
AIMMS Function Reference - GMPProgressWindow Procedures and Functions

This file contains only one chapter of the book. For a free download of the complete book in pdf format, please visit www.aimms.com

GMP::ProgressWindow Procedures and Functions

AIMMS supports the following procedures and functions for displaying progress information in the Progress Window:

- `GMP::ProgressWindow::DeleteCategory`
- `GMP::ProgressWindow::DisplayLine`
- `GMP::ProgressWindow::DisplayProgramStatus`
- `GMP::ProgressWindow::DisplaySolver`
- `GMP::ProgressWindow::DisplaySolverStatus`
- `GMP::ProgressWindow::FreezeLine`
- `GMP::ProgressWindow::UnfreezeLine`

GMP::ProgressWindow::DeleteCategory

The procedure `GMP::ProgressWindow::DeleteCategory` deletes a progress category.

```
GMP::ProgressWindow::DeleteCategory(  
    Category          ! (input) a progress category  
)
```

Arguments:

Category
An element in the set `AllProgressCategories`.

Return value:

The procedure returns 1 on success, or 0 otherwise.

See also:

The routines `GMP::Instance::CreateProgressCategory` and `GMP::SolverSession::CreateProgressCategory`.

GMP::ProgressWindow::DisplayLine

The procedure `GMP::ProgressWindow::DisplayLine` writes one line with progress information in the Progress Window. The *lineNo* argument gives the number of the line in which the information has to be shown. The title contains a string that will be displayed on the left side of the line; the value will be displayed on the right side.

```
GMP::ProgressWindow::DisplayLine(
  lineNo,      ! (input) a line number
  title,       ! (input) a title
  value,       ! (input) a value
  [Category]   ! (optional) a progress category
)
```

Arguments:

lineNo

The number of the line in which the information has to be shown. Its value should be a number between 1 and the maximum number of lines available in the Progress Window (currently 6).

title

The string that will be displayed on the left side of the line.

value

The value that will be displayed on the right side of the line.

Category

An element in the set `AllProgressCategories`.

Return value:

The procedure returns 1 on success, or 0 otherwise.

Remarks:

- If the *Category* argument is used then the element should be created with the function `GMP::SolverSession::CreateProgressCategory`.
- To freeze (lock) a line the procedure `GMP::ProgressWindow::FreezeLine` should be called. To unfreeze it thereafter the procedure `GMP::ProgressWindow::UnfreezeLine` should be called.

See also:

The routines `GMP::ProgressWindow::DisplaySolverStatus`, `GMP::ProgressWindow::DisplayProgramStatus`, `GMP::ProgressWindow::DisplaySolver`, `GMP::ProgressWindow::FreezeLine`, `GMP::ProgressWindow::UnfreezeLine` and `GMP::SolverSession::CreateProgressCategory`.

GMP::ProgressWindow::DisplayProgramStatus

The procedure `GMP::ProgressWindow::DisplayProgramStatus` writes the program status (or model status) to the Progress Window.

```
GMP::ProgressWindow::DisplayProgramStatus(
    status,          ! (input) a status
    [Category],     ! (optional) a progress category
    [lineNo]        ! (optional) a line number
)
```

Arguments:

status

An element in the set `AllSolutionStates`.

Category

An element in the set `AllProgressCategories`.

lineNo

The number of the line in which the program status has to be displayed. The default is 7.

Return value:

The procedure returns 1 on success, or 0 otherwise.

Remarks:

- If the *Category* argument is used then the element should be created with the function `GMP::SolverSession::CreateProgressCategory`.
- The program status can also be displayed by using the procedure `GMP::ProgressWindow::DisplayLine` with title 'Program Status'.

See also:

The routines `GMP::Solution::GetProgramStatus`, `GMP::ProgressWindow::DisplayLine`, `GMP::ProgressWindow::DisplaySolverStatus` and `GMP::SolverSession::CreateProgressCategory`.

GMP::ProgressWindow::DisplaySolver

The procedure `GMP::ProgressWindow::DisplaySolver` writes the solver name to the Progress Window.

```
GMP::ProgressWindow::DisplaySolver(  
    name,           ! (input) a solver name  
    [Category]     ! (optional) a progress category  
)
```

Arguments:

name

A scalar string representing the solver name.

Category

An element in the set `AllProgressCategories`.

Return value:

The procedure returns 1 on success, or 0 otherwise.

Remarks:

If the *Category* argument is used then the element should be created with the function `GMP::SolverSession::CreateProgressCategory`.

See also:

The routines `GMP::ProgressWindow::DisplaySolverStatus`, `GMP::ProgressWindow::DisplayProgramStatus`, `GMP::ProgressWindow::DisplayLine` and `GMP::SolverSession::CreateProgressCategory`.

GMP::ProgressWindow::DisplaySolverStatus

The procedure `GMP::ProgressWindow::DisplaySolverStatus` writes the solver status to the Progress Window.

```
GMP::ProgressWindow::DisplaySolverStatus(  
    status,          ! (input) a status  
    [Category],     ! (optional) a progress category  
    [lineNo]        ! (optional) a line number  
)
```

Arguments:

status

An element in the set `AllSolutionStates`.

Category

An element in the set `AllProgressCategories`.

lineNo

The number of the line in which the solver status has to be displayed.
The default is 8.

Return value:

The procedure returns 1 on success, or 0 otherwise.

Remarks:

- If the *Category* argument is used then the element should be created with the function `GMP::SolverSession::CreateProgressCategory`.
- The solver status can also be displayed by using the procedure `GMP::ProgressWindow::DisplayLine` with title 'Solver Status'.

See also:

The routines `GMP::Solution::GetSolverStatus`, `GMP::ProgressWindow::DisplayLine`, `GMP::ProgressWindow::DisplayProgramStatus` and `GMP::SolverSession::CreateProgressCategory`.

GMP::ProgressWindow::FreezeLine

The procedure `GMP::ProgressWindow::FreezeLine` freezes (or locks) a line in the Progress Window.

```
GMP::ProgressWindow::FreezeLine(
    lineNo,          ! (input) a line number
    [totalFreeze],  ! (optional) a binary
    [Category]      ! (optional) a progress category
)
```

Arguments:

lineNo

The number of the line that should be frozen.

totalFreeze

If it equals 1 (the default) then the line will never change (until the procedure `GMP::ProgressWindow::UnfreezeLine` is called). If it equals 0 then the line will only change if a `GMP::ProgressWindow` procedure is called for this line.

Category

An element in the set `AllProgressCategories`.

Return value:

The procedure returns 1 on success, or 0 otherwise.

Remarks:

- If the *Category* argument is used then the element should be created with the function `GMP::SolverSession::CreateProgressCategory`.
- If the *Category* argument is not specified then this procedure will freeze a line in the general AIMMS progress category for displaying solver progress, or in the solver progress category of the generated mathematical program in case function `GMP::Instance::CreateProgressCategory` was called.

See also:

The procedures `GMP::Instance::CreateProgressCategory`, `GMP::ProgressWindow::DisplayLine`, `GMP::ProgressWindow::DisplayProgramStatus`, `GMP::ProgressWindow::DisplaySolverStatus`, `GMP::ProgressWindow::UnfreezeLine` and `GMP::SolverSession::CreateProgressCategory`.

GMP::ProgressWindow::UnfreezeLine

The procedure `GMP::ProgressWindow::UnfreezeLine` unlocks a frozen line in the Progress Window.

```
GMP::ProgressWindow::UnfreezeLine(  
    lineNo,          ! (input) a line number  
    [Category]      ! (optional) a progress category  
)
```

Arguments:

lineNo

The number of the line that should be freed.

Category

An element in the set `AllProgressCategories`.

Return value:

The procedure returns 1 on success, or 0 otherwise.

Remarks:

- If the *Category* argument is used then the element should be created with the function `GMP::SolverSession::CreateProgressCategory`.
- If the *Category* argument is not specified then this procedure will unfreeze a line in the general AIMMS progress category for displaying solver progress, or in the solver progress category of the generated mathematical program in case function `GMP::Instance::CreateProgressCategory` was called.

See also:

The procedures `GMP::Instance::CreateProgressCategory`, `GMP::ProgressWindow::DisplayLine`, `GMP::ProgressWindow::FreezeLine` and `GMP::SolverSession::CreateProgressCategory`.