Code Contracts Release Notes

**May release**

Assembly Mode

* Makes it explicit if the rewriter is used on release builds or not. See section 5.1.1 in the documentation.

Contract Argument Validator Methods

* Factor legacy if-then-throw sequences and use them as if they were primitive contract methods

Contract Abbreviator Methods

* Factor non-legacy requires contracts and use them as if they were primitive contract methods

ContractOption attribute

* Provides on/off control over inheritance and runtime checking at assembly, type, or method granularity

Fixes

* Installer setting of environment variable problem fixed
* Proper extracting source text from Requires(Of …)(…) in VB
* Proper extracting source text from If .. Then legacy in VB
* Pop up box due to missing source context in error list fixed
* Some performance improvements in rewriter
* Complicated generics are now better handled in assembly reader
* Numerous fixes to static checker precision/correctness
* Recursion guards are not omitted when recursion is obviously not present

**March release**

Installer

* Should now properly recognize Architect edition
* Should install on build servers

Contract Extraction

* Better warnings when Invariant method contain bad contract uses

Runtime Checker

* Turn off invariant checking during any method which in the end will check its invariants
* Check invariant on exceptional exit from exceptions advertised via EnsuresOnThrow
* Emit proper pdb information to avoid spurious uncovered blocks in generated code when code coverage metrics are employed. For contracts, coverage is achieved when every condition is evaluated by the tests
* Fix incorrectly generated call-site wrapper for internal protected methods
* Don’t turn invariants on auto-properties that override/implement into pre/post
* Parameters appearing in ensures are now implicitly treated as wrapped inside OldValue
* /throwOnFailure is now the default

Contract Reference Assemblies

* Contract reference assemblies are now placed in a sub-directory called CodeContracts of the regular project bin output directory instead of the output directory itself. This change addresses an issue where some frameworks/tools (e.g. ASP.Net) would try to load the contract reference assemblies and fail.
* Added contracts to System.Web, System.Security

Static Checker

* Out and by-ref parameters of pure methods are now properly treated as assigned after the call
* Calling Monitor methods causes the static checker to assume that more of the world has changed than previously.
* Various performance and precision improvements and fixes
* Using a method not marked with Pure in a contract now only emits one warning, rather than repeated warnings and is treated as pure by the analyzer
* ContractVerification attribute is now recognized on properties rather than just on the getter/setter.
* Fix to string format issues in output messages

January 12, 2010 (notes for public release)

Contracts

* Silverlight 4 support added, Silverlight2 dropped.
* We now have a set of contract reference assemblies for the different target frameworks (3.5, 4.0, Silverlight 3.0, Silverlight 4.0). This allows contracts on newer APIs and avoids problems where the rewriter added the wrong references to the target assembly.
* Contract invariant methods now must be private.
* Invariants on auto properties are turned into pre and post conditions of the corresponding setter and getter. This permits using auto properties with contracts.
* Added more contracts to System.Linq, System.Net, System.Collections, System.Data, System.Math namespaces
* Fixed issues of referencing v3.5 assemblies with contracts from v4.0 projects
* Tweaks to extraction of contracts from VB constructors

Runtime Checker

* Makes sure contract invariant methods are not called directly
* Object invariants are now delayed until the end of the constructor to avoid checking them prematurely
* Fixes to contract inheritance that would generate bad IL.
* Avoid FxCop warnings about catching all exceptions in OldValue expressions
* Initial support for ContractIgnoredAttribute added (see documentation)
* Fix in rewriting attributes with arrays of enums
* Fixed codegen issues when removing Assert/Assumes from code

Documentation Generation

* Fixed path problems
* Fixed inheritance problems

Static Checker

* Improved handling of IsNan
* Improved error messages now include the condition in all cases (not just for other assemblies)
* Improved interval analysis
* Improved handling of box operations
* Squigglies now work in VS2010
* Fixed handling of unicode error messages in the output
* Better handling of IntPtr and UIntPtr
* Better handling of inferred mutations at method calls
* Masking of warnings at the assembly level using SuppressMessage now supported for global warnings

October 9, 2009 (notes for public release)

**Bug fixes**

Runtime Checker

* Fixed object invariant inheritance for runtime checking
* Old value evaluation now under try-catch to guard against conditionally needed old values
* Stricter checking of usage of contract methods
* Fixed generic method instantiation related to nested classes
* Better source contexts for some errors

Static Checker

* Fixed wrong reporting of unreached code
* Improved precision for handling lengths of generic collections
* Fixed several bugs caused by large numbers

Framework Contracts

* Added contracts in System.Windows.Forms
* Added contracts in System.Data

August 31st, 2009 (notes for public release)

**Feature updates**

* No API changes!
* Silverlight 3 supported.
* **New tool**:
  + Contract documentation generation via XML docs and Sandcastle
* **Static checking**
  + Support for floating point reasoning
  + Check for redundant assumptions supported
* **Runtime checking**
  + Object invariant can be split into multiple methods
* More contracts for framework assemblies

**Bug fixes**

* Lots of runtime instrumentation bug fixes
* Precision improvements for static checker
* Bad Contract Reference Assemblies for .NET have been fixed. This should fix problems with “method not found” failures and other verification problems in the rewritten code.
* Contract reference assemblies should now work for Silverlight 2 and 3. Warnings related to X509 certificate types should be gone.
* Runtime contract instrumentation and static checker made more robust.

May 18, 2009: Major refactoring, moved to minor version 2

* Added recursion guards during contract evaluation
* Added support for SuppressMessage attributes
* Fixed hang bug in IL /PDB reader
* Fixed handling of Abstract class contracts when generics and closures are involved
* Fixed handling of generic methods with interface/abstract class contracts
* Added support for Call-site Requires checking.
* RequiresAlways is deprecated
* Requires<TException> added
* Reworked AsmMeta to always normalize all contract to the \_\_ThreeArgVersion
* Reworked the rewriter to handle source strings and user messages consistently
* Reworked the rewriter to emit missing methods for runtime failure behavior
  + Split into RaiseFailureEvent and ShowFailure
  + Copies ContractException from library when internal
  + Emits Requires<TException> reflection code
  + Added /throwonfailure option to overwrite default assert box
  + Wired this option through to UI
* Added System.Linq.Expressions to Core contracts
* Fixed bug in key container handling in msbuild script
* Fixed bug in Heap analysis when seeing ldfieldtoken and ldmethodtoken. Added regression.

April 9, 2009 (notes for public release)

* More contract reference libraries for System.Xml, System.Data, System.Windows.Forms
* Contract class allows partially trusted callers
* Silverlight and WPF builds now supported
* Runtime Checker
  + Pure is now inherited and need not be repeated on overrides/implementations
  + Fixed problem with level of checking being applied incorrectly to inherited contracts
  + Provided more levels of checking
  + Allow erasure of all contracts not on public surface of an assembly
  + Fixed debugging issue with Silverlight applications
  + Fixed Ensures source context issues to improve stepping accuracy
  + Fixed problem with contract reference assemblies for executables
* Static Checker
  + Pure functions of multiple arguments are now handled
  + Strongly typed resources are assumed non-null
  + Unsatisfiable requires are now flagged
  + VB constructors can now have contracts (fixed extraction problem)
  + Property Pane custom arguments can now override default arguments

March 20, 2009

* Debugging issue with Silverlight apps resolved
* WPF Build issue work around
* VB constructors (New) can now have contracts
* Static checker warns about unsatisfiable preconditions
* Fixed command argument order so custom user options actually override fixed options
* More contracts for framework libraries

March 5, 2009

* Changed default contract failure behavior:
  + By default a ContractException is thrown
  + The behavior is fully configurable via the Contract.ContractFailed event hook
* More Contract libraries for the BCL
  + System.Core.dll now contains annotations for the extension methods in Enumerable and Queryable
* New static checking option for arithmetic checks appears in UI
* All Code snippets are now available in VB as well
* Support for Silverlight is now here. A special Microsoft.Contracts.dll built against Silverlight’s mscorlib is supplied in the installation directory. Adding the reference from a Silverlight project properly proposes this one.
  + Debugging of the rewritten code however is currently a problem, as VS fails to attach.
* Bug fixes
  + Out of band contract purity now visible
  + Changed declarative assembly build to use signing as original to allow InternalsVisibleTo accesses to succeed
  + Improved precision in static analyzer

Release 1.1.0113.1 1/13/2009

* Static analyzer deals better with boxing/unboxing in generic contexts. Removes false positives.
* Static analyzer can be run in the background
* Optional squigglies on source contexts.

Release 1.1.0106.2 1/6/2009

* More major fixes to rewriter handling generic contracts on interfaces. The rebasing of generic nested types wasn’t properly done.
* Removed warnings about impure methods by teaching checker more for the pure framework methods.
* Pulled in fixes from Herman to maintain undocumented dll flags

Release 1.0.1215.1 12/15/2008

* Major fixes to runtime contract rewriter to support inheriting interface contracts with closures.

Release 1.0.1125.1 11/25/2008

* Fixed out-of-date documentation (name change CodeContract -> Contract)

Release 1.0.1124.1 11/24/2008

* System.Diagnostics.Contracts.CodeContract class renamed back to Contract. Hopefully this is the final name.

Release 1.0.1113.7 11/13/2008

* Fixed interface contracts. They now work both in runtime and static checking
* Made rewriter robust even if no contract library is referenced
* Added UI option to select runtime contract checking level
* Many fixes to static checker

Release 1.0.1024.3 10/24/2008

* Fixed bug in non-null analysis warning unnecessarily for receivers of instance methods

Release 1.0.1017.5 10/17/2008

* Changed executable names to
  + ccrewrite.exe (for foxtrot.exe)
  + ccrefgen.exe (for asmmeta.exe)
  + cccheck.exe (for clousot.exe)
* ILMerged as many libraries as possible to cut down on the number of dlls shipped.
* Removed a VS dependency of Decompiler

Release 1.0.0815.1 8/15/2008

* Various rewriter fixes for running on mscorlib
* Clousot fixes and precision improvements
* Alternate mscorlib now passed to Clousot.

Release 1.0.0806.2 8/6/2008

* Fixes problems when runtime and static checking are enabled simultaneously
* Build Reference Assembly box now actually builds the reference assembly
* Added free form text box for additional options to static analyzer

Release 1.0.0801.1 8/1/2008

* BCL API review caused some changes to the Contract API
  + Namespace is now System.Diagnostics.Contracts
  + Class name is now CodeContracts
  + Old is now OldValue

You need to perform some global search and replace on your source.

* Microsoft.Contracts.dll has changed location in install
  + Please remove existing reference from your projects and add the reference anew.
  + Rebuilds are required