Connectivity Updates in Maple 2025

DocumentTools:-RunWorksheet

- DocumentTools:-RunWorksheet can now execute Maple Flow documents. Values can be passed into the Maple Flow document by overriding the first assignment to the given variable, and the returned result is the value computed in the last math container in the document. In addition the output option allows retrieving other values computed in the document.
- The example worksheet RunWorksheetExample.flow contains the following math containers:

  a := 1

  b := 2a

  c := 3

  a + b^2 + c^3
- This example can be executed as follows:

```
> flowfile := FileTools:-JoinPath( ["example", "RunWorksheetExample.
  flow"], base=datadir ):
```

```
> DocumentTools:-RunWorksheet( flowfile );
```

$$32 \quad (1)$$

```
> DocumentTools:-RunWorksheet( flowfile, [a=-1,b=4] );
```

$$42 \quad (2)$$

```
> DocumentTools:-RunWorksheet( flowfile, [a=2], outputs=[b,c] );
```

$$[b = 4, c = 3] \quad (3)$$

- For more about Maple Flow, see about,MapleFlow.

CodeGeneration

- The options supported by CodeGeneration translators now includes a new source option, which permits a variety of new sources of Maple code to translated by CodeGeneration.
- Specifying source=component indicates that the input expression is a string referencing a Maple embedded component such as a Code Edit Region, the content of

which is the Maple code to be translated.

```
Primes  := [2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47];
S := 0:
for p in Primes  do S += 1/p; end do;

```

```
> CodeGeneration:-Python("CodeEditRegion1", source=component);
```

- Specifying source=inert specifies that the input is a Maple inert representation of the form produced by ToInert and accepted by FromInert which encodes the code to be translated.

This allows such inert expressions to be constructed and provided to CodeGeneration without risk of accidental evaluation.

```
> statseq := _Inert_STATSEQ(_Inert_ASSIGN(_Inert_NAME("x"),
  _Inert_INTPOS(1)), _Inert_ASSIGN(_Inert_NAME("y"), _Inert_INTPOS
  (2)), _Inert_ASSIGN(_Inert_NAME("z"), _Inert_SUM(_Inert_NAME("x"),
  _Inert_NAME("y"))));
```

$$statseq := \_Inert\_STATSEQ(\_Inert\_ASSIGN(\_Inert\_NAME("x"), \_Inert\_INTPOS(1)), \_Inert\_ASSIGN(\_Inert\_NAME("y"), \_Inert\_INTPOS(2)), \_Inert\_ASSIGN(\_Inert\_NAME("z"), \_Inert\_SUM(\_Inert\_NAME("x"), \_Inert\_NAME("y")))) \quad (4)$$

```
> CodeGeneration:-Python(statseq);
```

```
x = 1
y = 2
z = x + y
```

Markdown

- As of this version, Maple has a Markdown parser integrated. Markdown is a lightweight document formatting language. The parser can be used by invoking the Worksheet:-FromMarkdown command.

```
> with(Worksheet):
```

```
> FromMarkdown("This is some *markdown*.", output = insert);
```

```
> file := FileTools:-JoinPath(["example", "markdown.md"], ':-base' =
  ':-datadir'):
```

```
> line := FileTools:-Text:-ReadLine(file):

> while line <> NULL do
     printf("%s\n", line);
     line := FileTools:-Text:-ReadLine(file);
  end do:

> FileTools:-Text:-Close(file);
```

- The following command will display the result as a new worksheet.

```
> FromMarkdown(file, output=display);
```