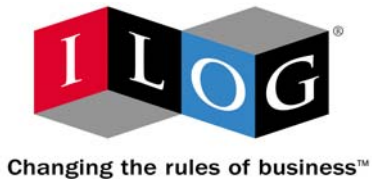

ILOG CPLEX 11.0

Release Notes

September 2007

COPYRIGHT NOTICE

General Use Restrictions

Trademarks

document version 11.0

Table of Contents

ILOG CPLEX 11.0 Release Notes

These release notes highlight improvements and new features in ILOG CPLEX 11.0. Please review these notes before using ILOG CPLEX 11.0.

Announcements: Port Changes

For a complete list of machine types and library formats (including version numbers of compilers and JDKs) see the file *yourCPLEXhome*/mptable.html.

Mac OS Darwin Port Available

ILOG CPLEX 11.0 is available on an additional x86 platform: Mac OS X 10.4, also known as Darwin 8, with the GNU g++ 4.0 (32-bit) compiler and JDK 5.0.

JDK 1.4 Deprecated

JDK 1.4, also known as J2EE 1.4, has been deprecated. That is, this development kit will no longer be supported after ILOG CPLEX 11.0. Instead, ILOG urges customers who currently use this deprecated development kit to migrate to a more recent version, such as JDK 6, as recommended by Sun Microsystems.

Visual Studio .NET 2003 Deprecated

Visual Studio .NET 2003 has been deprecated. That is, this compiler version will no longer be supported after ILOG CPLEX 11.0. Instead, ILOG urges customers who currently use this deprecated compiler to migrate to a more recent version, such as Visual Studio 8.

Conversion Notes for All Users

For users of earlier versions of ILOG CPLEX, the following topics offer guidelines for easy migration to ILOG CPLEX version 11.0. (Users of prior versions must first apply the Conversion Notes accompanying previous versions of ILOG CPLEX before upgrading to this one.)

The following topics cover changes that may affect all users of ILOG CPLEX 11.0:

Changes in Existing Parameters

Table 1 on page 7 lists parameters that have changed in behavior or possible values since the previous release. Navigational links in the table take you to the documentation of the changed parameter in the *ILOG CPLEX Parameters Reference Manual.*

The thread parameters for MIP optimization, barrier optimization, and strong branching have been consolidated into one parameter (`Threads`, `CPX_PARAM_THREADS`). Documentation of the threads parameter explains its effects on those activities that were formerly controlled by separate parameters.

Table 1 *Changed parameters*

Concert Parameter	Callable Library	Interactive Optimizer	Comment
`AdvInd`	`CPX_PARAM_ADVIND`	`advance`	setting 2 has new effect for MIP
`EpInt`	`CPX_PARAM_EPINT`	`mip tolerances integrality`	maximum value is now 0.5
`Symmetry`	`CPX_PARAM_SYMMETRY`	`preprocessing symmetry`	new more aggressive settings: 4, 5
`NodeLim`	`CPX_PARAM_NODELIM`	`mip limits nodes`	behavior has changed for 0 (zero) and 1 (one)
`Threads`	`CPX_PARAM_THREADS`	`threads`	new default; interaction with new parallel parameter
`MIPThreads`	`CPX_PARAM_MIPThreads`	`mip limits threads`	REMOVED: use threads parameter
`BarThreads`	`CPX_PARAM_BARTHREADS`	`barrier limits threads`	REMOVED: use threads parameter
`StrongThreadLim`	`CPX_PARAM_STRONGTHREADLIM`	`mip limits strongthreads`	REMOVED: use threads parameter

Dynamic Search

ILOG CPLEX 11.0 offers a new and innovative algorithmic approach for mixed integer programs. This new feature, known as *dynamic search*, is available by default. For many models, dynamic search finds feasible and optimal solutions more quickly than conventional branch & cut.

A new parameter (MIPSearch, CPX_PARAM_MIPSEARCH) allows you to turn off this feature in applications where you judge that conventional branch & cut search is more appropriate.

Query callbacks and control callbacks are **not** compatible with dynamic search. (For more about this point, see *Informational Callbacks* on page 8 in these release notes.) The error CPXERR_MIPSEARCH_WITH_CALLBACKS will be raised if a callback attempts to alter the search path while dynamic search is in progress. For more about dynamic search and callbacks, see also *Using Optimization Callbacks* on page 445 in the *ILOG CPLEX User's Manual*.

Informational Callbacks

Informational callbacks, new in ILOG CPLEX 11.0, provide a simplified approach to monitoring progress of your MIP solve. Informational callbacks are compatible with recent ILOG innovations such as dynamic search and parallel MIP optimization.

In previous versions of ILOG CPLEX, only query and control callbacks were available. Query callbacks (also known as diagnostic callbacks) and control callbacks are **not** compatible with dynamic search (the new default search). At default settings, in the presence of query callbacks or control callbacks, ILOG CPLEX 11.0 turns off dynamic search and turns off deterministic parallel MIP optimization.

In Concert Technology, new informational callback classes are available. For samples of their use, see:

- ilomipex4.cpp
- MIPex4.java
- MIPex4.cs or MIPex4.vb

In the Callable Library (C API), new routines, CPXsetinfocallbackfunc and CPXgetinfocallbackfunc, enable you to install and access informational callbacks. For a sample of their use, see mipex4.c.

The error CPXERR_IN_INFOCALLBACK will be raised if a callback calls any routine other than CPXgetcallbackincumbent or CPXgetcallbackinfo (the two routines allowed in informational callbacks in the C API).

Examples illustrating the use of callbacks have been revised to show you both how to replace old style callbacks with informational callbacks and how to set search parameters in the presence of callbacks. These revised examples are available in the standard distribution in *yourCPLEXhome*/examples in all APIs.

Using Optimization Callbacks on page 445 in the *ILOG CPLEX User's Manual* has been extensively revised to document these changes in callbacks and their interaction with dynamic search. Likewise, *Parallel Optimizers* on page 491 in the *ILOG CPLEX User's Manual* has been revised with respect to callbacks and parallel optimization.

Node Limits

Behavior has changed for the node limit parameter (`NodeLim`, `CPX_PARAM_NODELIM`).

Now when this parameter is set to 0 (zero), ILOG CPLEX completes processing at the root; that is, cuts are created and heuristics are applied at the root, but no nodes are created. When the parameter is set to 1 (one), ILOG CPLEX branches at the root; that is, nodes are created but not solved.

Previously, 0 (zero) solved the initial root LP relaxation and stopped before applying any cuts, heuristics, or other processing at the root node. The behavior for all other settings (including default) remains unchanged.

Interruption and Termination

ILOG CPLEX 11.0 has been improved, so that a user's application can intercept interrupt-signals and direct ILOG CPLEX to behave appropriately.

In Concert, there is a new class of objects representing termination. These objects to terminate the search can be attached to a CPLEX object and then called from inside or outside callbacks generally. These objects are useful in place of control callbacks, for example, when the user wants to terminate the search rather than redirect it.

- `IloCplex::Aborter` in the C++ API;
- `IloCplex.Aborter` in the Java API;
- `Cplex.Aborter` in the .NET API.

In the Callable Library, the new routine `CPXsetterminate` is available to enable your application to interrupt, abort, or terminate ILOG CPLEX gracefully.

File Format Removed: SOS

The SOS file format, deprecated in a previous release, has been removed. Instead of this formatted, ASCII-based file format for special ordered sets (SOS), ILOG recommends instead MPS or LP format.

The following routines of the Callable Library are no longer available:

- `CPXsoswrite`
- `CPXsosread`
- `CPXreadcopysos`

Error Codes Removed

The following error codes have been removed. Conditions that gave rise to these codes no longer exist.

- 1269 `CPXERR_ABORT_CONDITION_NO`
- 1100 `CPXERR_BOUNDS_INF`
- 3011 `CPXERR_BOUNDS_INT`
- 3022 `CPXERR_SEMI_BDS`

Error Codes Added

The following error codes have been added.

- 1123 `CPXERR_PRESLV_TIME_LIM`
- 1234 `CPXERR_THREAD_FAILED`
- 1804 `CPXERR_IN_INFOCALLBACK`
- 1805 `CPXERR_MIPSEARCH_WITH_CALLBACKS`
- 1806 `CPXERR_LP_NOT_IN_ENVIRONMENT`
- 3024 `CPXERR_NO_SOLNPOOL`
- 3414 `CPXERR_FILTER_VARIABLE_TYPE`

Conversion Notes for Concert Technology Users

A new version of Concert Technology (Concert Technology 2.5) accompanies this release of ILOG CPLEX 11.0. This new version requires you to recompile and relink your Concert Technology applications. In addition, the following conversion notes are of interest to users of Concert Technology.

- *Conversion Notes for Users of the C++ API* on page 10
- *Conversion Notes for Users of the Java API* on page 11

Conversion Notes for Users of the C++ API

The method `IloSolution::isBound` has been **deprecated**. It will be removed in a future release. If you are using this deprecated method, ILOG recommends that you use `IloSolution::isFixed` instead.

The following methods were deprecated in a previous release of the ILOG Concert Technology C++ API. They have been **removed** from the ILOG Concert Technology 2.5 C++ API.

- `IloRange::setCoef(const IloNumVar var, IloNum val)`
- `IloRange::setCoef(const IloNumVarArray vars, IloNumArray coefs)`
- `IloObjective::setCoef(const IloNumVar var, IloNum value)`
- `IloObjective::setCoef(const IloNumVarArray vars, IloNumArray vals)`

Conversion Notes for Users of the Java API

The following methods of the **Java API** were deprecated in ILOG CPLEX version 8.0. They have been removed from this version of the product. Use the parameters `RootAlgorithm` and `NodeAlgorithm` instead of these removed methods.

- `IloCplex.getRootAlgorithm`
- `IloCplex.setRootAlgorithm`
- `IloCplex.getNodeAlgorithm`
- `IloCplex.setNodeAlgorithm`

Conversion Notes for Callable Library Users

These changes in the Callable Library may affect existing ILOG CPLEX applications:

- *Environment and Problem Object Mismatch Raises Error* on page 11
- *CPXnewcols: Default Upper Bound* on page 12

Environment and Problem Object Mismatch Raises Error

In a Callable Library (C API) application, any attempt to use a problem object in an environment other than the environment where the problem object was created will raise the error `CPXERR_LP_NOT_IN_ENVIRONMENT`.

In the Callable Library, the `lp` pointer to a problem object (regardless of the type of the problem) returned by `CPXcreateprob` must remain within the environment returned by `CPXopenCPLEX`.

CPXnewcols: Default Upper Bound

`CPXnewcols(env, lp, 1, &obj, NULL, NULL, "B", "newbin")` now creates a new binary variable `newbin` with objective `obj`, lower bound 0 (zero) and upper bound 1 (one). Formerly, the upper bound was infinite by default.

New Features for All Users

In addition to the new features introduced with their implications for conversion in *Dynamic Search* on page 7 and *Informational Callbacks* on page 8, here is more detail about other major new features in this release:

Solution Pool

A new feature of ILOG CPLEX 11.0, known as the solution pool, enables you to generate and store multiple solutions to a mixed integer programming (MIP) problem. This new feature is available in all application programming interfaces (APIs) as well as in the Interactive Optimizer.

For an introduction to the solution pool, see *Solution Pool: Generating and Keeping Multiple Solutions* on page 301 of the *ILOG CPLEX User's Manual*. That introduction shows you how to use new methods or routines to generate multiple solutions and keep them in the solution pool. It also explains the interaction between conventional MIP optimization and the solution pool. In addition, the new chapter demonstrates how to use new parameters to manage the solution pool, as documented in the *ILOG CPLEX Parameters Reference Manual:*

- `PopulateLim`, `CPX_PARAM_POPULATELIM`
- `SolnPoolCapacity`, `CPX_PARAM_SOLNPOOLCAPACITY`
- `SolnPoolIntensity`, `CPX_PARAM_SOLNPOOLINTENSITY`
- `SolnPoolGap`, `CPX_PARAM_SOLNPOOLGAP`
- `SolnPoolAGap`, `CPX_PARAM_SOLNPOOLAGAP`
- `SolnPoolReplace`, `CPX_PARAM_SOLNPOOLREPLACE`

Sample applications in *`yourCPLEXhome`*`/examples` show you how to invoke the populate procedure and how to manage the solution pool.

- `ilopopulate.cpp`
- `Populate.java`
- `Populate.cs or Populate.vb`
- `populate.c`

There are also examples in that chapter of the user's manual to show various approaches to filtering the solutions accumulated in the solution pool. For additional information about solution pool filters, see also *Filtering the Solution Pool* on page 323 in the *ILOG CPLEX User's Manual* and *FLT File Format: Filter Files for the Solution Pool* on page 39 in the *ILOG CPLEX File Format Reference Manual*.

Tuning Tool

To help you analyze whether default parameter settings provide the best performance for your particular model or models and to aid you in adjusting parameter settings efficiently, ILOG CPLEX 11.0 offers a new performance tuning tool. This tool is available in all application programming interfaces (APIs) as well as in the Interactive Optimizer.

For an introduction to the tuning tool, see *Tuning Tool* on page 161 of the *ILOG CPLEX User's Manual*. That chapter includes sample sessions in the Interactive Optimizer, as well as references to examples available in the standard distribution of the product. Additionally, there are sample applications using the tuning tool in *yourCplexhome*`/examples`:

- `ilotuneset.cpp`
- `TuneSet.java`
- `TuneSet.cs or TuneSet.vb`
- `tuneset.c`

Deterministic Parallel MIP

ILOG CPLEX 11.0 extends the capabilities of the Parallel MIP Optimizer to include a new deterministic mode of operation in addition to the existing opportunistic mode.

In this context, *deterministic* means that multiple runs with the same model at the same parameter settings on the same platform will reproduce the same solution path and results. In contrast, *opportunisitc* search skips synchronization between threads and thus entails less waiting and consequently performs faster. Lack of synchronization often provides improved performance but leads to variations in the search path and consequently possibly different solution vectors, though not a difference in the optimal objective value.

A new parameter (`ParallelMode` or `CPX_PARAM_PARALLELMODE`) allows you to control which mode ILOG CPLEX uses to solve your model. This new parameter has three possible settings: automatic (where ILOG CPLEX decides which mode to apply), deterministic, and

opportunistic. In most cases, the automatic setting of this parameter selects deterministic mode if your platform supports multiple threads.

Parallel Optimizers on page 491 in the *ILOG CPLEX User's Manual* has been extensively revised to document changes due to the introduction of deterministic parallel mode. See particularly *Parallel MIP Optimizer* on page 497 for more detail.

New Algorithm for MIQCPs

ILOG CPLEX 11.0 offers a new algorithm for the solution of quadratically constrained mixed integer programs (MIQCPs). A new parameter (`MIQCPStrat`, `CPX_PARAM_MIQCPSTRAT`) sets the strategy that ILOG CPLEX uses to solve these problems.

This parameter lets you control the type of relaxation used to solve a MIQCP. At the default setting of 0 (zero), ILOG CPLEX automatically chooses a strategy based on characteristics of your problem.

When you set this parameter to the value 1 (one), you tell ILOG CPLEX to solve a QCP **relaxation** of the model at each node. This is the behavior available in previous versions of ILOG CPLEX.

When you set this parameter to the value 2, you tell ILOG CPLEX to attempt to solve only an **LP relaxation** of the model at each node.

For more general information about the types of quadratically constrained models that ILOG CPLEX solves, see *Identifying a Quadratically Constrained Program (QCP)* on page 239 in the *ILOG CPLEX User's Manual*.

Feasibility Pump

ILOG CPLEX 11.0 makes the feasibility pump heuristic available to you through a new parameter (`FPHeur, CPX_PARAM_FPHEUR`). This heuristic has demonstrated its ability to find an initial feasible solution even in certain very hard mixed integer programming problems (MIPs). This new parameter allows you to select how the feasibility pump looks for a feasible solution.

For more detail about the feasibility pump heuristic, see research reported by Fischetti, Glover, and Lodi (2003, 2005), by Bertacco, Fischetti, and Lodi (2005), and by Achterberg and Berthold (2005, 2007).

New Status Code for FeasOpt

In each application programming interface (API), there is a new status code, `CPXMIP_FEASIBLE` in the enumeration `CPX_STAT_FEASIBLE`. This new status code is returned whenever FeasOpt in phase 1 finds a model to be feasible and installs a feasible solution.

Solution Information Available for MIPs and FeasOpt

Existing methods and routines, such as `getValues`, have been extended so that solution information is available in ILOG CPLEX 11.0 for mixed integer programming problems (MIPs) as well as for FeasOpt.

New Parameters

The release notes introduce these new parameters. More detail about each of them is available in the *ILOG CPLEX Parameters Reference Manual* and in the *ILOG CPLEX User's Manual*.

Table 2 *New Parameters*

Parameter in Concert Technology	Parameter in Callable Library	Parameter in Interactive Optimizer
EachCutLim	CPX_PARAM_EACHCUTLIM	mip limit eachcutlimit
FPHeur	CPX_PARAM_FPHEUR	mip strategy fpheur
MIPSearch	CPX_PARAM_MIPSEARCH	mip strategy search
MIQCPStrat	CPX_PARAM_MIQCPSTRAT	mip strategy miqcpstrat
ParallelMode	CPX_PARAM_PARALLELMODE	parallel
PopulateLim	CPX_PARAM_POPULATELIM	mip limits populate
SolnPoolCapacity	CPX_PARAM_SOLNPOOLCAPACITY	mip pool capacity
SolnPoolReplace	CPX_PARAM_SOLNPOOLREPLACE	mip pool replace
SolnPoolGap	CPX_PARAM_SOLNPOOLGAP	mip pool relgap
SolnPoolAGap	CPX_PARAM_SOLNPOOLAGAP	mip pool absgap
TuningDisplay	CPX_PARAM_TUNINGDISPLAY	tune display
TuningMeasure	CPX_PARAM_TUNINGMEASURE	tune measure
TuningRepeat	CPX_PARAM_TUNINGREPEAT	tune repeat
TuningTiLim	CPX_PARAM_TUNINGTILIM	tune timelimit
ZeroHalfCuts	CPX_PARAM_ZEROHALFCUTS	mip cuts zerohalfcut

New Features in Concert Technology

In addition to the new features of general interest to all users of ILOG CPLEX 11.0, the following topics introduce new features in Concert Technology:

Status Output Operator in C++ API

There is a new output operator for the enumeration `IloCplex::CplexStatus` in the **C++ API**.

Visitor for Extractables in C++ API

There is a new class of visitors for extractable objects `IloExtractableVisitor` in the **C++ API**. Instances of this class support introspection of extractable objects in an instance of `IloModel`.

Iterator for Parameter Set in Concert Technology

A group of parameters can be treated as a set. To aid you in managing the parameters in a set, ILOG CPLEX provides an iterator over a parameter set.

- In the C++ API, the iterator is an instance of the class `IloCplex::ParameterSet::Iterator`.
- In the Java API, the iterator is an instance of the class `java.util.Iterator`.
- In the .NET API, the iterator is an instance of the class `System.Collections.IEnumerator`.

For samples of a parameter set and its iterator in use, see these examples in the standard distribution of the product:

- `ilotuneset.cpp`
- `TuneSet.java`
- `TuneSet.cs`
- `TuneSet.vb`

New Features in the Callable Library

In addition to the new features of general interest to all users of ILOG CPLEX 11.0, the following topic may be of particular interest to users of the Callable Library (C API).

Data Cleaning Tool in the Callable Library

ILOG CPLEX 11.0 offers a tool to change small values of coefficients in a model to zero. You may find this tool helpful in reducing numerical error introduced by floating-point and finite-precision arithmetic. The practice implemented by this tool is also known as *zero-ing out negligible coefficients*.

In the Callable Library (C API), use the routine `CPXcleanup`. It accepts a tolerance (that is, an epsilon value) within which small coefficients will be zero-ed out in a problem that your application has already created with `CPXcreateprob`.

New Features in the Interactive Optimizer

In addition to the new features of general interest to all users of ILOG CPLEX 11.0, the following topics may be of particular interest to users of the Interactive Optimizer:

- *Data Cleaning Tool in the Interactive Optimizer* on page 18
- *Zero-ing out the Objective* on page 18

Data Cleaning Tool in the Interactive Optimizer

ILOG CPLEX 11.0 offers a tool to change small values of coefficients in a model to zero. You may find this tool helpful in reducing numerical error introduced by floating-point and finite-precision arithmetic. The practice implemented by this tool is also known as *zero-ing out negligible coefficients*.

In the **Interactive Optimizer**, after you have read or entered a problem and detected negligible values among the coefficients, use the command `change values`. The Interactive Optimizer will then prompt you for a tolerance (that is, an epsilon value) within which you want small values to be changed to 0 (zero).

Zero-ing out the Objective

It is now possible to set the value of coefficients in the objective function to zero in the Interactive Optimizer. (This change was already available in the APIs of ILOG CPLEX.) This practice is also known familiarly as *zero-ing out the objective*. It does not delete the

objective function, which is necessary to the optimization algorithms. The command to use in the Interactive Optimizer to achieve this effect is the following:

```
change delete constraint 0
```

In a problem in which the objective function includes a quadratic term and thus is a problem of type QP, this change of setting the objective function to zero will result in a change as well in the type of problem.

Documentation for IDEs

ILOG CPLEX 11.0 provides reference documentation that can be integrated into and accessed from integrated development environments (IDEs) such as Microsoft Visual Studio or Eclipse.

For users of Microsoft **Visual Studio**, the documentation to support IntelliSense is available (as in previous releases) in the XML files accompanying `ILOG.CPLEX.dll` and `ILOG.Concert.dll`.

For users of **Eclipse**, the documentation to support tooltips is available:

- on UNIX and GNU/Linux platforms, in *`yourCPLEXinstallation`*`/doc/doc/html/refjavacplex/html`
- on Microsoft Windows, in `C:\ILOG\CPLEX110\doc\refjavacplex`.

Index