



**PathMATE
Transformation Map for C
Version 8.0.2
Release Notes**

November 5, 2009

Pathfinder Solutions
www.pathfinder-mdd.com
+1 508-568-0068

Table Of Contents

1. Introduction	1
Online Documentation	1
Demo	1
2. Technical Support	2
3. Installation	2
4. Compatibility with Other Pathfinder Solutions Releases	2
5. Impact to Existing Projects	2
6. New Features	3
Version 8.0.2.....	3
7. Defects Repaired	5
Version 8.0.2.....	5

1. Introduction

Online Documentation

PathMATE help is available from the Eclipse Help system - see the PathMATE topics under the Eclipse environment Help->Help Contents.

Additional Online product documentation and technical information may be found in the doc subdirectory of the base PathMATE installation (default c:\pathmate\doc). Many of the more sophisticated features are documented in feature-focused Technical Notes in the TechNotes subdirectory below doc. The documentation is stored in Adobe Acrobat (PDF) format. An Acrobat reader can be downloaded free of charge from www.adobe.com.

Demo

In addition to the product demo available from www.pathfinder.mdd.com, be sure to see the PathMATE Quick Start Guide for RSM, and the PathMATE Quick Start Guide (for Rose users), located in the doc directory.

2. Technical Support

Pathfinder Solutions Technical Support
Telephone: 888-662-7284 x103
Email: support@pathfindermd.com

3. Installation

PathMATE Transformation Map for C must be installed from the same root directory as base PathMATE so the C transformation maps in the platform models delivered with PathMATE can find their template files. If base PathMATE is installed in its default location, then the PathMATE Transformation Map for C default location is correct.

4. Compatibility with Other Pathfinder Solutions Releases

Requires PathMATE 8.0.2 or higher

5. Impact to Existing Projects

None

6. New Features

Version 8.0.2

REQ02014 – Service handle copy optimization

Rather than copy service handles during invocation, they can be marked as non-transient to reduce overhead. See c:\pathmate\doc\TechNotes\TechNote_Non-Transient Incident Handles.pdf for more information on using this feature.

REQ02010 – Support FineGrainedTime

Support the FineGrainedTime datatype in C. See c:\pathmate\doc\TechNotes\TechNote_UsingFineGrainedTime.pdf for more information on using this feature.

REQ2048 – Support SuppressGeneration

Set the SuppressGeneration marking on entire domains or individual domain services to exclude these model elements from the generated code.

REQ2049 - Optimize FIND CLASS for Singletons

This new feature allows you mark a class with Singleton set to "TRUE", and a simple static variable is created to point to the single class instance. Anywhere a plain FIND CLASS is used (no WHERE clause), it is translated into code that uses this variable. A limitation is you should only create one instance of a Singleton. No optimization is applied if you use a FIND CLASS with a WHERE clause (not necessary for a singleton) or if you navigate via an association to the singleton.

To test this with SimpleOven apply this marking:

```
Object,*.MicrowaveCooking.Door,Singleton,TRUE
```

REQ 2050 - Synchronous Signal Dispatch

This new feature allows you mark a signal with SynchronousDispatch set to "TRUE" and everywhere it is generated a simple synchronous function call is generated to directly call the appropriate transition handler instead of instantiating and queueing an event. It also has the added benefit of not actually creating or deleting an event (sw_incident) instance. Limitations:

- the signal can't have parameters
- the signal can only trigger one transition on its state machine
- the signal can only trigger a transition to a plain state (not a history state)
- the generate can't have a delay

To test this with SimpleOven apply this marking:

```
Event , * .MicrowaveCooking .Door .IsOpen , SynchronousDispatch , TRUE
```

REQ2051 - Single Task – Tighter Trimming

The self-trimming of multi-tasking mechanisms has been extended to eliminate additional ram elements and code fragments not needed for single-task contexts.

REQ2052 - Selectable Trimming of Command Line Artifacts

The compiler symbol `PATH_NO_COMMAND_LINE` can be set to mask code artifacts that allow the access of command line parameters.

REQ2053 - Specifiable Domain Initialization Order

The `InitializeSequence` marking can be specified for domains to control the order of when their initialize actions is invoked. The default for `InitializeSequence` is "last", and it is expected the numeric values are used to override this, such as "1", "2", etc.

REQ2054 - Buffer Manager Static Initialization

All memory apparently allocated dynamically via model constructs like `CREATE` and `GENERATE` can instead draw from a pool of statically pre-allocated memory blocks, eliminating the run-time impacts of dynamic memory while retaining model-level flexibility. The new module `sw_buf_mgr_static.c` statically instantiates all buffer manager constructs, including one instance of `SW_BufferManager`, and instances of `SW_AllocationCluster_` and `SW_ClusterManager_` as needed. For further detail on this feature please refer to the detailed instructions in this module.

7. Defects Repaired

Version 8.0.2

Design - C

- BUG01996** Production error in `obj_static_find_row` when using properly formatted csv file.
- BUG2047** Correct compiler error in generated Spotlight instrumentation when creating instance of object marked with an `IndexID`