

R2009b New Features for Stateflow

R2009b

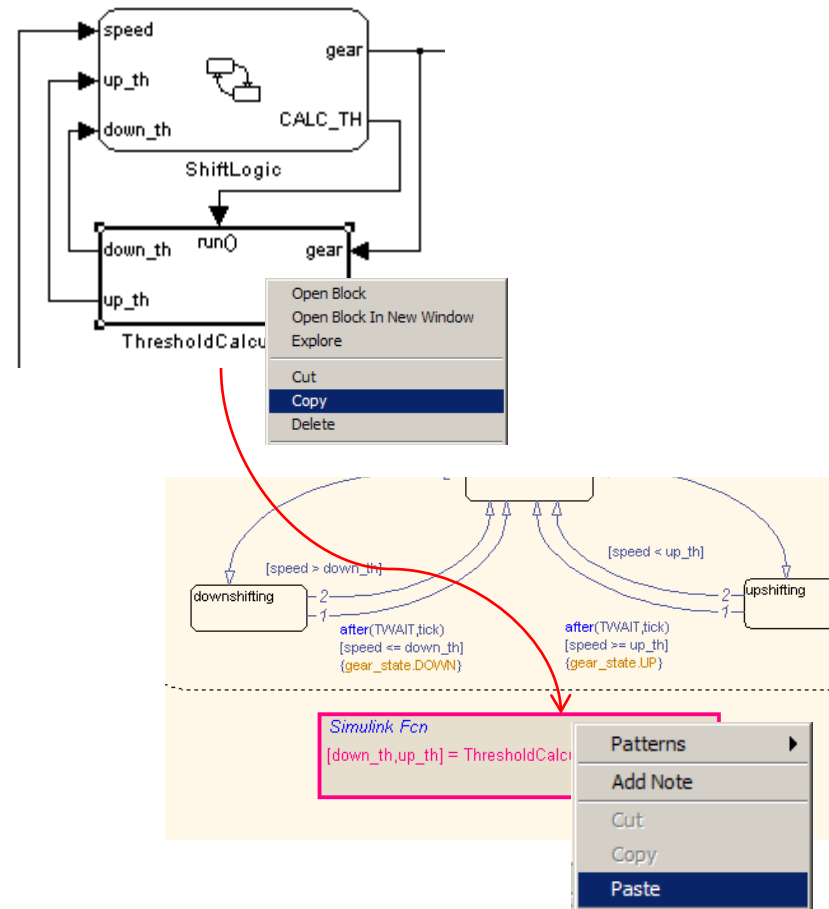
New Features

- Easily copy and paste Simulink functions into Stateflow and vice versa
- Create switch-case flowchart patterns within Pattern Wizard GUI
- Create generated code that contains easy-to-read switch statements
- Know exactly what Simulink does before the simulation runs
- Create large-scale charts with as many events as you need
- Easily step through complex logic when debugging charts

Copy and Paste Between Stateflow and Simulink

Easily copy and paste Simulink functions into Stateflow and vice versa

- Copy and paste Simulink function-call subsystems into Stateflow and vice versa
- Easily embed Simulink functions within your Stateflow charts
- This operation previously required several steps – now two steps

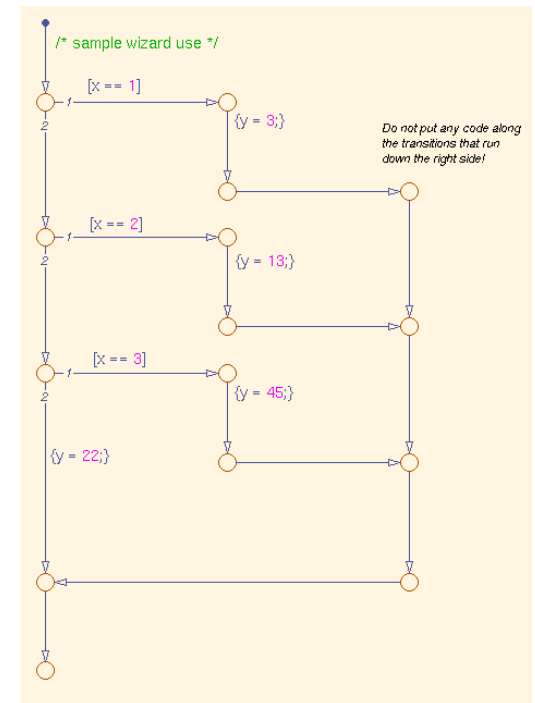
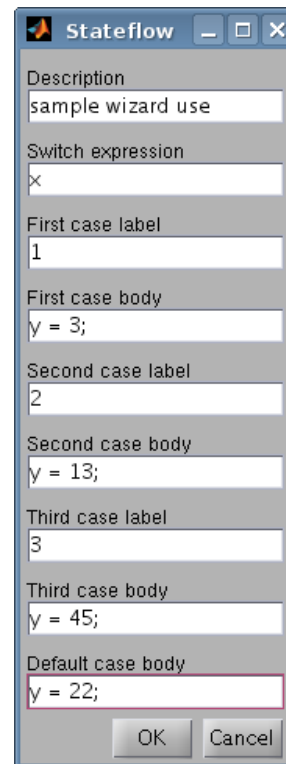


Only Two steps required to transfer an existing Simulink function into Stateflow

Switch Statement in Pattern Wizard

Create switch-case flowchart patterns with Pattern Wizard GUI

- New pattern added to the Stateflow Pattern Wizard to construct flow graphs from switch-case statements
- If generating code with Real-Time Workshop Embedded Coder, resulting flow graphs will generate switch-case statements



Switch Statements in Generated Code


Create generated code that contains easy-to-read switch statements

- Switch statements can now be generated from flow charts
- Generates clearer code that more accurately represents the user's intent
- Requires Real-Time Workshop Embedded Coder

```

/* Model step function */
void mif_to_switch_no_extra_junction_step(void)
{
    if (mif_to_switch_no_extra_juncti_U.In1 == 1) {
        mif_to_switch_no_extra_juncti_Y.Out1 = 1;
    } else {
        if (mif_to_switch_no_extra_juncti_U.In1 == 2) {
            mif_to_switch_no_extra_juncti_Y.Out1 = 2;
        } else {
            if (mif_to_switch_no_extra_juncti_U.In1 == 3) {
                mif_to_switch_no_extra_juncti_Y.Out1 = 3;
            } else {
                mif_to_switch_no_extra_juncti_Y.Out1 = 4;
            }
        }
    }
}

```



```

/* Model step function */
void mif_to_switch_no_extra_junction_step(void)
{
    int32_T rtb_y;

    switch (mif_to_switch_no_extra_juncti_U.In1) {
        case 1:
            rtb_y = 1;
            break;

        case 2:
            rtb_y = 2;
            break;

        case 3:
            rtb_y = 3;
            break;

        default:
            rtb_y = 4;
            break;
    }

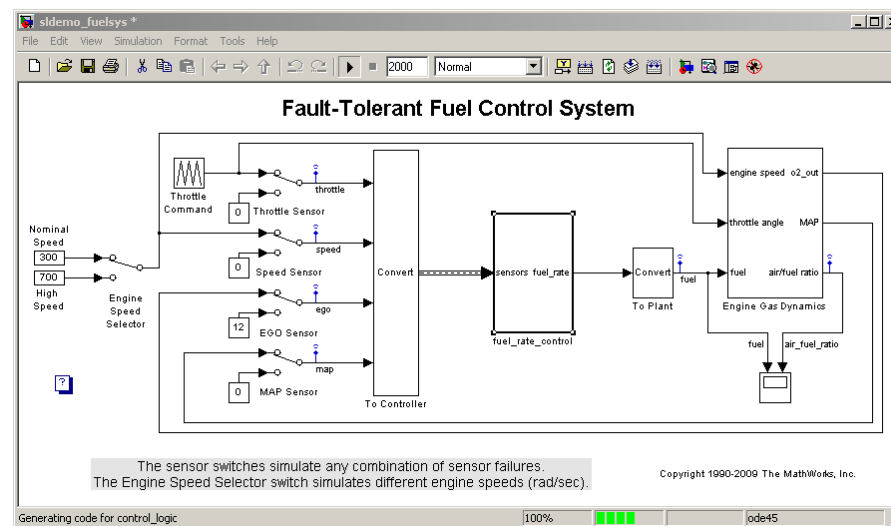
    mif_to_switch_no_extra_juncti_Y.Out1 = rtb_y;
}

```

Simulink Status Bar Shows Stateflow Compilation Status

Know exactly what Simulink does before the simulation runs

- Stateflow compilation status has been added to the Simulink status bar
- Know when Stateflow charts are compiled

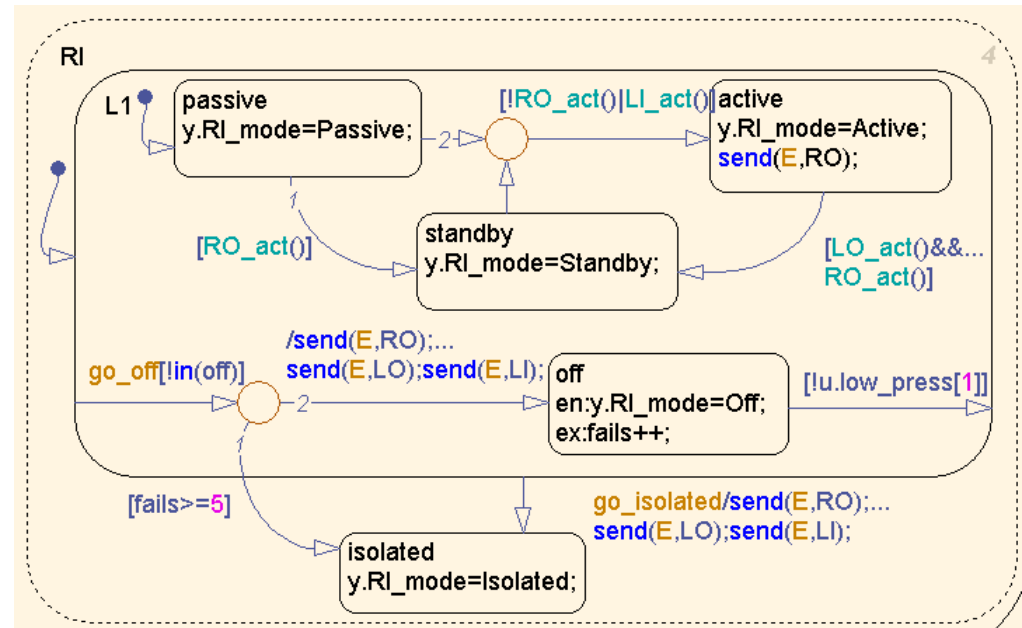


Generating code for control_logic

Support for More than 254 Events

Create large-scale charts with as many events as you need

- Stateflow now has no limit on the number of events that can be used in a Stateflow chart
 - Actual limit imposed by C-compiler is $2^{31} - 1$
- Facilitates large-scale modeling in Stateflow charts

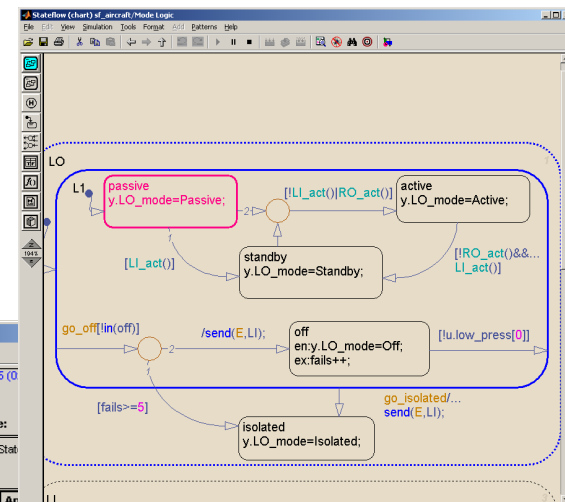


Improved Stateflow Debugger

Easily step through complex logic when debugging charts

- Zoom level is now maintained during debugging
- Retain state context and minimize visual disruptions

Before
R2009b



Stateflow Debugging sf_aircraft

File View

Stopped: Just after activation of State passive #25 (0.000000)

Executing: Mode Logic #18 (0.0:0.0)

Current Event: Simulink call event

Simulink Time: 0.000000 (Major Time Step) Coverage:

Breakpoints: Chart Entry Event Broadcast State

Continue Step Break

Error checking options

State Inconsistency Data Range

Transition Conflict Detect Cycles

Breakpoints Browse Data Active States

MATLAB Command:

R2009b

