

MC-3020 3.1 Release Notes

Overview

MC-3020 3.1 completes the support for BridgePoint Model Debugger and PreExisting Instances (PEIs) defined in data. Other optimizations and a few bug fixes are included.

How to use this document

This document has been divided into the following sections for easier reference.

Support for MC-3020

Support resources available for DesignPoint® and BridgePoint® products and MC-3020 specifically. (BridgePoint® and DesignPoint® are registered trademarks of Project Technology, Inc.)

Supported Platforms

Multiple platforms and operating systems are supported by this release.

Installation Procedure

Follow these instructions to successfully install for the first time or upgrade from a previous version.

Change Highlights

Read on overview of what is new for 3.1.

Known Issues

Learn about known bugs or modeling restrictions.

Support for MC-3020

World class support is provided for the BridgePoint and DesignPoint products. Support is available via the web, email, telephone, and fax. Each of the support options is described at our support web address: *Project Technology Support* (<http://www.projtech.com/support.html>).

Web Support

The Project Technology, Inc. web site is available 24 hours a day, 7 days a week. You can access the following features at the Project Technology Customer Service web site by clicking on the "Customers Only" icon.

Problem Report Submission

Use a pre-defined form to submit a problem report and be assured you are providing enough information for our support technicians to help you. The information that you provide on this form is emailed to our support organization at <support@projtech.com>.

Enhancement Idea Submission

Submit your enhancement ideas directly from your workstation as you think of them. This information will be immediately entered in our enhancements database for consideration in future releases. Your ideas drive our product development ONLY when we hear them.

Technical Tips

Read technical tips that will help you get more out of your Project Technology software products. New tips will be posted on a regular basis and old tips will be archived for your easy reference.

Application Notes

Read technical papers that treat topics of interest in depth. Many are written by Project Technology developers, instructors, and consultants, but we encourage you to submit your own contributions.

BridgePoint Users Mailing List

We have extended our popular users' mailing list service by adding a list for BridgePoint users. This mailing list allows you to communicate with a community of BridgePoint users, who are solving many of the same problems you encounter each day. You'll have to sign-up for this service.

The "Customers Only" section utilizes authentication technology to limit this service to Project Technology, Inc. customers who have support contracts.

To access this area, you will need your Project Technology Customer ID and your web site password. You'll find your Customer ID on packing lists and/or invoices you have received from us. Your initial password is the first eight characters (or all, if eight or fewer) of the name of the city your product was shipped to, capitalized appropriately, with spaces compressed out.

Project Technology, Inc. is your first stop for customer support. If what you need is not available there, the web site of ROX Software, Inc. has additional support material. Visit *ROX Software, Inc.* (<http://www.roxsoftware.com>) This site contains sample models, generated code and an

online version of the MC-3020 Users Guide. All of the most recent and up-to-date material is found at this web site.

Email

The customer support email address is <support@projtech.com>. This is the preferred method for non-critical questions.

Phone

Customers within the US can call our toll-free number, 800-482-3853. The direct line, which can be used by US and International customers is +1-520-544-0808.

Facsimile

Our support fax number is +1-520-544-2912. Please specify preferred method for response.

Supported Platforms

MC-3020 translation runs on all platforms supported by BridgePoint. See your BridgePoint release notes for details on versions of the operating systems. The list includes Solaris and Windows XP/2000/NT/98SE/ME.

Targets for running the translated models include almost any platform with an ANSI compliant C compiler. Compilers known to work include:

- gcc (which supports Linux, Windows, Mac, H8 and many, many microprocessors)
- Microsoft Visual C/C++ (Visual Studio)
- Texas Instruments TI470 C
- Sparcworks Workshop
- Tasking 8051, C167 and others
- Borland C/C++
- Yellowsoft for H8
- Cosmic
- others

Installation

Find here step by step instructions for installing MC-3020 onto the development platform.

Documentation is found in the `docs` directory of the installed model compiler. Directions presented by the installation program that differ from this manual supercede the instructions listed here. The installation program will always provide the most recent and timely installation information. The text in the on-line user guide at www.roxsoftware.com (<http://roxsoftware.com/>) will be even more up to date.

Uninstall any previous version of MC-3020 before beginning the installation process.

Note MC-3020 will need to be uninstalled and reinstalled after upgrading BridgePoint. MC-3020 relies on components within BridgePoint. This uninstall/reinstall must occur any time that BridgePoint is altered substantially (e.g upgrade, change in location of model repository). ROX_* environment variables need to be set to point to the correct (new) locations. Specifically, the variables ROX_PT_HOME and ROX_PT_REPOSITORY_HOME need to link to the correct directories. These variables use the old 8.3 format names. **DIR /X** is the DOS command to see the underlying file and path names.

Note that MC-3020 may not install onto a system that does not have the appropriate licensing.

Windows XP/2000/NT/98

Follow these steps and the directions on the screen. Cygwin, ksh and MC-3020 will be installed as needed.

1. On XP/2000/NT, log in as administrator.
2. Insert MC-3020 compact disk into the CD ROM drive of the development platform.
3. If necessary, select Program->Run under the Start button.
4. Run the Setup program (ksh `winsetup.ksh`) on the CD.
5. Follow the instructions.
6. Reboot.

Cygwin Support

The MKS Toolkit has served MC-3020 well. However, superior technology exists in the freely available Cygwin. *Cygwin* (<http://www.cygwin.com/>). MC-3020 3.1 has Cygwin (and ksh) support right on the CD. Installation is automatic.

If you already have Cygwin installed, the MC-3020 installation CD will detect it and not install Cygwin. However, you may need to upgrade your Cygwin installation to include packages such as `make`, `perl` and `xsltproc` if you do not have them.

Permissions issues with Cygwin can pose an issue. Once Cygwin is installed executables may be configured to allow only the owner of the file to read them and execute them. By default,

BridgePoint is set such that only an administrator can run the `repository_bin` executables. Keep this in mind if you seem to be having difficulty getting some build steps to execute correctly. Access can be granted to the executables using the **chmod** command (in the bin directory in question type **chmod ugo+rx *.exe**).

The 3.1 release CD of MC-3020 has automatic installation of Cygwin and ksh. These instructions will serve to document where these files are expected to be.

If you run MC-3020 on Windows, uninstall MKS.

MC-3020 uses the Korn shell in the build environment. Korn shell does not (as of this writing) come standard with Cygwin. A distribution of the public domain Korn shell (**ksh**) is provided with MC-3020. Find it in `mech/windows/pdksh-5.2.14-1.tar.gz` (or similar) in the mc3020 installation directory. Untar the the archive and place the executable in the `/bin` of your Cygwin installation. (CD should have done this automatically.)

Solaris

Follow these steps and the directions on the screen.

1. Insert MC-3020 compact disk into the CD ROM drive of the development platform.
2. Ensure that CD is mounted as an ISO-9660 type filesystem.
3. Run the `setup.ksh` program on the CD.
4. Follow the instructions.
5. Establish environment variables as directed by the installation program.

Change Highlights

MC-3020 3.1 delivers UML Debugger capability and support for preexisting instances (PEIs) defined in data. In concert with these two major enhancements, support for initialization functions is supplied. Some additional coloring options are added and a few performance enhancements. The installation package has been enhanced to automatically install Cygwin and ksh. And a few bugs have been fixed.

Upgrade Notes

There are some changes to the build environment in 3.0 that require attention. This section will outline these changes. See the installation appendix for more details.

Support for Cygwin

The MKS Toolkit has served MC-3020 well. However, superior technology exists in the freely available Cygwin (*Cygwin* (<http://www.cygwin.com/>)). MKS is no longer supported.

Cygwin provides most of what MKS supplies and more. The utilities needed for the build environment are part of Cygwin with the exception of **ksh** which is supplied in the MC-3020 distribution (in `mech/windows/pdksh*`).

The installation CD configures Cygwin and ksh onto Windows platforms automatically.

Coloring Class Operations (Transformers)

In 3.1, the coloring option `TagTransformerGeneration` is no longer supported.

`TagTransformerGeneration` was originally supplied as a testing and debug color to provide scaffolding for normally unmodeled function. Now that BridgePoint Model Builder supports class operations standard, it does not make sense to translate only those operations colored with `TagTransformerGeneration`. This coloring function will now do nothing.

As of version 3.1, all class (and instance) operations will be translated by default.

`TagClassOperationTranslationOff` is supplied to disable transformer/operation translation. This allows integration of hand-coded operations (transformers).

This change may you require to edit the coloring files of existing models in order for translation to occur correctly.

System File Name Changes

Internal issue 91 documented a problem with the naming of some system include files. (See below.) The names of generated include files are now different. This change will require **make clean_all_src**, **make clean_sys_src** and **make all** to retranslate existing system nodes.

Enhancements

Preexisting Instances Defined in Data

Support for preexisting instances defined in data is in MC-3020 3.1.

The new make target **make dom_xml** will perform an XML extraction from the generation database. This extracted XML fully describes/defines the user application system. The format of the XML is suitable for delivery to the BridgePoint Model Debugger.

In 3.0, the Instance subsystem was added to the MC-3020 meta-model. This subsystem schema has tables for instances, links, attribute values, etc.

Following is an example of the gain (reduced code size) achieved by defining preexisting instances in data. Below are size measurements of a test case. The first case shows the output of the **size** command for a normal init-object-style generation (`gcc/Cygwin`). The second shows using `PEIsDefinedInData`.

To interpret the table:

- *text*: code space
- *data*: constants

- *bss*: global and static variables

Table 1. PEI Comparison

> > > > >

text	data	bss	dec	hex	filename
27136	2560	11104	40800	9f60	s2t/bin/rox.exe
20992	10240	3744	34976	88a0	s2s/bin/rox.exe

This is a conservative test case. There are no state machines in the model. However there is procedural action language to access all of the instances either created or defined in data. The test case creates a total of 84 instances of 23 different objects with 39 links.

This simple test demonstrates a 15% reduction in code space and a significant speed-up in terms of initialization. Models with many preexisting instances will see very large reductions in code size.

Constructs Supported

In MC-3020 3.1, preexisting instances can be defined in data for any normal modeling construct supported by MC-3020 For example, data initialization can be defined for:

- simple 1-1 associations
- simple 1-* associations
- associatives (1-1, 1-*, *-*)
- subtypes
- supertypes
- sub/supertypes
- simple reflexive associations
- simple 1-* reflexives
- reflexive associatives
- asymmetric reflexive associatives

BridgePoint Model Debugger

Code generated with MC-3020 is tooled for use with BridgePoint Model Debugger. Use the coloring function `TagModelDebuggingOn` to enable the generation of this tooling code.

Real-time debugging of translated models is now possible. See the appropriate BridgePoint Model Debugger documentation from Project Technology, Inc. for details.

With the addition of debugging with BridgePoint Model Debugger another touch point between plain ASCII character sets and Unicode character sets is exposed. During a debug-enabled translation, the model compiler access and embeds various constructs from within the BridgePoint UML model into the generated code. The constructs include the names and values of UML modeling elements. Many of these elements can be encoded in international character sets (for example Shift_JIS).

Domain Function Coloring

Two new colors have been added to facilitate the process of preparing the XML representation of preexisting instances. These colors are also useful in general.

`TagInitializationFunction` marks function(s) to be executed during bring-up. This effectively replaces initialization objects. (Initialization objects are still supported, but discouraged.) Functions tagged to run at initialization bring-up can also be called during run time processing. Functions tagged for initialization are run in alphabetical order.

`TagFunctionTranslationOff` disables the translation of functions. This is convenient for enabling/disabling test cases. It is especially nice for PEI support. An init function can be tagged for running with model debugger. It can then be tagged off for the production compile using preexisting instances (PEIs) defined in data.

Create/Delete/Link/Unlink Optimization

MC-3020 has extended its optimization capability with regard to code reduction. Create, delete and link accessors are not generated unless actually needed. Previous versions of MC-3020 were able to optimize out some delete and link accessors. Create accessors were generated unconditionally.

With the advent of preexisting instances defined in data it is possible for a system to have instances that were not created by action language in the system but were defined in data. As such, some classes may not need create accessors.

During first pass translation, MC-3020 identifies all creates, deletes, links and unlinks that are needed. All accessors are assumed to be not needed until action language is identified that uses the accessor. Only those accessors actually used in action language are generated.

With a static instance population composed totally of preexisting instances defined in data, it is possible to have a large number of instances that are rich with attributes and related with many other instances that have no create, delete, link or unlink accessors!

Empty Handle Checking

Additional checks for empty handles have been eliminated. This will result in smaller code. The relate/unrelate (link/unlink) code formerly tested the input parameters unconditionally. These checks are now generated only when `TagEmptyHandleDetectionOn()` has been executed. All of the checks that had formerly been made are still available with this color.

Tagging Operations and Transformers

As of version 3.1, all class (and instance) operations will be translated by default.

`TagClassOperationTranslationOff` is supplied to disable transformer/operation translation. This allows integration of hand-coded operations (transformers).

New User Callout Functions

New user callout functions `UserNonSelfEventQueueEmptyCallout` and `UserSelfEventQueueEmptyCallout` were added. These give the user the opportunity to get control of processing when event queues are empty.

Other Enhancements

The installation package now includes automatic installation of Cygwin and ksh.

The `schema/colors` directory now has coloring files with file extension `.jap`. These are Japanese language versions of the coloring files. Some have English place-holders for the new coloring options.

Fixes

Internal Issue 91: `e_object.h` interferes with using `E` for registered name.

If `E` is used as the registered domain prefix, a file `e_objects.h` will be generated. This interferes with `e_objects.h` in the model compiler mechanism include directory.

This is fixed. Names of system includes are changed. Note that this will require retranslation of existing system nodes after upgrading to 3.1.

Schema Changes

The MC-3020 meta-model has been enhanced to support these features and fixes. The changes are as listed:

- A `CharacterEncoding` attribute has been added to `S_DOM`.
- An `Order` attribute has been added to `S_SYNC` to allow for alphabetical ordering of initialization functions.
- `PEIsDefinedInData` is added to `O_OBJ` to tag those objects that are eligible to have preexisting instances defined in data.
- The `IsFixedPopulation` attribute is added to `O_OBJ` to identify those classes having static instance populations (no creates or deletes during normal execution).
- `ForcePriorityEvents` has been added to support the Model Debugger's need to have sender information on all events. Normally only prioritized events carry sender information.

