

Outros Comandos

Descreveremos neste capítulo comandos não enquadrados em nenhum dos capítulos anteriores. Isto inclui comandos para controle de projeto, listagem de variáveis, mensagens, definição de parâmetros de controle e acesso ao sistema operacional.

PROJETO

Dentro dos sistemas TQS, Projeto significa um lote de elementos para processamento. Assim, um projeto de vigas contém um conjunto de vigas; um projeto de formas contém todos os elementos que definem uma planta de formas. Projetos são identificados por números de 4 dígitos, com zeros à esquerda se necessário.

No DP um projeto agrupa desenhos quaisquer. O comando PROJETO associa os desenhos sendo gerados pelo programa DP atual a um número de projeto. Mais tarde, a plotagem em impressora e em plotter poderá ser feita referenciando-se apenas o número de projeto; todos os desenhos do projeto serão plotados.

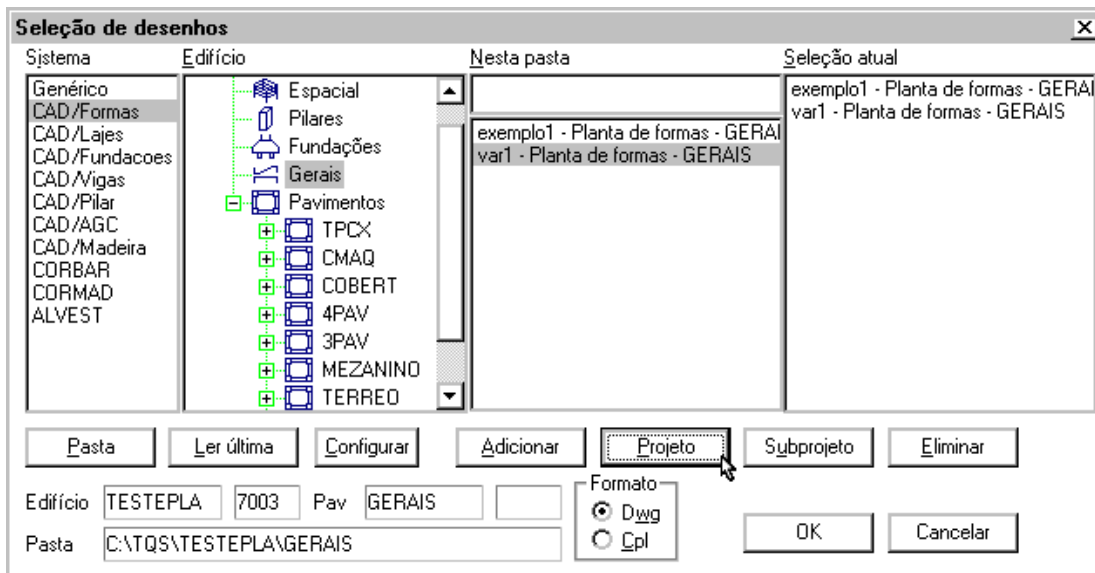
Vários arquivos DP diferentes podem referenciar o mesmo número de projeto; neste caso, os desenhos gerados em cada um destes arquivos farão parte do mesmo projeto. A sintaxe do comando é:

```
PROJETO num
```

onde num é um número de até 4 dígitos. Vamos supor, por exemplo, que os arquivos 1000.DP, 1001.DP e 1002.DP gerem cada um 1 desenho, e que em todos seja declarado

```
PROJETO 1000
```

Ao gerar plantas, utilize o botão “Projeto” para selecionar todos os desenhos do projeto:



O “Subprojeto” consiste num subconjunto de desenhos do projeto, que pode ser montado através do programa gerenciador (veja no manual TQS-AGC&DP Manual de comandos e funções).

MENSAGEM

O comando MENSAGEM emite mensagens na listagem de saída:

```
MENSAGEM 'texto'
```

onde 'texto' é a mensagem entre apóstrofes que será impressa. Por exemplo:

```
N RAI0 = 10
N PI = 3.141593
N AREA = %PI * (%RAIO ** 2)
MENSAGEM 'Raio = %RAIO Area = %.1AREA'
```

resultará em:

```
Raio = 10 Area = 314.2
```

AVISO

Avisos são mensagens especiais, onde o projetista no final é advertido na tela sobre a sua existência. Por exemplo,

```
SE%BITOLA = 0
N BITOLA= 8
```

```
AVISO 'Bitola não fornecida, adotada '  
FIM
```

Se a variável BITOLA valer zero, na listagem aparecerá:

```
*****01 AVISO: Bitola não fornecida, adotada
```

Na tela do computador, no final do processamento aparecerá a mensagem:

```
ATENCAO: emitidos 1 aviso(s).
```

A vantagem do AVISO sobre a MENSAGEM é que nem sempre o projetista examina a listagem de saída a procura de erros e avisos. O comando AVISO adverte o projetista diretamente na tela.

ERRO

Funciona de forma análoga ao aviso. Por exemplo:

```
SE %BITOLA = 0  
ERRO 'Bitola não fornecida'  
SAIDA  
FIM
```

Se a variável BITOLA valer zero, na listagem de saída aparecerá:

```
***** ERRO: Bitola não fornecida
```

e na tela do computador:

```
ATENCAO: Emitidos 1 erro(s)
```

LISTAR

Este comando permite listar nós, variáveis, distâncias e expressões aritméticas ou lógicas. A sintaxe é:

```
LISTAR nno1 nno2 nno3 nno4 ....  
nno1 A nno2 nno3 A nno4 ...
```

DISTANCIA nno1 nno2

DRETA nno1 nno2 nno3

NOS

VARIAVEIS

TUDO

EXPRESSAO valor

Listagem de nós

Pode-se listar os nós desejados (e suas coordenadas) um a um ou uma faixa de nno1 A nno2.

A listagem é feita sempre pelo sistema de coordenadas locais atuais. No caso de necessidade de listagem pelo sistema global, é necessário antes voltar a este sistema através do comando ORIGEM. Um exemplo de uso do comando:

```
LISTAR 1 10 15 30
```

Este comando lista os nós 1, 10, 15 e todos os existentes entre os números 20 e 30, inclusive. Todos os nós existentes podem ser listados através do comando

```
LISTAR NOS
```

A listagem se refere sempre aos nós locais do programa ou subprograma que acionou o comando. Nós (e outras variáveis) globais devem ser listados fora da seção DESENHO.

LISTAR DISTancia

A listagem de distância entre os nós 100 e 200 pode ser obtida pelo comando:

```
LISTAR DIS 100 200
```

LISTAR Distância de ponto a RETA

A distância de um ponto a uma reta formada por 2 pontos é obtida por:

```
LISTAR DRETA nno1 nno2 nno3
```

A distância listada pelo programa se refere à distância do nó nno1 à reta formada pelos nós nno2 e nno3.

LISTAR VARIÁVEIS

Todas as variáveis numéricas e alfanuméricas locais são listadas por este comando. A listagem de variáveis globais deve ser feita fora da seção DESENHO.

LISTAR TUDO

Este comando equivale a LISTAR NOS seguido de LISTAR VARIÁVEIS.

LISTAR EXPRESSÃO

Permite listar valores numéricos resultantes de expressões aritméticas (o que inclui expressões lógicas). Por exemplo:

```
LISTAR EXP (100 * SIN (30.5)) + (200 + 5)
```

```
LISTAR EXP CMP ('%TITULO', 'V101')
```

DEFINE

O comando DEFINE define parâmetros de controle do sistema. A maioria das definições possível já foi vista ao longo do manual.

DEFINE LISTA / NLISTA

Habilita ou desabilita a listagem de saída, gerada pelo programa. Se o arquivo processado for TESTE.DP, a listagem gerada, quando direcionada para o disco, será TESTE.LST.

Normalmente o DP lista todas as linhas do arquivo .DP, após a macro-substituição; os subprogramas tipo .DPS não são listados. As principais situações onde se habilita ou desabilita a listagem são:

Quando comandos do tipo REPETE ou ENQUANTO são colocados no programa .DP, pode-se desabilitar a listagem antes do comando e reabilita-las após o FIM, para que as iterações do laço não sejam todas listadas;

Quando a chamada de subprogramas é simplesmente simulada através de INCLUIs, pode-se desabilitar a listagem no início do arquivo de inclusão e reabilita-la no final;

Quando um subprograma .DPS está em fase de desenvolvimento, pode ser interessante habilitar a listagem no início do subprograma, para facilitar o exame dos erros encontrados.

DEFINE ERROS num

Define o limite máximo de erros que podem ser encontrados antes do DP interromper a execução de um programa. O limite existe, pois em linguagens interpretadas do tipo DP os erros se propagam em uma reação em cadeia, e a partir de um certo ponto se torna inútil a sua análise. Por exemplo:

```
21 @ DELTAX, DELTAY
```

```
*  
***** ERRO: Coordenadas esperadas  
32 @ 100, 0  
***** ERRO: No' indefinido
```

No primeiro erro, o projetista esqueceu os sinais "%" para efetuar a substituição das variáveis por seus valores. O DP acusou erro na interpretação das coordenadas. O segundo erro decorreu do primeiro, não foi causado diretamente. Em programas normais estes erros tendem a se propagar; por default o limite de erros antes do DP interromper a execução é de 15. O DEFINE ERROS modifica este limite, e o comando:

```
DEFINE ERROS 0
```

desliga este limite.

DEFINE ESCALA valor

Durante o processo de plotagem, o controlador de plotter divide todas as coordenadas de desenho pelo chamado fator de escala, resultando em um valor em centímetros, que a pena deverá se deslocar. As unidades de desenho, por convenção, são armazenadas internamente na escala 1:1.

No DP os tamanhos de texto não são definidos em termos de unidades de desenho, mas em centímetros plotados, o que é mais prático. Para que o tamanho dos textos seja dividido pelo fator de escala durante a plotagem, é necessário que antes sejam multiplicados pelo mesmo fator, para que tenham o tamanho em centímetros definido pelo projetista.

Por default, o fator de escala vale 50. Para redefinir este valor para 1:20, em um desenho em centímetros, fazemos:

```
DEFINE ESCALA 20
```

Imediatamente após a definição da escala, a variável global `_ESCALA` assume o valor definido.

DEFINE TABPLT 'nome'

No caso de plotagem inteligente, a tabela de plotagem pode ser definida por este comando. O tipo da tabela deve ser sempre .DAT. Por exemplo, um subprograma pode gerar um desenho que acessa a tabela PONTES.DAT na pasta \DP\PONTES com o comando:

```
DEFINE TABPLT ' \DP\PONTES\PONTES.DAT '
```

Toda vez que um desenho é gerado, internamente ele contém propriedades tais como: sistema, subsistema fator de

escala e, entre outros, o nome da tabela de plotagem inteligente. Estas propriedades são lidas automaticamente pelo editor de plantas e programas de plotagem.

Um recurso interessante é associar uma tabela de plotagem exclusiva para um determinado tipo de desenho; o nome da tabela, as convenções seguidas, e a sua utilização pode ser totalmente controlada pelo subprograma .DPS.

A variável global _TABPLT contém a última tabela de plotagem inteligente definida.

DEFINE IAPLIC num

O DP pode gerar desenhos de qualquer tipo de aplicação. Os programas de plotagem e extração de tabelas de ferros TQS estão preparados para tomar ações automaticamente conforme o tipo de sistema / subsistema. Por exemplo, o programa de plotagem elimina a letra C de um texto se escrita em seguida a um número entre os níveis 1 e 50 para desenhos de vigas. Este estranho procedimento apenas "limpa" o desenho de símbolos identificadores de certos tipos de ferros, que aparecem na tela, mas não na impressora ou plotter.

O parâmetro IAPLIC define o tipo de sistema associado a um desenho; o valor default é (9), que significa desenho genérico de armação de concreto. Os tipos de aplicação são:

Código	Sistema
(1)	Genérico
(2)	TQS-Formas
(3)	TQS-Lajes
(4)	TQS-Fundações
(7)	TQS-Vigas
(8)	TQS-Pilar
(9)	TQS-AGC&DP
(11)	TQS-Madeira
(15)	CORBAR
(16)	CORMAD
(17)	TQS-Alvest

A variável global _IAPLIC tem o número da aplicação atual definida.

DEFINE ISUBAPLIC num

Como já foi dito no parágrafo anterior, conforme o tipo de sistema / subsistema, podemos ter ações distintas no momento da plotagem: desde a “eliminação” de letras indicativas de dobras à escolha da tabela de plotagem.

Para cada sistema temos vários subsistemas. A tabela de correlação entre sistemas e subsistemas está na tabela de Sistemas e Subsistemas da página 70.

A variável global `_ISUBAPLIC` tem o número da aplicação atual definida.

DEFINE TAMTTX valor

O comando `TEXTO` pode ser usado sem a definição do tamanho de texto; neste caso, vale o definido por `TAMTTX`.

O tamanho default de texto é de . A variável global `_TAMTTX` tem o tamanho de texto atual.

DEFINE TAMTEI valor

O comando `EIXOS` rotula eixos por um tamanho fixo, definido por `TAMTEI`. Por default, `TAMTEI` vale 0.4. A variável global `_TAMTEI` contém o valor da última definição.

DEFINE NÍVEL num

Todos os elementos de desenho gerados pelo DP podem ter o seu nível explicitamente declarado pela palavra `NVD`. Quando esta declaração não é feita, o elemento é gerado com nível default, controlado pelo parâmetro `NÍVEL`. Por default, este nível vale zero; a variável global `_NIVEL` contém a definição atual do nível default.

DEFINE NIVCOR

O parâmetro `NIVCOR` atribui uma cor a um nível de desenho:

```
DEFINE NIVCOR nível cor
```

Parâmetros de cotagem

Os seguintes parâmetros controlam a cotagem feita pelo comando `COTAGEM`:

```
DEFINE TAMTCO valor Tamanho do texto de cotagem
```

```
DEFINE NCOTLCH Cotagem sem linha de chamada
```

```
DEFINE COTLCH Cotagem com linha de chamada
```

```
DEFINE COTEXE val Extensão da linha de chamada
```

```
DEFINE COTEXO val Gap da linha de chamada
```



```
DEFINE COTTSZ val Tamanho da marca da linha de cotagem
DEFINE COTDLE val Extensão da linha de cotagem
DEFINE COTNIV ival Nível de cotagem
DEFINE COTNIC ival Nível da linha de cotagem
DEFINE COTNIL ival Nível das linhas de chamada
DEFINE COTMUL val Multiplicador de dimensões
DEFINE COTPRC ival Precisão de cotagem, c/supressão de zeros
DEFINE COTBLO 'nome' Nome do bloco de marca cotagem
DEFINE BIBBLO 'nome' Biblioteca de blocos para bloco de marca
```

Definição de Bibliotecas

O termo "Biblioteca" no nosso caso designa um conjunto organizado de arquivos com determinada finalidade. Do ponto de vista prático, o DP permite que arquivos de subprogramas DPS, blocos de desenho e inclusões DP sejam separados em pastas, onde possam ser organizados e documentados sem interferir fisicamente com as pastas onde o projetista realiza seus projetos.

Quando em um processamento, um arquivo é referenciado, o DP pesquisa a localização do arquivo nas várias bibliotecas declaradas. Por default, o arquivo `INSTAL.DP` localizado na pasta `%_SUPORTE\DP` tem a declaração da localização da biblioteca DPS (`%_SUPORTE\DP\DPS`) e da biblioteca de blocos (`%_SUPORTE\BLOCOS\GERAIS`).

Os 3 tipos de biblioteca são definidos pelos comandos:

```
DEFINE BIBDP 'pasta(s) '
DEFINE BIBBLO 'pasta(s) '
DEFINE BIBINC 'pasta(s) '
```

que definem respectivamente, as bibliotecas de subprogramas (DPS) e máscaras, blocos de desenho e inclusões DP. A última definição é sempre armazenada nas variáveis globais `_BIBDP`, `_BIBBLO` e `_BIBINC`.

As pastas declaradas entre apóstrofes devem seguir as convenções do MS-DOS, podendo ser declarado mais de uma, separadas por ponto e vírgula, tal como:

```
DEFINE BIBDP '\DP\PROG;\DP\PROG\PONTES '
```

O modo mais prático de utilizar estes comandos é declarando as pastas de bibliotecas no arquivo de critérios, `%_SUPORTE\DP\INSTAL.DP`. Assim, as definições serão conhecidas automaticamente em todos os processamentos de programas DP.

DEFINE SEMENTE 'nome'

Conforme comentamos no capítulo 4.16. Semente de desenho, arquivos de desenho gerados pelo programa tem sempre como ponto de partida o chamado arquivo semente de desenho. Este arquivo tem pré-definido níveis de desenho a serem usados, cores, etc, sendo sobrepostos pelos definidos pelo Sistema / Subsistema. Por default, a semente de desenho é o arquivo %_SUPORTE\NGE\SEED.DWG. Havendo necessidade de mudar a semente, pode-se usar o comando acima, onde 'nome' é o nome da semente, conforme as convenções do MS-DOS, mas sem o tipo ".DWG".

A definição de semente deve ser feita sempre antes do comando DESENHO. A variável global _SEMENTE contém o nome da última semente definida.

DOS 'comando'

O comando DOS aciona um comando do MS-DOS entre apóstrofes. Por exemplo, suponha que o arquivo PRJ-0000.ING deva se copiado do pasta \DP\EXEC caso não exista na pasta atual:

```
DOS 'IF NOT EXIST DESARM.DAT COPY \TQSW\SUPORTE\NGE\DESARM.DAT'
```

Este é um tipo de controle que pode ser feito pelo próprio subprograma. Outro exemplo: Apagar os arquivos FOR1000.DWG, FOR1001.DWG, ... até o FOR1010.DWG:

```
REPETE I = 1010  
DOS 'DEL FOR%I.DWG'  
FIM
```