ILOG CPLEX 9.0 Parameters

October 2003

Table of Contents

Parameters of ILOG CPLEX

The behavior of ILOG CPLEX is controlled by a variety of parameters that are each accessible and settable by the user. This manual lists these parameters and explains their settings in the ILOG CPLEX Component Libraries and the Interactive Optimizer. It also explains how to read and write parameter settings of the Callable Library to a file in *Saving Parameter Settings to a File* on page 7.

The following methods set and access parameters for objects of the Concert Technology class `IloCplex` in C++ and Java:

```
setParam
getParam
getMin
getMax
getDefault
setDefaults
```

The names of the corresponding accessors in the class `Cplex` in C#.NET follow the usual conventions of names and capitalization in that language and framework.

Callable Library programs (C and other languages) access and set parameters with the following routines:

`CPXgetdblparam`	Accesses a parameter of type double
`CPXsetdblparam`	Changes a parameter of type double
`CPXinfodblparam`	Gets the default value and range of a parameter of type double
`CPXgetintparam`	Accesses a parameter of type integer
`CPXsetintparam`	Changes a parameter of type integer
`CPXinfointparam`	Gets the default value and range of a parameter of type integer
`CPXgetstrparam`	Accesses a parameter of type string
`CPXsetstrparam`	Changes a parameter of type string
`CPXinfostrparam`	Gets the default value of a parameter of type string
`CPXsetdefaults`	Resets all parameters to their standard default values

Parameter Names

In the parameter table, each parameter has a name (that is, a symbolic constant) to refer to it within a program.

- For the Callable Library these constants are capitalized and start with `CPX_PARAM_`; for example, `CPX_PARAM_ITLIM`. They are used as the second argument in all parameter routines (except `CPXsetdefaults` which does not require them).
- For C++ applications using Concert Technology, the parameters are defined in nested enumeration types for Boolean, integer, floating-point, and string parameters. The enum names use mixed (lower and upper) case letters and must be prefixed with the class name `IloCplex::` for scope. For example, `IloCplex::ItLim` is the `IloCplex` equivalent of `CPX_PARAM_ITLIM`.
- For Java applications using Concert Technology, the parameters are defined as final static objects in nested classes called `IloCplex.BooleanParam`, `IloCplex.IntParam`, `IloCplex.DoubleParam`, and `IloCplex.StringParam` for Boolean, integer, floating-point, and string parameters, respectively. The parameter object names use mixed (lower and upper) case letters and must be prefixed with the appropriate class for scope. For example, `IloCplex.IntParam.ItLim` is the object representing the parameter `CPX_PARAM_ITLIM`.
- For C#.NET applications using Concert Technology, the parameters follow the usual conventions for capitalizing attributes and defining scope.

An integer that serves as a reference number for each parameter is shown in the table. That integer reference number corresponds to the value that each symbolic constant represents, as found in the `cplex.h` header file, but it is strongly recommended that the symbolic constants be used instead of their integer equivalents whenever possible, for the sake of portability to future versions of ILOG CPLEX.

Correspondence of Parameters

Some parameters available for the Callable Library are not supported as parameters for `IloCplex`. In particular:

- Logging output is controlled by a parameter in the Callable Library (`CPX_PARAM_SCRIND`), but when using Concert Technology, you control logging by configuring the output channel
  - `IloCplex::out` in C++
  - `IloCplex.out` in Java
  - `Cplex.Out` in C#.NET
- The parameters:

  ```
  IloCplex::RootAlg
  IloCplex::NodeAlg
  ```

  are used where parameters `CPX_PARAM_STARTALG` and `CPX_PARAM_SUBALG` would be used for the Callable Library.

Saving Parameter Settings to a File

It is possible to read and write a file of parameter settings with the Callable Library. The file extension is `.prm`. The Callable Library routine `CPXreadcopyparam` reads parameter values from a file with the `.prm` extension. The routine `CPXwriteparam` writes a file of the current non-default parameter settings to a file with the `.prm` extension. Here is the format of such a file:

```
CPLEX Parameter File Version <number>
  <parameter_name>   <parameter_value>
```

ILOG CPLEX reads the entire file before changing any of the parameter settings. After successfully reading a parameter file, the Callable Library first sets all parameters to their default value. Then it applies the settings it read in the parameter file. No changes are made if the parameter file contains errors, such as missing or illegal values. There is no checking for duplicate entries in the file. In the case of duplicate entries, the last setting in the file is applied.

When you write a parameter file from the Callable Library, only the non-default values are written to the file. String values may be double-quoted or not, but are always written with double quotation marks.

The comment character in a parameter file is #. ILOG CPLEX ignores the rest of the line.

The Callable Library issues a warning if the version recorded in the parameter file does not match the version of the product. A warning is also issued of a non-integral value is given for an integer-valued parameter.

Here is an example of such a file:

```
CPLEX Parameter File Version 9.0
CPX_PARAM_EPPER                 3.45000000000000e-06
CPX_PARAM_IISIND                1
CPX_PARAM_OBJULIM               1.23456789012345e+05
CPX_PARAM_PERIND                1
CPX_PARAM_SCRIND                1
CPX_PARAM_WORKDIR               "tmp"
```

Parameter Table

The CPLEX parameters and their types, options, and default values are listed in the following table. The Callable Library name for each parameter is listed first, followed by the Concert Technology name, followed by the name in the Interactive Optimizer. Some CPLEX parameters are not used in the Concert Technology Library, and in those cases, no Concert Technology Library name appears.

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_ADVIND IloCplex::AdvInd advance	1001	int	0 Off : do not use advanced start information 1 On: CPLEX will use an advanced basis supplied by the user 2 On: CPLEX will crush an advanced basis or starting vector supplied by the user Default: 0
Description: Advanced start indicator. An indicator which, if set to 1 or 2, uses advanced starting information when optimization is initiated. Setting 2 may be effective for MIPs in which the percentage of integer constraints is low. It may also reduce the solution time of fixed MIPs.			
CPX_PARAM_AGGCUTLIM IloCplex::AggCutLim mip limits aggforcut	2054	int	Any nonnegative integer Default: 3
Description: Constraint aggregation limit for cut generation. Limits the number of constraints that can be aggregated for generating flow cover and mixed integer rounding cuts.			
CPX_PARAM_AGGFILL IloCplex::AggFill preprocessing fill	1002	int	Any nonnegative integer Default: 10
Description: Preprocessing aggregator fill. Limits variable substitutions by the aggregator. If the net result of a single substitution is more nonzeros than this value, the substitution is not made.			
CPX_PARAM_AGGIND IloCplex::AggInd preprocessing aggregator	1003	int	-1 Automatic (1 for LP, infinite for MIP) 0 Do not use any aggregator Any positive integer Default: -1
Description: Preprocessing aggregator application limit. Invokes the aggregator to use substitution where possible to reduce the number of rows and columns before the problem is solved. If set to a positive value, the aggregator is applied the specified number of times or until no more reductions are possible.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_BARALG IloCplex::BarAlg barrier algorithm	3007	int	0 Default setting 1 Infeasibility-estimate start 2 Infeasibility-constant start 3 Standard barrier Default: 0
Description: Barrier algorithm. The default setting 0 uses the “infeasibility - estimate start” algorithm (setting 1) when solving subproblems in a MIP problem, and the standard barrier algorithm (setting 3) in other cases. The standard barrier algorithm is almost always fastest. However, on problems that are primal or dual infeasible (common for MIP subproblems), the standard algorithm may not work as well as the alternatives. The two alternative algorithms (settings 1 and 2) may eliminate numerical difficulties related to infeasibility, but are generally slower.			
CPX_PARAM_BARCOLNZ IloCplex::BarColNz barrier colnonzeros	3009	int	0 Dynamically calculated or, any positive integer Default: 0
Description: Barrier column nonzeros. Used in the recognition of dense columns. If columns in the presolved and aggregated problem exist with more entries than this value, such columns are considered dense and are treated specially by the CPLEX Barrier Optimizer to reduce their effect. If the problem contains fewer than 400 rows, dense column handling is NOT initiated.			
CPX_PARAM_BARCROSSALG IloCplex::BarCrossAlg barrier crossover	3018	int	-1 No crossover 0 Automatic 1 Primal crossover 2 Dual crossover Default: 0
Description: Barrier crossover algorithm. Determines which, if any, crossover is performed at the end of a barrier optimization called via CPXhybbaropt.			
CPX_PARAM_BARDISPLAY IloCplex::BarDisplay barrier display	3010	int	0 No progress information 1 Normal setup and iteration information 2 Diagnostic information Default: 1
Description: Barrier display information. Determines the level of barrier progress information to be displayed.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_BAREPCOMP IloCplex::BarEpComp barrier convergetol	3002	double	Any positive number ≥ $1e^{-12}$ Default: $1e^{-8}$
Description: Convergence tolerance for LP and QP problems. For problems with quadratic constraints (QCP), see CPX_PARAM_BARQCPEPCOMP. Sets the tolerance on complementarity for convergence. The barrier algorithm terminates with an optimal solution if the relative complementarity is smaller than this value. Changing this tolerance to a smaller value may result in greater numerical precision of the solution, but also increases the chance of a convergence failure in the algorithm and consequently may result in no solution at all. Therefore, caution is advised in deviating from the default setting.			
CPX_PARAM_BARGROWTH IloCplex::BarGrowth barrier limits growth	3003	double	1.0 or greater Default: $1e^{12}$
Description: Barrier growth limit. Used to detect unbounded optimal faces. At higher values, the barrier algorithm is less likely to conclude that the problem has an unbounded optimal face, but more likely to have numerical difficulties if the problem has an unbounded face.			
CPX_PARAM_BARITLIM IloCplex::BarItLim barrier limits iterations	3012	int	0 No Barrier iterations or, any positive integer Default: BIGINT
Description: Barrier iteration limit. Sets the number of barrier iterations before termination. When set to 0, no barrier iterations occur, but problem "setup" occurs and information about the setup is displayed (such as Cholesky factoring information).			
CPX_PARAM_BARMAXCOR IloCplex::BarMaxCor barrier limits corrections	3013	int	-1 Automatically determined 0 None or, any positive integer Default: -1
Description: Barrier maximum correction limit. Sets the maximum number of centering corrections done on each iteration. An explicit value greater than 0 may improve the numerical performance of the algorithm at the expense of computation time.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_BAROBJRNG IloCplex::BarObjRng barrier limits objrange	3004	double	Any nonnegative number Default: $1e^{20}$
Description: Barrier objective range. Sets the maximum absolute value of the objective function. The barrier algorithm looks at this limit to detect unbounded problems.			
CPX_PARAM_BAROOC IloCplex::BarOOC barrier outofcore	3019	int	0 [CPX_OFF] Off 1 [CPX_ON] On Default: 0
Description: Out-of-core barrier indicator. Specifies whether the barrier optimizer should use out-of-core storage (on disk) for the Cholesky factoring. Disk use is controlled by the parameters CPX_PARAM_WORKMEM and CPX_PARAM_WORKDIR.			
CPX_PARAM_BARORDER IloCplex::BarOrder barrier ordering	3014	int	0 Automatic 1 Approximate minimum degree (AMD) 2 Approximate minimum fill (AMF) 3 Nested dissection (ND) Default: 0
Description: Barrier ordering algorithm. Sets the algorithm to be used to permute the rows of the constraint matrix in order to reduce fill in the Cholesky factor.			
CPX_PARAM_BARQCPEPCOMP IloCplex::BarQCPEpComp set bar qcpconvergetol	3020	double	Any positive number $\geq 1e^{-12}$ Default: $1e^{-6}$
Description: Convergence tolerance for QCP problems. That is, for quadratically constrained problems. For LPs and for QPs (that is, when all the constraints are linear) see CPX_PARAM_BAREPCOMP. Sets the tolerance on complementarity for convergence. The barrier algorithm terminates with an optimal solution if the relative complementarity is smaller than this value. Changing this tolerance to a smaller value may result in greater numerical precision of the solution, but also increases the chance of a convergence failure in the algorithm and consequently may result in no solution at all. Therefore, caution is advised in deviating from the default setting.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_BARSTARTALG IloCplex::BarStartAlg barrier startalg	3017	int	1 Dual is 0 2 Estimate dual 3 Average of primal estimate, dual 0 4 Average of primal estimate, estimate dual Default: 1
Description: Barrier starting point algorithm. Sets the algorithm to be used to compute the initial starting point for the barrier optimizer.			
CPX_PARAM_BARTHREADS IloCplex::BarThreads barrier limits threads	3016	int	0 Determined by global thread default >0 upper limit on threads for Parallel Barrier Default 0
Description: Barrier thread limit. Determines the maximum number of parallel processes (threads) that will be invoked by the parallel barrier optimizer. The default value of 0 means that the limit will be determined by the value of CPX_PARAM_THREADS, the global thread limit parameter. A positive value will override the value found in CPX_PARAM_THREADS.			
CPX_PARAM_BASINTERVAL IloCplex::BasInterval simplex basisinterval	1004	int	Any positive integer Default: BIGINT
Description: Basis file saving frequency. Establishes the number of iterations between writes of the CPLEX backup simplex basis file in .xxx format.			
CPX_PARAM_BBINTERVAL IloCplex::BBInterval mip strategy bbinterval	2039	int	0 Best estimate node always selected or, any positive integer Default: 7
Description: MIP strategy best bound interval. When you set nodeselect 2, the bbinterval is the interval at which the best bound node, instead of the best estimate node, is selected from the tree. A bbinterval of 0 means to never select the best bound node. A bbinterval of 1 means always to select the best bound node, and is thus equivalent to nodeselect 1. Higher values of bbinterval mean that the best bound node will be selected less frequently; experience has shown it to be beneficial to occasionally select the best bound node, and therefore the default bbinterval is 7.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_BNDSTRENIND IloCplex::BndStrenInd preprocessing boundstrength	2029	int	-1 Automatically determined 0 Do not apply bound strengthening 1 Apply bound strengthening Default: -1
Description: Bound strengthening indicator. Used when solving mixed integer programs. Bound strengthening tightens the bounds on variables, perhaps to the point where the variable can be fixed and thus removed from consideration during branch & cut. This reduction is usually beneficial, but occasionally, due to its iterative nature, takes a long time.			
CPX_PARAM_BRDIR IloCplex::BrDir mip strategy branch	2001	int	-1 [CPX_BRDIR_DOWN] Down branch selected first 0 [CPX_BRDIR_AUTO] Automatically determined 1 [CPX_BRDIR_UP] Up branch selected first Default: 0
Description: MIP branching direction. Used to decide which branch, the up or the down branch, should be taken first at each node.			
CPX_PARAM_BTTOL IloCplex::BtTol mip strategy backtrack	2002	double	Any number from 0.0 to 1.0 Default: 0.9999
Description: Backtracking tolerance. Controls how often backtracking is done during the branching process. The decision when to backtrack depends on three values that change during the course of the optimization: - the objective function value of the best integer feasible solution (“incumbent”) - the best remaining objective function value of any unexplored node (“best node”) - the objective function value of the most recently solved node (“current objective”). If a cutoff tolerance (see CPX_PARAM_CUTUP and CPX_PARAM_CUTLO) has been set by the user then that value is used as the incumbent until an integer feasible solution is found. The “target gap” is defined to be the absolute value of the difference between the incumbent and the best node, multiplied by this backtracking parameter. CPLEX does not backtrack until the absolute value of the difference between the objective of the current node and the best node is at least as large as the target gap. Low values of this backtracking parameter thus tend to increase the amount of backtracking, which makes the search process more of a pure best-bound search. Higher parameter values tend to decrease backtracking, making the search more of a pure depth-first search. The backtracking value has effect only after an integer feasible solution is found or when a cutoff has been specified. Note that this backtracking value merely permits backtracking but does not force it; CPLEX may choose to continue searching a limb of the tree if it seems a promising candidate for finding an integer feasible solution.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_CLIQUES IloCplex::Cliques mip cuts cliques	2003	int	-1 Do not generate clique cuts 0 Automatically determined 1 Generate clique cuts moderately 2 Generate clique cuts aggressively Default: 0
Description: MIP cliques indicator. Determines whether or not clique cuts should be generated for the problem. Setting the value to 0, the default, indicates that the attempt to generate cliques should continue only if it seems to be helping.			
CPX_PARAM_CLOCKTYPE IloCplex::ClockType clocktype	1006	int	1 CPU time 2 Wall clock time (total physical time elapsed) Default: 1
Description: Computation time reporting. Determines how computation times are measured on UNIX platforms. Computation time on Windows systems is always measured as wall clock time. Small variations in measured time on identical runs may be expected on any computer system under either setting of this parameter.			
CPX_PARAM_COEREDIND IloCplex::CoeRedInd preprocessing coeffreduce	2004	int	0 Do not use coefficient reduction 1 Reduce only to integral coefficients 2 Reduce all potential coefficients Default: 2
Description: Coefficient reduction setting Determines how coefficient reduction is used. Coefficient reduction improves the objective value of the initial (and subsequent) LP relaxations solved during branch & cut by reducing the number of non-integral vertices.			
CPX_PARAM_COLGROWTH IloCplex::ColGrowth read variables	1047	int	Any integer from 0 to 268,435,450 Default: 100
Description: Variable (column) memory growth. Sets the extra space allocated for subsequent modifications of the problem.			
CPX_PARAM_COLREADLIM IloCplex::ColReadLim read variables	1023	int	Any integer from 0 to 268,435,450 Default: Depends on the computer and operating system
Description: Variable (column) read limit. Sets the number of variables that can be read.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_COVERS IloCplex::Covers mip cuts covers	2005	int	-1 Do not generate cover cuts 0 Automatically determined 1 Generate cover cuts moderately 2 Generate cover cuts aggressively Default: 0
Description: MIP covers indicator. Determines whether or not cover cuts should be generated for the problem. Setting the value to 0, the default, indicates that the attempt to generate covers should continue only if it seems to be helping.			
CPX_PARAM_CRAIND IloCplex::CraInd simplex crash	1007	int	**LP Primal**: 0 Ignore objective coefficients during crash -1 or 1 Alternate ways of using objective coefficients **LP Dual**: 1 Default starting basis 0 or -1 Aggressive starting basis **QP Primal**: -1 Slack basis 0 Ignore Q terms and use LP solver for crash 1 Ignore objective and use LP solver for crash **QP Dual**: -1 Slack basis 0 or 1 Use Q terms for crash
Description: Simplex crash ordering. Determines how CPLEX orders variables relative to the objective function when selecting an initial basis.			
CPX_PARAM_CUTLO IloCplex::CutLo mip tolerances lowercutoff	2006	double	Any number Default: $-1e^{+75}$
Description: Lower cutoff. When the problem is a maximization problem, the LOWERCUTOFF parameter is used to cut off any nodes that have an objective value below the lower cutoff value. On a continued mixed integer optimization, the larger of these values and the updated cutoff found during optimization are used during the next mixed integer optimization. A too-restrictive value for the LOWERCUTOFF parameter may result in no integer solutions being found.			
CPX_PARAM_CUTPASS IloCplex::CutPass mip limits cutpasses	2056	int	-1 None 0 Automatically determined Positive values give number of passes to perform Default: 0
Description: Number of cutting plane passes. Sets the upper limit on the number of passes CPLEX performs when generating cutting planes on a MIP model.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_CUTSFACTOR IloCplex::CutsFactor mip limits cutsfactor	2033	double	Any nonnegative number Default: 4.0
Description: Row multiplier factor for cuts. Limits the number of cuts that can be added. The number of rows in the problem with cuts added is limited to CUTSFACTOR times the original number of rows. If the problem is presolved, the original number of rows is that from the presolved problem. A CUTSFACTOR of 1.0 or less means that no cuts will be generated. Because cuts can be added and removed during the course of optimization, CUTSFACTOR may not correspond directly to the number of cuts seen during the node log or in the summary table at the end of optimization.			
CPX_PARAM_CUTUP IloCplex::CutUp mip tolerances uppercutoff	2007	double	Any number Default: $1e^{+75}$
Description: Upper cutoff. Cuts off any nodes that have an objective value above the upper cutoff value, when the problem is a minimization problem. When a mixed integer optimization problem is continued, the smaller of these values and the updated cutoff found during optimization are used during the next mixed integer optimization. A too-restrictive value for the UPPERCUTOFF parameter may result in no integer solutions being found.			
CPX_PARAM_DATACHECK IloCplex::DataCheck read datacheck	1056	int	0 [CPX_OFF] Off (do not check) 1 [CPX_ON] On (check) Default: 0
Description: Data consistency checking indicator. When set to CPX_ON, the CPXcopy____, CPXread____ and CPXchg____ functions perform extensive checking on data in the array arguments, such as checking that indices are within range, that there are no duplicate entries and that values are valid for the type of data or are valid numbers. This is useful for debugging applications.			
CPX_PARAM_DEPIND IloCplex::DepInd preprocessing dependency	1008	int	-1 automatic: let CPLEX choose when to use dependency checking 0 Off : do not use dependency checker 1 turn on only at the beginning of preprocessing 2 turn on only at the end of preprocessing 3 turn on at the beginning and at the end of preprocessing Default: 0
Description: Dependency indicator. Determines whether to activate the dependency checker. If on, the dependency checker searches for dependent rows during preprocessing. If off, dependent rows are not identified.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_DISJCUTS IloCplex::DisjCuts mip cuts disjunctive	2053	int	-1 Do not generate disjunctive cuts 0 Automatically determined 1 Generate disjunctive cuts moderately 2 Generate disjunctive cuts aggressively 3 Generate disjunctive cuts very aggressively Default: 0
Description: MIP disjunctive cuts indicator. Determines whether or not disjunctive cuts should be generated for the problem. Setting the value to 0, the default, indicates that the attempt to generate disjunctive cuts should continue only if it seems to be helping.			
CPX_PARAM_DIVETYPE IloCplex::DiveType mip strategy dive	2060	int	0 automatic 1 traditional dive 2 probing dive 3 guided dive Default: 0
Description: MIP dive strategy. The MIP traversal strategy occasionally performs probing dives, where it looks ahead at both children nodes before deciding which node to choose. The default (automatic) setting lets CPLEX choose when to perform a probing dive, 1 directs CPLEX never to perform probing dives, 2 always to probe, 3 spend more time exploring potential solutions that are similar to the current incumbent. Setting 2, always to probe, is helpful for finding integer solutions.			
CPX_PARAM_DPRIIND IloCplex::DPriInd simplex dgradient	1009	int	0 [CPX_DPRIIND_AUTO] Determined automatically 1 [CPX_DPRIIND_FULL] Standard dual pricing 2 [CPX_DPRIIND_STEEP] Steepest-edge pricing 3 [CPX_DPRIIND_FULL_STEEP] Steepest-edge pricing in slack space 4 [CPX_DPRIIND_STEEPQSTART] Steepest-edge pricing, unit initial norms 5 [CPX_DPRIIND_DEVEX] devex pricing Default: 0
Description: Dual simplex pricing algorithm. The default pricing (0) usually provides the fastest solution time, but many problems benefit from alternate settings.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_EPAGAP IloCplex::EpAGap mip tolerances absmipgap	2008	double	Any nonnegative number Default: $1e^{-06}$
Description: Absolute mipgap tolerance. Sets an absolute tolerance on the gap between the best integer objective and the objective of the best node remaining. When this difference falls below the value of the ABSMIPGAP parameter, the mixed integer optimization is stopped.			
CPX_PARAM_EPGAP IloCplex::EpGap mip tolerances mipgap	2009	double	Any number from 0.0 to 1.0 Default: $1e^{-04}$
Description: Relative mipgap tolerance. Sets a relative tolerance on the gap between the best integer objective and the objective of the best node remaining. When the value $\|bestnode-bestinteger\|/(1e\text{-}10+\|bestinteger\|)$ falls below the value of the MIPGAP parameter, the mixed integer optimization is stopped. For example, to instruct CPLEX to stop as soon as it has found a feasible integer solution proved to be within five percent of optimal, set the relative mipgap tolerance to.05.			
CPX_PARAM_EPINT IloCplex::EpInt mip tolerances integrality	2010	double	Any number from 0.0 to 1.0 Default: $1e^{-05}$
Description: Integrality tolerance. Specifies the amount by which an integer variable can be different from an integer and still be considered feasible. A value of zero is permitted and the optimizer will attempt to meet this tolerance. However, in some models, computer roundoff may still result in small, nonzero deviations from integrality.			
CPX_PARAM_EPMRK IloCplex::EpMrk simplex tolerances markowitz	1013	double	Any number from 0.0001 to 0.99999 Default: 0.01
Description: Markowitz tolerance. Influences pivot selection during basis factoring. Increasing the Markowitz threshold may improve the numerical properties of the solution.			
CPX_PARAM_EPOPT IloCplex::EpOpt simplex tolerances optimality	1014	double	Any number from $1e^{-9}$ to $1e^{-1}$ Default: $1e^{-06}$
Description: Optimality tolerance. Influences the reduced-cost tolerance for optimality. This parameter governs how closely CPLEX must approach the theoretically optimal solution.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_EPPER IloCplex::EpPer simplex perturbation	1015	double	Any positive number ≥ $1e^{-8}$ Default: $1e^{-6}$
Description: Perturbation constant. Sets the amount by which CPLEX perturbs the upper and lower bounds on the variables when a problem is perturbed. This parameter can be set to a smaller value if the default value creates too large a change in the problem.			
CPX_PARAM_EPRHS IloCplex::EpRHS simplex tolerances feasibility	1016	double	Any number from $1e^{-9}$ to $1e^{-1}$ Default: $1e^{-06}$
Description: Feasibility tolerance. The feasibility tolerance specifies the degree to which a problem's basic variables may violate their bounds. FEASIBILITY influences the selection of an optimal basis and can be reset to a higher value when a problem is having difficulty maintaining feasibility during optimization. You may also wish to lower this tolerance after finding an optimal solution if there is any doubt that the solution is truly optimal. If the feasibility tolerance is set too low, CPLEX may falsely conclude that a problem is infeasible. If you encounter reports of infeasibility during Phase II of the optimization, a small adjustment in the feasibility tolerance may improve performance.			
CPX_PARAM_FINALFACTOR IloCplex::FinalFactor simplex finalfactor	1080	bool	0 [CPX_OFF] Off (IloFalse) 1 [CPX_ON] On (IloTrue) Default: On
Description: Final factor, the indicator for basis final factorization after uncrush. When preprocessing changes the model prior to optimization, a reverse operation (uncrush) occurs at termination to restore the full model with its solution. With default settings, the simplex optimizers perform a final basis factorization on the full model before terminating. If you turn off this parameter, the final factorization after uncrushing will be skipped; on large models this can save some time, but computations that require a factored basis after optimization (for example, for the computation of the condition number Kappa) may be unavailable, depending on the operations performed during preprocessing. If you run out of memory at the end of a simplex optimization, consider turning off final factorization.			
CPX_PARAM_FLOWCOVERS IloCplex::FlowCovers mip cuts flowcuts	2040	int	-1 Do not generate flow cover cuts 0 Automatically determined 1 Generate flow cover cuts moderately 2 Generate flow cover cuts aggressively Default: 0
Description: MIP flow cover cuts indicator. Determines whether or not to generate flow cover cuts for the problem. Setting the value to 0, the default, indicates that the attempt to generate flow cover cuts should continue only if it seems to be helping.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_FLOWPATHS IloCplex::FlowPaths mip cuts pathcut	2051	int	-1 Do not generate flow path cuts 0 Automatically determined 1 Generate flow path cuts moderately 2 Generate flow path cuts aggressively Default: 0
Description: MIP flow path cut indicator. Determines whether or not flow path cuts should be generated for the problem. Setting the value to 0, the default, indicates that the attempt to generate flow path cuts should continue only if it seems to be helping.			
CPX_PARAM_FRACCAND IloCplex::FracCand mip limits gomorycand	2048	int	Any positive integer Default: 200
Description: Candidate limit for generating Gomory fractional cuts. Limits the number of candidate variables for generating Gomory fractional cuts.			
CPX_PARAM_FRACCUTS IloCplex::FracCuts mip cuts gomory	2049	int	-1 Do not generate Gomory fractional cuts 0 Automatically determined 1 Generate Gomory fractional cuts moderately 2 Generate Gomory fractional cuts aggressively Default: 0
Description: MIP Gomory fractional cuts indicator. Determines whether or not Gomory fractional cuts should be generated for the problem. Setting the value to 0, the default, indicates that the attempt to generate Gomory fractional cuts should continue only if it seems to be helping.			
CPX_PARAM_FRACPASS IloCplex::FracPass mip limits gomorypass	2050	int	0 Automatic or, any positive integer Default: 0
Description: Pass limit for generating Gomory fractional cuts. Limits the number of passes for generating Gomory fractional cuts. At the default setting of 0, CPLEX decides. The parameter is ignored if the Gomory fractional cut parameter, CPX_PARAM_FRACCUTS, is set to a nonzero value.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_GUBCOVERS IloCplex::GUBCovers mip cuts gubcovers	2044	int	-1 Do not generate GUB cuts 0 Automatically determined 1 Generate GUB cuts moderately 2 Generate GUB cuts aggressively Default: 0
Description: MIP GUB cuts indicator. Determines whether or not to generate GUB cuts for the problem. Setting the value to 0, the default, indicates that the attempt to generate GUB cuts should continue only if it seems to be helping.			
CPX_PARAM_HEURFREQ IloCplex::HeurFreq mip strategy heuristicfreq	2031	int	-1 None 0 Automatic or, any positive integer Default: 0
Description: MIP heuristic frequency. Determines how often to apply the periodic heuristic. Setting the value to -1 turns off the periodic heuristic. Setting the value to 0, the default, applies the periodic heuristic at an interval chosen automatically. Setting the value to a positive number applies the heuristic at the requested node interval. For example, setting HEURISTICFREQ to 20 dictates that the heuristic be called at node 0, 20, 40, 60, etc.			
CPX_PARAM_IISIND IloCplex::IISInd simplex iisfind	1018	int	0 Algorithm with minimum computation time 1 Algorithm generating smaller IIS set Default: 0
Description: IIS algorithm indicator. Determines the algorithm to be used to identify the IIS set (see the *ILOG CPLEX User's Manual* for a description of the CPLEX Infeasibility Finder). The default algorithm is faster and works best for most problems. However, if the size of the resulting IIS is large, the alternative algorithm may be useful. The resulting IIS is smaller, although more computation time is usually needed.			
CPX_PARAM_IMPLBD IloCplex::ImplBd mip cuts implied	2041	int	-1 Do not generate implied bound cuts 0 Automatically determined 1 Generate implied bound cuts moderately 2 Generate implied bound cuts aggressively Default: 0
Description: MIP implied bound cuts indicator. Determines whether or not to generate implied bound cuts for the problem. Setting the value to 0, the default, indicates that the attempt to generate implied bound cuts should continue only if it seems to be helping.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_INTSOLLIM IloCplex::IntSolLim mip limits solutions	2015	int	Any positive integer Default: BIGINT
Description: MIP solution limit. Sets the number of MIP solutions to be found before stopping.			
CPX_PARAM_ITLIM IloCplex::ItLim simplex limits iterations	1020	int	Any nonnegative integer Default: BIGINT
Description: Simplex maximum iteration limit. Sets the maximum number of iterations to be performed before the algorithm terminates without reaching optimality. When set to 0 (zero), no simplex method iteration occurs. However, CPLEX factors the initial basis from which solution routines provide information about the associated initial solution.			
CPX_PARAM_LPMETHOD IloCplex::RootAlg lpmethod	1062	int	0 [CPX_ALG_AUTOMATIC] Automatic 1 [CPX_ALG_PRIMAL] Primal Simplex 2 [CPX_ALG_DUAL] Dual Simplex 3 [CPX_ALG_NET] Network Simplex 4 [CPX_ALG_BARRIER] Barrier 5 [CPX_ALG_SIFTING] Sifting 6 [CPX_ALG_CONCURRENT] Concurrent (Dual, Barrier, and Primal) Default: 0
Description: Algorithm for linear optimization. Determines which algorithm is used when CPXlpopt (or optimize in the Interactive Optimizer) is invoked. Currently, the behavior of the Automatic setting is that CPLEX almost always invokes the dual simplex algorithm. The one exception is when solving the relaxation of an MILP model when multiple threads have been requested. In this case, the Automatic setting will use the concurrent optimization algorithm. The Automatic setting may be expanded in the future so that CPLEX chooses the algorithm based on additional problem characteristics.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_MIPDISPLAY IloCplex::MIPDisplay mip display	2012	int	0 No display 1 Display integer feasible solutions 2 Display nodes under CPX_PARAM_MIPInterval 3 Same as 2 with information on node cuts 4 Same as 3 with LP subproblem information at root 5 Same as 4 with LP subproblem information at nodes Default: 2
Description: MIP node log display information. Determines what CPLEX reports to the screen during mixed integer optimization. The amount of information displayed increases with increasing values of this parameter. A setting of 0 causes no node log to be displayed until the optimal solution is found. A setting of 1 displays an entry for each integer feasible solution found. Each entry contains the objective function value, the node count, the number of unexplored nodes in the tree, and the current optimality gap. A setting of 2 also generates an entry for every n-th node (where n is the setting of the MIP INTERVAL parameter). A setting of 3 additionally generates an entry for every nth node giving the number of cuts added to the problem for the previous INTERVAL nodes. A setting of 4 additionally generates entries for the LP root relaxation according to the set simplex display setting. A setting of 5 additionally generates entries for the LP subproblems, also according to the set simplex display setting.			
CPX_PARAM_MIPEMPHASIS IloCplex::MIPEmphasis mip emphasis	2058	int	0 [CPX_MIPEMPHASIS_BALANCED] Balance optimality and feasibility 1 [CPX_MIPEMPHASIS_FEASIBILITY] Emphasize feasibility over optimality 2 [CPX_MIPEMPHASIS_OPTIMALITY] Emphasize optimality over feasibility 3 [CPX_MIPEMPHASIS_BESTBOUND] Emphasize moving best bound 4 [CPX_MIPEMPHASIS_HIDDENFEAS] Emphasize hidden feasibility Default: 0
Description: MIP emphasis indicator. With the default setting of BALANCED, CPLEX works toward a rapid proof of an optimal solution, but balances that with effort toward finding high quality feasible solutions early in the optimization. When set to FEASIBILITY, CPLEX frequently will generate more feasible solutions as it optimizes the problem, at some sacrifice in the speed to the proof of optimality. When set to OPTIMALITY, less effort may be applied to finding feasible solutions early. With the seting BESTBOUND, even greater emphasis is placed on proving optimality through moving the best bound value, so that the detection of feasible solutions along the way becomes almost incidental. When set to HIDDENFEAS, the MIP optimizer works hard to find high quality feasible solutions that are otherwise very difficult to find, so consider this setting when the FEASIBILITY emphasis has difficulty finding solutions of acceptable quality.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_MIPINTERVAL IloCplex::MIPInterval mip interval	2013	int	Any positive integer Default: 100
Description: MIP node log interval. Controls the frequency of node logging when CPX_PARAM_MIPDISPLAY is set higher than 1.			
CPX_PARAM_MIPORDIND IloCplex::MIPOrdInd mip strategy order	2020	int	0 [CPX_OFF] Off (do not use order information) 1 [CPX_ON] On (use order information if it exists) Default: 1
Description: MIP priority order indicator. When set to on, uses the priority order (if it exists) for the next mixed integer optimization.			
CPX_PARAM_MIPORDTYPE IloCplex::MIPOrdType mip ordertype	2032	int	0 Do not generate a priority order 1 [CPX_MIPORDER_COST] Use decreasing cost 2 [CPX_MIPORDER_BOUNDS] Use increasing bound range 3 [CPX_MIPORDER_SCALEDCOST] Use increasing cost per coefficient count Default: 0
Description: MIP priority order generation. Used to select the type of generic priority order to generate when no priority order is present.			
CPX_PARAM_MIPSTART IloCplex::MIPStart mip strategy mipstart	2035	int	0 [CPX_OFF] Do not use starting values 1 [CPX_ON] Use starting values at node 0 Default: 0
Description: Indicator for starting MIP values. Used to indicate how the MIP advanced starting values are used. A setting of 1 indicates that the values should be checked to see if they provide an integer feasible solution before starting optimization.			
CPX_PARAM_MIPTHREADS IloCplex::MIPThreads mip limits threads	2014	int	0 determined by global thread default >0 upper limit on threads for Parallel MIP Default: 0
Description: MIP thread limit Determines the maximum number of parallel processes (threads) that will be invoked by the Parallel MIP optimizer. The default value of 0 means that the limit will be determined by the value of CPX_PARAM_THREADS, the global thread limit parameter. A positive value will override the value found in CPX_PARAM_THREADS.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_MIRCUTS IloCplex::MIRCuts mip cuts mircut	2052	int	-1 Do not generate MIR cuts 0 Automatically determined 1 Generate MIR cuts moderately 2 Generate MIR cuts aggressively Default: 0
Description: MIP MIR (mixed integer rounding) cut indicator. Determines whether or not to generate MIR cuts for the problem. Setting the value to 0, the default, indicates that the attempt to generate MIR cuts should continue only if it seems to be helping.			
CPX_PARAM_NETDISPLAY IloCplex::NetDisplay network display	5005	int	0 [CPXNET_NO_DISPLAY_OBJECTIVE] No display 1 [CPXNET_TRUE_OBJECTIVE] Display true objective values 2 [CPXNET_PENALIZE_OBJECTIVE] Display penalized objective values Default: 2
Description: Network logging display indicator. Settings 1 and 2 differ only during Phase I. Setting 2 shows monotonic values, whereas 1 usually does not.			
CPX_PARAM_NETEPOPT IloCplex::NetEpOpt network tolerances optimality	5002	double	Any number from $1e^{-11}$ to $1e^{-1}$ Default: $1e^{-6}$
Description: Optimality tolerance for CPXNETprimopt. The optimality tolerance specifies the amount a reduced cost may violate the criterion for an optimal solution.			
CPX_PARAM_NETEPRHS IloCplex::NetEpRHS network tolerances feasibility	5003	double	Any number from $1e^{-11}$ to $1e^{-1}$ Default: $1e^{-6}$
Description: Feasibility tolerance for CPXNETprimopt. The feasibility tolerance specifies the degree to which a problem's flow value may violate its bounds. This tolerance influences the selection of an optimal basis and can be reset to a higher value when a problem is having difficulty maintaining feasibility during optimization. You may also wish to lower this tolerance after finding an optimal solution if there is any doubt that the solution is truly optimal. If the feasibility tolerance is set too low, CPLEX may falsely conclude that a problem is infeasible. If you encounter reports of infeasibility during Phase II of the optimization, a small adjustment in the feasibility tolerance may improve performance.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_NETFIND IloCplex::NetFind network netfind	1022	int	1 [CPX_NETFIND_PURE] Extract pure network only 2 [CPX_NETFIND_REFLECT] Try reflection scaling 3 [CPX_NETFIND_SCALE] Try general scaling Default: 2
Description: Simplex network extraction level. Establishes the level of network extraction for network simplex optimizations. The default value is suitable for recognizing commonly used modeling approaches when representing a network problem within an LP formulation.			
CPX_PARAM_NETITLIM IloCplex::NetItLim network iterations	5001	int	Any nonnegative integer Default: BIGINT
Description: Network simplex iteration limit. Sets the maximum number of iterations to be performed before the algorithm terminates without reaching optimality.			
CPX_PARAM_NETPPRIIND IloCplex::NetPPriInd network pricing	5004	int	0 [CPXNET_PRICE_AUTO] Automatic 1 [CPXNET_PRICE_PARTIAL] Partial pricing 2 [CPXNET_PRICE_MULT_PART] Multiple partial pricing 3 [CPXNET_PRICE_SORT_MULT_PART] Multiple partial pricing with sorting Default: 0
Description: Network Simplex pricing algorithm. The default (0) shows best performance for most problems, and currently is equivalent to 3.			
CPX_PARAM_NODEFILEIND IloCplex::NodeFileInd mip strategy file	2016	int	0 No node file 1 Node file in memory and compressed 2 Node file on disk 3 Node file on disk and compressed Default: 1
Description: Node storage file indicator. Used when working memory, WORKMEM, has been exceeded by the size of the tree. If the node file parameter is set to zero when the tree memory limit is reached, optimization is terminated. Otherwise, a group of nodes is removed from the in-memory set as needed. By default, CPLEX transfers nodes to node files when the in-memory set is larger than 128 MBytes, and it keeps the resulting node 'files' in compressed form in memory. At settings 2 and 3, the node files are transferred to disk, in compressed and uncompressed form respectively, into a directory named by the WORKDIR parameter, and CPLEX actively manages which nodes remain in memory for processing. The use of node files is described in more detail in the *ILOG CPLEX User's Manual*.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_NODELIM IloCplex::NodeLim mip limits nodes	2017	int	Any nonnegative integer Default: BIGINT
Description: MIP node limit. Sets the maximum number of nodes solved before the algorithm terminates, without reaching optimality. When set to 0 (zero), CPLEX does not create any nodes, but it does solve the root node LP relaxation and repeatedly apply cuts and resolve this LP.			
CPX_PARAM_NODESEL IloCplex::NodeSel mip strategy nodeselect	2018	int	0 [CPX_NODESEL_DFS] Depth-first search 1 [CPX_NODESEL_BESTBOUND] Best-bound search 2 [CPX_NODESEL_BESTEST] Best-estimate search 3 [CPX_NODESEL_BESTEST_ALT] Alternative best-estimate search Default: 1
Description: MIP node selection strategy. Used to set the rule for selecting the next node to process when backtracking. The depth-first search strategy chooses the most recently created node. The best-bound strategy chooses the node with the best objective function for the associated LP relaxation. The best-estimate strategy selects the node with the best estimate of the integer objective value that would be obtained from a node once all integer infeasibilities are removed. An alternative best-estimate search is also available.			
CPX_PARAM_NZGROWTH IloCplex::NzGrowth read nonzeros	1048	int	Any integer from 0 to 268,435,450 Default: 500
Description: Nonzero element memory growth. Sets the growth policy for subsequent modifications of the problem.			
CPX_PARAM_NZREADLIM IloCplex::NzReadLim read nonzeros	1024	int	Any integer from 0 to 268,435,450 Default: Depends on the computer and operating system
Description: Nonzero element read limit. Sets the number of nonzeros that can be read.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_OBJDIF IloCplex::ObjDif mip tolerances objdifference	2019	double	Any number Default: 0.0
Description: Absolute objective difference cutoff. Used to update the cutoff each time a mixed integer solution is found. This absolute value is subtracted from (added to) the newly found integer objective value when minimizing (maximizing). This forces the mixed integer optimization to ignore integer solutions that are not at least this amount better than the one found so far. The OBJDIFFERENCE parameter can be adjusted to improve problem solving efficiency by limiting the number of nodes; however, setting this parameter at a value other than zero (the default) can cause some integer solutions, including the true integer optimum, to be missed. Negative values for this parameter can result in some integer solutions that are worse than or the same as those previously generated, but does not necessarily result in the generation of all possible integer solutions.			
CPX_PARAM_OBJLLIM IloCplex::ObjLLim simplex limits lowerobj	1025	double	Any number Default: $-1e^{+75}$
Description: Lower objective value limit. Setting a lower objective function limit causes CPLEX to halt the optimization process once the minimum objective function value limit has been exceeded. This limit applies only during Phase II of the simplex algorithm.			
CPX_PARAM_OBJULIM IloCplex::ObjULim simplex limits upperobj	1026	double	Any number Default: $1e^{+75}$
Description: Upper objective value limit. Setting an upper objective function limit causes CPLEX to halt the optimization process once the maximum objective function value limit has been exceeded. This limit applies only during Phase II of the simplex algorithm.			
CPX_PARAM_PERIND IloCplex::PerInd simplex perturbation	1027	int	0 [CPX_OFF] Off 1 [CPX_ON] On Default: 0
Description: Simplex perturbation indicator. Setting this parameter to 1 causes all problems to be automatically perturbed as optimization begins. A setting of 0 allows CPLEX to determine dynamically, during solution, whether progress is slow enough to merit a perturbation. The situations in which a setting of 1 helps are rare and restricted to problems that exhibit extreme degeneracy.			
CPX_PARAM_PERLIM IloCplex::PerLim simplex limits perturbation	1028	int	0 Determined automatically or, any positive integer Default: 0
Description: Simplex perturbation limit. Sets the number of stalled iterations before perturbation is performed.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_PPRIIND IloCplex::PPriInd simplex pgradient	1029	int	-1 [CPX_PPRIIND_PARTIAL] Reduced-cost pricing 0 [CPX_PPRIIND_AUTO] Hybrid reduced-cost & devex pricing 1 [CPX_PPRIIND_DEVEX] Devex pricing 2 [CPX_PPRIIND_STEEP] Steepest-edge pricing 3 [CPX_PPRIIND_STEEPQSTART] Steepest-edge pricing with slack initial norms 4 [CPX_PPRIIND_FULL] Full pricing Default: 0
Description: Primal Simplex pricing algorithm. The default pricing (0) usually provides the fastest solution time, but many problems benefit from alternative settings.			
CPX_PARAM_PRECOMPRESS IloCplex::PreCompress preprocessing compress	1066	int	-1 Off 0 Automatic 1 On Default: 0
Description: Compression of original model after presolve. Specifies whether CPLEX should compress the original model after presolve is performed. This can save considerable storage space for large models. Under the automatic setting, CPLEX will decide whether to perform the compression based on model characteristics.			
CPX_PARAM_PREDUAL IloCplex::PreDual preprocessing dual	1044	int	-1 Off 0 Automatic 1 On Default: 0
Description: Presolve dual setting. Determines whether CPLEX Presolve should pass the primal or dual linear programming problem to the linear programming optimization algorithm. By default, CPLEX chooses automatically. If the DUAL indicator is set to 1, the CPLEX presolve algorithm is applied to the primal problem, but the resulting dual linear program is passed to the optimizer. This is a useful technique for problems with more constraints than variables.			
CPX_PARAM_PREIND IloCplex::PreInd preprocessing presolve	1030	int	0 [CPX_OFF] Off (do not use presolve) 1 [CPX_ON] On (use presolve) Default: 1
Description: Presolve indicator. When set to 1, invokes the CPLEX Presolve to simplify and reduce problems.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_PRELINEAR IloCplex::PreLinear preprocessing linear	1058	int	0 Only linear reductions 1 Full reductions Default: 1
Description: Linear reduction indicator. If only linear reductions are performed, each variable in the original model can be expressed as a linear form of variables in the presolved model. This guarantees, for example, that users can add their own custom cuts to the presolved model.			
CPX_PARAM_PREPASS IloCplex::PrePass preprocessing numpass	1052	int	-1 Determined automatically 0 Do not use Presolve or, any positive integer Default: -1
Description: Limit on the number of Presolve passes made. When set to a nonzero value, invokes the CPLEX Presolve to simplify and reduce problems. When set to a positive value, the Presolve is applied the specified number of times, or until no more reductions are possible. At the default value of -1, Presolve should continue only if it seems to be helping.			
CPX_PARAM_PRESLVND IloCplex::PreslvNd mip strategy presolvenode	2037	int	-1 No node presolve 0 Automatic 1 Force node presolve Default: 0
Description: Node presolve selector. Indicates whether node presolve should be performed at the nodes of a mixed integer programming solution. Node presolve can significantly reduce solution time for some models. The default setting is generally effective at determining whether to apply node presolve, although runtimes can be reduced for some models by turning node presolve off.			
CPX_PARAM_PRICELIM IloCplex::PriceLim simplex pricing	1010	int	0 Determined automatically or, any positive integer Default: 0
Description: Simplex pricing candidate list size. Sets the maximum number of variables kept in the pricing candidate list.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_PROBE IloCplex::Probe mip strategy probe	2042	int	-1 No probing 0 Automatic 1-3 Probing level Default: 0
Description: MIP probe. Determines the amount of probing on variables to be performed before MIP branching. Higher settings perform more probing. Probing can be very powerful but very time consuming at the start. Setting the parameter to values above the default of 0 (automatic) can result in dramatic reductions or dramatic increases in solution time, depending on the model.			
CPX_PARAM_QPMAKEPSDIND IloCplex::QPmakePSDInd preprocessing qpmakepsd	4010	int	0 [CPX_OFF] Off 1 [CPX_ON] On Default: On
Description: Indefinite MIQP indicator. Determines whether CPLEX will attempt to adjust a MIQP formulation, in which all the variables appearing in the quadratic term are binary. When this feature is active, adjustments will be made to the elements of a quadratic matrix that is not nominally positive semi-definite (PSD, as required by CPLEX for all QP formulations), to make it PSD, and will also attempt to tighten an already PSD matrix for better numerical behavior. The default setting of 1 means yes, but you can turn it off if necessary; most models should benefit from the default setting.			
CPX_PARAM_QPMETHOD IloCplex::RootAlg qpmethod	1063	int	0 [CPX_ALG_AUTOMATIC] Automatic 1 [CPX_ALG_PRIMAL] Primal Simplex 2 [CPX_ALG_DUAL] Dual Simplex 3 [CPX_ALG_NET] Network Simplex 4 [CPX_ALG_BARRIER] Barrier Default: 0
Description: Algorithm for continuous quadratic optimization. Determines which algorithm is used when CPXqpopt (or optimize in the Interactive Optimizer) is invoked. Currently, the behavior of the Automatic setting is that CPLEX invokes the barrier optimizer for continuous QP models, and the dual simplex optimizer for root relaxations of MIQP models. The Automatic setting may be expanded in the future so that CPLEX chooses the algorithm based on additional problem characteristics.			
CPX_PARAM_QPNZGROWTH IloCplex::QPNzGrowth read qpnonzeros	4002	int	Any integer from 0 to 268,435,450 Default: 200
Description: QP Q matrix memory growth. Sets the growth policy for subsequent modifications of the problem.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_QPNZREADLIM IloCplex::QPNzReadLim read qpnonzeros	4001	int	Any integer from 0 to 268,435,450 Default: 500
Description: QP Q matrix nonzero read limit. Sets the number of Q matrix nonzeros that can be read.			
CPX_PARAM_REDUCE IloCplex::Reduce preprocessing reduce	1057	int	0 No primal and dual reductions 1 Only primal reductions 2 Only dual reductions 3 Both primal and dual reductions Default: 3
Description: Primal and dual reduction type. Determines whether primal reductions, dual reductions, or both, are performed during preprocessing.			
CPX_PARAM_REINV IloCplex::ReInv simplex refactor	1031	int	0 Determined automatically or, any integer from 1 to 10,000 Default: 0
Description: Simplex refactoring frequency. Sets the number of iterations between refactoring of the basis matrix.			
CPX_PARAM_RELAXPREIND IloCplex::RelaxPreInd preprocessing relax	2034	int	0 [CPX_OFF] Off (do not use presolve on initial relaxation) 1 [CPX_ON] On (use presolve on initial relaxation) Default: 0
Description: Relaxed LP presolve indicator. Determines whether LP presolve is applied to the root relaxation in a mixed integer program. Sometimes additional reductions can be made beyond any MIP presolve reductions that were already done.			
CPX_PARAM_RELOBJDIF IloCplex::RelObjDif mip tolerances relobjdifference	2022	double	Any number from 0.0 to 1.0 Default: 0.0
Description: Relative objective difference cutoff. Used to update the cutoff each time a mixed integer solution is found. The value is multiplied by the absolute value of the integer objective and subtracted from (added to) the newly found integer objective when minimizing (maximizing). This forces the mixed integer optimization to ignore integer solutions that are not at least this amount better than the one found so far. The relative objective difference parameter can be adjusted to improve problem solving efficiency by limiting the number of nodes; however, setting this parameter at a value other than zero (the default) can cause some integer solutions, including the true integer optimum, to be missed. If both RELOBJDIFFERENCE and OBJDIFFERENCE are nonzero, the value of OBJDIFFERENCE is used.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_REVERSEIND IloCplex::ReverseInd read reverse	1032	int	0 [CPX_OFF] Off (do not reverse bytes) 1 [CPX_ON] On (reverse bytes) Default: 0
Description: SAV file reading byte-reverse indicator. If set to 1, reverses the byte ordering when reading SAV files. This is useful when a SAV file was created on one system, but is to be read on another system which uses a different byte ordering convention (for example, PCs versus many UNIX systems).			
CPX_PARAM_RINSHEUR IloCplex::RINSHeur mip strategy rinsheur	2061	int	-1 None 0 Automatic (default) or, any positive integer Default: 0
Description: Relaxation induced neighborhood search heuristic determines how often to apply the relaxation induced neighborhood search heuristic (RINS heuristic). Setting the value to -1 turns off the RINS heuristic. Setting the value to 0, the default, applies the RINS heuristic at an interval chosen automatically by CPLEX. Setting the value to a positive number applies the RINS heuristic at the requested node interval. For example,setting RINSHeur to 20 dictates that the RINS heuristic be called at node 0, 20, 40, 60, etc.			
CPX_PARAM_ROWGROWTH IloCplex::RowGrowth read constraints	1046	int	Any integer from 0 to 268,435,450 Default: 100
Description: Constraint (row) memory growth. Sets the growth policy for subsequent modifications of the problem.			
CPX_PARAM_ROWREADLIM IloCplex::RowReadLim read constraints	1021	int	Any integer from 0 to 268,435,450 Default: Depends on the computer and operating system
Description: Constraint (row) read limit. Sets the number of constraints that can be read.			
CPX_PARAM_SCAIND IloCplex::ScaInd read scale	1034	int	-1 No scaling 0 Equilibration scaling 1 More aggressive scaling Default: 0
Description: Scale parameter. Indicates how to scale the problem matrix.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_SCRIND	1035	int	0 [CPX_OFF] Off 1 [CPX_ON] On Default: 0
Description: Messages to screen indicator. Indicates whether or not results messages are displayed on screen.			
CPX_PARAM_SIFTALG IloCplex::SiftAlg sifting algorithm	1077	int	0 Automatic 1 Primal simplex 2 Dual simplex 3 Network simplex 4 Barrier Default: 0
Description: Sifting subproblem algorithm Sets the algorithm to be used for solving sifting subproblems.			
CPX_PARAM_SIFTDISPLAY IloCplex::SiftDisplay sifting display	1076	int	0 No display 1 Display major iterations 2 Display LP subproblem information within each sifting iteration Default: 1
Description: Sifting display information. Determines the amount of sifting progress information to be displayed.			
CPX_PARAM_SIFTITLIM IloCplex::SiftItLim sifting iterations	1078	int	Any nonnegative integer Default: BIGINT
Description: Upper limit on sifting iterations. Sets the maximum number of sifting iterations that may be performed if convergence to optimality has not been reached.			
CPX_PARAM_SIMDISPLAY IloCplex::SimDisplay simplex display	1019	int	0 No iteration messages until solution 1 Iteration info after each refactoring 2 Iteration info for each iteration Default: 1
Description: Simplex iteration display information. Determines how often CPLEX reports during simplex optimization.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_SINGLIM IloCplex::SingLim simplex limits singularity	1037	int	Any nonnegative integer Default: 10
Description: Simplex singularity repair limit. Restricts the number of times CPLEX attempts to repair the basis when singularities are encountered. Once this limit is exceeded, CPLEX replaces the current basis with the best factorable basis that has been found.			
CPX_PARAM_STARTALG IloCplex::RootAlg mip strategy startalgorithm	2025	int	0 [CPX_ALG_AUTOMATIC] Automatic 1 [CPX_ALG_PRIMAL] Primal Simplex 2 [CPX_ALG_DUAL] Dual Simplex 3 [CPX_ALG_NET] Network Simplex 4 [CPX_ALG_BARRIER] Barrier 5 [CPX_ALG_SIFTING] Sifting 6 [CPX_ALG_CONCURRENT] Concurrent Dual, Barrier and Primal Default: 0
Description: MIP starting LP algorithm. Determines which LP algorithm should be used to solve the initial relaxation of the MIP.			
CPX_PARAM_STRONGCANDLIM IloCplex::StrongCandLim mip limits strongcand	2045	int	Any positive number Default: 10
Description: MIP candidate list Controls the length of the candidate list when CPLEX uses the setting strong branching variable selection (set mip strategy variableselect 3).			
CPX_PARAM_STRONGITLIM IloCplex::StrongItLim mip limits strongit	2046	int	0 Automatic: Let CPLEX determine automatically or any positive integer Default: 0
Description: MIP simplex iterations Controls the number of simplex iterations performed on each variable in the candidate list when CPLEX uses the setting strong branching variable selection (set mip strategy variableselect 3). The default setting 0 chooses the iteration limit automatically.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_STRONGTHREADLIM IloCplex::StrongThreadLim mip limits strongthreads	2047	int	Any positive number Default: 1
Description: MIP parallel threads Controls the number of parallel threads used to perform strong branching. Note that this parameter does nothing if the MIP thread limit (set mip limits threads) is greater than 1. Note also that the global thread limit, CPX_PARAM_THREADS, does not affect this parameter.			
CPX_PARAM_SUBALG IloCplex::NodeAlg mip strategy subalgorithm	2026	int	0 [CPX_ALG_AUTOMATIC] Let CPLEX choose 1 [CPX_ALG_PRIMAL] Primal Simplex 2 [CPX_ALG_DUAL] Dual Simplex 3 [CPX_ALG_NET] Network Simplex 4 [CPX_ALG_BARRIER] Barrier 5 [CPX_ALG_SIFTING] Sifting Default: 0
Description: MIP subproblem LP algorithm. Sets the algorithm to be used on MIP subproblems.			
CPX_PARAM_SUBMIPNODELIM IloCplex::SubMIPNodeLim mip limits submipnodelim	2062	int	Any positive integer Default: 500
Description: MIP subnode limit. Restricts the number of nodes searched, during application of the relaxation induced neighborhood search (RINS) heuristic.			
CPX_PARAM_SYMMETRY IloCplex::Symmetry preprocessing symmetry	2059	int	0 [CPX_OFF] Off 1 [CPX_ON] On Default: Off
Description: Symmetry breaking cuts. Determines whether symmetry breaking cuts may be added, during the preprocessing phase, to a MIP model.			
CPX_PARAM_THREADS IloCplex::Threads threads	1067	int	Minimum: 1 Maximum: determined by license key and computer Default: 1
Description: Global default thread count. Determines the default number of parallel processes (threads) that will be invoked by any CPLEX parallel optimizer. This provides a convenient way to control parallelism with a single parameter setting. The value in place for this parameter can be overridden for any particular CPLEX parallel optimizer by setting the appropriate thread limit (CPX_PARAM_BARTHREADS, CPX_PARAM_MIPTHREADS, or).			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_TILIM IloCplex::TiLim timelimit	1039	double	Any nonnegative number Default: $1e^{+75}$
Description: Global time limit. Sets the maximum time, in seconds, for a call to an optimizer. This time limit applies also to the infeasibility finder. The time is measured in terms of either CPU time or elapsed time, according to the setting of the CLOCKTYPE parameter. The time limit for an optimizer applies to the sum of all its steps, such as preprocessing,crossover, and internal calls to other optimizers. In a sequence of calls to optimizers, the limit is not cumulative but applies to each call individually. For example, if you set a time limit of 10 seconds, and you call mipopt twice then there could be a total of (at most) 20 seconds of running time if each call consumes its maximum allotment.			
CPX_PARAM_TRELIM IloCplex::TreLim mip limits treememory	2027	double	Any nonnegative number Default: $1e^{+75}$
Description: Tree memory limit. Sets an absolute upper limit on the size (in megabytes) of the branch & cut tree. If this limit is exceeded, CPLEX terminates optimization.			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_VARSEL IloCplex::VarSel mip strategy variableselect	2028	int	-1 [CPX_VARSEL_MININFEAS] Branch on variable with minimum infeasibility 0 [CPX_VARSEL_DEFAULT] Branch variable automatically selected 1 [CPX_VARSEL_MAXINFEAS] Branch on variable with maximum infeasibility 2 [CPX_VARSEL_PSEUDO] Branch based on pseudo costs 3 [CPX_VARSEL_STRONG] Strong branching 4 [CPX_VARSEL_PSEUDOREDUCED] Branch based on pseudo reduced costs Default: 0
Description: MIP variable selection strategy. Used to set the rule for selecting the branching variable at the node which has been selected for branching. The maximum infeasibility rule chooses the variable with the largest fractional value; the minimum infeasibility rule chooses the variable with the smallest fractional value. The minimum infeasibility rule (-1) may lead more quickly to a first integer feasible solution, but is usually slower overall to reach the optimal integer solution. The maximum infeasibility rule (1) forces larger changes earlier in the tree, which tend to produce faster overall times to reach the optimal integer solution. Pseudo cost (2) variable selection is derived from pseudo-shadow prices. Strong branching (3) causes variable selection based on partially solving a number of subproblems with tentative branches to see which branch is the most promising. This strategy can be effective on large, difficult MIP problems. Pseudo reduced costs (4) are a computationally less-intensive form of pseudo costs. The default value (0) allows CPLEX to select the best rule based on the problem and its progress.			
CPX_PARAM_WORKDIR IloCplex::WorkDir workdir	1064	string	Default: '.'
Description: Directory for working files. Specifies the name of an existing directory into which CPLEX may store temporary working files, such as for MIP node files or for out-of-core barrier.			
CPX_PARAM_WORKMEM IloCplex::WorkMem workmem	1065	double	Any nonnegative number, in megabytes Default: 128.0
Description: Memory available for working storage. Specifies an upper limit on the amount of central memory, in megabytes, that CPLEX is permitted to use for working files (see CPX_PARAM_WORKDIR).			

Parameter Name	Code	Type	Value [Symbolic Constants]
CPX_PARAM_XXXIND IloCplex::XXXInd simplex xxxstart	1041	int	0 [CPX_OFF] Off (disable xxx file reading) 1 [CPX_ON] On (enable xxx file reading) Default: 0

Description: Indicator for reading .xxx files.
Used to enable/disable the reading of .xxx files. When solving a linear program using a simplex optimizer option (PRIMOPT or TRANOPT), if for some reason the optimization as well as the CPLEX session were terminated before completion, it may be useful to read an .xxx file to resume optimization. However, if preprocessing was used during the optimization, just reading in this basis file does not produce the desired behavior since the '.xxx' file was generated relative to the presolved problem. The XXXSTART indicator provides an alternative approach.
If this indicator is turned on, CPLEX activates its presolve and turns the advanced-start indicator off (so that no internally stored advanced start is used). It then attempts to find a file with a .xxx extension in the working directory. The name of the file preceding the .xxx extension must match the name of the problem being optimized.

Index

I

M

N

P

Q

S