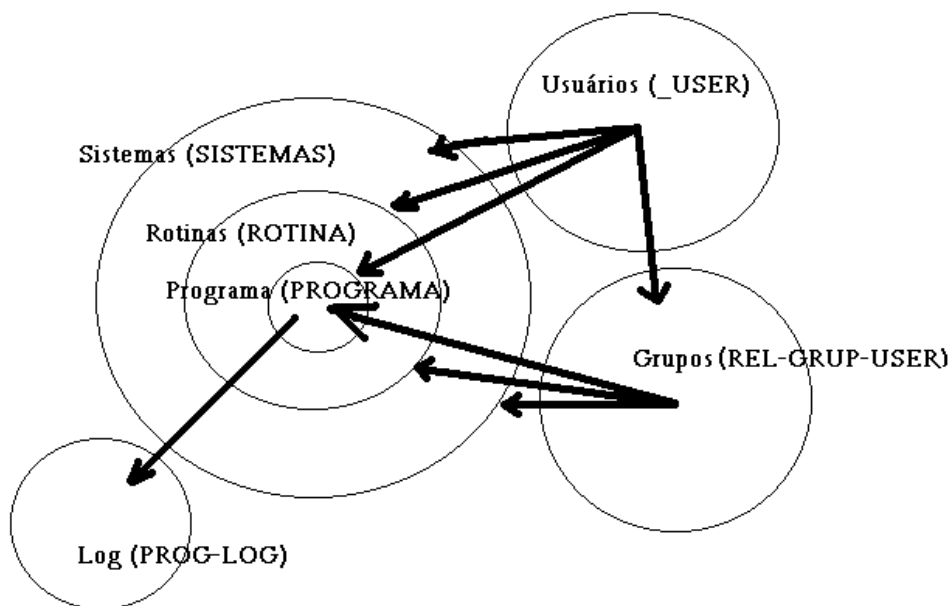
OUTROS...

Funcionamento do MENU

Hierarquia/Relacionamento

SISTEMA → ROTINA → PROGRAMA

O gráfico abaixo demonstra a lógica dos relacionamentos e permissões de acesso a cada SISTEMA, ROTINA e/ou PROGRAMA.



Exemplo

```
/* ex0036.p */  
  
define variable c-programas as character.  
  
for each sistema:  
    display sistema.sigla  
    sistema.nome  
    sistema.permissao format "x(40)".
```

```
for each rotina of sistema:
    display rotina.sigla
        rotina.rotina
        rotina.permissao format "x(40)".

    for each programa where
        programa.programa = rotina.programa[01] or
        programa.programa = rotina.programa[02] or
        programa.programa = rotina.programa[03] or
        programa.programa = rotina.programa[04] or
        programa.programa = rotina.programa[05] or
        programa.programa = rotina.programa[06] or
        programa.programa = rotina.programa[07] or
        programa.programa = rotina.programa[08] or
        programa.programa = rotina.programa[09] or
        programa.programa = rotina.programa[10] or
        programa.programa = rotina.programa[11] or
        programa.programa = rotina.programa[12] or
        programa.programa = rotina.programa[12]:

        assign c-programas = rotina.programa[01] '+' '+'
            rotina.programa[02] '+' '+'
            rotina.programa[03] '+' '+'
            rotina.programa[04] '+' '+'
            rotina.programa[05] '+' '+'
            rotina.programa[06] '+' '+'
            rotina.programa[07] '+' '+'
            rotina.programa[08] '+' '+'
            rotina.programa[09] '+' '+'
            rotina.programa[10] '+' '+'
            rotina.programa[11] '+' '+'
            rotina.programa[12] '+' '+'
            rotina.programa[12].

        display entry(lookup(programa.programa,
            c-programas),
            'a,b,c,d,e,f,g,h,i,j,k,l,m')
        programa.programa
        programa.permissao format "x(40)"
        with no-box no-labels.

    end.
end.
end.
```

Validando o acesso ao Programa

Este código abaixo valida a permissão para uso de um programa de acordo com as permissões concedidas ao grupo do usuário ou individualmente, pelo nome.

Exemplo

```

/*****
* Programa...: ACESSOMG.P
* Programador: Marcio Brener
* Finalidade.: Consistencia das permissoes para execucao
*              de programas
* Data.....:
*****/
define input parameter c-programa like programa.programa.

/* Ignora verificacao se usuario ADM ou SUPER */
if userid("mgadm") = "adm" or
   userid("mgadm") = "super" then return.

/* Verifica se o usuario atual tem USERID do MAGNUS */
find _user where
   _user._userid = userid("mgadm") no-lock no-error.
if not available _user then do:
   bell.
   message "Usuario nao autorizado no MAGNUS"
           view-as alert-box warning title "Acesso negado".
   return error.
end.

/* Verifica se o programa esta cadastrado no MAGNUS
   para acessos */
find programa where
   programa.programa = c-programa no-lock no-error.
if not available programa then do:
   bell.
   message "Programa nao cadastrado no MAGNUS"
           view-as alert-box warning title "Acesso negado".
   return error.
end.

/* Verifica se e permitido para todos */
if lookup("?", programa.permissao) > 0 then return.

/* Verifica se tem permissao para o USERID do MAGNUS */
if lookup(userid("mgadm"), programa.permissao) > 0 then return.

```

```
/* Verifica permissao programa-grupos X usuario-grupos */
for each rel-grup-user where
    rel-grup-user.usuario = userid("mgadm") no-lock:
    if lookup(rel-grup-user.cd-grupo, programa.permissao) > 0
then return.
end.

/* Retorna em ERRO se nenhuma condicao satisfeita */
bell.
message "Usuario sem permissao para executar este programa"
    view-as alert-box warning title "Acesso negado".
return error.

/* fim do programa */
```

Bancos Históricos

Quando é encerrado um período fiscal, 1 ano, é necessário fazer os fechamentos contábeis do período e iniciar um novo exercício fiscal. Um problema comum às empresas é o histórico deste exercício encerrado, até mesmo por não ter o módulo de Bancos Históricos do MAGNUS, então a solução é manter os bancos de dados deste período encerrado à parte para futuras consultas, conforme exigências legais.

Os bancos de dados históricos devem ser consultados, porém nunca alterado. Então como garantir acesso somente para consulta?

Para isso a melhor forma é garantir o conteúdo dos dados inalterados a partir do próprio banco de dados com permissão somente para leitura pelos clientes, o programa abaixo mostra como fazer isso.

Exemplo

```
/* ex0034.p */

/*
Este programa concede permissão somente aos usuários
ADM e SUPER para modificação de dados
*/

/* Banco Administrativo */
for each mgadm._file
where mgadm._file._file-name <> "maxusadm"      and
      mgadm._file._file-name <> "prog-log"      and
      mgadm._file._file-name <> "log-mensagem"
exclusive-lock:

    assign mgadm._file._can-create = "adm,super"
    mgadm._file._can-delete = "adm,super"
    mgadm._file._can-write = "adm,super"
    mgadm._file._can-read = "*"
    mgadm._file._can-dump = "adm,super"
    mgadm._file._can-load = "adm,super".

    for each mgadm._field of mgadm._file exclusive-lock:
        assign mgadm._file._can-write = "adm,super"
        mgadm._file._can-read = "*".
    end.
end.

/* Banco Comercial */
for each mgcom._file exclusive-lock:
    assign mgcom._file._can-create = "adm,super"
    mgcom._file._can-delete = "adm,super"
    mgcom._file._can-write = "adm,super"
    mgcom._file._can-read = "*"
    mgcom._file._can-dump = "adm,super"
    mgcom._file._can-load = "adm,super".

    for each mgcom._field of mgcom._file exclusive-lock:
        assign mgcom._file._can-write = "adm,super"
        mgcom._file._can-read = "*".
    end.
end.

/* Banco Industrial */
```

```
for each mgind._file exclusive-lock:
    assign mgind._file._can-create = "adm,super"
    mgind._file._can-delete = "adm,super"
    mgind._file._can-write = "adm,super"
    mgind._file._can-read = "*"
    mgind._file._can-dump = "adm,super"
    mgind._file._can-load = "adm,super".

    for each mgind._field of mgind._file exclusive-lock:
        assign mgind._file._can-write = "adm,super"
        mgind._file._can-read = "*".
    end.
end.

/* fim do programa */
```

Você poderia também iniciar a sessão cliente com o parâmetro somente leitura -RO, mas no banco de dados administrativo do MAGNUS, mgadm, é necessário a leitura, gravação e deleção de registros para se utilizar o sistema.

Desenvolver em Windows para usar no UNIX...

Caso você use MAGNUS em ambiente UNIX via telnet, geralmente o software NETTERM, tanto para uso do Sistema como também desenvolvimento de programas específicos a edição é bastante difícil, talvez nem tanto, mas de modo difícil com certeza. Em ambiente UNIX o Procedure Editor não lhe dá uma agilidade para escrita tão boa como qualquer editor Windows ou DOS, com facilidades como copiar e colar, seleção rápida, uso do mouse, help on-line, entre outros diversos atrativos disponíveis no Windows.

Para isso primeiramente você tem ter uma versão do Progress para Windows, com suporte a editar procedures e manipular dados (Procedure Editor e

Dicionário de Dados), caso você tenha faça os seguintes passos:

Configure a sessão servidor UNIX adequadamente para a conexão do Windows:

1. Edite o arquivo /etc/services e inclua ao final deste os serviços dos bancos de dados que serão utilizados para identificações por parte das conexões clientes Windows, exemplo:

mgadm 8030/tcp

mgcom 8031/tcp

mgind 8032/tcp

2. A carga do banco de dados inclua a informação do serviço a ser utilizado pelo banco, exemplo:

Carga do banco Administrativo

proserve -db i00adm -S mgadm -N tcp ...

Configure a sessão cliente Windows adequadamente para o MAGNUS:

1. Crie um diretório local na máquina de desenvolvimento ou então no servidor para serem instalados os arquivos de programas do MAGNUS, isso ocupará em média 300 megabytes;
2. Transfira os programas por FTP para o diretório já criado. Será mais fácil se você transferir o diretório inteiro do MAGNUS de uma só vez para o Windows;

3. Você pode apagar todos os arquivos .R, eles não serão necessários para desenvolvimento porque apenas ocuparão mais espaço, e você já tem os programas .P que funcionam;
4. Edite um arquivo .INI para o MAGNUS (MAGNUS.INI) igual ao descrito no capítulo Configuração do Produto, Listagem do arquivo Progress.ini, ao qual será utilizado para iniciar o MAGNUS. Lembre-se de configurar corretamente os valores para o PROPATH, DLC PROCFG, PROMSG, etc;
5. Edite o arquivo **host** localizado no diretório do Windows, caso não exista o crie e informe o número do IP do Servidor dos Bancos de Dados e o Nome/Alias do Servidor, exemplo:

172.35.1.10 servidor_magnus

6. Edite o arquivo **services** localizado no diretório do Windows, caso não exista o crie e informe os Nomes/Aliases dos serviços e os Números dos Serviços iniciados, igual editado no arquivo /etc/services do Servidor UNIX.

mgadm 8030/tcp

mgcom 8031/tcp

mgind 8032/tcp



Em Windows NT a localização é Diretório do NT\SYSTEM32\DRIVERS\ETC\hosts e services. Saiba também que o padrão utilizado pelo Progress é o arquivo HOSTS e não LMHOSTS!

7. No arquivo de parâmetros (.pf) de conexão do MAGNUS o faça da seguinte maneira:

Arquivo MAGNUS.PF

-db i00adm

-ld mgadm

-H servidor_magnus #igual ao nome definido no arquivo hosts

-S mgadm #igual ao nome definido no arquivo services

-N tcp #protocolo TCP/IP

... (Outros Bancos de Dados)

-p menu7.p

Refaça estes procedimentos para os bancos Comercial e Industrial.

8. Faça um atalho com a seguinte sintaxe:

ALVO

PROWIN32.EXE -basekey "ini" -ininame
MAGNUS.INI -pf MAGNUS.PF

INICIAR EM

Diretório do MAGNUS



O parâmetro **-basekey "ini"** é necessário para forçar o uso do arquivo INI, ao invés de valores padrões na Registry do Windows. Caso você esteja utilizando Progress 7 não o especifique.



Lembre-se que no MAGNUS Versão I para Windows o programa inicial do menu é chamado **MENU7.P**, e caso você não tenha os arquivo das imagens de bitmap, localizadas no diretório IGP, você precisa obtê-las pela BBS Datasul e colocas neste diretório!



Saiba que mesmo em Windows ao indicar um arquivo deve-se usar sempre barra estilo UNIX (run **esp/es0001.p**) e os diretórios do PROPATH. Sempre!

Pronto! Você já pode desenvolver em Windows e depois de editadas e testadas as procedures apenas as transfira por FTP para o UNIX e compile-as.

Criar Bancos para Testes/Desenvolvimento

Como é praxe para a área técnica a utilização de bancos teste e/ou desenvolvimento e sessão teste e/ou de desenvolvimento. Será apresentada algumas sugestões e maneiras de melhorar ou facilitar o desenvolvimento e/ou manutenção do sistema.

Criando os bancos

Obviamente é ideal criá-los a partir dos bancos de atuais em uso, para maior fidelidade à produção, então sigas os passos a seguir:

1. Tire do ar todas as sessões que estão utilizando os bancos de testes atuais, como exitam;
2. Tire os bancos de testes do ar;
3. Copie os bancos produção para o bancos teste. Lembre-se sempre de tirar do ar os bancos de produção antes de fazer as cópias para a área de testes e se possível trunque o Before-Image do banco. Use o PROCOPY preferencialmente, ou então por meio do próprio sistema operacional (COPY, CP, EXPLORER, etc);



Jamais faça cópia de bancos de dados ativos, isso pode danificá-los!



Caso você precise fazer uma cópia para testes, mas não possa finalizar o banco em produção faça um **BACKUP ON-LINE** e o restaure na área de testes! Sintaxe:

probkup online Banco Destino, exemplo: probkup online i00adm d:\desenv\i00adm, em UNIX AIX probkup online i00adm /dev/rmt0. Restaurando: prorest i00adm d:\desenv\i00adm, UNIX AIX prorest i00adm /dev/rmt0.



Em UNIX o backup em fita destrói todos os dados já existentes na mesma! (PROBKUP, CPIO, TAR)

4. Ative os bancos de produção e bancos de teste;
5. Inicie a sessão do MAGNUS TESTE e acrescente em todas as descrições dos sistemas a palavra TESTE, por exemplo: Conta a Pagar - TESTE, Caixa e Bancos TESTE, Contabilidade TESTE. Isso informa ao usuário que aquele sistema é para testes, evitando qualquer problema...
6. Caso você tenha o Progress a partir da versão 8 identifique a sessão com o parâmetro -param no arquivo .pf de início. Exemplo: -param TESTE ou -param DESENV. Você pode obter o status da sessão por meio de código, exemplo: DISPLAY SESSION:PARAM, isso identificará a sessão.

BACKUP

Existem duas opções para backup, pelo utilitário do Progress PROBACKUP ou por aplicativos próprios do Sistema Operacional ou ainda terceiros.

O backup do Progress, PROBACKUP, tem diversas funções específicas para esta tarefa, nele é possível especificar o número de redundância de gravação do banco, afim de evitar algum problema físico em um pedaço da fita; ele pode ser restaurado em qualquer local específico, como também apenas incremental.

Em UNIX não é aconselhável utilizar o PROBACKUP, caso você precise armazenar mais que 1(um) banco por fita. Isso porque ele apagará todo o conteúdo da mesma para o backup de um banco.

Antes de iniciar o backup você deve, sempre, baixar o banco, se possível também truncar o Before-Image. O script abaixo demonstra como proceder em UNIX e/ou Windows NT:



Os scripts mostrados assumem que os arquivos físicos dos bancos de dados têm a nomenclatura de I00[sigla], como Administrativo I00adm, Comercial I00com, etc.

UNIX AIX

```
#Script para Backup dos Bancos de Dados MAGNUS
```

```
wall "Senhores usuários, o backup iniciara em 1 minuto! Por  
favor finalizem seus trabalhos"  
sleep 60
```

```
cd /bases/magnus  
DLC=/sistemas/dlc ; export DLC  
PATH=/usr/bin:$DLC:$DLC/bin ; export PATH
```

```
#Baixa os bancos de dados  
proshut i00adm -by  
proshut i00com -by  
proshut i00ind -by
```

```
#Trunca o Before-Image  
proutil i00adm -C truncate bi  
proutil i00com -C truncate bi  
proutil i00ind -C truncate bi
```

```
#Use CPIO ou TAR para backup  
find ./ -name i00* | cpio -ovcB > /dev/rmt0 > backup.log  
tar -cvf i00* /dev/rmt0 > backup.log
```

```
#Marque o banco de dados como "backupado"  
rfutil i00adm -C mark backedup
```

```
rfutil i00com -C mark backedup
rfutil i00ind -C mark backedup

#Carregue os bancos de dados
proserve i00adm ... parâmetros
proserve i00com ... parâmetros
proserve i00ind ... parâmetros
wall "Senhores usuarios! O sistema MAGNUS ja esta no ar
novamente!"

#fim do script
```

Windows NT

REM Script para Backup dos Bancos de Dados MAGNUS

```
@echo off
net send DOMÍNIO "Senhores usuários o backup esta send
iniciado. Finalize seu trabalho!"
```

```
d:
cd \bases\magnus
set DLC=c:\sistemas\dlc
set PATH=%PATH%;%DLC%;%DLC%/bin
```

```
REM Baixa os bancos de dados
proshut i00adm -by
proshut i00com -by
proshut i00ind -by
```

```
REM Trunca o Before-Image
proutil i00adm -C truncate bi
proutil i00com -C truncate bi
proutil i00ind -C truncate bi
```

```
backup i00* [DESTINO] /F:backup.log
```

```
REM Marque o banco de dados como "backupado"
rfutil i00adm -C mark backedup
rfutil i00com -C mark backedup
rfutil i00ind -C mark backedup
```

```
REM Carregue os bancos de dados
proserve i00adm ... parâmetros
proserve i00com ... parâmetros
proserve i00ind ... parâmetros
```

```
net send DOMÍNIO "Senhores usuarios! O sistema MAGNUS ja  
esta no ar!
```

```
REM fim do script
```


Utilizando zoom de cadastro Datasul

Os programas do MAGNUS têm janelas zoom para facilitar a manipulação dos dados de cadastros diversos. Os programas CDZxxxx.p são os zooms dos cadastros diversos de utilização genérica em todos os programas.

Estes programas zoom requerem algumas variáveis pré-declaradas (NEW SHARED) para o posterior retorno do dado escolhido, como abaixo:

```
/* variáveis solicitadas pelo programa de zoom de
Empresas (cdz0104.p) */
```

Exemplo

```
/* ex0035.p */

define new shared variable l-implanta as logical init yes.
define new shared variable i-cod030 like empresa.ep-codigo.

define variable v-ep-codigo like empresa.ep-codigo.

form v-ep-codigo at row 1 column 8 colon-aligned
    validate(can-find(empresa where empresa.ep-codigo =
input frame f-empresa v-ep-codigo),
    "Empresa nao cadastrada")
    empresa.nome at row 1 column 14 left-aligned no-
labels view-as text
    with frame f-empresa row 10 centered overlay side-
labels title "Empresa".

update v-ep-codigo with frame f-empresa editing:
readkey.
if frame-field = "v-ep-codigo" and
lookup(keyfunction(lastkey), "tab,return,get") > 0 then
do:
    if keyfunction(lastkey) = "get" then do:
        run cdp/cdz0104.p.
        assign v-ep-codigo = i-cod030.
        display i-cod030 @ v-ep-codigo
```

```

        with frame f-empresa.
    end.
else apply lastkey.

    find empresa where
        empresa.ep-codigo = input frame f-empresa v-ep-
codigo no-lock no-error.

    if available empresa then
        display empresa.nome with frame f-empresa.
    else
        display "" @ empresa.nome with frame f-empresa.
    end.
else apply lastkey.
end.

```

Para se saber quais são as variáveis solicitadas pelo zoom simplesmente execute o programa (run) e observe os nomes das variáveis não criadas no retorno do erro.

Shared variable [nome da variável] has not been created. (392)

variáveis requeridas: l-implanta e i-cod030 para retorno.

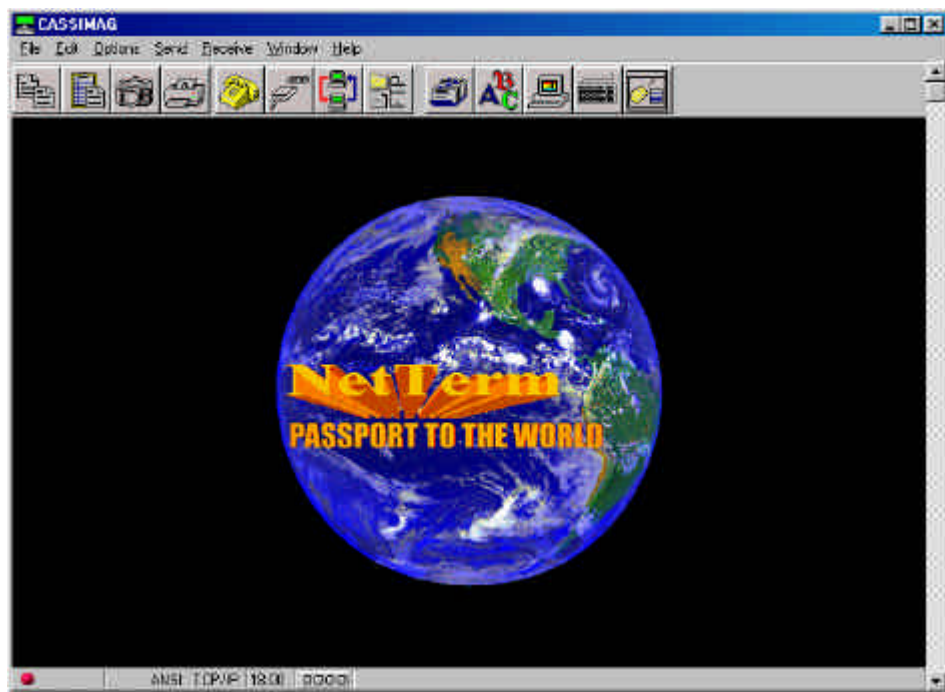
Empresas - CD20104 - 1.00

| Emp | Empresa | Endereco |
|-----|------------|------------------------------|
| 1 | Modelo S/A | Rua Marques de Olinda, 9280 |
| 2 | Vila Velha | Rua Pres Getulio Vargas 2323 |
| 3 | Cia Delta | Av. Procopio Gomes |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Capítulo n

Softwares, Utilitários & Sites diversos

NetTerm



O NetTerm é o aplicativo de telnet apresentado neste livro é certamente um dos melhores telnets existentes, e melhor ainda, ele é shareware.

Escrito por Márcio Brener - <http://www.geocities.com/marcio-brener>

Mude o logotipo padrão do NetTerm para a marca da sua empresa ou pessoal, para tal apenas substitua o arquivo logo.bmp localizado no diretório de instalação do aplicativo para um bitmap do seu logotipo.

Configure as teclas de acordo com o MAGNUS

| Configuração de teclas: ^[Escape, [Control | | |
|-----------------------------------------------------|-------|--------------------|
| F1 | ^[OP | GO |
| F2 | ^[OQ | HELP |
| F3 | ^[OR | INSERT-MODE |
| F4 | ^[OS | EXIT |
| F5 | ^G | GET |
| F6 | ^P | PUT |
| F7 | ^[18~ | RECALL |
| F8 | ^[19~ | CLEAR |
| F9 | ^N | NEW-LINE |
| F10 | ^D | DELETE-LINE |
| F11 | ^[B | BREAK-LINE |
| F12 | ^A | APPEND-LINE |

Obtenha uma cópia em:

InterSoftware@compuserve.com

Kea!X



O Kea!X também foi utilizado na edição deste livro, as telas do Progress em ambiente X-Windows (OSF-Motif) foram capturadas através dele.

O Kea!X é um tipo de telnet gráfico, ele simula um console X-Windows exatamente como se você estivesse em frente a estação servidora em uma janela Windows 95/98/NT.

Obtenha uma cópia de avaliação em:

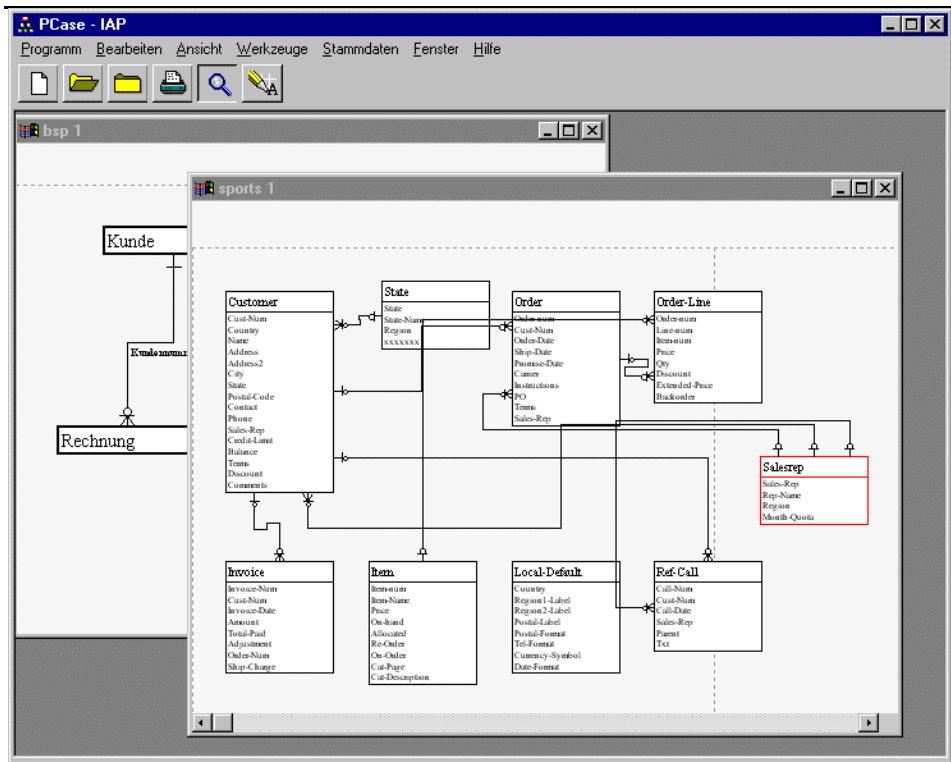
<http://www.attachmate.com/>

PCase



O PCase é uma ferramenta CASE desenvolvida em Progress v8 utilizando, também, OCX e DLL's.

Ela possui engenharia inversa e reversa para bancos de dados Progress.



<http://tools4progress.com/>

Viper

Editor de relatórios gráficos para Progress

Obtenha um cópia em <http://tools4progress.com>

ERWin

Outra ferramenta CASE bastante popular que tem também disposição para Progress, através de arquivos de definições (.df).

Visual PCL

Ferramenta para pré-visualização de arquivos tipo PCL, muito útil para quem usa impressão neste formato.

Obtenha uma cópia em <http://www.visual.co.uk>

Sites sobre Progress

PROGRESS SOFTWARE USA

<http://www.progress.com>

Este é o site oficial da Progress Software proprietária criadora da linguagem. Neste site você encontrará as últimas informações sobre Progress e outros aplicativos.

PROGRESS SOFTWARE BRASIL

Site oficial da Progress no Brasil totalmente em português. Este é um ótimo referencial para informações sobre cursos certificados pela Progress, consultoria, compra e atualização de softwares Progress e qualquer outro serviço sobre Progress no Brasil.

<http://www.progress-software.com.br>

DATASUL

O site do fabricante dos Sistema MAGNUS e MAGNUS EMS, com suporte ao usuário, dados sobre os sistemas, informações de marketing e diversos outros temas sobre Gestão Empresarial.

<http://www.datasul.com.br>

PROGRESS E-MAIL GROUP

Sem dúvida o melhor endereço na Internet para obtenção de ajuda e informações sobre qualquer tema Progress.

O Progress User Mail é dividido em várias partes: linguagem, banco de dados, smart-objects, WebSpeed, AppServers entre outros.

Cadastre-se no PEG e receba diariamente diversos e-mail sobre perguntas e respostas Progress.

<http://www.peg.com>

INSCREVA-SE NO USER GROUP

Faça sua inscrição no Progress User Mail neste link:

<http://www.peg.com/lists/subscribe.html>

PROGRESS HOME PAGE SOBRE ESTE LIVRO

O meu site na Internet, lá você encontrará vários temas e um Forum ON-LINE de perguntas e respostas sobre Progress.

<http://www.geocities.com/marcio-brener/>

mail: brener@zipmail.com.br

USANDO PROGRESS NO LINUX

Informações sobre Progress no Linux

<http://marathon-man.com/pl/Progress-Linux-FAQ.html>

Escrito por Márcio Brener - <http://www.geocities.com/marcio-brener>

<http://www.peg.com/techpapers/redhat.txt>

PERFORMANCE

Artigos, monografias e estudos sobre performance do Progress em vários sistemas operacionais, ajuste de parâmetros e técnicas de programação.

<http://www.westnet.com/~gsmith/ppperform.htm>

<http://www.peg.com/techpapers/monographs/>

SLIDES E APRESENTAÇÕES

Diversos slides, gráficos, anúncios e propagandas sobre Progress.

<http://www.peg.com/techpapers/1999Conf/>

USANDO API'S DO WINDOWS

Página dedicada à utilização de APIs Windows no Progress.

<http://home.wxs.nl/~jurjen.dijkstra/prodevring/index.html>

DESENVOLVEDORES EM PROGRESS

Site de diversos desenvolvedores progress em todo o mundo.

<http://home.wxs.nl/~jurjen.dijkstra/prodevring/about.html>

**BASE DE CONHECIMENTOS PROGRESS
(KNOWLEDGE BASE PROGRESS)**

Site da Progress americana para pesquisa de erros e soluções Progress.

<http://techweb.progress.com/services/support/cgi-bin/techweb-kbase.cgi/kbase.w>