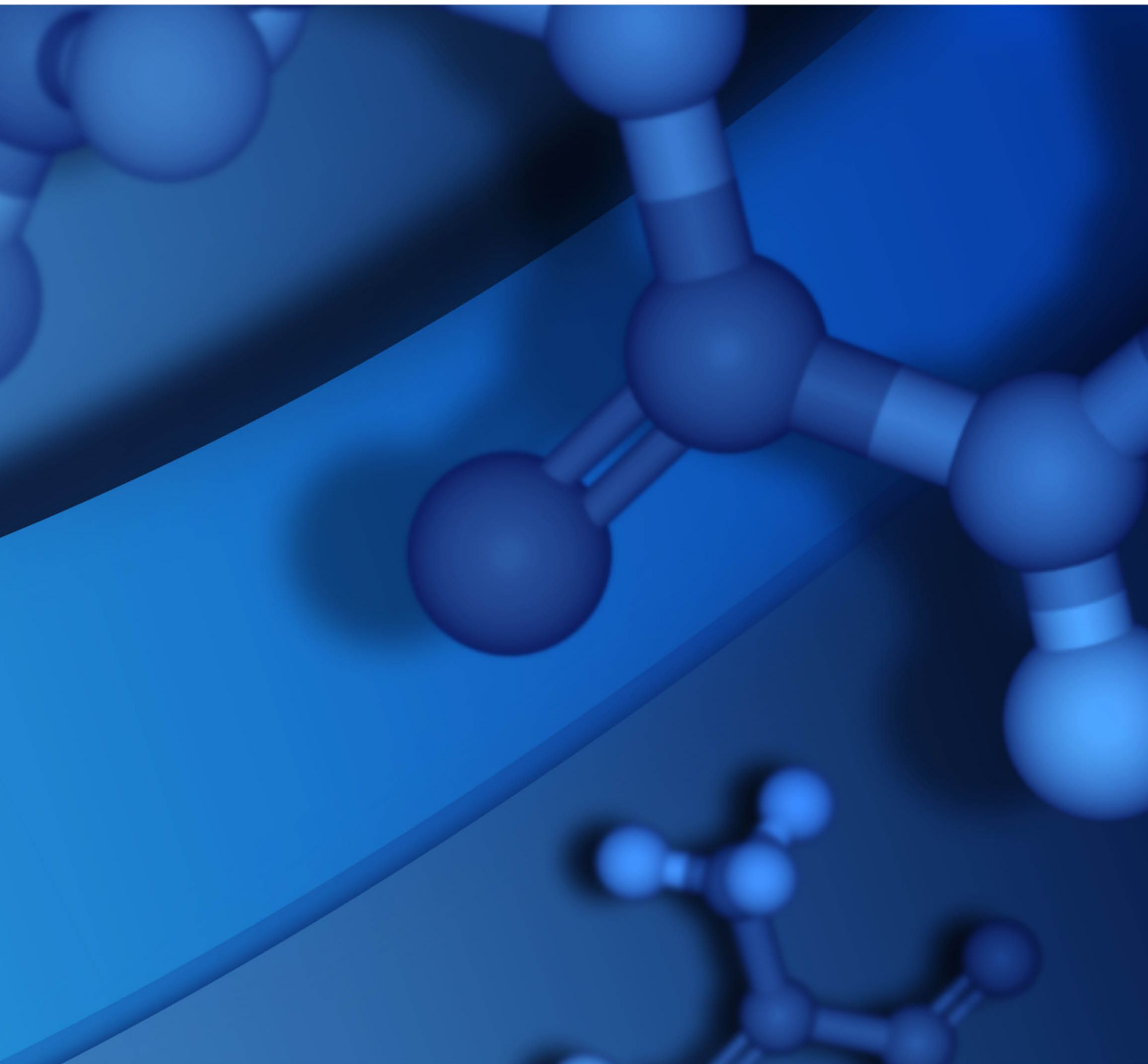


BUILDING TOOLS GUIDE

MATERIALS VISUALIZER
8.0



Copyright Notice

©2014 Dassault Systèmes. All rights reserved. 3DEXPERIENCE, the Compass icon and the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, DELMIA, SIMULIA, GEOVIA, EXALEAD, 3D VIA, BIOVIA and NETVIBES are commercial trademarks or registered trademarks of Dassault Systèmes or its subsidiaries in the U.S. and/or other countries. All other trademarks are owned by their respective owners. Use of any Dassault Systèmes or its subsidiaries trademarks is subject to their express written approval.

Acknowledgments and References

To print photographs or files of computational results (figures and/or data) obtained using BIOVIA software, acknowledge the source in an appropriate format. For example:

"Computational results obtained using software programs from Dassault Systèmes Biovia Corp..
The *ab initio* calculations were performed with the DMol³ program, and graphical displays generated with Materials Studio."

BIOVIA may grant permission to republish or reprint its copyrighted materials. Requests should be submitted to BIOVIA Support, either through electronic mail to support@accelrys.com, or in writing to:

BIOVIA Support
5005 Wateridge Vista Drive, San Diego, CA 92121 USA

Contents

Building Tools	1	If analog building fails	24
Introduction	1	If setting the connection point fails	25
Further Information	1	Build Analogs dialog	26
Polymer Builder	2	Cores tab	26
Tasks in Polymer Builder	2	Fragments tab	27
Creating and using repeat units	2	Choose Fragments dialog	28
Fragments and repeat units	2	Options tab	28
Existing and sketched repeat units	3	Nanostructure Builder	30
Building a random copolymer using probabilities	3	Tasks in Nanostructure Builder	30
Building a block copolymer with branches ..	4	Building nonperiodic and periodic single-wall nanotubes	30
If build dendrimer fails	5	Building nonperiodic multi-wall nanotubes ..	31
Theory in Polymer Builder	6	Building nonperiodic single-wall nanotube bundles (nanoropes)	34
Introduction	6	Building nanoclusters	36
Conditional probabilities	7	Dialogs in Nanostructure Builder	37
Reactivity ratios	7	Build Single-Wall Nanotube dialog	37
Copolymer statistics	8	Build Multi-Wall Nanotube dialog	38
Branched polymers	9	Setup tab	38
Dialogs in Polymer Builder	10	Sequence tab	39
Block Copolymer dialog	10	Individual tab	40
Polymerize tab	11	Build Nanorope dialog	41
Add Block Definition dialog	12	Build Nanocluster dialog	42
Advanced tab	12	Shape tab	42
Branches tab	13	Substitution tab	44
Branch Points dialog	13	Transport Device Builder	46
Branch Points Properties dialog	14	Building a transport device	46
Edit Branch Points dialog	14	Dialogs in Transport Device Builder	47
Dendrimer dialog	15	Build Electrode dialog	47
Construct tab	15	Cleave Electrode dialog	48
Connection Points tab	16	Transport Device dialog	49
Homopolymer dialog	16	Mesostructure Builder	51
Polymerize tab	17	Beads and Bead Types	51
Advanced tab	18	Coarse Grainer	51
Branches tab	18	Mesomolecule Builder	51
Random Copolymer dialog	18	Mesostructure Template Builder	52
Polymerize tab	19	Mesostructure Builder	52
Add Repeat Unit dialog	20	Tasks in Mesostructure Builder	52
Probabilities tab	20	Creating and editing bead types	52
Reactivity Ratios tab	21	Coarse graining atoms to beads	54
Advanced tab	21	Patterns	54
Branches tab	22	Motion groups	55
Repeat Unit dialog	22	Subunits	55
Analog Builder	23	Using the Coarse Grainer	55
Creating an analog library	23	Potential problems in coarse graining ...	56

Building mesoscale polymers	57	Dialogs in Crystal Builder	90
Building mesoscale branched polymers	58	Build Crystal dialog	91
Building mesoscale dendrimers	59	Space Group tab	92
Constructing a mesostructure template	61	Lattice Parameters tab	92
Setting up a structured mesoscale system	63	Options tab	93
Dialogs in Mesostructure Builder	65	Build Vacuum Slab Crystal dialog	93
Bead Types dialog	66	Vacuum Slab tab	94
Bead Type Properties dialog	66	Options tab	95
Bead Type Defaults dialog	67	Add Atoms dialog	95
Coarse Grain dialog	67	Atoms tab	95
Build Mesomolecule dialog	69	Options tab	96
Mesomolecule Branches dialog	70	Surface Builder	97
Build Mesostructure Template dialog	70	Tasks in Surface Builder	97
Add Formers tab	71	Building and rebuilding surfaces	97
Fillers tab	73	Building a surface	98
Formers tab	73	Rebuilding a surface	100
Build Mesostructure dialog	74	Selecting a plane group	100
Components tab	75	Specifying the surface lattice	100
Options tab	75	Adding atoms to surfaces	101
Bead Packing Options dialog	76	Cleaving surfaces from crystals	102
Reduced Units dialog	76	Building a layered surface or interface	103
Crystal Builder	78	Removing surface periodicity	103
Tasks in Crystal Builder	78	Making a nonperiodic superstructure	104
Building and rebuilding crystals	78	Unbuilding a surface	104
Building a crystal	79	Redefining surface symmetry	104
Rebuilding a crystal	81	Creating a surface supercell	105
Selecting a space group	82	Redefining a surface lattice	105
Specifying the crystal lattice	82	Converting a surface to p1 symmetry	105
Adding atoms to crystals	83	Converting a surface to primitive representation	106
Adding atoms to structures	84	Converting a surface to conventional representation	106
Building crystals from surfaces	84	Dialogs in Surface Builder	107
Building a layered crystal	85	Build Surface dialog	107
Rebuilding a crystal	86	Plane Group tab	108
Removing crystal periodicity	86	Lattice Parameters tab	108
Making a nonperiodic superstructure	86	Options tab	109
Unbuilding a crystal	87	Cleave Surface dialog	109
Redefining crystal symmetry	87	Surface Box tab	110
Creating a crystal supercell	87	Surface Mesh tab	111
Redefining a crystal lattice	88	Options tab	112
Converting a crystal to P1 symmetry	88	Layer Builder	113
Converting a crystal to primitive representation	89	Building a layered crystal	113
Converting a crystal to conventional representation	89	Building a layered surface or interface	114
Converting a crystal to rhombohedral representation	90	Build Layers dialog	115
Converting a crystal to hexagonal representation	90	Define Layers tab	116
		Layer Details tab	116

Matching tab117

Options tab118

Building Tools

Introduction

In Materials Studio, the following Building tools are provided:

- [Polymer Builder](#)
- [Analog Builder](#)
- [Nanostructure Builder](#)
- [Transport Device Builder](#)
- [Mesostructure Builder](#)
- [Crystal Builder](#)
- [Surface Builder](#)
- [Layer Builder](#)

Further Information

For more information about the Materials Studio and other Accelrys software products, visit BIOVIA Support on the Web: <https://community.accelrys.com/index.jspa>

Polymer Builder

The purpose of polymer building in Materials Studio is to construct realistic atomistic models of polymer chains. These are then used as input to the property prediction and simulation modules of Materials Studio.

Chains are constructed from repeat units built using the Sketcher or taken from the extensive repeat unit library. Polymers can be built with specified tacticity and repeat unit orientation. Branched (comb and star) structures can also be built automatically.

Tasks in Polymer Builder

A feature of polymer molecules is their configurational variety. Even a simple vinyl homopolymer may have a range of different repeat unit configurations (head-to-tail, head-to-head, a mixture of the two) and chirality (tacticity). For copolymers, the combinations are even more complex. It is important to create a realistic polymer model, because repeat unit configuration often has a profound effect on the conformation (or shape), properties, and behavior of a polymer. Indeed, it is frequently the purpose of modeling to investigate these effects. A range of tools is available to help you construct and validate reasonable polymer configurations.

The following topics provide information on building polymers.

Creating and using repeat units

The repeat unit represents the structure of the monomer after it is chemically bonded into a polymer. In Materials Studio the head of the repeat unit is highlighted with a cyan wireframe and the tail is highlighted with a magenta wireframe. When the polymer is constructed, terminal dangling bonds are replaced with the original atoms (usually hydrogens) or terminal repeat units.

Fragments and repeat units

A repeat unit can be created from any fragment which is either sketched or imported into Materials Studio. A repeat unit is different from a fragment in that it has a defined head, tail and backbone. Initiators and terminators are special types of repeat units. They have a head atom and a backbone defined but no designated tail. The initiators and terminators will bond to the existing polymer chain at the endpoints, removing their head atoms in the process.

To create a repeat unit from a fragment

1. Sketch a fragment or open a 3D atomistic document containing one.
2. Choose *Build | Build Polymers | Repeat Unit* from the menu bar to open the [Repeat Unit](#) dialog.
3. Select the desired head atom and click the *Head Atom* button. A cyan wireframe is displayed around the atom to indicate that it is the head atom.
4. Select a different atom and click the *Tail Atom* button. A magenta wireframe is displayed around the atom to indicate that it is the tail atom. A set of atoms is now selected, this is the automatically defined backbone. Whenever you define a head and a tail, the backbone is automatically defined as the shortest connecting path between the two atoms. To modify the backbone path, select a set of atoms and click the *Set backbone atoms* button. The *Clear backbone atoms* button removes all of the selected atoms from the backbone definition.
5. Select a pseudo-chiral backbone atom, with different non-backbone substituents, and click the *Chiral Center* button to assign the atom as a chiral center. Chiral inversion and tacticity pertain to repeat units that contain pseudo-chiral centers.

Existing and sketched repeat units

Simple polymer chains can be constructed from repeat units stored in repeat unit libraries, or from sketched repeat units.

To use an existing repeat unit to build a polymer chain

1. Choose *Build | Build Polymers | Homopolymer* from the menu bar to open the Homopolymer dialog.
2. Select a repeat unit library from the *Library* dropdown list.
3. Select a repeat unit from the *Repeat unit* dropdown list.
4. Set other construction parameters on the [Polymerize](#) and [Advanced](#) tabs.
5. Click the *Build* button to create the polymer chain. The new chain will be displayed in a new 3D Atomistic Document.

To use a user defined repeat unit to build a polymer chain

1. With the desired repeat unit displayed in the 3D Viewer, choose *Build | Build Polymers | Homopolymer* from the menu bar to open the Homopolymer dialog.
2. Select the [Current project](#) entry in the *Library* dropdown list. This indicates that you want to use a repeat unit that is in your project, rather than one of the repeat units in the set of libraries that is provided.
3. Select the name of the document which contains your repeat unit in the *Repeat unit* dropdown list. If the document you have selected does not contain a valid repeat unit, you will be notified.
4. Set any construction parameters as above.
5. Click the *Build* button to construct the polymer chain. The constructed chain will be displayed in a new 3D Atomistic Document.

Building a random copolymer using probabilities

The distribution of repeat units in random (statistical) copolymers is not purely random but depends on conditional probabilities. These probabilities can be supplied directly or they can be calculated from reactivity ratios.

The conditional probabilities for chain growth can be supplied directly in Materials Studio as an $n \times n$ matrix. Each entry is a number less than 1 and the sum of the entries in each row is equal to 1. Summing the values to 1 guarantees that something is added to any given chain end.

For example, suppose there are three participants, *A*, *B*, *C*. Participant *A* has three probabilities: *A* bonding to *A*, *B* bonding to *A*, and *C* bonding to *A*. Each probability is less than or equal to 1, and they must sum to 1.

Alternatively, the conditional probabilities may be supplied in terms of monomer concentrations, M_i and [reactivity ratios](#), r_{ij} .

To build a linear random polymer using probabilities

1. Choose *Build | Build Polymers | Random Copolymer* from the menu bar to open the Random Copolymer dialog. Select the *Polymerize* tab.
2. Now create a list of repeat units that will be used in the polymer chain. Click in the empty *Repeat unit* cell in the *Define Repeat Units* grid, to open the [Add Repeat Unit](#) dialog.
3. Select a *Library* and a *Repeat unit* from the dropdown lists. If desired, specify chiral inversion and flip probability.
4. Click the *Add* button to add the specified repeat unit to the *Define Repeat Units* grid.
5. Continue adding repeat units until your list is complete.
6. The *Probabilities* matrix on the *Probabilities* tab allows you to view and edit the bonding probabilities of individual repeat units. Click in the matrix cell of the repeat unit you wish to modify and type in a new value.
7. When you have finished modifying probabilities, you may wish to set other construction parameters using the *Polymerize*, *Advanced* and *Branches* tabs.
8. When all of the parameters are set, click the *Build* button on the Random Copolymer dialog to construct the polymer chain. A new 3D Atomistic Document, containing the polymer, is created.

Building a block copolymer with branches

In the Materials Studio polymer builder, a bonding site that can be used as a branch point is a heavy atom bonded to one or more hydrogen atoms. An atom designated as a bonding site does not have to be in the backbone path. Materials Studio allows you to define specific bonding sites or random bonding sites.

If specific branch points are used, a bonding site is created at each atom specified. However, if random bonding sites are used only one atom is specified. Every atom in the backbone chain that matches this representative atom is a potential branch point. The number of potential branch points that become branch points is controlled by a user-defined probability.

The bonding limit is an attribute of a bonding site. It specifies the number of bonds that can be formed at a bonding site. The bonding limit of a bonding site atom must be less than or equal to the number of hydrogen atoms bonded to the atom.

The steric penalty is another attribute of a bonding site. It is directly related to bonding limits. It specifies the relative probability of multiple reactions at a bonding site. For each hydrogen bonded to a bonding site, there is an associated steric penalty value. For example, if a bonding site has a bonding limit of 2, the first steric penalty controls the first reaction at the bonding site, and the second steric penalty specifies the relative probability of the second reaction at that bonding site.

To build a block copolymer with branches you must first [construct the block copolymer](#), then [define the branches](#).

To construct the block copolymer

1. Select *Build | Build Polymers | Block Copolymer* from the menu bar to open the Block Copolymer dialog, then choose the *Polymerize* tab.
2. Create a list of repeat units that will be used in the polymer chain. Click in the empty *Repeat unit* cell in the *Block Definition* grid to open the *Add Block Definition* dialog.
3. Select a *Library* and a *Repeat unit* from the dropdown lists. Specify the *Block size*. If desired, specify chiral inversion and flip probability parameters.
4. Click the *Add* button to add the specified block to the *Block Definition* grid.
5. Continue adding blocks until your list is complete.
6. You can specify additional construction parameters via the controls on the [Advanced](#) tab.
7. When all of the parameters are set, click the *Build* button to construct the polymer chain. A new 3D Atomistic Document, containing the polymer, is created.

To define branch points

1. Select *Build | Build Polymers | Branch Points* from the menu bar to open the Branch Points dialog.
2. Select the backbone atoms that you want to designate as branch points and click the *Create* button to create the branch points. An orange wireframe is displayed to highlight the bonding atoms of branch points.
3. You can delete branch points by deleting the wireframe object.
4. Branch points are created with default steric penalties unless you specify a different value in advance. To specify different values click the *Properties...* button on the Branch Points dialog to open the [Branch Points Properties](#) dialog.
5. You can view and modify steric penalties for individual branch points after they have been created. Select a branch point in the structure and click the *Edit...* button on the Branch Points dialog to display the [Edit Branch Points](#) dialog.
6. When you have defined the branch points, select *Build | Build Polymers | Homopolymer* from the menu bar to open the Homopolymer dialog.
7. Choose a repeat unit from the one of the listed libraries on the *Polymerize* tab.
8. If desired, make changes to other polymerization parameters. Remember that the chain being constructed will become a branch on the existing block copolymer.
9. Check the *Attach constructed polymer chain to branch points* checkbox on the *Branches* tab to specify that the new chain is a branch and may be attached to the specified branch points.
10. Click the *Build* button on the Homopolymer dialog to construct the branched polymer chain. A new 3D Atomistic Document, containing the polymer, is created.

If build dendrimer fails

When you click the *Build* button on the [Dendrimer](#) dialog a variety of tests are performed both on the seed structure and on the specified repeat unit to determine their suitability for the requested dendrimer construction. If any of these tests fails then the construction of the dendrimer will not proceed.

Seed structure

The seed structure must contain one or more connection points. To mark specific atoms as connection points, you should select the desired atoms and use the [Connection Points](#) tab of the Dendrimer dialog to mark these as connection points.

Repeat unit

The repeat unit to be used for the dendrimer build must contain exactly one dendrimer repeat unit. If the *Repeat unit* is set to [Selection](#), then the currently selected portion of the current 3D Atomistic document will be used. If there is no selection in this document then the whole document will be used.

Among the possible reasons for the build dendrimer operation to fail, or produce unexpected results are:

- The selection does not contain a single dendrimer repeat unit.
- The selection contains a single dendrimer repeat unit plus some other atoms not in the dendrimer repeat unit. Those atoms outside the dendrimer repeat unit will be ignored.
- The single dendrimer repeat unit does not contain a head atom, or contains more than one head atom.
- The *Number of generations* is greater than 1 and there is no connection point in the selected dendrimer repeat unit.

If you are trying to use as a repeat unit a structure that was created using the dendrimer builder, it will probably be necessary to redefine the dendrimer repeat unit before attempting to build the new dendrimer. The following procedures may be useful:

- Polymer hierarchy objects (such as dendrimer repeat units) may be removed from a structure. To do this, use the right mouse to obtain the shortcut menu, and select the *Remove polymer hierarchy* item.
- If, when you select an atom for creating a connection point, the selected atom does not belong to a dendrimer repeat unit then the action of creating a connection point will also create a dendrimer repeat unit from the whole fragment to which the atom belongs.

Theory in Polymer Builder

The following topics contain information supporting the theory behind polymer building in Materials Studio.

Introduction

The synthesis of polymer molecules requires consideration of the reaction kinetics that control the insertion of monomer units into growing chain ends. The product of a synthesis can be vastly different, depending on the purity of the feedstock and the choice of catalyst, temperature, additives, and type of reactor. The microstructure of the chains, the probability of ring formation and/or branching, and the molecular weight distributions can all vary with these factors.

Materials Studio's polymer building implementation provides the capability to build molecules with structures that are accurate representations of real molecules. Although it does not include detailed kinetic models, it does provide sufficient flexibility so that kinetic information can be used to construct molecules of choice.

For molecules of regular structure, including ideal homopolymers, alternating copolymers, and block copolymers, the rules for addition of repeat units to the ends of growing chains are well-defined. Once the structure is specified in detail, the software can easily build the molecule. However, when an element of randomness enters into the building process, decisions must be made as to how the statistics of the addition process are to be handled.

Any algorithm that builds a linear system containing statistical correlations must function by the sequential selection of events that is based on the outcome of previous steps in the building process. This means that kinetic data must be converted, either automatically or manually, to a Markov transition probability scheme. A decision must be made, on the basis of chemical information, regarding

how far the correlations extend in order to define the kinetic or statistical scheme for software implementation. In Materials Studio, attention is restricted to the instantaneous terminal model of polymerization kinetics. Penultimate and antepenultimate models, which make heavy demands on you to supply the kinetic information required by the software to grow the chains one repeat unit at a time, have not been implemented.

The conversion of conditional probabilities to monomer concentration and reactivity ratio information or vice versa is not difficult (see [Eq. Th-2](#) and [Eq. Th-3](#)). Polymer building in Materials Studio provides both types of data entry to control the sequence distribution. You have the option of controlling tacticity and orientation of each repeat unit with conditional probabilities that operate independent of the sequence distribution probabilities or of using a full conditional probability or kinetic scheme for sequence, tacticity, and orientation simultaneously. The latter option is accomplished by defining the various conformers of a given repeat unit as different repeat units. These options provide great flexibility in building molecules of interest.

Conditional probabilities

As noted in the [Introduction](#), the sequence in a statistical copolymer depends on the growing chain's last repeat unit and the given conditional probabilities. These conditional probabilities are denoted by:

$$P(A_j | A_i)$$

They give the probability for adding repeat unit A_j given that the chain ends in repeat unit A_i . Since a repeat unit of one kind or another must follow any given repeat unit in an infinite chain, it follows that:

Eq. Th-1

$$\sum_{j=1}^N P(A_j | A_i) = 1$$

where the sum is over the entire set of repeat units that may be added to the growing chain.

Reactivity ratios

A reactivity ratio is the quotient of the rate constant for a given chain end bearing a repeat unit of type i to add the same type i of repeat unit with respect to the rate constant for adding a repeat unit of type j . That is:

Eq. Th-2

$$r_{ij} = \frac{k_{ii}}{k_{ij}}$$

where k_{ii} is the rate constant for the addition of monomer i to a chain ending in repeat unit i , and k_{ij} is the rate constant for the addition of monomer j to a chain ending in repeat unit i .

Note: Smaller values of r_{ij} imply a greater probability of a type j repeat unit attaching to a chain ending with a type i repeat unit. Similarly, higher values of r_{ij} result in fewer attachments of type j repeat units to growing chains ending with a type i repeat unit.

From the given reactivity ratios and concentrations M_i conditional probabilities are obtained from:

Eq. Th-3

$$P(A_j | A_i) = \frac{M_j / r_{ij} M_i}{1 + \sum_{\substack{k=1 \\ k \neq j}}^n M_k / r_{ik} M_i}$$

according to a generalization of the equations in Chapter 4 of Koenig (1980).

Note: In the software, any given reactivity ratio of zero is replaced with an arbitrarily chosen small number in order to avoid divisions by zero.

Copolymer statistics

The appearance of runs of repeat units in statistical chains is characterized by a sequence of probabilities for the occurrence of *monads* (single repeat units) with probabilities $P_1(A_i)$, *dyads* (pairs of repeat units) with probabilities $P_2(A_i A_j)$, *triads* with probabilities $P_3(A_i A_j A_k)$, and so on. The definition of conditional probabilities is sufficient to verify the relations:

Eq. Th-4

$$P_2(A_i A_j) = P_1(A_i) P(A_j | A_i)$$

Eq. Th-5

$$P_3(A_i A_j A_k) = P_2(A_i A_j) P(A_k | A_i A_j)$$

and so on. The dyad and triad probabilities are expressed in terms of the monad and conditional probabilities. The chance of getting i, j is the product of the probability of getting i and the probability of getting j given i . Likewise, the chance of getting i, j, k is the product of the probability of getting i and the probability of getting j given i , and the probability of getting k given j (based on the instantaneous assumption).

The kinetics model that the software uses is the terminal model, so that the identity of A_i in the term $P(A_k | A_i A_j)$ in [Eq. Th-5](#) is of no consequence. It follows that $P(A_k | A_i A_j) = P(A_k | A_j)$ for the terminal model, and hence that:

Eq. Th-6

$$P_3(A_i A_j A_k) = P_1(A_i) P(A_j | A_i) P(A_k | A_j)$$

with use of [Eq. Th-4](#) for the dyad term in [Eq. Th-5](#). It is easy to see how higher-order sequence probabilities are constructed from the monad distributions and conditional probabilities.

The monad probabilities $P_1(A_i)$ are calculated from conditional probabilities with use of the law of total probability. This law states that, for the infinite sequences of interest here, the probability of finding repeat i is the sum of all probabilities for the dyads $A_j A_i$ or $A_i A_j$, since a repeat unit A_i must be followed by or preceded by another repeat unit:

Eq. Th-7

$$P_1(A_i) = \sum_{j=1}^N P_2(A_j A_i) = \sum_{j=1}^N P_2(A_i A_j)$$

Upon substituting [Eq. Th-4](#) into [Eq. Th-7](#), defining conditional probability for the dyad in terms of the monad, you obtain:

Eq. Th-8

$$P_1(A_i) = \sum_{j=1}^N P_1(A_i)P(A_j | A_i) = \sum_{j=1}^N P_1(A_j)P(A_i | A_j)$$

The first equality is the tautological $P_1(A_i) = P_1(A_i)$, because of [Eq. Th-1](#). The second part of [Eq. Th-8](#) gives a useful relation between the monad and conditional probabilities.

The essential part of [Eq. Th-8](#) may be rewritten as:

Eq. Th-9

$$\sum_{j=1}^N P_1(A_j) [\delta_{ij} - P(A_i | A_j)] = 0$$

where δ_{ij} is the Kronecker delta having the value 1 when $i = j$ and 0 otherwise. This equation is solved by the software using Gaussian elimination on the matrix $\delta_{ij} - P(A_i | A_j)$, and the values of $P_1(A_i)$ are calculated and reported. With these values in hand, the program goes on to evaluate [Eq. Th-4](#) for the dyad probabilities and [Eq. Th-6](#) for the triad probabilities.

The first repeat unit in an n-copolymer chain, whether it be initiated or not, is chosen in proportion to its monad probability by a balloting procedure. Chains that are constructed with initiators and terminators are given the same statistical treatment as if they were infinite. That is, probabilities for occurrence of initiators and terminators are set to zero. Should this scheme be unsatisfactory for your application, it is relatively easy for you to correct the output of the module manually by adding in terms for the initiator and terminator and then renormalizing the probabilities. The equations given above supply all that you need to effect this transformation.

Branched polymers

Branched polymers fall into four major categories: combs, stars, starburst (dendrimer), and random. Comb polymers consist of a backbone chain with side chains pendant on the backbone at more or less regular intervals (for example, at every repeat unit of the backbone with probability 0.95, which is probably dependent on the chemistry that is used to graft the side chains onto the backbone chain). The structure of this class of molecules is illustrated in [Figure 1\(a\)](#).



Figure 1. Sketch of Typical (a) Comb and (b) Star Polymers

Stars contain a single branch point. Good examples of molecules of this type are provided by the polyoxypropylene polyols, in which the central branch point might be glycerol or pentaerythritol, and the branches (three or four, respectively) are polyoxypropylene chains. Molecules in this class look a bit like the sketch in [Figure 1\(b\)](#).

Dendrimers or starburst polymers have a carefully controlled structure. They consist of a number of tiers, as shown in [Figure 2\(a\)](#). Dendrimers invariably have a limited molecular weight, because the branches eventually reach a tier that is so crowded that further reactions to create the next tier are sterically forbidden, and further growth cannot be achieved.

Random branched structures are generally formed when monomers, prepolymers, or macromers bearing more than two reactive groups are polymerized or cured with a crosslinking agent. The resulting structures have complex topology (connectivity), as illustrated in [Figure 2\(b\)](#). Since the reactions are controlled by statistical mechanisms, the resulting materials are invariably mixtures of molecules having a broad distribution of both topology and molecular weight. If the extent of bond formation in these systems is sufficiently large, they form a gel molecule that is of macroscopic dimensions. Randomly branched systems are comprehensible with statistical theory (Flory 1941, 1953; Stockmayer 1943).

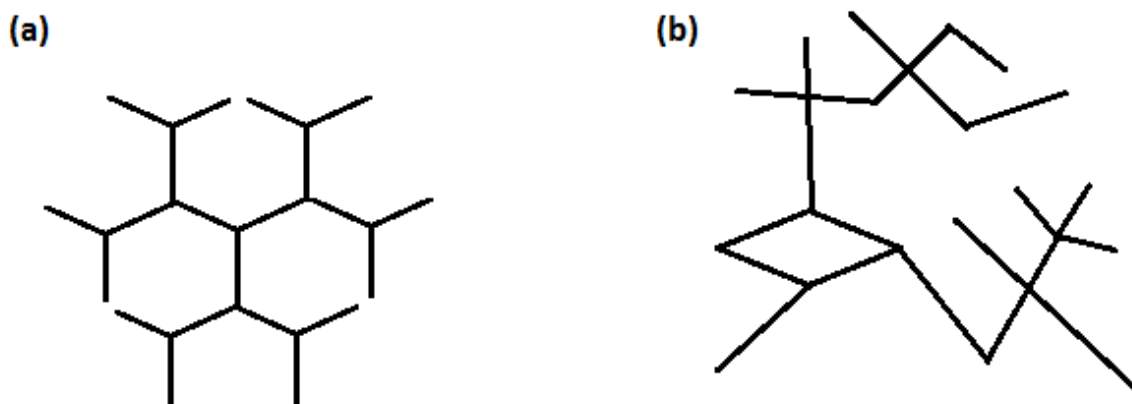


Figure 2. Typical Sketch of (a) Starburst (dendrimer) and (b) Random Polymers

Besides linear molecules, Materials Studio can build comb and star molecules, since neither is particularly crowded and both are of potentially infinite size. A laboratory synthesis of a comb polymer might begin with a backbone molecule containing repeat units that are susceptible to subsequent reaction. To these sites are grafted other chains, which in general have a different composition than the backbone. This is the construction process that Materials Studio enables you to emulate for building combs.

Ideal reactions that go to completion without side reactions are seldom realized in practice. When building polymers, Materials Studio therefore accepts probabilities for bond formation as input into the step in which the branches of a comb or a star are attached to the backbone or coupling agent, respectively. Furthermore, sequential reactions at a crowded site might entail decreasing probabilities for the addition of the second, third, etc. chain branch, and this can be simulated with the use of steric penalties.

Dialogs in Polymer Builder

Materials Studio's polymer building functionality allows you to create polymers of nearly any type or configuration. By manipulating the inputs to the various Polymer Builder dialogs, polymers of various geometries can be generated.

Block Copolymer dialog

The Block Copolymer dialog allows you to build block and alternating copolymer chains with various orientations and tacticities.

It contains the following tabs:

- [Polymerize](#): Contains the controls commonly used for polymerization.
- [Advanced](#): Contains less frequently used controls for polymerization.
- [Branches](#): Allows you to specify that the polymer currently being built will be used as a branch in the construction process.

Build: Generates a new 3D Atomistic Document containing the constructed polymer.

Note: Typically, the nature of active document has no relevance to the *Build* operation. However, if the *Attach constructed polymer chain to branch points* option is checked, the active 3D Atomistic document must contain a polymer with branch points defined. The new polymer is created in a new 3D Atomistic document, which contains a copy of the previous active model, to which the new chains have been added.

Build | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Note: Use of a user-defined, system-defined, or random torsion can lead to situations where rings are speared by the growing chain. Simple minimization of the structure will not eliminate such features and the polymer should either be re-built with a different choice of torsion, or built using an Amorphous Cell, which constructs a cell containing the structure. Amorphous Cell has a specific checking tool for spearrings and catenations, the result of this checking will cause the calculation to fail, reasons for which are shown in the Project Log.

Help: Displays the Help topic for the current tab.

Access methods

Menu	<i>Build Build Polymers Block Copolymer</i>
------	---

Polymerize tab

The *Polymerize tab* includes options for building typical block copolymers.

The *Block definition* grid lists all of the repeat units currently defined, together with additional information about each repeat unit.

Repeat Unit: Displays the name of the repeat unit.

Block Size: Displays the number of times the specified repeat unit will be duplicated in the construction of this block. Default = 5. Range = 1 to 1000.

Invert: Displays the probability of a chiral inversion between repeat units of a block. Default = 0.0. Range = 0.0 to 1.0. The default of 0.0 corresponds to an isotactic construction. A value of 1.0 corresponds to an syndiotactic construction. Any value between 0.0 and 1.0 will result in an atactic construction.

Flip: Displays the probability that a repeat unit will flip its orientation. Default = 0.0. Range = 0.0 to 1.0. The default of 0.0 corresponds to a head to tail construction, while a value of 1.0 corresponds to a head to head construction.

Add new repeat units by clicking in the *Repeat unit* column on the last (empty) row of the grid. The [Add Block Definition](#) dialog is displayed. This allows you to specify repeat units and their properties. New repeat units are then displayed in the *Block definition* grid.

To edit existing repeat unit parameters, click in the appropriate grid cell and type the new value.

Delete repeat units by clicking on the *Repeat Unit* name in the grid and press the DELETE key.

Number of superunits: Specifies how many superunits are built. Default = 1. Range = 1 to 2000.

Number of chains: Specifies the number of chains to build. Default = 1. Range = 1 to 2000.

Access methods

Menu	<i>Build Build Polymers Block Copolymer Polymerize</i>
------	--

Add Block Definition dialog

The Add Block Definition dialog allows you to specify information about a specific repeat unit which is then used to construct the block copolymer.

Library: Select the library to which the repeat unit belongs from the list of system libraries. This selection controls which repeat units are available in the *Repeat unit* list.

Repeat unit: Select the name of the repeat unit from the specified library or from an existing structure on the screen.

Block size: Specify the number of times the repeat unit will be duplicated when this block is constructed. Default = 5. Range = 1 to 1000.

Chiral inversion: Specify the probability of a chiral inversion between repeat units of a block. Default = 0.0. Range = 0.0 to 1.0. The default of 0.0 corresponds to an isotactic construction. A value of 1.0 corresponds to an syndiotactic construction. Any value between 0.0 and 1.0 will result in an atactic construction.

Flip probability: Specify probability that a repeat unit will flip its orientation. Default = 0.0. Range = 0.0 to 1.0. The default of 0.0 corresponds to a head to tail construction, while a value of 1.0 corresponds to a head to head construction.

Add: Adds the newly defined repeat unit to the *Block definition* grid and keeps the Add Block Definition dialog open.

Cancel: Closes the Add Block Definition dialog without applying any further changes.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build</i> <i>Build Polymers</i> <i>Block Copolymer</i> <i>Polymerize</i> <i>Repeat unit</i>
------	--

Advanced tab

Initiator: Specifies a repeat unit to bond to the initial tail atom of the polymer chain. Default = None (the tail atom designator is removed and the existing atom becomes the initiator).

Terminator: Specifies a repeat unit to bond to the final head or tail atom of the polymer chain. (Whether the final atom is a head atom or a tail atom will depend on the *Flip* probability.) Default = None (the head or tail atom designator is removed and the existing atom becomes the terminator).

Torsion: Specifies the torsion angle in degrees between new repeat units. Default = 180.0. Range = -180.0 to 180.0.

Note: This option is enabled only if *Random* is unchecked.

Random: When checked, indicates that the torsion angle between the repeat units will be generated at random, in the range -180.0 to 180.0.

Use simple names for constructed polymers: Indicates whether the name of the constructed polymer should be defined with simple rules. Default = unchecked (use more complicated rules in the naming scheme).

Use random number seed: Indicates whether the construction method should use a defined random number seed. Since there is a randomness in all of the construction techniques, this provides a way to duplicate reliably results from one session to the next. Default = unchecked (the random seed is not specified).

If the *Use random number seed* checkbox is checked, you can enter a value to seed the random number generator for the polymerization. Range = 0.0 to 1.0.

Access methods

Menu *Build | Build Polymers | Block Copolymer | Advanced*

Branches tab

Attach constructed polymer chain to branch points: Specifies whether or not the polymer under construction is to be used as a branch that will be bonded to the specified branch points in the displayed polymer. Default = [unchecked](#).

Access methods

Menu *Build | Build Polymers | Block Copolymer | Branches*

Branch Points dialog

The Branch Points dialog allows you to add branch points to an existing repeat unit or to modify previously defined branch points.

Branch points specification

These controls determine how the branch points will be specified when the polymer is constructed.

Specific: Indicates that only the selected atoms in the polymer will be used as branch points.

Representative: Indicates that a selected atom will be used as a representative branch point. All equivalent atoms in similar monomers in the polymer will also be potential branch points.

Note: Only one atom may be selected as the representative atom.

Probability: Specify the probability that any representative branch point actually becomes a branch point.

Note: This option is enabled only if *Representative* branch points are used.

Properties...: Displays the [Branch Points Properties](#) dialog, which allows you to modify the default properties of potential branch points.

Branch points manipulation

Create: Creates branch points using the options specified.

Note: When the *Representative* option is chosen, the *Create* button is enabled only if a single atom is selected.

Delete: Deletes the currently selected branch points from the polymer.

Edit...: Displays the [Edit Branch Points](#) dialog, which allows you to modify the properties of selected branch points.

Note: The *Edit...* button is enabled only if branch points in the current polymer are selected.

Branch points markers

Display: Highlights the branch points in the current polymer.

Hide: Hides the branch points in the current polymer.

Help: Displays the Help topic in a browser.

Access methods

Menu *Build | Build Polymers | Branch Points*

Branch Points Properties dialog

The Branch Points Properties dialog allows you to modify the default steric penalties used to create new branch points. The parameters are displayed in a grid.

Branch Number: Displays the number of the given branch point.

Steric Penalty: Specify the steric penalty value for the given branch point. Range = 0.0 to 1.0. A steric penalty of 1.0 indicates a maximum penalty, and no branch will occur. A penalty of 0.0 corresponds to no penalty, and a branch will occur.

To edit steric penalty parameters, click in the appropriate grid cell and type the new value.

To add additional branches to the grid, type a non-unit value in the *Steric penalty* grid cell of the last row in the grid.

Note: You can specify steric penalties for a maximum of 8 branches.

OK: Updates the branch point properties with any changes and closes the dialog.

Cancel: Closes the dialog without editing the branch point properties.

Help: Displays the Help topic in a browser.

Access methods

Menu *Build | Build Polymers | Branch Points | Properties...*

Edit Branch Points dialog

The Edit Branch Points dialog allows you to modify the steric penalties of existing branch points. The parameters are displayed in a grid.

Branch Number: Displays the number of the given branch point.

Steric Penalty: Specify the steric penalty value for the given branch point. Range = 0.0 to 1.0. A steric penalty of 1.0 indicates a maximum penalty, and no branch will occur. A penalty of 0.0 corresponds to no penalty, and a branch will occur.

Note: If no branch point is selected, then editing branch point steric penalties will act on all branch points in the active document. If one or more branch points are selected, any changes to the steric penalties are applied only to those selected branch points.

To edit steric penalty parameters, click in the appropriate grid cell and type the new value.

Tip: If multiple branches with conflicting steric penalties are selected, the *Steric Penalty* grid cell for the corresponding branch point will be blank.

OK: Updates the branch points with any changes and closes the dialog.

Cancel: Closes the dialog without editing the branch points.

Help: Displays the Help topic in a browser.

Access methods

Menu *Build | Build Polymers | Branch Points | Edit...*

Dendrimer dialog

The Dendrimer dialog allows you to build dendrimers.

It contains the following tabs:

- **Construct:** Contains commonly used controls for dendrimer construction.
- **Connection Points:** Allows you to define a structure as a dendrimer repeat unit and to add, remove and display the connection points in either the dendrimer repeat unit or the dendrimer.

Build: Builds the polymer according to the specified options on the dialog.

Note: When building dendrimers, if the *Add generations to constructed dendrimer* option is checked, the active document must contain a dendrimer with connection points defined. The generations are then added to the structure in the active document.

Build | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Note: Use of a user-defined, system-defined, or random torsion can lead to situations where rings are speared by the growing chain. Simple minimization of the structure will not eliminate such features and the polymer should either be re-built with a different choice of torsion, or built using an Amorphous Cell, which constructs a cell containing the structure. Amorphous Cell has a specific checking tool for spearings and catenations, the result of this checking will cause the calculation to fail, reasons for which are shown in the Project Log.

Help: Displays the Help topic for the current tab.

Access methods

Menu *Build | Build Polymers | Dendrimer*

Construct tab

The *Construct* tab contains the options most commonly used when constructing a dendrimer.

The construction method used is controlled by the *Build option* you choose:

Create a new dendrimer: When selected, indicates that a new dendrimer will be created using the options specified. The new polymer will be created in a new 3D Atomistic Document.

Add generations to constructed dendrimer: When selected, indicates that the specified building block will be added to the defined connection points on the existing dendrimer.

Seed structure: Select a seed structure from either the list of dendrimer seeds provided or a structure on the screen. A dendrimer seed is any fragment that contains at least one connection point.

Repeat unit: Select a repeat unit to use in dendrimer construction either from the list of repeat units provided or a dendrimer repeat unit on the screen. A dendrimer repeat unit is any fragment that has a head atom (which will be attached to the existing dendrimer). Dendrimer repeat units may also contain one or more connection points.

Note: Entries in the provided list of dendrimer repeat units that start with `end_` do not have connection points. Since these structures terminate the dendrimer you cannot add more than one generation.

Number of generations: Specify how many generations will be added to each connection point during the construction process.

Use simple names for constructed polymers: Indicate whether the constructed dendrimer should have a simple default name, or the name of the dendrimer should be based upon the names of the repeat units. Default = [checked](#).

Access methods

Menu [Build](#) | [Build Polymers](#) | [Dendrimer](#) | [Construct](#)

Connection Points tab

The *Connection Points* tab contains mechanisms which allow you to manipulate a fragment, which can then be used in the dendrimer building process, or the connection points of an existing dendrimer to influence growth in the next generation.

Connection point definition

The *Set* and *Clear* buttons in the *Connection point definition* section allow you to set and clear the connection points on the existing fragment. The fragment may be a seed or an existing dendrimer with previously defined connection points.

Set: In the *Connection point definition* section, defines the currently selected atoms in the structure as connection points. If atoms are already defined as connection points, the currently selected atoms are added to the list.

Clear: Clears the selected connection points from the current dendrimer or seed.

Head atom definition

Set: In the *Head atom definition* section, defines the currently selected atom to be the head atom in a fragment. The head atom is the atom which will be removed in a bonding operation when the fragment is bonded to an existing seed or dendrimer. While any number of connection points are allowed on a fragment, only one head may be defined.

Connection point display

Display: Displays the connection points in the specified fragment or dendrimer.

Hide: Hides the connection points on the specified fragment or dendrimer.

Access methods

Menu [Build](#) | [Build Polymers](#) | [Dendrimer](#) | [Connection Points](#)

Homopolymer dialog

The Homopolymer dialog allows you to build homopolymer chains with various orientations and tacticities.

It contains the following tabs:

- [Polymerize](#): Contains the commonly used parameters for homopolymerization.
- [Advanced](#): Contains less frequently used parameters for homopolymerization.
- [Branches](#): Allows you to specify that the polymer currently being built will be used as a branch in the construction process.

Build: Generates a new 3D Atomistic Document containing the constructed polymer.

Note: Typically, the nature of active document has no relevance to the *Build* operation. However, if the *Attach constructed polymer chain to branch points* option is checked, the active 3D Atomistic document must contain a polymer with branch points defined. The new polymer is created in a new 3D Atomistic document, which contains a copy of the previous active model, to which the new chains have been added.

Build | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Note: Use of a user-defined, system-defined, or random torsion can lead to situations where rings are speared by the growing chain. Simple minimization of the structure will not eliminate such features and the polymer should either be re-built with a different choice of torsion, or built using an Amorphous Cell, which constructs a cell containing the structure. Amorphous Cell has a specific checking tool for spearings and catenations, the result of this checking will cause the calculation to fail, reasons for which are shown in the Project Log.

Help: Displays the Help topic for the current tab.

Access methods

Menu	<i>Build Build Polymers Homopolymer</i>
------	---

Polymerize tab

The Homopolymer dialog controls the variables used in constructing a homopolymer. The *Polymerize* tab includes options for most typical polymerizations.

Library: Select a library from the list of system libraries. This selection controls which repeat units are available in the *Repeat unit* list.

Repeat unit: Select the name of the repeat unit from the specified library or from an existing structure on the screen. All repeat units must have defined head and tail atoms.

Tacticity: Specify the tacticity of the repeat units in the constructed polymer. Possible values are *Isotactic* (default), *Syndiotactic*, and *Atactic*.

If you specify *Isotactic* tacticity, the chiral inversion value changes to **0.0**. If you specify *Syndiotactic* tacticity, the chiral inversion value changes to **1.0**. Likewise, modifications of the chiral inversion value may force changes to the tacticity type.

Chiral inversion: Specify the chiral inversion probability. This value is closely related to *Tacticity*. The chiral inversion is inactive if *Isotactic* or *Syndiotactic* are selected. In order to set the chiral inversion to a value other than **0.0** or **1.0**, *Tacticity* must be changed to *Atactic*. See the discussion of Tacticity above. Range = **0.0** to **1.0**. No values outside the range are accepted.

Chain length: Specify the length of the chain in terms of multiples of repeat units. Default = **10**. Range = **1** to **2000**.

Number of chains: Specify the number of chains to be built using currently defined values. Default = **1**. Range = **1** to **2000**.

Access methods

Menu	<i>Build Build Polymers Homopolymer Polymerize</i>
------	--

Advanced tab

The *Advanced* tab includes parameters and options that are less commonly used to build a homopolymer.

Orientation: Specifies the orientation of the repeat units with respect to each other in the constructed chain. Possible values = [Head-to-Tail](#) (default), [Head-to-Tail](#), or [Random](#).

Flip probability: Specifies the probability that a repeat unit will flip its orientation. Default = [0.0](#). Range = [0.0](#) to [1.0](#).

Initiator: Specifies a repeat unit to bond to the initial tail atom of the polymer chain. Default = [None](#) (the tail atom designator is removed and the existing atom becomes the initiator).

Terminator: Specifies a repeat unit to bond to the final head or tail atom of the polymer chain. (Whether the final atom is a head atom or a tail atom will depend on the *Flip probability*.) Default = [None](#) (the head or tail atom designator is removed and the existing atom becomes the terminator).

Torsion: Specifies the torsion angle in degrees between new repeat units. Default = [180.0](#) degrees. Range = [-180.0](#) to [180.0](#).

Note: This option is enabled only if *Random* is unchecked.

Random: When checked, indicates that the torsion angle between the repeat units will be generated at random, in the range [-180.0](#) to [180.0](#).

Use random number seed: Indicates whether the construction method should use a defined random number seed. Since there is a randomness in all of the construction techniques, this provides a way to duplicate reliably results from one session to the next. Default = [unchecked](#) (the random seed is not specified).

If the *Use random number seed* checkbox is checked, you can enter a value to seed the random number generator for the polymerization. Range = [0.0](#) to [1.0](#).

Access methods

Menu [Build](#) | [Build Polymers](#) | [Homopolymer](#) | [Advanced](#)

Branches tab

Attach constructed polymer chain to branch points: Specifies whether or not the polymer under construction is to be used as a branch that will be bonded to the specified branch points in the displayed polymer. Default = [unchecked](#).

Access methods

Menu [Build](#) | [Build Polymers](#) | [Homopolymer](#) | [Branches](#)

Random Copolymer dialog

The Random Copolymer dialog allows you to build random copolymer chains with various orientations, tacticities and conditional probabilities.

It contains the following tabs:

- [Polymerize](#): Contains commonly used parameters for a polymerization.
- [Probabilities](#): Allows you to specify the conditional reaction probabilities.
- [Reactivity Ratios](#): Provides access to the reactivity ratios matrix.
- [Advanced](#): Contains less commonly used parameters for a polymerization.
- [Branches](#): Allows you to specify that the polymer currently being built will be used as a branch in the construction process.

Build: Generates a new 3D Atomistic Document containing the constructed polymer.

Note: Typically, the nature of active document has no relevance to the *Build* operation. However, if the *Attach constructed polymer chain to branch points* option is checked, the active 3D Atomistic document must contain a polymer with branch points defined. The new polymer is created in a new 3D Atomistic document, which contains a copy of the previous active model, to which the new chains have been added.

Similarly, when building dendrimers, if the *Add generations to constructed dendrimer* option is checked, the active document must contain a dendrimer with connection points defined.

Build | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Note: Use of a user-defined, system-defined, or random torsion can lead to situations where rings are speared by the growing chain. Simple minimization of the structure will not eliminate such features and the polymer should either be re-built with a different choice of torsion, or built using an Amorphous Cell, which constructs a cell containing the structure. Amorphous Cell has a specific checking tool for spearings and catenations, the result of this checking will cause the calculation to fail, reasons for which are shown in the Project Log.

Help: Displays the Help topic for the current tab.

Access methods

Menu [Build | Build Polymers | Random Copolymer](#)

Polymerize tab

The *Polymerize* tab includes options for building typical random copolymers.

The *Define repeat units* grid lists all of the repeat units currently defined, together with additional information about each repeat unit.

Repeat Unit: Displays the name of the repeat unit.

Invert: Displays the probability of a chiral inversion between repeat units of a block. Default = 0.0. Range = 0.0 to 1.0. The default of 0.0 corresponds to an isotactic construction. A value of 1.0 corresponds to an syndiotactic construction. Any value between 0.0 and 1.0 will result in an atactic construction.

Flip: Displays the probability that a repeat unit will flip its orientation. Default = 0.0. Range = 0.0 to 1.0. The default of 0.0 corresponds to a head to tail construction, while a value of 1.0 corresponds to a head to head construction.

Add new repeat units by clicking in the *Repeat unit* column on the last (empty) row of the grid. The [Add Repeat Unit](#) dialog is displayed. This allows you to specify repeat units and their properties. New repeat units are then displayed in the *Define Repeat Units* grid.

To edit existing repeat unit parameters, click in the appropriate grid cell and type the new value.

Delete repeat units by clicking on the *Repeat unit* name in the grid and press the DELETE key.

Propagate using: Specifies whether the [Probabilities](#) or [Reactivity ratios](#) method of determining which repeat unit is to be added to the chain will be used. Probabilities are defined on the [Probabilities](#) tab. Reactivity ratios are defined on the [Reactivity Ratios](#) tab.

Force concentrations: When checked, requires the constructed polymer to have exactly the concentration specified in either the *Reactivity Ratios* or *Probability* controls. Default = [unchecked](#).

Chain length: Specifies number of repeat units in chain. Default = [10](#). Range = [1](#) to [2000](#).

Number of chains: Specifies the number of chains to build. Default = [1](#). Range = [1](#) to [2000](#).

Access methods

Menu	Build Build Polymers Random Copolymer Polymerize
------	--

Add Repeat Unit dialog

The Add Repeat Unit dialog allows you to specify information about a specific repeat unit which is then used to construct the random copolymer.

Library: Select the library to which the repeat unit belongs from the list of system libraries. This selection controls which repeat units are available in the *Repeat unit* list.

Repeat unit: Select the name of the repeat unit from the specified library or from an existing structure on the screen.

Chiral inversion: Specify the probability of a chiral inversion between repeat units of a block. Default = [0.0](#). Range = [0.0](#) to [1.0](#). The default of [0.0](#) corresponds to an isotactic construction. A value of [1.0](#) corresponds to an syndiotactic construction. Any value between [0.0](#) and [1.0](#) will result in an atactic construction.

Flip probability: Specify probability that a repeat unit will flip its orientation. Default = [0.0](#). Range = [0.0](#) to [1.0](#). The default of [0.0](#) corresponds to a head to tail construction, while a value of [1.0](#) corresponds to a head to head construction.

Add: Adds the newly defined repeat unit to the *Define Repeat Unit* grid and keeps the Add Repeat Unit dialog open.

Close: Closes the Add Repeat Unit dialog without applying any further changes.

Help: Displays the Help topic in a browser.

Access methods

Menu	Build Build Polymers Random Copolymer Polymerize Repeat unit
------	--

Probabilities tab

The *Probabilities* tab allows you to define and modify reaction probabilities. The probabilities are displayed in matrix form. The row headings denote which repeat unit is at the growing end of the chain, the next repeat unit to be added is represented by the column headings. The repeat units are specified on the [Polymerize](#) tab.

Probabilities: Specify the relative probabilities of specific repeat units following any given repeat unit during chain construction. Range = any real number greater than [0.0](#) and less than or equal to [1.0](#).

Note: The probabilities must sum to 1.0 across a row.

To edit existing probability parameters, click in the appropriate matrix cell and type the new value.

Access methods

Menu [Build](#) | [Build Polymers](#) | [Random Copolymer](#) | [Probabilities](#)

Reactivity Ratios tab

The *Reactivity Ratios* tab allows you to define and modify the reactivity ratios and concentrations of the various repeat units. The parameters are displayed in matrix form. The row headings denote which repeat unit is at the growing end of the chain, the next repeat unit to be added is represented by the column headings. The repeat units are specified on the [Polymerize](#) tab.

Reactivity Ratios: Specify the reactivity ratios of specific repeat units with respect to a given repeat unit and the concentrations of each repeat unit. Range = any real number greater than [0.0](#).

Note: Since the definition of reactivity ratios is such that $r(i,i)$ must be 1, the diagonal elements of the reactivity ratios portion of the matrix are fixed at [1.0](#).

To edit existing reactivity ratios and concentrations, click in the appropriate matrix cell and type the new value.

Access methods

Menu [Build](#) | [Build Polymers](#) | [Random Copolymer](#) | [Reactivity Ratios](#)

Advanced tab

Initiator: Specifies a repeat unit to bond to the initial tail atom of the polymer chain. Default = [None](#) (the tail atom designator is removed and the existing atom becomes the initiator).

Terminator: Specifies a repeat unit to bond to the final head or tail atom of the polymer chain. (Whether the final atom is a head atom or a tail atom will depend on the *Flip* probability.) Default = [None](#) (the head or tail atom designator is removed and the existing atom becomes the terminator).

Torsion: Specifies the torsion angle in degrees between new repeat units. Default = [180.0](#). Range = [-180.0](#) to [180.0](#).

Note: This option is enabled only if *Random* is unchecked.

Random: When checked, indicates that the torsion angle between the repeat units will be generated at random, in the range [-180.0](#) to [180.0](#).

Report construction statistics: Indicates whether the construction statistics will be saved and displayed. Default = [unchecked](#) (don't display the construction statistics).

Use simple names for constructed polymers: Indicates whether the name of the constructed polymer should be defined with simple rules. Default = [unchecked](#) (use more complicated rules in the naming scheme).

Use random number seed: Indicates whether the construction method should use a defined random number seed. Since there is a randomness in all of the construction techniques, this provides a way to duplicate reliably results from one session to the next. Default = [unchecked](#) (the random seed is not specified).

If the *Use random number seed* checkbox is checked, you can enter a value to seed the random number generator for the polymerization. Range = [0.0](#) to [1.0](#).

Access methods

Menu [Build](#) | [Build Polymers](#) | [Random Copolymer](#) | [Advanced](#)

Branches tab

Attach constructed polymer chain to branch points: Specifies whether or not the polymer under construction is to be used as a branch that will be bonded to the specified branch points in the displayed polymer. Default = [unchecked](#).

Access methods

Menu	<i>Build Build Polymers Random Copolymer Branches</i>
------	---

Repeat Unit dialog

The Repeat Unit dialog contains options for defining a repeat unit from a selected fragment.

Head Atom: Defines the currently selected atom as the head atom in a repeat unit. If a head atom has already been defined, the previous definition is cleared.

Tail Atom: Defines the currently selected atom as the tail atom in a repeat unit. If a tail atom has already been defined, the previous definition is cleared.

Chiral Center: Defines the currently selected atom as a chiral center atom in a repeat unit. If a chiral center atom has already been defined, the previous definition is cleared.

Set: Defines the selected atoms as backbone atoms. If backbone atoms have already been defined, the previous definition is cleared.

Clear: Removes the currently selected atoms from the backbone definition.

Display: Highlights head, tail, chiral and backbone atoms in the repeat unit.

Hide: Hides indicators for head, tail, chiral and backbone atoms.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build Build Polymers Repeat Unit</i>
------	---

Analog Builder

The Analog Builder allows you to create a set of analogous molecules using a Markush-style enumeration.

The Analog Builder takes a number of core structures that have one or more defined connection points and attaches an assortment of substituents (fragments) to each connection point to generate a series of related molecules (analogues). This means that you can quickly produce a suite of similar molecules with a range of properties, from which you can identify the best candidate molecules for further investigation using the bulk processing capabilities of the study table.

Creating an analog library

The Analog Builder takes a number of core structures that have one or more defined connection points and attaches an assortment of fragments to each connection point to generate a series of related analogues using a Markush-style enumeration. This means that you can quickly produce a suite of similar molecules with a range of properties.

To build a library of analogues

1. Before you begin an Analog Builder run, you should ensure that the Fragment Library contains all the fragments that you wish to use. Fragments can be added to the Fragment Library using the Define Fragment dialog.
2. Either import core structures from pre-existing files or construct new molecules using the sketching tools in the Materials Visualizer. To be used as cores in an Analog Builder enumeration, input structures must be molecular. Periodic structures cannot be used in the Analog Builder.
3. Choose *Build / Build Analogs* from the menu bar to display the Build Analogs dialog.
4. Select the [Cores](#) tab.
5. Make a core structure file the active document. Click in the *Core* column of the grid and select the active core structure file from the dropdown tree view of the current project.
6. In the active core structure document, select one or more singly bonded terminal atoms that you wish to define as a connection point group. Select a name for the connection point group from the dropdown list and click the *Set* button. When an atom is included in a connection point group, it will be surrounded with a cage and labeled with the name of the connection point group. If the command fails, see [Set connection point failures](#).

Tip: Once connection point groups have been defined for a structure, they can be saved in the .xsd file. This means that you only need to prepare a core once and not every time you want to use it in an enumeration. In this way, you will build up a 'library' of core structures ready for use in subsequent Analog Builder runs.

7. Repeat steps 5 and 6 to select further core structures and define connection point groups on each, as required.

Tip: Connection points fall into the *Linkage* category of objects within Materials Studio. In common with other such objects, the properties of connection points can be viewed and edited using the Properties Explorer.

8. Select the [Fragments](#) tab. The core structures and the connection point groups defined for them will be shown in the *Connection point groups* list, although the exact format depends on the connection point group option selected.

9. Choose the attachment strategy you wish to use by selecting the appropriate radio button. *All groups separate* will associate each connection point group on every core with a different set of fragments. *Match groups across cores* will associate connection point groups with the same name that are on different cores with the same set of fragments. *Match all groups* will associate all connection point groups on every core with the same set of fragments.
10. To assign a set of fragments to a connection point group or a collection of groups, double-click on an item in the *Connection point groups* list to display the [Choose Fragments dialog](#).
11. Select fragments in the Fragment Library from the *Available fragments* list and click > to add them to the *Selected fragments* list. When the fragment list is complete, click the *OK* button. The *Selected fragments* list on the *Fragments* tab will be updated accordingly.
12. Repeat steps 10 and 11 to assign sets of fragments to further connection point groups, as required.

Tip: While the inputs and parameters for the building process are being defined, the *Total analogs* field is constantly updated with the number of analogs that will be produced by an enumeration using the current settings.

13. Select the [Options](#) tab.
14. Select the format in which you want the analogs to be presented from the *Return results as* dropdown list. Select the naming system to be used to label the analogs from the *Name structures by* dropdown list.
15. If you wish to optimize the bond lengths and angles of each analog structure after it is constructed, check the *Clean structures* checkbox. Enabling this option will significantly increase the time required to create the analogs.
16. If you wish the number and position of hydrogen atoms on each analog structure to be recalculated and updated after it is constructed, check the *Add hydrogens* checkbox.
17. Click the *Build* button. Should the build fail, common failures and their solutions are described in [Build failures](#).

Note: When analogs are built, all the connection points within a connection point group will always receive the same fragment. For example, if the core is methane with two connection points defined within a single connection point group and the selected fragments are F and Cl, the analogs that will be generated are CH_2F_2 and CH_2Cl_2 , but not CH_2ClF .

The build process constructs analogs with every allowed combination of core and fragments. The analogs will be returned in a study table or a trajectory document, or both, according to the formats you specified on the *Options* tab.

The study table infrastructure provides extensive functionality for sorting, filtering, and processing the analog structures in the study table. It also enables you to calculate additional properties. You can use the tools on the Animation toolbar to step through the analogs in the trajectory file.

If analog building fails

The following situations can prevent analogs from being built. A warning message is displayed to alert you to these conditions.

Structure has no linkages

Linkages, or connection points, define how fragments are attached to a core structure. A connection point on a core is an atom that is to be replaced by a fragment.

This usually happens because one of the core structures has no connection points defined. Connection points may be defined on the [Cores tab](#).

Add hydrogens removed connection points

The *Add hydrogens* option will automatically add the correct number of hydrogens to your core and fragment structures before the build begins. However, if there are too many hydrogens on a core or fragment, it may remove a hydrogen with a connection point. Without a connection point, the fragment cannot be attached to form an analog, and the build fails.

This may be solved by unchecking the *Add hydrogens* checkbox on the [Options](#) tab, or by manually removing hydrogens from the problem structure.

Structure contains atoms of undefined element

Usually, this means that some of the fragments are metal templates. These structures contain a template atom that may be replaced with a transition metal atom. To include such a fragment in the build, the element of the template atom must be set:

1. Create a new 3D Atomistic document.
2. Open the Fragment Browser.
3. Select the Metal Template fragment.
4. Select an element for the undefined atom.
5. Use the *Sketch Fragment* tool to place a copy of the fragment into the document.
6. Open the Define Fragment dialog.
7. Select the library to which the new fragment will be added.
8. Enter the name you wish to give to the fragment.
9. Click the *Add* button to add the fragment to the library.

The fragment will now be available for use with the Analog Builder.

Pathname is too long

This can occur if the Workspace folder is too deeply nested. You can change the location of this folder through the Locations tab.

If setting the connection point fails

The following situations can prevent the simulation from initializing or from being saved. A warning message is displayed to alert you to these conditions and allow you to continue the calculation or abort.

A connection point group name must be of the set 'R1' to 'R999'

All connection point groups must have a name between R1 and R999. The [Cores tab](#) of the Build Analogs dialog will automatically offer suitable names. Should you wish to use a specific name, you may type it in. The names must begin with 'R', and must have no leading zeros.

A core structure should have no symmetry

Core structures for use with Analog Builder must have no symmetry defined. Symmetry may be removed using the [Unbuild Crystal](#) command.

Atom selection failures

Analog builder will attempt to add all the atoms in the current selection to the connection point group. Objects other than atoms (for example bonds) will be ignored. Every atom in the selection must meet the following criteria:

- An atom must have only one bond.
- The bond must be a single bond.
- An atom must not be bonded to another connection point atom in this or any other connection point group.

Build Analogs dialog

The Build Analogs dialog allows you to create a set of analogous molecules using a Markush-style enumeration.

The Build Analogs dialog contains the following tabs:

- **Cores:** Allows you to select the core structures to be used and define the connection points on these cores.
- **Fragments:** Allows you to specify the set of fragments to be attached to the core structures at each connection point.
- **Options:** Allows you to choose how the analogs will be named and presented, and offers some additional processing steps.

Build: Generates a new Study Table and optionally a 3D Atomistic Trajectory Document containing the constructed analog structures.

Help: Displays the Help topic for the current tab.

Access methods

Menu	<i>Build Build Analogs</i>
------	------------------------------

Cores tab

The *Cores* tab on the Build Analogs dialog allows you to select the core structures to be used and define the connection points on these cores.

Select the input documents by clicking in the *Core* column of the grid and selecting a 3D Atomistic document from the dropdown tree view of the current project.

Note: To be used as cores in an Analog Builder enumeration, input structures must be molecular. Periodic structures cannot be used in the Analog Builder.

3D Atomistic documents containing the input molecular structures must be part of the current project. Either create the desired structure using the sketching tools in the Materials Visualizer or import the required molecule into the project before selecting cores for the enumeration.

As connection points are defined for the cores, the *Number of CPGs* column of the grid will be updated to show the number of groups of connection points (which is not necessarily the same as the number of connection points) that have been defined for each structure. This column is read-only.

Define connection point group: Select a name for the connection point group from the dropdown list. Connection point group names must have the format Rx, where x is an integer between 1 and 999.

Set: Defines the selected atom(s) in the active 3D Atomistic document as a connection point group. The selected atoms must all be singly bonded terminal atoms. If you attempt to define a connection point group that includes a multiply bonded atom, a warning message will be displayed.

Note: If the specified connection point group has already been already defined, it will be replaced by the current selection. Similarly, if any of the selected atoms have been defined as part of another connection point group, they will be removed from the previous group and added to the new one.

Tip: While the inputs and parameters for the building process are being defined, the *Total analogs* field is constantly updated with the number of analogs that will be produced by an enumeration using the current settings.

Access methods

Menu *Build | Build Analogs | Cores*

Fragments tab

The *Fragments* tab allows you to specify the set of fragments to be attached to the core structures at each connection point. A set of analogs is created by cycling through the available fragments for each connection point group in turn. Existing fragments may be chosen from a fragment library which can be viewed using the Fragment Browser. New fragments can be added to a library using the Define Fragments dialog.

Connection point groups: Lists the connection point groups defined on each core. Double-click on a connection point group to access the [Choose Fragments](#) dialog, which allows you to specify the range of fragments to be attached to that connection point group. The way that the connection point groups are grouped together in the list depends on which combination setting is chosen.

Tip: Connection point groups which have not had fragments associated with them are shown in bold in the list.

Selected fragments: Lists the fragments associated with the selected connection point group. Double-click on any item in the list to access the Choose Fragments dialog, which allows you to add or remove fragments from the list.

Tip: You can rapidly delete items from the *Selected fragments* list by selecting them and pressing the DELETE key.

All groups separate: When selected, indicates that each connection point group on every core will be listed separately and can be associated with a different set of fragments.

Match groups across cores: When selected, indicates that connection point groups with the same name that are on different cores will be listed together and will be associated with the same set of fragments.

Match all groups: When selected, indicates that all connection point groups on every core will be listed together and will be associated with the same set of fragments. This is the simplest setting and is useful for rapidly setting up an enumeration.

Note: If you have selected either the *All groups separate* or *Match groups across cores* options and assigned fragments to the various connection point groups, you should be aware that if you then choose one of the other options, then the fragments will be automatically assigned across the connection point groups to comply with the new setting. However, if you then reselect the original option, the fragment assignment will not change - the original settings will not be reinstated. Therefore, it is advisable that you decide what attachment scheme to use before you start selecting fragments to be attached to the connection point groups.

Tip: While the inputs and parameters for the building process are being defined, the *Total analogs* field is constantly updated with the number of analogs that will be produced by an enumeration using the current settings.

Access methods

Menu *Build | Build Analogs | Fragments*

Choose Fragments dialog

The Choose Fragments dialog allows you to specify a range of fragments to be attached to one or more connection point groups.

Available fragments: Lists the fragments available within the Fragment Library.

Selected fragments: Lists the fragments currently selected for the connection point group(s).

Tip: The *Available fragments* and *Selected fragments* fields support multiple selection. Select multiple items by holding down the CTRL key and clicking on them individually or by holding down SHIFT and selecting the first and last items in a range. Alternatively, click and drag to the right of the tree view items to describe a rectangular selection region or click on a parent item to select all its children.

Use the  and  buttons to add and remove items from the *Selected fragments* list.

OK: Updates the fragment selection with any changes and closes the dialog.

Cancel: Closes the dialog without updating any fragment selection.

Help: Displays the Help topic in a browser.

Access methods

Menu *Build | Build Analogs | Fragments | Connection point groups item or Selected fragments*

Options tab

The *Options* tab allows you to choose how the analogs will be named and presented, and offers some additional processing steps.

Return results as: Select the format in which the enumerated series of analogs will be output from the dropdown list. Available options are:

- **Study table only** - the structures of the analogs will be listed in a study table document (.std)
- **Trajectory only** - the structures of the analogs will be presented in a 3D Atomistic Trajectory document (.xtd)
- **Study table and trajectory** - the structures of the analogs will be returned into both a study table and a 3D Atomistic Trajectory document

Note: The study table and trajectory documents referred to here store the analogs within the project as a hidden .sd file. The .std and .xtd files that may be produced by an Analog Builder run provide different views onto the data contained in the .sd file.

Name structures by: Select the naming system to be used to label the analogs, based on the cores and fragments used to construct them, from the dropdown list. Available options are:

- **Sequential number** - for example, core1_1, core1_2, etc.
- **Core and fragment numbers** - for example, core1_1_1, core1_1_2, ... core1_2_1, core1_2_2, etc.
- **Core and fragment names** - for example, core1_core1_F, core1_core1_Cl, ... core1_core2_F, core1_core2_Cl, etc.

Note: In all cases, the names of the analogs will always begin with the name of the first core structure listed in the grid on the [Cores](#) tab, regardless of the actual core structure that the analog is based on.

Clean structures: When checked, indicates that the bond lengths and angles of each analog structure will be optimized using the Clean functionality in the Materials Visualizer. Enabling this option will significantly increase the time required to create the analogs.

Add hydrogens: When checked, indicates that the number and position of hydrogen atoms on each analog structure will be recalculated and updated accordingly using the Adjust Hydrogen functionality in the Materials Visualizer.

Access methods

Menu	<i>Build / Build Analogs / Options</i>
------	--

Nanostructure Builder

The purpose of nanostructure building in Materials Studio is to construct realistic atomistic models of the structures of relevance to nanotechnology modeling: nanotubes (single- and multi-walled), nanoropes, nanoclusters of various shapes, etc. These are then used as input to the property prediction and simulation modules of Materials Studio.

Tasks in Nanostructure Builder

The Nanostructure Builder provides a range of tools enabling you to build nanotubes and various related structures, including periodic and nonperiodic single-wall nanotubes, nonperiodic multi-wall nanotubes, and nonperiodic single-wall nanotube bundles (nanoropes).

Nanoclusters represent another important class of objects frequently used in nanotechnology. The Nanostructure Builder allows you to generate clusters with different shapes and chemical compositions, including defect clusters with specific concentrations of vacancies.

The following topics provide information on building nanostructures.

Building nonperiodic and periodic single-wall nanotubes

A nanotube is formed by rolling a graphene sheet into a cylinder. Its structure can be uniquely characterized by defining the indices N and M of the chiral vector:

$$\mathbf{Ch} = N \cdot \mathbf{a}_1 + M \cdot \mathbf{a}_2$$

The vectors \mathbf{a}_1 and \mathbf{a}_2 are the principle lattice vectors of a graphene sheet.

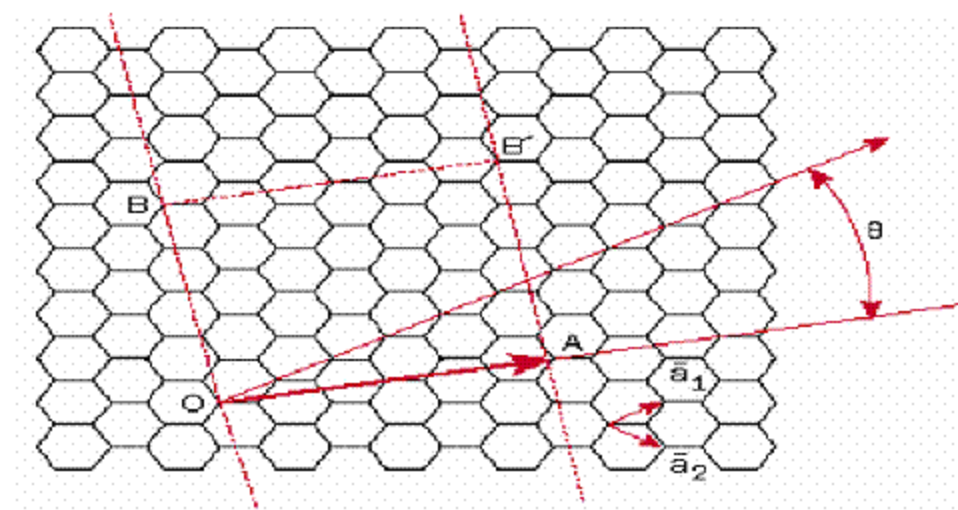


Figure from Rafi-Tabar, 2004.

In the above diagram, the vector \mathbf{OA} is the chiral vector, with a chiral angle of ϑ , for the $(N,M) = (5,2)$ nanotube. The rectangle $\mathbf{OBB'A}$ is the repeat unit for the nanotube. It is this repeat unit that is rolled into a cylinder to form the nanotube.

There are three basic types of nanotubes, named for the shape of the graphene sheet at the end of the tube: "armchair", where $N = M$, "zig-zag", where $M = 0$, and "chiral", with any other combination of N and M .

It is conventional, although not necessary, to define the nanotube so that $N > M$. As a result one can show that the chiral angle is largely controlled by the value of the M index (the chiral angle is zero when

$M=0$). Both indices affect the diameter of the nanotube in equal measure. In view of the $N > M$ condition it is natural to think of the N index as of the parameter which determines the overall size of a nanotube.

Nanotube building process in Materials Studio can be logically separated in two parts: determining the positions of the atoms and establishing the bonding pattern in the nanotube. Atomic positions are defined unambiguously by the chiral vector and by the bond length of the ideal graphene sheet. The actual bonds are created between the nanotube atoms based on the settings of the connectivity options for bond calculations.

To build a single-wall nanotube

1. Choose *Build | Build Nanostructure | Single-Wall Nanotube* from the menu bar to open the [Build Single-Wall Nanotube dialog](#).
2. In the N field, specify a value for the parameter N . This parameter must have a value of at least 1.
3. In the M field, specify a value for the parameter M that controls the chiral angle (twist) of the graphite sheet used to construct the nanotube. This parameter must have a value of 0 or greater.
4. Select the type of nanotube (carbon or boron nitride) you wish to create from the *Type* dropdown list.
5. Optionally, specify the length of the bonds between non-hydrogen atoms in the nanotube using the *Bond length* field.

Note: The bond length value refers to the bonds in the ideal graphene sheet. Since nanotubes are obtained by rolling up the sheet, the resultant distortions generate the structure with the bond lengths deviating from the ideal value. The smaller the N value, the higher is the curvature and the deviations are bigger. This explains why, if the binding energy per atom is calculated using QM or FF tools, one finds that the stability increases monotonically as the diameter increases - this is due to the release of the strain from bond distortion.

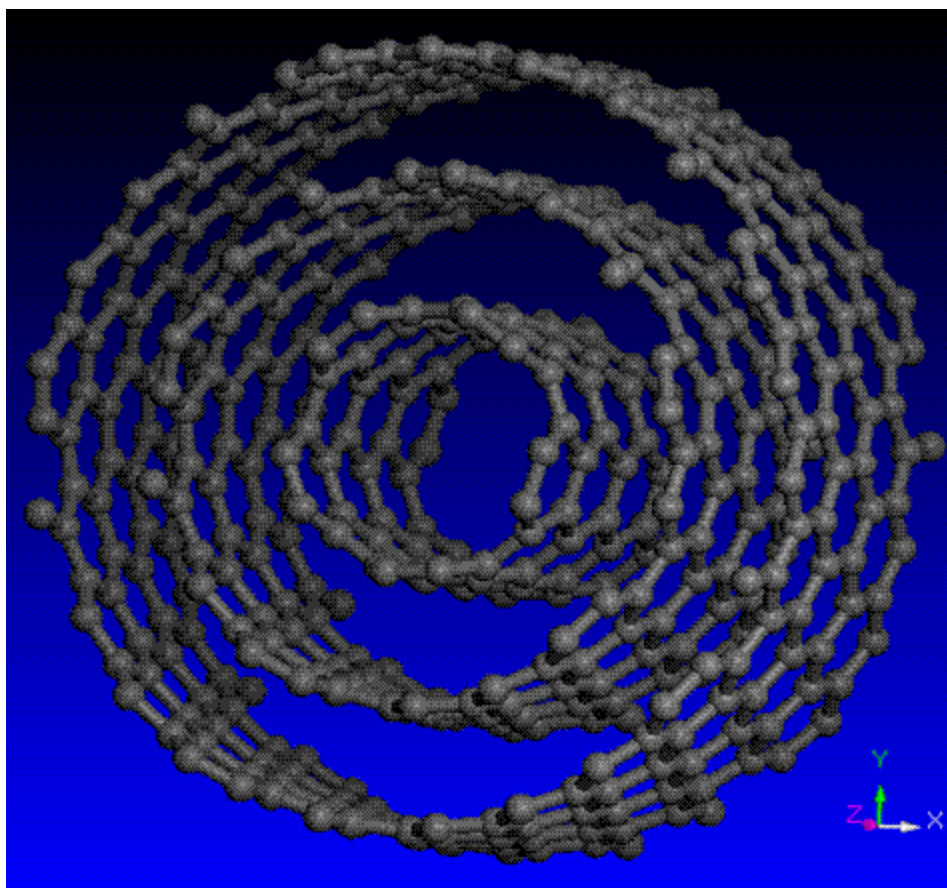
6. If you wish to generate a periodic nanotube, check the *Periodic nanotube* checkbox.
7. Set the overall length of a nonperiodic nanotube by specifying the number of nanotube repeat units to be used to construct it in the *Repeat units* field.
8. For a nonperiodic nanotube, select whether you want dangling bonds to be terminated with hydrogen atoms at one end of the nanotube only, both ends, or neither end by selecting the appropriate option from the *Hydrogen termination* dropdown list.

Tip: The diameter of the nanotube that will be produced using the current parameters is shown at the bottom of the Build Single-Wall Nanotube dialog. In the case of a nonperiodic nanotube, the length of the nanotube that will be produced is also shown. Both these values are updated automatically as the parameters on the dialog are altered.

9. Click the *Build* button to create a new 3D Atomistic document containing a single-wall nanotube built according to the specified parameters.

Building nonperiodic multi-wall nanotubes

A nonperiodic multi-wall nanotube is a concentric set of single-wall nanotubes aligned along the same z -axis. It is commonly assumed that the typical distance between the tubes is similar to the distance between graphene sheets in graphite (Saito et al., 2001). However, a recent experimental study of double-wall carbon nanotubes (DWNT) using high resolution transmission electron microscopy showed that the typical difference in diameters between the inner and outer tubes is in fact around 7.5 Å, or twice the graphite interlayer spacing (Hashimoto et al., 2005). There is no evidence for a correlation between the chiral angles of the constituent nanotubes in concentric DWNTs, which implies that any combination of chiral indices, (N,M) , is possible.



Two methods of building multi-wall nanotubes are available:

- Specify an inner nanotube, then specify the number of further nanotubes to be built around it and the distance between the walls of each nanotube.
- Specify N and M for each nanotube individually.

Note: When building a multi-wall nanotube, the length of the structure is determined by the innermost nanotube - all other nanotubes are truncated to the length of the innermost nanotube. This may result in a structure which is not chemically reasonable, i.e., there may be dangling bonds, incorrect bonding, or non-bonded atoms.

Tip: It is possible in either scenario to specify the length of the bonds between non-hydrogen atoms in the nanotube. The bond length value refers to the bonds in the ideal graphene sheet. Since nanotubes are obtained by rolling up the sheet, the resultant distortions generate the structure with the bond lengths deviating from the ideal value. The smaller the N value, the higher is the curvature and the deviations are bigger.

To build a multi-wall nanotube by specifying an inner nanotube, a number of walls, and a wall separation

1. Choose *Build* | *Build Nanostructure* | *Multi-Wall Nanotube* from the menu bar to open the [Build Multi-Wall Nanotube dialog](#).
2. Choose the *Setup* tab.
3. Select *Sequence* from the *Nanotube definition* dropdown list.
4. Select what type of nanotubes (carbon or boron nitride) you wish to create from the *Type* dropdown list.
5. Optionally, specify the length of the bonds between non-hydrogen atoms in the nanotubes using the *Bond length* field.
6. To rotate the nanotubes relative to one another in order to minimize the strain energy, check the *Minimize energy* checkbox.
7. Specify the quality of the energy minimization using the slider.
8. Choose the *Sequence* tab.
9. In the *N* field, specify a value for the parameter N. This parameter must have a value of at least 1.
10. In the *M* field, specify a value for the parameter M that controls the chiral angle (twist) of the graphite sheet used to construct the innermost nanotube. This parameter must have a value of 0 or greater.





Tip: The diameter and length of the initial nanotube that will be produced using the parameters currently specified on both the *Setup* and *Sequence* tabs is shown in the center of the *Sequence* tab. Both these values are updated automatically as the parameters on both tabs are altered.

11. Set the overall length of the innermost nanotube by specifying the number of nanotube repeat units to be used to construct it in the *Repeat units* field.
12. Specify the number of subsequent nanotubes enclosing the innermost nanotube in the *Number of walls* field.
13. Specify the desired distance between the walls of the nanotubes in the *Wall separation* field.

Note: The separation between nanotubes is determined as a difference between their radii. Since the diameter is completely determined by the set of discrete *N* and *M* values, it follows that only certain values of intertube separations can be realized. Materials Studio attempts to create the nanotubes with separations that are as close to the requested value as possible; the actual separation will be higher than the requested value.

14. Click the *Build* button to create a new 3D Atomistic document containing a multi-wall nanotube built according to the specified parameters.

To build a multi-wall nanotube by specifying each nanotube individually

1. Choose *Build* | *Build Nanostructure* | *Multi-Wall Nanotube* from the menu bar to open the [Build Multi-Wall Nanotube dialog](#).
2. Choose the *Setup* tab.
3. Select *Individual* from the *Nanotube definition* dropdown list.
4. Select what type of nanotubes (carbon or boron nitride) you wish to create from the *Type* dropdown list.
5. Optionally, specify the length of the bonds between non-hydrogen atoms in the nanotubes using the *Bond length* field.
6. To rotate the nanotubes relative to one another in order to minimize the strain energy, check the *Minimize energy* checkbox.
7. Specify the quality of the energy minimization using the slider.
8. Choose the *Individual* tab.
9. The grid contains a single row with default values for *N* (the parameter that controls the overall size of the nanotube) and *M* (the parameter that controls the chiral angle, or twist, of the graphite sheet used to construct the nanotube). Click on the appropriate cells and enter the desired values for these parameters. The *Diameter* column in the grid shows the diameter of the nanotube that will be created using the settings in the corresponding row.
10. To add a new row, representing another nanotube, to the grid, click the *Add* button . To delete a highlighted row in the grid, click the *Delete* button . To select the row above or below the currently highlighted row in the grid, click the *Move up*  or *Move down*  buttons, respectively.
11. Set the overall length of the nanotubes by specifying the number of nanotube repeat units to be used to construct them in the *Inner repeat units* field.

Note: It is not required to order the nanotubes in ascending order of the nanotube diameter.

Tip: The length of the nanotubes that will be produced using the parameters currently specified on both the *Setup* and *Individual* tabs is shown at the bottom of the *Individual* tab. This value is updated automatically as the parameters on both tabs are altered.

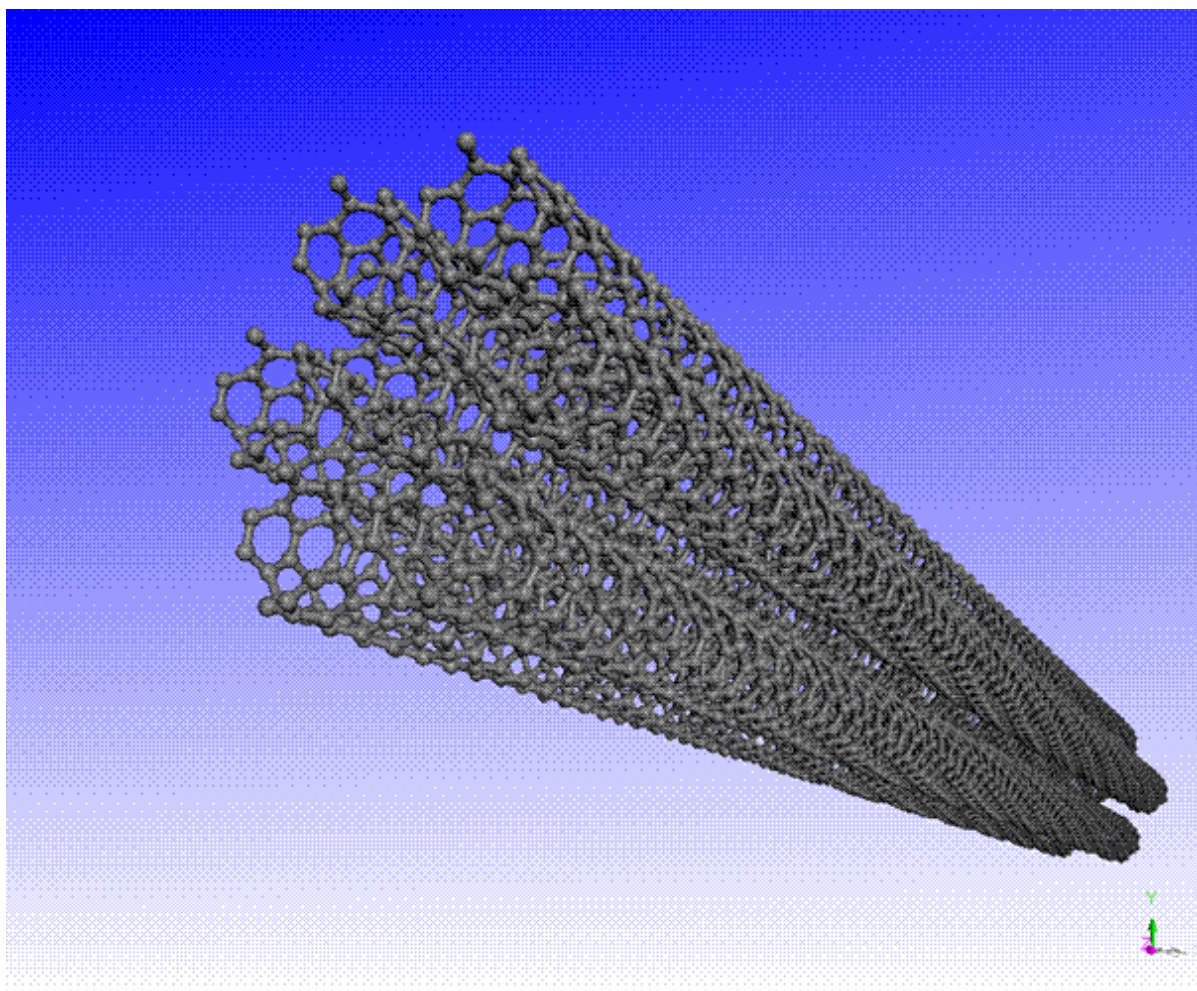
12. Click the *Build* button to create a new 3D Atomistic document containing a multi-wall nanotube built according to the specified parameters.

Note: Depending on the parameters you specify for the multi-wall nanotube, you may encounter an error message.

The error message *Tubes too close* indicates that the nanotube diameters specified bring the walls too close together for the energy minimization to produce an accurate result. The nanotubes may not be oriented in a minimum energy arrangement. You can either proceed or abort construction of the multi-wall nanotube.

Building nonperiodic single-wall nanotube bundles (nanoropes)

A nonperiodic single-wall nanotube bundle, or nanorope, is a collection of identical single-wall nanotubes having parallel axes arranged in a close-packed geometry.



To build a nanorope

1. Choose *Build | Build Nanostructure | Nanorope* from the menu bar to open the [Build Nanorope dialog](#).
2. In the *N* field, specify a value for the parameter *N*. This parameter must have a value of at least 1.
3. In the *M* field, specify a value for the parameter that controls the chiral angle (twist) of the graphite sheet used to construct each nanotube comprising the nanorope. This parameter must have a value of 0 or greater.
4. From the *Type* dropdown list, select what type of nanotubes (carbon or boron nitride) the nanorope will be composed of.
5. Optionally, in the *Bond length* field, specify the length of the bonds between non-hydrogen atoms in the nanotubes comprising the nanorope.

Note: The bond length value refers to the bonds in the ideal graphene sheet. Since nanotubes are obtained by rolling up the sheet, the resultant distortions generate the structure with the bond lengths deviating from the ideal value. The smaller the *N* value, the higher is the curvature and the deviations are bigger.

6. Set the overall length of the nanotubes comprising the nanorope by specifying the number of nanotube repeat units to be used to construct them in the *Repeat units* field.

7. Select whether you want dangling bonds to be terminated with hydrogen atoms at one end only of each nanotube comprising the nanorope, both ends, or neither end by selecting the appropriate option from the *Hydrogen termination* dropdown list.
8. Set the geometric arrangement of nanotubes in the nanorope by specifying a value for the *Number of nanotubes*.
9. Optionally, specify the distance between the walls of the nanotubes comprising the nanorope in the *Nanotube separation* field.

Tip: The diameter and length of the nanotubes that will be used to construct the nanorope are shown at the bottom of the Build Nanorope dialog. Both these values are updated automatically as the parameters on the dialog are altered.

10. Click the *Build* button to create a new 3D Atomistic document containing a nanorope composed of nonperiodic single-wall nanotubes built according to the specified parameters.


Building nanoclusters


A nanocluster is a collection of atoms with a specific geometric shape. These shapes include simple (rectangular) box, sphere, cylinder, cone, frustum, tetrahedron, and pyramid. The clusters are constructed by cutting the specified shape out of an infinite crystal. This may sometimes result in the inclusion of seemingly extraneous atoms in the cluster, depending on how the shape boundaries correspond to the crystal structure of the material.

To build a nanocluster

1. Make the periodic structure document you wish to use as a basis for the nanocluster the active document.
2. Choose *Build | Build Nanostructure | Nanocluster* from the menu bar to open the Build Nanocluster dialog.
3. Choose the [Shape](#) tab.
4. Select the shape of the nanocluster you wish to construct from the *Shape* dropdown list.
5. Specify additional dimensions for the nanocluster in the appropriate fields. The fields that are present on the dialog depend on the selected nanocluster shape.
6. Specify the proportion of vacancies in the *Vacancy concentration* text box.

Note: The *Vacancy concentration* is approximate and is not reproducible.

7. If you wish to cap bonds that are cleaved at the shape boundaries, check the *Cap broken bonds* checkbox and select the required atom type from the dropdown list. If you wish to use an element that is not present on the list, click the  button to display the Periodic Table dialog and select the desired element.
8. Select a scheme for handling atoms at the shape boundaries from the *Clipping rule* dropdown list.
9. Choose the [Substitution](#) tab.
10. If required, select the *Distribution* for the concentration gradient and the *Functional form* for the distribution from the appropriate dropdown lists.
11. If you are building a nanocluster with a concentration gradient for substituted species, specify the initial and final values of the gradient in the *From* and *To* fields.
12. Specify the substitute species in the *Substitution* column of the grid by clicking in the appropriate row and selecting the required atom type from the dropdown list. If you wish to use an element that

is not present on the list, click the  button to display the Periodic Table dialog and select the desired element.

13. Click the *Build* button to construct a nanocluster in the active document according to the specified parameters.

Dialogs in Nanostructure Builder

Materials Studio's nanostructure building functionality provides a range of dialogs that allow you to create models of various types of objects, including periodic and nonperiodic single-wall nanotubes, nonperiodic multi-wall nanotubes, nonperiodic single-wall nanotube bundles (nanoropes), and nanoclusters with specific shapes.

The following sections describe each of these dialogs in more detail.

Build Single-Wall Nanotube dialog

The Build Single-Wall Nanotube dialog allows you to control all of the variables used to construct single-wall nanotubes.

N: Specify the [integer parameter, N](#), that controls the overall size of the nanotube. The minimum value for this parameter is 1. Default = 6.

M: Specify the [integer parameter, M](#), that controls the chiral angle, or twist, of the graphite sheet used to construct the nanotube. The minimum value for this parameter is 0. Default = 6.

Type: Select the type of nanotube you wish to create. Available options are:

- Carbon
- Boron Nitride

Bond length: Specify the length, in Å, of the bonds between atoms in the [ideal graphene sheet](#) that is used to generate the nanotube. This value is not used for bonds to any terminal hydrogen atoms. Default = 1.42 Å, which corresponds to carbon nanotubes (Ciraci et al., 2004).

Note: The default bond length for boron-nitrogen bonds is 1.44 Å.

Periodic nanotube: When checked, indicates that a periodic lattice, comprising one repeat unit of a single-wall nanotube, will be built. Default = [checked](#).

A standard hexagonal packing arrangement of nanotubes is used, in which the $a = b$ dimension is determined by the nanotube diameter plus half the graphite interlayer separation. This means that the nanotubes are at approximately the same separation as the layers in graphite, 3.347 Å (Zhang et al., 2004). When nanotubes with very large diameters (and therefore very small curvatures) are used, the graphite structure is the limiting case.

Tip: You can change the cell parameters using the Lattice Parameters dialog. It is important, however, to keep the Cartesian coordinates fixed by unchecking the *Keep fractional coordinates fixed during changes to the lattice* check box on the Advanced tab.

Repeat units: Specify the number of repeat units for the nanotube. This value determines the overall length of the nanotube. Default = 2.

Note: This option is disabled when the *Periodic nanotube* checkbox is checked.

Hydrogen termination: Select whether you want dangling bonds at the ends of a nonperiodic nanotube capped with hydrogen atoms. Available options are:

- **None** - no terminating hydrogen atoms are added
- **Both ends** - terminating hydrogen atoms are added to both ends of the nanotube
- **One end** - terminating hydrogen atoms are added to only one end of the nanotube

Note: This option is disabled when the *Periodic nanotube* checkbox is checked.

Diameter: Displays the diameter, in Å, of the nanotube currently specified. The value is automatically updated if any of the parameters are changed.

Length: Displays the overall length, in Å, of the nanotube currently specified. The value is automatically updated if any of the parameters are changed.

Note: The *Length* value shown does not include any terminal hydrogen atoms and it is not displayed at all when the *Periodic nanotube* checkbox is checked.

Build: Creates a new 3D Atomistic document containing a single-wall nanotube built according to the specified parameters.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build Build Nanostructure Single-Wall Nanotube</i>
------	---

Build Multi-Wall Nanotube dialog

The Build Multi-Wall Nanotube dialog allows you to control all of the variables used to construct a nonperiodic multi-wall nanotube.

The Build Multi-Wall Nanotube dialog contains the following tabs:

- **Setup:** Allows you to specify whether you wish to create a sequence or individual nanotube, the composition of the nanotube, the length of the non-hydrogen bonds, and whether the strain energy of the structure should be minimized.
- **Sequence:** Allows you to specify how the multi-wall nanotube will be constructed and its dimensions.
- **Individual:** Allows you to construct a multi-wall nanotube by specifying *N* and *M* for each nanotube.

Build: Creates a new 3D Atomistic document containing a multi-wall nanotube built according to the specified parameters.

Help: Displays the Help topic for the current tab.

Access methods

Menu	<i>Build Build Nanostructure Multi-Wall Nanotube</i>
------	--

Setup tab

The *Setup* tab allows you to specify how you would like to define the nanotube (by using a sequence mechanism or by specifying each individual nanotube). It also allows you to specify the composition of the nanotube, the length of the non-hydrogen bonds, and whether the strain energy of the structure should be minimized. More detailed explanation of the approach is given in the [Building nonperiodic multi-wall nanotubes](#) topic.

Nanotube definition: Select how the multi-wall nanotube will be defined. Available options are:

- **Sequence**
- **Individual**

The [Sequence](#) option uses a predefined algorithm to generate a multi-wall nanotube based on the properties of the innermost single-wall nanotube and on the requested number of walls. The [Individual](#) option allows you to specify the chiral vectors of each wall in the multi-wall nanotube.

Type: Select the type of nanotubes you wish to create. Available options for the chemical composition are:

- [Carbon](#)
- [Boron Nitride](#)

Bond length: Specify the length, in Å, of the bonds between atoms in the [ideal graphene sheet](#) that is used to generate the nanotubes. This value is not used for bonds to any terminal hydrogen atoms. Default = [1.42 Å](#) which corresponds to carbon nanotubes (Ciraci et al., 2004).

Note: The default bond length value for boron-nitrogen bonds is [1.44 Å](#).

Minimize Energy: When checked, indicates that the nanotubes will be rotated relative to one another in order to minimize the strain energy. Specify the quality of the energy minimization using the slider.

The potential that is used to determine the minimum energy for carbon is a Lennard-Jones-type potential (Zhang et al., 2004):

$$\text{energy} = -A/r^6 + B/r^{12}$$

with $A = 77.15 \text{ eV Å}^6$ and $B = 1.14548 \times 10^5 \text{ eV Å}^{12}$.

The same potential is used for boron nitride, except that the B term in the above equation is doubled for like species (i.e., B-B or N-N bonds). This prevents like species overlap.

Access methods

Menu [Build](#) | [Build Nanostructure](#) | [Multi-Wall Nanotube](#) | [Setup](#)

Sequence tab

The *Sequence* tab allows you to specify how the multi-wall nanotube will be constructed and its dimensions.

Note: The *Sequence* tab is accessible only if the *Nanotube definition* is set to [Sequence](#) on the [Setup](#) tab.

Note: Currently, multi-wall nanotubes are constructed without hydrogen termination.

Inner nanotube: These options allow you to construct a multi-wall nanotube from an inner nanotube and a minimum wall separation. Each successive outer wall is selected by keeping M constant and increasing N until the distance between the walls exceeds the specified *Wall separation*.

- **N:** Specify the [integer parameter, N](#), that controls the overall size of the innermost nanotube. The minimum value for this parameter is [1](#). Default = [6](#).
- **M:** Specify the [integer parameter, M](#), that controls the chiral angle, or twist, of the graphite sheet used to construct the innermost nanotube. The minimum value for this parameter is [0](#). Default = [6](#).
- **Repeat units:** Specify the number of repeat units for the innermost nanotube. This value determines the overall length of the nanotube. Default = [2](#).
- **Length:** Displays the overall length, in Å, of the innermost nanotube currently specified. The value is automatically updated if any of the parameters are changed.
- **Diameter:** Displays the diameter, in Å, of the innermost nanotube currently specified. The value is automatically updated if any of the parameters are changed.

Tip: All other (outer) nanotubes are trimmed to the length of the innermost tube. This may cause some outer tubes to have a final length that is a non-integer multiple of their repeat unit length.

Overall: These options allow you to specify the number of walls and their separation.

- **Number of walls:** Specify the number of walls for the multi-wall nanotube. Default = 2.
- **Wall separation:** Specify the minimum distance, in Å, between the successive nanotubes. Default = 3.347 Å, the interlayer distance of graphite (Zhang et al., 2004).

Note: The separation between nanotubes is determined as a difference between their radii. Since the diameter is completely determined by the set of discrete N and M values, it follows that only certain values of intertube separations can be realized. Materials Studio attempts to create the nanotubes with separations that are as close to the requested value as possible; the actual separation will be higher than the requested value.

Access methods

Menu | Build | Build Nanostructure | Multi-Wall Nanotube | Sequence

Individual tab

The *Individual* tab allows you to construct a multi-wall nanotube by specifying N and M for each nanotube.

Note: The *Individual* tab is accessible only if the *Nanotube definition* is set to **Individual** on the [Setup](#) tab.

Note: Currently, multi-wall nanotubes are constructed without hydrogen termination.

N: Specify the [integer parameter N](#) which controls the overall size of the nanotube in the selected row. The minimum value for this parameter is 1. Default = 6.

M: Specify the [integer parameter M](#) which controls the chiral angle, or twist, of the graphite sheet used to construct the nanotube in the selected row. The minimum value for this parameter is 0. Default = 6.

Diameter: Displays the diameter, in Å, of the nanotube in the selected row. The value is automatically updated if any of the parameters are changed.



Add: Adds a new row below the currently selected row in the grid, allowing parameters to be entered for a new nanotube. The initial values of N and M for the new nanotube are selected to maintain the specified *Wall separation*.



Delete: Removes the currently selected row in the grid.



Move up: Moves the currently selected row one row upward in the grid.



Move down: Moves the currently selected row one row downward in the grid.

Inner repeat units: Specify the number of repeat units for the innermost nanotube. This value determines the overall length of the nanotube. Default = 2.

Length: Displays the overall length, in Å, of the innermost nanotube currently specified. The value is automatically updated if any parameters are changed.

Tip: All other (outer) nanotubes are trimmed to the length of the innermost tube. This may cause some outer tubes to have a final length that is a non-integer multiple of their repeat unit length.

Note: It is not necessary to order the nanotubes in ascending order of diameter.

Depending on the parameters specified, you may encounter the following error message: [Tubes too close](#). The energy minimization algorithm assumes van der Waals forces. If the nanotube diameters are too close, this assumption may not be valid and the nanotubes may not be oriented in a minimum energy arrangement. You can either proceed or abort construction of the multi-wall nanotube.

Access methods

Menu | *Build* | *Build Nanostructure* | *Multi-Wall Nanotube* | *Individual*

Build Nanorope dialog

The Build Nanorope dialog allows you to control all of the variables used to construct a nonperiodic single-wall nanotube bundle, or nanorope. This is a collection of identical single-wall nanotubes having parallel axes and origins which are arranged in a close-packed geometry. The nanotubes will not be rotated relative to one another in order to minimize the strain energy.

N: Specify the [integer parameter, N](#), that controls the overall size of each nanotube in the nanorope. The minimum value for this parameter is 1. Default = 6.

M: Specify the [integer parameter, M](#), that controls the chiral angle, or twist, of the graphite sheet used to construct each nanotube in the nanorope. The minimum value for this parameter is 0. Default = 6.

Type: Select the type of nanotubes you wish to create the nanorope from. Available options are:

- [Carbon](#)
- [Boron Nitride](#)

Bond length: Specify the length, in Å, of the bonds between atoms in the [ideal graphene sheet](#) that is used to generate the nanotubes comprising the nanorope. This value is not used for bonds to any terminal hydrogen atoms. Default = 1.42 Å, which corresponds to carbon nanotubes (Ciraci et al., 2004).

Note: The default bond length for boron-nitrogen bonds is 1.44 Å.

Repeat units: Specify the number of repeat units for the nanotubes comprising the nanorope. This value determines the overall length of the nanotubes. Default = 2.

Hydrogen termination: Select whether you want dangling bonds at the ends of the nanotubes comprising the nanorope capped with hydrogen atoms. Available options are:

- [None](#) - no terminating hydrogen atoms are added
- [Both ends](#) - terminating hydrogen atoms are added to both ends of each nanotube in the nanorope
- [One end](#) - terminating hydrogen atoms are added to only one end of each nanotube in the nanorope

Number of nanotubes: Specify the number of nanotubes in the nanorope. This parameter determines the geometric arrangement of the nanotubes. Default = 6.

Nanotube separation: Specify the minimum distance, in Å, between the walls of each nanotube in the nanorope. Default = 3.347 Å, the interlayer distance of graphite (Zhang et al., 2004).

Diameter: Displays the diameter, in Å, of the currently specified nanotubes in the nanorope. The value is automatically updated if any of the parameters are changed.

Length: Displays the overall length, in Å, of the currently specified nanotubes in the nanorope. The value is automatically updated if any of the parameters are changed.

Note: The *Length* value shown does not include any terminal hydrogen atoms.

Build: Creates a new 3D Atomistic document containing a nanorope built according to the specified parameters.

Help: Displays the Help topic in a browser.

Access methods

Menu *Build | Build Nanostructure | Nanorope*

Build Nanocluster dialog

The Build Nanocluster dialog allows you to control all of the variables used to construct nanoclusters (i.e., collections of atoms) of various shapes.

The Build Nanocluster dialog contains the following tabs:

- [Shape](#): Allows you to select the shape and size of the nanocluster that you wish to construct, along with options that determine how atoms at the shape boundaries will be handled.
- [Substitution](#): Allows you to specify the options that control the concentration gradient for a substituted species in a nanocluster.

Build: Constructs a nanocluster in the active document according to the specified parameters.

Note: The *Build* button is enabled only if the active document contains a periodic structure.

Help: Displays the Help topic for the current tab.

Access methods

Menu *Build | Build Nanostructure | Nanocluster*

Shape tab

The *Shape* tab allows you to select the shape and size of the nanocluster that you wish to construct, along with options that determine how atoms at the shape boundaries will be handled. Available cluster shapes include simple (rectangular) box, sphere, cylinder, cone, frustum, tetrahedron, and pyramid. The clusters are constructed by cutting the specified shape out of an infinite crystal. This may sometimes result in the inclusion of seemingly extraneous atoms in the cluster, depending on how the shape boundaries correspond to the crystal structure of the material.

The periodic structure document you wish to use as a basis for the nanocluster must be made the active document before a nanocluster can be constructed.

Shape: Select the desired cluster shape from the dropdown list. Available options are:


- [Simple box](#)
- [Sphere](#)
- [Cylinder](#)
- [Cone](#)
- [Frustum](#)
- [Tetrahedron](#)
- [Square pyramid](#)

Definitions of the shapes and information about the additional parameters that must be specified in each case are given in the sections below.

Vacancy concentration: Specify the proportion of vacancies in the cluster. 0 = no vacancies, 1 = no atoms. Default = 0.0.

As each atom is added to the cluster, a random number is generated. If the random number does not exceed the specified vacancy concentration, the atom is deleted.

Note: The *Vacancy concentration* is approximate and is not reproducible.

Cap broken bonds with: When checked, indicates that bonds which are cleaved at the shape boundaries will be capped with atoms of the specified type. Select an atom type from the dropdown list, which contains the most commonly used elements. Alternatively, click the  button to display the Periodic Table dialog, from which you can select elements not provided in the dropdown list.

Clipping rule: Select a scheme for handling atoms at the shape boundaries. Available options are:

- **Default** - atoms are included in the cluster if the center of geometry of the connected fragment to which they belong lies within the shape's boundary. Adjustments are made for atoms belonging to infinitely connected networks.
- **Atomic** - all atoms that fall outside the boundaries of the specified shape will be removed.

Note: A clipping rule is only valid for molecular materials. If the active document does not contain a material that is composed of discrete molecules, the *Clipping rule* list box will appear blank. When this is the case, you will still be able to select one of the available options from the dropdown list, but this selection will be ignored when the Nanostructure Builder constructs the nanocluster.

Simple box

This is a rectangular box, or cuboid, centered on the origin and extending in the x, y, and z directions.

X, Y, and Z directions: Specify the measurements of the box in the x, y, and z directions in Å. Default = 5.0 Å.

Sphere

This is a sphere centered on the origin.

Radius: Specify the radius of the sphere in Å. Default = 5.0 Å.

Cylinder

This is a cylinder, centered on the origin and extending in the specified direction.

Radius: Specify the radius of the cylinder in Å. Default = 5.0 Å.

Height: Specify the distance between the ends of the cylinder in Å. Default = 15.0 Å.

Base plane (h k l): Specify the base plane for the cylinder in terms of Miller indices. Default = (0 0 1).

Cone

This is a cone, centered on the origin and extending in the specified direction.

Radius: Specify the radius of the base of the cone in Å. Default = 10.0 Å.

Height: Specify the distance between the apex and the base of the cone in Å. Default = 15.0 Å.

Base plane (h k l): Specify the base plane for the cone in terms of Miller indices. Default = (0 0 1).

Frustum

This is a cylindrical frustum, or truncated cone, centered on the origin and extending in the specified direction. The cone is truncated parallel to its base.

Top radius: Specify the radius of the top surface of the frustum in Å. Default = 10.0 Å.

Base radius: Specify the radius of the base of the frustum in Å. Default = 15.0 Å.

Height: Specify the distance between the top surface and the base of the frustum in Å. Default = 10.0 Å.

Base plane (h k l): Specify the base plane for the frustum in terms of Miller indices. Default = (0 0 1).

Tetrahedron

This is a tetrahedron (a pyramid with a triangular base), centered on the origin and extending in the specified direction.

Height: Specify the vertical distance between the base and the apex of the tetrahedron in Å. Default = 10.0 Å.

Base edge: Specify the length of one edge of the base of the tetrahedron in Å. Default = 10.0 Å.

Base plane (h k l): Specify the base plane for the tetrahedron in terms of Miller indices. Default = (0 0 1).

Square pyramid

This is a pyramid with a square base, centered on the origin and extending in the specified direction.

Height: Specify the vertical distance between the base and the apex of the pyramid in Å. Default = 10.0 Å.

Base edge: Specify the length of one edge of the base of the pyramid in Å. Default = 10.0 Å.

Base plane (h k l): Specify the base plane for the pyramid in terms of Miller indices. Default = (0 0 1).

Access methods

Menu	Build Build Nanostructure Nanocluster Shape
------	---

Substitution tab

The *Substitution* tab allows you to specify the options that control the concentration gradient for a substituted species in a nanocluster.

The periodic structure document you wish to use as a basis for the nanocluster must be made the active document before a nanocluster can be constructed.

The cluster shape that has been selected on the [Shape](#) tab is displayed at the top of the *Substitution* tab.

Distribution: Select the distribution of the concentration gradient throughout the nanocluster from the dropdown list. The available options will include some or all of the following, depending on the cluster shape selected on the *Shape* tab:

- **No Substitution** - no atom substitution will take place
- **Axial** - the concentration of the substituted species will increase from a particular plane. This option is only available for shapes with a linear aspect, for example, simple box, cylinder, tetrahedron.
- **Radial** - the concentration of the substituted species will increase equally in all directions from a central point. This option is only available for shapes with a circular aspect, for example sphere, cylinder, cone.


Functional form: Select the functional form of the concentration gradient distribution. Available options are:

- **Linear**
- **Quadratic**
- **Exponential**

From: Specify the initial percentage value of the concentration gradient.

To: Specify the final percentage value of the concentration gradient.

The elements present in the basic structural unit in the active document are listed in the *Species* column of the grid. Enter the elements to be substituted for the existing species in the *Substitution* column by clicking on the appropriate row in the grid and selecting the required element from the dropdown list.

Alternatively, click the  button to display the Periodic Table dialog, from which you can select elements not provided in the dropdown list. The default entries in the *Substitution* column are the same as those in the *Species* column.

Access methods

Menu	<i>Build / Build Nanostructure / Nanocluster / Substitution</i>
------	---

Transport Device Builder

The Transport Device Builder allows you to build transport devices which can later be used in electron transport calculations. A valid transport device consists of two or more Electrode objects that are connected to a central device region. The Transport Device Builder creates both nonperiodic transport devices and periodic devices. The periodic devices are constructed as 2D periodic documents and are restricted to two electrodes.

Note: Only the DMol³ and DFTB+ modules' Electron Transport tasks can work with documents containing electrodes or transport devices. In addition, some Materials Visualizer functions are not enabled for these objects.

Building a transport device

A transport device is a structure which consists of a central device region connected to two or more semi-infinite electrodes. The device structure can either be periodic or nonperiodic. If geometry optimization is required you should optimize the source structures before constructing the electrodes.

A periodic device structure is defined in a 2D periodic document where only two electrodes are allowed. The two electrodes will be oriented perpendicular to the 2D periodic plane making the structure fully periodic.

For a nonperiodic device structure the number of electrodes are limited to ten and the periodic direction of the electrodes must be parallel to the X, Y, or Z axis.

To build a periodic transport device

1. Import or create a source crystal from which the electrode should be cleaved.
2. Choose *Build | Build Transport Device | Cleave Electrode* from the menu bar to open the [Cleave Electrode](#) dialog.
3. Select the *Cleave plane (h k l)* Miller indices.
4. Specify *Cleave position* and *Tip width*.
5. Click the [Cleave](#) button.
6. Choose *Build | Build Transport Device | Transport device* from the menu bar to open the [Transport Device](#) dialog.
7. Choose a document to be inserted in between of two electrodes:
 - If the document contains a molecule, use *Direction* to specify the alignment direction and *Min. super cell size* to set the minimum size for the electrode supercell.
 - If the document contains a crystal, set the *Thickness* for the crystal layer.
8. Click the *Build* button.

Note: For insertion of multiple layers the [Build Layers tool](#) can be used.

To build a nonperiodic transport device

1. Create a central device region with electrode stumps.
2. Choose *Build | Build Transport Device | Build Electrode* from the menu bar to open the [Build Electrode](#) dialog.
3. Select the *Electrode direction* along which the electrode should be oriented
4. Optionally, select a region that should be used for the electrode definition.
 - With a region selected, check *Add selection to tip* to specify if the selected region should be added to the tip region of the electrode.
5. Click the Build button.

Note: If the *Build* button is disabled the builder was unable to determine the periodicity of the electrode.

Note: Only the DMol³ and DFTB+ modules' Electron Transport tasks can work with documents containing electrodes or transport devices. In addition, some Materials Visualizer functions are not enabled for these objects.

To remove an electrode

1. Open the document containing the electrode object(s) which you want to remove.
2. Select *Build | Build Transport Device | Unbuild Electrode* from the menu bar. Alternatively, right-click on the document and select *Unbuild Electrode*. The electrode definition is removed while the structure remains.

Dialogs in Transport Device Builder

Materials Studio's transport device building functionality provides a range of dialogs that allow you to create a transport device i.e. a document containing electrodes which can be used further in electron transport calculations.

The following sections describe each of these dialogs in more detail:

- [Build Electrode dialog](#): Allows you to create an electrode in an existing document.
- [Cleave Electrode dialog](#): Allows you to create a new 2D periodic document containing an electrode or two electrodes.
- [Transport Device dialog](#): Allows you to add a device section to a 2D periodic document containing two electrodes.

Access methods

Menu	<i>Build Build Transport Device</i>
------	---------------------------------------

Build Electrode dialog

The Build Electrode dialog allows you to create an electrode in an existing document. In order to define an electrode, starting from the end or selected part of the structure, at least two periodic cells of atoms along the electrode direction must be present.

The evaluation of the periodic cell starts from the outermost part of the structure and looks for a periodic repeat. To enable a successful prediction of the periodic cell, ensure that there are no surface reconstructions or extra surface atoms in this part of the structure as this would destroy the periodicity of the structure.

Electrode direction: Direction along which the periodic part of the electrode extends. Available options are:

- **+X** - Electrode extends along the positive direction of the x-axis (nonperiodic)
- **-X** - Electrode extends along the negative direction of the x-axis (nonperiodic)
- **+Y** - Electrode extends along the positive direction of the y-axis (nonperiodic)
- **-Y** - Electrode extends along the negative direction of the y-axis (nonperiodic)
- **+Z** - Electrode extends along the positive direction of the z-axis (nonperiodic)
- **-Z** - Electrode extends along the negative direction of the z-axis (nonperiodic)
- **Forward** - Electrode extends along the positive direction of the nonperiodic axis (2D periodic)
- **Backward** - Electrode extends along the negative direction of the nonperiodic axis (2D periodic)

Add selection to tip: When **checked**, any selected atoms that are not already part of the Electrode Wire section will be added to the Electrode Tip section of the newly created Electrode.

Tip: Existing electrodes cannot be modified using the *Build Electrode* dialog.

Build: Builds a new electrode. The new electrode will be positioned to the far end of the structure or selected part of the structure in the electrode direction.

Tip: If the *Build* button is disabled, this is because there is no periodicity in the chosen direction.

Tip: Electrodes can only be created in nonperiodic and 2D periodic documents and cannot overlap with existing electrodes. Electrodes cannot be positioned inside a structure. Any existing hierarchy which overlaps with the electrode must be removed before the electrode can be created.

Note: Only the DMol³ and DFTB+ modules' Electron Transport tasks can work with documents containing electrodes or transport devices. In addition, some Materials Visualizer functions are not enabled for these objects.

Help: Displays the Help topic in a browser.

Access methods

Menu | *Build* | *Build Transport Device* | *Build Electrode*

Cleave Electrode dialog

The Cleave Electrode dialog allows you to create a new 2D periodic document containing an electrode, or two electrodes facing in opposite directions, which can later be used to create a transport device document.

Cleave plane (h k l): Specify the cleave plane in terms of the Miller indices h, k, and l.

Construct orthogonal basis: If checked requests creation of the 2D periodic lattice with orthogonal basis vectors as required by the DFTB+ module. Otherwise creates minimal possible in plane basis which could be useful for creating nonperiodic devices by [making a nonperiodic superstructure](#). Default = **checked**.

Cleave position: Specify the distance from the origin to the top of the cleave as a fraction of the d-spacing of the Miller family of the cleave plane. The corresponding distance in Å is also displayed. Default = **0.0**.

Tip width: Specify the thickness of the electrode tip as a fraction of the d-spacing. Default = **1.0**.

Build two electrodes: If checked two electrodes facing outwards of the interface will be constructed. Otherwise only one electrode is created. Default = [checked](#).

Cleave: Performs a cleave operation on the current active 3D model using the parameters specified.

Tip: If the *Cleave* button is disabled, hover the cursor over it to display a tooltip explaining why you cannot currently cleave an electrode.

Note: Only the DMol³ and DFTB+ modules' Electron Transport tasks can work with documents containing electrodes or transport devices. In addition, some Materials Visualizer functions are not enabled for these objects.

Help: Displays the Help topic in a browser.

Note: Any existing hierarchy must be removed before the electrodes can be created.

Access methods

Menu *Build | Build Transport Device | Cleave Electrode*

Transport Device dialog

The Transport Device dialog allows you to add a device section to a 2D-periodic document containing a single electrode or two electrodes. The document should not contain atoms that are not part of an electrode. If the document contains a single electrode a second mirrored copy of the electrode will be created to complete the device.

Insert: Choose a document to insert between the two electrodes. It can be either a molecular document (nonperiodic) or a 2D or 3D periodic document provided its lattice parameters match the lattice parameters of the host.

Note: The document chooser filters the files in the current project so that only currently loaded 3D model documents are displayed.

Insert molecular fragment

This section controls insertion of nonperiodic documents.

Direction: Choose a direction which would be aligned with the repeat vectors of the electrodes.

- X (default)
- Y
- Z

Min. supercell size: Specify the minimum supercell size required to avoid self-interaction of the molecule(s) placed between electrodes. This value gives an estimate of the distance between molecules. Default = [10 Å](#).

Insert periodic fragment

This section controls insertion of a periodic system.

Thickness: Specify the (approximate) thickness of the material which will be inserted. Available only for 3D periodic documents. Default = [4.00 Å](#).

Max. lattice mismatch: Specify the maximum percentage by which the lattice parameters of the inserted system are allowed to vary from the host values. Default = [4.00 %](#).

Max. supercell size: Specify the maximum host supercell size to be considered while trying to match lattices of the host and inserted document. Default = 2. To the right of this setting the actual supercell which best satisfies the specified mismatch parameter is displayed, provided there is a match. For 3D periodic documents the best matching cleave surface is chosen automatically.

Build: Builds the transport device.

Note: This button will be disabled if the selected fragment cannot be inserted into the electrode document.

Tip: If the *Build* button is disabled, hover the cursor over it to display a tooltip explaining why you cannot currently build a transport device.

Note: Only the DMol³ and DFTB+ modules' Electron Transport tasks can work with documents containing electrodes or transport devices. In addition, some Materials Visualizer functions are not enabled for these objects.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build Build Transport Device Transport Device</i>
------	--

Mesostructure Builder

The Mesostructure Builder brings together tools for building mesoscale molecules and template structures for use in mesoscale applications, such as Mesocite. The Mesostructure Builder supports four different tasks: Defining and editing [bead types](#), building [mesomolecules](#), building mesostructure [templates](#), and creating templated [mesostructures](#). The Mesostructure Builder tools allow you to:

- Create complex topologies for molecules, without the need to use or learn a string notation.
- Set up a structure in abstract terms prior to specifying molecular detail and use such mesostructure templates with different mixtures or types of molecules.
- Reduce the equilibration time of a mesoscale simulation, by using templates representing the expected equilibrium configuration.
- Simulate novel structures that would otherwise be impossible or too time-consuming to create in a mesoscale simulation.
- Convert atomistic structures to mesoscale bead structures for use in simulations.

Beads and Bead Types

Beads are fundamental building blocks in mesoscale modeling, somewhat analogous to atoms in atomistic modeling. Each bead has many of its properties defined according to its bead type, in the same way that an atom derives many of its properties from the element type. Since a new Materials Studio project has no knowledge of bead types, appropriate bead types first have to be [created and edited](#). Bead types created in this way are then available for use in other applications, such as the Mesomolecule Builder.

Note: Visualizer tools that specifically act on atoms and bonds, such as the Sketching or Bond Calculation tools, will ignore any bead content in the document. Other Visualizer tools such as the [Crystal Builder](#) will generally process both beads and atoms as appropriate. Similarly application modules will, unless otherwise specified, ignore any bead content of documents and process just the atomistic content.

Tip: To select all beads of the same type, press and hold the ALT key and double-click on a bead.

Coarse Grainer

Coarse graining is the conversion of a molecular structure, containing atoms and bonds, to a mesoscale structure, containing beads and connectors. Three methods of defining groups of atoms to be converted to beads are available and can be used in any combination; Patterns, Motion groups, and Subunits. These methods identify groups of atoms with bead types, the resulting configuration of patterns and bead type names can then be used to translate a structure from an atomistic to a mesoscale representation.

Mesomolecule Builder

The Mesomolecule Builder allows you to build molecules by connecting individual beads. With the Mesomolecule Builder it is easy to construct molecules with common topologies, such as [linear polymers](#), [branched polymers](#) and [dendrimers](#). More complex topologies, such as comb or star polymers, can be created using the branch point functionality. Within a molecule you can arrange bead types in blocks or in random sequences.

Mesostructure Template Builder

The Mesostructure Template Builder allows you to [design a mesoscale system](#) in terms of abstract shapes, called "formers". Various types of formers are available: Droplet, Rod, Slab, Shell, and Tube formers, allowing you to template many different structures, from simple lamellar arrangements to complex coated nanotubes. You can associate a filling material, or "filler", with each former in the template. Different formers can be filled by the same material, making it easy to distribute a material over different phases in a mesoscale system.

Mesostructure Builder

The Mesostructure Builder allows you to pack a mesostructure template with mesoscale molecules to produce a structured system by selecting one or more mesoscale molecules for each filler material. All formers of a given filler are be packed with the selected mesoscale molecules. You can control the packing process in many ways, for example by randomizing the configurations, changing the density, or using constraints to pack specified beads onto surfaces. The Mesostructure Builder can output the structure in terms of molecules, density fields, or both, providing input for mesoscale particle and field methods, such as Mesocite.

Tasks in Mesostructure Builder

The following topics provide information on building mesostructures.

Creating and editing bead types

Beads are fundamental objects in certain types of mesoscale modeling, to some extent analogous to atoms in atomistic modeling. In the same way that each atom has many of its properties defined by its element type, each bead has many of its properties defined by its bead type.

Unlike elements, of which there is a fixed number, there is no defined set of bead types, nor standard property values. Consequently, a new Materials Studio project will initially have no known bead types. Any bead types required for subsequent modeling must be defined, and must have their properties set to values appropriate for the type of modeling to be undertaken.

To create a new bead type

1. Choose *Build | Build Mesostructure | Bead Types* from the menu bar to open the [Bead types](#) dialog.
2. If any bead types have already been defined in the project, either through this dialog or by importing a mesoscale structure file containing beads, these beads types will be shown in the list, together with their defined colors.
3. To create a new bead type, enter a name in the final empty row of the *Bead Type* column of the grid, and then either click the mouse elsewhere or press the TAB key.
4. The list of bead types will be refreshed, with the new bead type shown in its alphabetical position, together with its [assigned color](#).
5. Click the *Properties...* button to open the [Bead Type Properties](#) dialog. The properties have the values shown in the [Bead Type Defaults](#) dialog.
6. The properties of the new bead type should be checked, as the values of some properties may affect subsequent building operations.
7. If the properties of the new bead type are not as desired, they can be [modified](#) at this point.
8. Alternatively, you can [change the default bead type properties](#) before creating further bead types.

To edit the properties of an existing bead type

1. Choose *Build | Build Mesostructure | Bead Types* from the menu bar to open the Bead Types dialog.
2. From the list of bead types, choose the type whose properties you wish to edit, by clicking on the name.
3. Click the *Properties...* button to open the [Bead Type Properties](#) dialog.
4. Make any changes that you require to the property values.
5. These changes will apply to any documents using the chosen bead type. The relevant document views will be refreshed to reflect the changes.

Note: It is not possible to change the name of an existing bead type because the name is the primary identification method for the bead type.

If you mistyped the name of a new bead type during creation, you should [delete](#) the mistyped one and create a new one with the correct name.

To change the default bead type properties

1. Choose *Build | Build Mesostructure | Bead Types* from the menu bar to open the Bead Types dialog.
2. Click the *Properties...* button to open the [Bead Type Properties](#) dialog.
3. In the *Bead Type Properties* dialog, click the *Defaults...* button to open the [Bead Type Defaults](#) dialog.
4. Make any changes that you desire to the default property values.

Note: Changes to default property values have no effect on existing bead types, unless the *Reset* button is used on the [Bead Type Properties](#) dialog.

Colors of bead types

When a new bead type is created using the [Bead Types](#) dialog, a color is automatically derived from the name. This color is then compared with the colors of all existing bead types. If it is too similar to another bead color then the algorithm is repeated to create another color, until a suitably unique color is obtained.

This means that generating a new bead type with a specific name in different projects will usually generate the same color. However, this cannot be guaranteed because one project may already contain bead types whose colors are sufficiently close to that derived for the new bead type that they cause a conflict, in which case an alternative color is chosen.

Names of bead types

Bead type names are case sensitive, so it is possible to have two bead types whose names differ only in their case.

Rules for bead type names are as follows:

- The first character must be alphabetic (either upper or lower case).
- After the first character, the name can contain any combination of alphabetic (upper or lower case) or numeric characters, or any of the following characters: () [] _

To delete a bead type

Bead types can be removed from a project, but only if they are not being used in an open document.

1. Choose *Build / Build Mesosstructure / Bead Types* from the menu bar to open the Bead Types dialog.
2. From the list of bead types, choose the type which you want to delete.
3. Press either DELETE or BACKSPACE on the keyboard.
4. If the bead type is not being used by an open document, it will be deleted from the project and the list of bead types will be refreshed.
5. If the bead type is currently in use by an open document a warning message to this effect is shown and the deletion will not proceed.

Coarse graining atoms to beads

Coarse graining is the conversion of a molecular structure containing atoms and bonds to a mesoscale structure containing beads and connectors. Each bead represents a group of atoms. The center of mass of the group of atoms is preserved as the position of the bead. Atoms can be grouped together in any manner according to the system being investigated and the intentions of the user.

Before an atomistic structure can be coarse grained, the groups of atoms that will be associated with bead types must be defined in a typing document. This is a Study Table document in which column A should contain 3D Atomistic documents describing the structures represented by each bead type and column B should contain the bead type names. The typing document can be created manually, or it can be created and populated using the coarse grainer. Three methods of creating structures for typing are available and can be used in any combination.

Note: The methods the Coarse Grainer uses to group the atoms are applied according to the order: [Patterns](#), [Motion groups](#), [Subunits](#). Atoms that have been grouped with one method are not available for subsequent methods.

Patterns

If the structure to be coarse grained contains multiple identical groups, patterns can be used to identify those groups in the structure. A pattern document is required as a Study Table containing structures in column A. Each structure in the pattern document will be matched against the input structure, and the resulting groups are converted to beads.

A pattern structure can be partitioned using motion groups, for example a lipid molecule in which the atoms in the polar head and apolar tails are grouped by separate motion groups. The partitioning in the pattern structure is carried over to each match in the input structure and the resulting groups are converted to beads. This allows you to coarse grain structures containing many molecules of the same kind, such as polymer melts and bilayers. If no motion groups are defined in the pattern structure, the entire structure will be converted to beads.

The pattern document can contain more than one structure, for example polymer and solvent patterns. The pattern structures are applied according to the row order in the Study Table. Atoms covered by one pattern are not available for matching by subsequent patterns.

Note: The row order of structures in the pattern document can be important if one pattern matches another. For example, in a polymer chain the difference in groups at the end of the chain and those in the middle is often just a single hydrogen atom. In this case the pattern for the end groups should be applied before the pattern for the chain groups, to prevent the head and tail groups being matched by the inner group pattern, leaving unmatched hydrogen atoms and subsequent failed coarse graining.

Motion groups

The input structure can be prepared with motion groups, such that each motion group corresponds to a group of atoms to be converted to a bead. Motion groups can be added to the structure using the Motion Groups dialog.

Subunits

Subunits are the building blocks of structures containing multiple equivalent components. For example, subunits may be the repeat units used to construct [polymers and dendrimers](#) or the amino acid residues and nucleotides in proteins and nucleic acids respectively, which can be imported in .pdb format.

Using the Coarse Grainer


Coarse Graining an atomistic structure involves three distinct stages:

1. Selecting the [matching method\(s\)](#)
2. Creating and reviewing the [typing document](#)
3. Configuring and running the [coarse grainer](#)

In order for coarse graining to proceed successfully, it is important that you should:

- Take into account the hierarchy of the matching methods
- If using patterns, ensure that the pattern structures are inserted into the Study Table document in the required order.
- Only uncheck the *Automatically update typing document* checkbox if you have already confirmed that the structures in the selected *Typing document* are appropriate and their associated bead types are appropriately named.

To select the method(s)

1. Import an existing document containing atoms or create new molecules such as linear polymers, branched polymers, and dendrimers, using the [building](#) or sketching tools.
2. Select *Build | Build Mesostructure | Coarse Grain* from the menu bar or click the *Coarse Grain*  button on the Mesostructure toolbar to open the [Coarse Grain](#) dialog.
3. Choose the method(s) you wish to use to define atom groups, options are:
 - [Patterns](#)
 - [Motion groups](#)
 - [Subunits](#)

Note: Atoms that have been grouped with one method are not available for subsequent methods, so it is important to deselect the methods you do not wish to use.

4. Click the *Preview as motion groups* button to display motion groups around the portions of the molecule that will form the beads. Atoms that could not be grouped with any of the chosen methods can then be selected.

Tip: Use the *Preview* function to investigate coarse graining with different methods. Previewing is entirely optional and will not affect the typing document or project bead types.

To create and review the typing document

Note: These steps are optional but may be useful to ensure a smooth coarse graining process and to check the method(s) used. If you are confident in the structures for typing and bead type names defined by the selected method(s) you can skip to [creating](#) the coarse grained structure.

1. If there is no typing document in the current project, click the *Create* button. This creates and opens a new study table document called *Bead Typing.std* containing two columns. The *Typing document* dropdown list is automatically updated with the name of the new study table document, even if a different typing document had previously been selected.

Note: If a Study Table document named *Bead Typing.std* already exists, the new study table document will be called *Bead Typing (2).std* and so on. Previous Study Table documents will not be overwritten.

Tip: To clear a specified typing document and start afresh, select the document from the *Typing document* dropdown list and press the DELETE key. This will not delete the typing document from your project.

2. In *Bead Typing.std* double-click on a cell in the Structure column. This opens a Detail View containing the structure of the fragment.
3. Click in a cell in the *Bead Type Name* column and, if necessary, edit the name of the bead type.

Note: You can edit the names of the bead types in the study table. When converting to a bead structure, this name will be used to create bead types in the project. Since you cannot change the name of a project bead type once it is created, it is important to ensure all names in the typing document are suitable for use in the project.

4. If there is already a typing document present, ensure that it is selected from the *Typing document* dropdown list and click the *Update* button. Any bead structures in the active 3D Atomistic document that are not already in the typing document will be added with an associated bead type name.

To create the coarse grained structure

1. Select the [method](#) and set up and review the [typing document](#) if desired.
2. Ensure that the *Automatically update typing document* checkbox is checked. If no typing document is selected and this checkbox is unchecked the *Build* button will be disabled.
3. When you are satisfied with the structures for typing and bead type names in the typing document, double-click on the 3D Atomistic document in the Project Explorer and click the *Build* button on the Coarse Grain dialog. A new document is created containing beads and connectors representing the atoms and bonds in the input structure. Bead types are created in the project using the names in the typing document.

Tip: If the *Automatically update typing document* checkbox is checked, the *Build* step automatically creates a new typing document if none is selected, or updates a selected list. This will create bead types in the project with the default naming.

Potential problems in coarse graining

There are a few common problems which may be encountered during coarse graining:

■ Not all atoms matched to beads

During coarse graining, or preview, the message *Some atoms in the document cannot be matched to beads using the defined matching criteria for coarse-graining* is reported. This could be as a result of the order in which patterns are matched from a Pattern document - atoms matched using the first row, for example, are not available for matching by the second. For polymers it is important that terminal patterns are matched before chain groups. You may be able to rectify the problem by changing the order of the rows in the pattern document. If this occurred when creating or updating the typing document, it will be updated with those bead types that were found. If it occurred during creation of a coarse grain document, the task will fail and no new coarse grained document will be created.

■ Generated bead type names are unsuitable

During coarse graining the message *Some bead type names have been modified to conform to the permitted naming convention for bead types* may be reported. This occurs because the permitted names for Bead Types are quite restrictive. During the creation or updating of the typing document some bead type names were encountered that do not conform to the standard. These names have been modified to adhere to the standard. This is just a warning, and the task will have completed successfully.

■ Invalid bead type names

If you have specified bead type names which are inappropriate the message *Invalid name for bead type in row <number> of the bead typing document* is reported. This occurs when the bead type names in the typing study table have been modified to include any characters not permitted for bead type names. Allowed characters are all standard alphabetic characters (upper and lower case), numbers, and ()[]_. The name must start with an alphabetic character. To proceed, you must change the names so that they all conform to the described standard.

Building mesoscale polymers

A number of different types of mesoscale structure can be built using the [Mesomolecule Builder](#) dialog. This topic discusses building simple, unbranched mesoscale structures of the following types: [homopolymers](#), [block copolymers](#) and [random copolymers](#). Other topics describe building [mesoscale branched polymers](#) and [mesoscale dendrimers](#).

To build a mesoscale homopolymer

1. [Create](#) the necessary bead type, or ensure that it is known to the project, for example by importing an existing mesoscale structure document containing this bead type.
2. Choose *Build | Build Mesoscale | Mesomolecule* from the menu bar to open the Build Mesomolecule dialog.
3. Check that the *Repeat unit definition* list is empty.
 - If the list is not empty, clear it by selecting each component in turn and pressing the BACKSPACE or DELETE key to remove them from the list.
4. Click in the empty *Component Name* cell and, from the dropdown list, choose the desired bead type for the polymer.
5. Change the value in the *Number* cell to the desired number of copies of this bead for the chain.
6. Ensure that the *Number of repeat units* is set to **1**.
 - For homopolymers, the individual values for the *Number* and *Number of repeat units* are not important; instead, their product will give the chain length.
7. Ensure that the *Randomize order within repeat unit* checkbox is [unchecked](#).
8. Ensure that the *Add to branch points* checkbox is either disabled or is [unchecked](#).
9. Click the *Build* button.
10. A new document will be created containing the desired homopolymer.

To build a mesoscale block copolymer

1. [Create](#) the necessary bead types, or ensure that they are known to the project, for example by importing an existing mesoscale structure document containing these bead types.
2. Choose *Build | Build Mesoscale | Mesomolecule* from the main menu.
3. Check that the *Repeat unit definition* list is empty.
 - If the list is not empty, clear it by selecting each component in turn and pressing the BACKSPACE or DELETE key to remove them from the list.
4. Click in the empty *Component Name* cell and, from the dropdown list, choose the desired bead type for the first component of the block copolymer.
5. Change the value in the *Number* cell to the desired number of copies of this bead within the repeat unit block.
6. Repeat the previous two steps to define the remaining components for another repeat unit block.
7. Set the *Number of repeat units* to the desired value to give the desired total chain length.
 - The total chain length will be the sum of the *Number* values for each component, multiplied by the *Number of repeat units*.
8. Ensure that the *Randomize order within repeat unit* checkbox is [unchecked](#).
9. Ensure that the *Add to branch points* checkbox is either disabled or is [unchecked](#).
10. Click the *Build* button.
11. A new document will be created containing the desired block copolymer.

To build a mesoscale random copolymer

Mesoscale random copolymers are created by specifying the composition of a repeat unit, and then randomly changing the bead type for each copy of the repeat unit within the chain.

1. [Create](#) the necessary bead types, or ensure that they are known to the project, for example by importing an existing mesoscale structure document containing these bead types.
2. Choose *Build | Build Mesoscale | Mesomolecule* from the main menu.
3. Check that the *Repeat unit definition* list is empty.
 - If the list is not empty, clear it by selecting each component in turn and pressing the BACKSPACE or DELETE key to remove them from the list.
4. Click in the empty *Component Name* cell, and from the dropdown list choose the desired bead type for the first component of the random copolymer.
5. Change the value in the *Number* cell to the desired number of copies of this bead within a single copy of the repeat unit.
6. Repeat the previous two steps to define the remaining components within the repeat unit.
7. Set the *Number of repeat units* to the desired value to give the desired total chain length.
 - The total chain length will be the sum of the *Number* values for each component, multiplied by the *Number of repeat units*.
8. Ensure that the *Randomize order within repeat unit* checkbox is [checked](#).
9. Ensure that the *Add to branch points* checkbox is either disabled or is [unchecked](#).
10. Click the *Build* button.
11. A new document will be created containing the desired random copolymer.

Building mesoscale branched polymers

A number of different types of mesoscale structure can be built using the [Build Mesomolecule](#) dialog. This topic describes building branched mesoscale polymer structures. It is also possible to build

[mesoscale polymers](#) (homopolymers, random copolymers and block polymers) and [mesoscale dendrimers](#).

To build a branched mesoscale polymer, first you must [create the backbone](#) of the structure and then [define and add the branches](#). Using these steps, it is possible to create [more complicated structure](#) built up from chains of beads.

To create the backbone for a mesoscale branched polymer

- Follow the instructions for [building mesoscale polymers](#) to build a suitable polymer backbone. This can be a homopolymer, a block copolymer, or a random copolymer.

To define and add branches for a mesoscale branched polymer

1. In the document containing the backbone bead structure for the branched polymer, select the individual beads onto which branches are to be added.
2. Choose *Build | Build Mesostructure | Mesomolecule* from the menu bar to open the Build Mesomolecule dialog.
3. Specify the desired *Component Names* and *Numbers* for the repeat unit of the chains to be added.
4. Ensure that the *Randomize order within repeat unit* checkbox is set as desired.

Note: If this is [checked](#), the randomization is performed independently for each repeat unit, and also for each chain that will be generated.

5. Set the *Number of repeat units* to the desired value.
6. Check the *Add to branches* checkbox.
7. Click the *More...* button to open the [Mesomolecule Branches](#) dialog.
8. Ensure that the *Branch from selected beads* option is selected.
9. Set the *Number of branches to attach* to have the desired value. This is the number to attach to each selected bead.
10. Click the *Build* button on the *Build Mesomolecule* dialog.
11. New chains, each corresponding to the specification on the *Build Mesomolecule* dialog, will be added, sprouting from each of the selected beads. The geometry for the new chains is arbitrary, and is designed to give a reasonable schematic view of the structure rather than a properly minimized structure.

To create complicated structures from bead chains

Using the *Branch from selected beads* option on the [Mesomolecule Branches](#) dialog, it is possible to create a wide variety of more complicated structures. There are no restrictions on how many branches can be attached to any particular selected bead, new branches can be added to a bead that already has beads attached to it. The new branches can be different from the other chains in the document.

Each time that a new set of branches is added to a particular selected bead, a new orientation is chosen for the new beads, so that chains do not overlap. However, for highly branched mesoscale object there is no guarantee of this. If this is important that all chains are isolated, a complicated structure should be built up a small portion at a time, so that you can *Undo* an addition that gives an undesirable geometry. Rotating the existing structure relative to the Cartesian framework (using SHIFT + right-click) will give a different set of initial conditions from which the new geometry will be calculated.

Building mesoscale dendrimers

A number of different types of mesoscale structure can be built using the [Mesomolecule Builder](#) dialog. This topic discusses building mesoscale dendrimer structures. Other topics describe building [mesoscale](#)

[polymers](#) (homopolymers, random copolymers and block polymers) and [branched mesoscale polymers](#).

A mesoscale dendrimer is built in two stages. You must first [create the seed structure](#) onto which the dendrimer will be grown and then [define and add the dendrimer branches](#).

Note: In this context, a *terminal bead* is one that has no more than one other bead connected to it. Isolated beads count as terminal beads.

To create a seed structure

The seed structure can be:

- A single bead. This can be created by following the instructions for [Building mesoscale polymers](#), but using values of **1** for both the *Number* of a component within a repeat unit and also for the *Number of repeat units*
- A mesoscale polymer chain
- A branched polymer
- A dendrimer

The important criterion for a seed structure is that it must have one or more terminal beads.

To define and add dendrimer branches

1. Make the document containing the seed structure for the mesoscale dendrimer the active document.
2. If desired, select a portion of this structure. The dendrimer will grow from terminal beads in the selected portion, or, if nothing is selected, from all terminal beads in the seed structure.
3. Choose *Build | Build Mesomolecule | Mesomolecule* from the menu bar to open the Build Mesomolecule dialog.
4. Specify the desired *Component Names* and *Numbers* for the repeat unit of the chains to be added.
5. Ensure that the *Randomize order within repeat unit* checkbox is set as desired.

Note: If this is [checked](#), then the randomization is performed independently for each repeat unit, and also for each chain that will be generated.

6. Set the *Number of repeat units* to the desired value
7. Check the *Add to branches* checkbox.
8. Click the *More...* button to open the [Mesomolecule Branches](#) dialog.
9. Ensure that the *Branch from terminal beads* option is selected.
10. Set the desired value for the *Number of generations*. This is the number of times that the building process will be repeated, as described below.
11. Set the *Number of branches to attach* to have the desired value. This is the number of chains to attach to each selected bead.
12. Click the *Build* button on the *Build Mesomolecule* dialog.
13. For the first generation of the dendrimer building operation new chains, each corresponding to the specification on the *Build Mesomolecule* dialog, will be added, sprouting from each of the specified terminal beads. The geometry for the new chains is arbitrary, and is designed to give a reasonable schematic view of the structure rather than a properly minimized structure.
14. For each subsequent generation new chains will be added to the terminal beads of the chains from the previous generation of the dendrimer building process.

15. In this way you can add multiple generations of a dendrimer structure to a single point on a seed structure in one step, after which a different dendrimer structure can be added to the other end of the seed structure.

Constructing a mesostructure template

The Mesostructure Template Builder allows you to design a mesoscale structure by representing the phases in the system in terms of abstract shapes, called "formers". You can choose a variety of shapes to template many different arrangement of phases, from simple lamellar to complex coated nanotube structures. The former types currently supported are:

- System former: Defines the simulation cell and is the first object to be defined when building a template.
- Droplet former: A spherical object with a specified radius.
- Rod former: A cylindrical object with a specified radius, spanning the entire length of the simulation cell.
- Slab former: A slab of specified thickness spanning the entire simulation cell in the other two dimensions.
- Shell former: A spherical shell with specified inner radius and outer radius.
- Tube former: A cylindrical shell with specified inner radius and outer radius, spanning the entire length of the simulation cell.

Each former can be assigned a "filler", defining, in abstract terms, the material this former is composed of.

Note: Several formers can have the same filler; for example, to create a template for an oil/water emulsion, a filler called [water](#) could be assigned to the System former, and a filler called [oil](#) assigned to all droplet formers created within it.

Formers can be assembled to form more complex composite structures; for example, you could place a smaller droplet former inside a larger one and assign them different fillers. This structure simulates a vesicle, where the phase in the center is separated from the surrounding environment by a second phase occupying the outer layer.

To build a mesostructure template

1. Choose *Build | Build Mesostructure | Mesostructure Template* from the menu bar to open the [Build Mesostructure Template](#) dialog.
2. Ensure that there is either no active document, or that the active document does not contain a System former already.
3. Select the [Add Formers](#) tab, the *Former type* is set to [System](#) by default.
4. Specify the dimensions of the simulation cell you wish to create in the *Extent (X,Y,Z)* text boxes.
5. Optionally, specify a filler for the new system former by entering a name in the *Filler* text box or selecting an existing filler from the dropdown list.
6. Click the *Build* button.

A new document is created with a System Former, visualized as a cuboid wireframe. If a filler was selected on creation, the former is colored with the filler color, otherwise it will be white. You can now add additional formers to the newly created template.

7. To add a former to the system ensure that the active document contains the System Former.
8. On the [Add Formers](#) tab, choose the type of former to add from the *Former type* dropdown list.
9. Specify the *Orientation*, *Radius* (or radii in case of shelled formers), and/or *Depth*, as appropriate, for the new former.

10. Select whether to use Cartesian or fractional coordinate notation from the *Coordinates* dropdown list. Specify the *Position* of the center of the former in the selected coordinate notation in the appropriate text boxes.
11. If you want the filler component to be packed on the surface of the former, check the *Enable surface packing* checkbox and, if the selected former type has more than one surface, set the *Packing surface*.
12. Optionally assign a filler, either by typing in a new name, or by selecting a previously defined filler from the dropdown list.
13. Click the *Add* button.

Each time you click the *Add* button, a new former will be added to the template in the active document. Repeat the previous steps until all required formers have been created.

To modify the fillers in an existing template:

If you chose not to assign fillers on creation of the formers, or if you want to modify the filler properties including assignment later, you can do so using the [Fillers](#) and [Formers](#) tabs of the Mesostructure Template Builder.

1. Make the template document the active document.
2. Select the [Fillers](#) tab on the Build Mesostructure Template dialog. All fillers previously defined in the template are listed.
3. To add a new filler to the active document, click the *Add* button to create a new filler in the list at the top of the dialog. Once added, edit the name for the filler to represent its functional role within the bulk system, for example [Solvent](#), [Surfactant](#), [Substrate](#) and so on.
4. Optionally, click on the color control to display the Color dialog. Choose a color for the new filler from the palette and click the *OK* button.

Repeat the above steps to add more fillers to the document, if required. Having defined all fillers, you can now continue to assign them to selected formers in the active document.

Tip: It is advisable to choose names for the fillers that reflect their role in the system you are creating. The same filler names should be used to characterize a particular phase type across all template documents. This approach allows you to consistently associate the same mixtures of mesoscale molecules to particular phase types when packing in different template documents.

5. Select the [Formers](#) tab. All formers defined in the active document are listed. If you wish, you can control the visibility of the formers using the checkboxes, rename the formers by clicking on the name and entering a new one, or delete formers from the current document by selecting them in the list and pressing the DELETE key.
6. Select the former(s) from the list to which you wish to assign a filler. By selecting formers, the corresponding object in the active document is highlighted. Alternatively, select formers by clicking on the objects in the active document.
7. Select the [Fillers](#) tab. The number of formers currently selected in the active document is displayed next to the *Assign* button.
8. Choose the filler that you wish to fill the selected former with from the list at the top of the dialog. The name of the currently selected filler is displayed in the *Color* field.
9. Click the *Assign* button to associate the filler with the selected former(s).

Repeat the above steps to fill any remaining unfilled formers in the system, including the system former itself.

To change the former hierarchy in an existing template:

A point in the simulation cell can be owned by one, and only one, former; there is no shared ownership of space. Where two formers intersect, the former hierarchy is used to decide which former claims overlap in space. This hierarchy is shown on the *Formers* tab as an ordered list, the ownership is assigned from bottom to top, meaning that if a former is lower in the hierarchy than another, it will own any intersection space between them.

The Mesostructure Builder uses the ownership hierarchy when packing the template with molecules. Modifying the hierarchy can have a dramatic effect on the resulting structure. For example, to simulate a vesicle by placing a larger droplet around a smaller droplet, it is necessary to ensure that the larger outer droplet is above the smaller inner droplet in the hierarchy so that the inner droplet will assume control over the intersection between the two, which is effectively the entire volume of the inner droplet. If the outer droplet is lower in the hierarchy, it will claim the space of the inner droplet as well, making the inner droplet redundant.

1. Make the template document the active document.
2. Select the [Formers](#) tab. All formers defined in the active document are listed. By default the hierarchy corresponds to the order in which the formers were created in the template.
3. Select one or more formers from the hierarchy list and use the up and down arrow buttons to move the selected formers in the list.

Note: The system former cannot be moved within the hierarchy, it must remain at the root of the hierarchy. This implies that the system former will own all the space not occupied by any other former.

The structured mesoscale system that you have constructed can now be packed with specific mixtures of mesoscale molecules and the resulting 3D Atomistic or 3D Mesoscale document can be used as the input for a mesoscale simulation, such as Mesocite.

Setting up a structured mesoscale system

The Mesostructure Builder allows you to pack mesoscale molecules into a mesostructure template to create a structured input for mesoscale applications such as Mesocite. To build a templated structure, you need a template as well as have available documents with individual molecules of all components you wish to pack the template with. For each filler present in the template document, you select one or more molecules. The Mesostructure Builder will then pack all formers that contain this filler with the associated molecules, returning a structured input document for subsequent mesoscale simulations.

By default, the Mesostructure Builder will pack molecules in the provided conformation, such that their center-of-geometry is within the volume of the former. The builder will also randomize the orientations of the molecules. However, you can control the packing algorithm in a variety of ways, for example, to achieve surface packing, orientated packing, or packing with randomized conformations. Furthermore, you can control the output, to include or exclude components, depending on the mesoscale application with which you intend to use the structure.

Note: The Mesostructure Builder interprets the dimension of template formers in Å. This is suitable for use in Mesocite module, including the DPD task.

To build a templated mesostructure:

1. Import an existing template document or [construct a mesostructure template](#) using the [Mesostructure Template Builder](#).
2. Import existing documents with molecules for all components, or create new molecules such as [linear polymers](#), [branched polymers](#), and [dendrimers](#) using the [Mesomolecule Builder](#).
3. Choose *Build* / *Build Mesostructure* / *Mesostructure* from the menu bar to open the [Build Mesostructure](#) dialog. Ensure that the mesostructure template document is the active document. Select the [Components](#) tab.

The components tab lists all the fillers defined in the template. It is assumed at this stage that during the construction of the template, each filler has been assigned to one or more formers. Because each filler has a different color, you can easily check if this is the case. If necessary, you can modify the assignment using the [Build Mesostructure Template dialog](#).

Note: The Mesostructure Builder stores the most recent composition information for each filler. If the template document in focus contains a filler for which a mixture of mesoscale molecules has been assigned previously and all the mesoscale molecules are still defined, the grid will show the persisted filler composition by default.

4. For each filler, click in the *Mesoscale Molecule* column. Select the document containing the mesoscale molecule of the desired species.
5. In the column *Relative Amount*, specify the amount of the selected component in arbitrary units. The amount will be normalized and displayed in *Fraction* column of the grid.
6. To assign more than one mesoscale molecule to one filler, repeat the previous steps using the empty row at the bottom of each filler section.

These steps should be repeated for each filler defined in the template. Ensure that all fillers have been assigned at least one molecule, as otherwise formers of this material will not be packed.

Pressing Build at this stage will create a mesoscale structure document using the default packing algorithm and output settings. Before doing so, you may want to set the density:

7. In the *Density* box, set the target density in g/cm^3 . This bulk density will be used for all formers in the simulation cell. You can also set the density in reduced units, by selecting [Reduced](#) from the *Units* dropdown list on the Options tab. The conversion between the two sets of units can be specified on the [Options](#) dialog.
8. Select the [Options](#) tab.
9. Uncheck the *Randomize conformations* checkbox if you want the packed molecules to all have the same conformation as the input molecule. By default the Mesostructure Builder randomizes the angles and torsions in the input structure.

Finally you can specify how the Mesostructure Builder should output the structure. By default, the bead structure is returned as a configuration of molecules in a 3D Atomistic structure document (.xsd). However, you can also create templated structures as density fields, or have both molecules and fields in the same document.

10. In the *Content* dropdown list select whether to return the structure as molecules, fields or both.
11. If fields are part of the content, set the *Field resolution* in Å. The resolution is the same in all directions. From the resolution the Mesostructure Builder will calculate the number of grid points in each direction.
12. Click the *Build* button.

Note: It is possible to specify or move formers in a mesostructure template such that they extend beyond the boundaries of the simulation cell, as defined by the system former. While the Materials Visualizer will render and display regions that fall outside the simulation cell in order to help you visualize the system, you should note that when the template is packed with mesoscale molecules, only the volume enclosed by the system former will be filled.

To build a templated mesostructure using surface packing:

To pack molecules onto the surface of a former, the former must be enabled for surface packing as explained in [constructing a mesostructure template](#); if the former has more than one surface, you must specify which surface to enable. Enabled surfaces are visualized as dots. Secondly, you must define a packing center in each molecule and define the molecular orientation. This is done by collecting beads whose center-of-geometry is to be used as packing center or as orientation center into named sets. Creation and deletion of those sets, as well as selection of beads within predefined sets, can be done using the Bead Packing Options dialog.

Note: If the molecule does not have a packing center defined, or if surface packing is not enabled for the former, the Mesostructure Builder will use the default, volume, packing. Combined surface and volume packing can be achieved by filling a surface-enabled former with a mixture of molecules, with and without constraints.

1. Set up a structure as above.
2. On the [Options](#) tab, click the *More...* button to open the [Bead Packing Options](#) dialog.
3. Make the molecule whose packing you wish to constrain the active document.
4. Select the bead(s) in the molecule whose center should be on the former surface.
5. Select [Head](#) from the *Bead tag* dropdown list and click the *Create* button. A bead *Head set* is created.
6. Select the beads whose center should be used to orientate the molecule. Molecules will be packed such that the vector from head to tail is perpendicular to the former surface, directed inside the former volume.
7. Select [Tail](#) from the *Bead tag* dropdown list and click the *Create* button. A bead *Tail set* is created.
8. Repeat for all molecules and close the Bead Packing Options dialog.

When you now click the *Build* button all formers that are surface packing enabled will be packed with molecules, with the packing center on the surface, and oriented perpendicular to the formers surface, directed inside the former volume. If a molecule has a bead *Head set* defined, but no bead *Tail set*, the molecules are still packed onto the surface, but with random orientation.

Tip: The number of molecules in surface and volume packing is the same. Thus, you can control the surface density by changing the volume of the former.

In some mesoscale applications it can be useful to fix certain beads during a simulation. For example, Mesocite will fix beads by starting from zero velocities and subsequently neglecting all forces that act on them. You can fix beads by selecting them in the structure and checking *Fix Cartesian Position* the checkbox on the Edit Constraints dialog.

Dialogs in Mesostructure Builder

Materials Studio's mesostructure building functionality allows you to create mesoscale objects of a wide range of types and to edit the bead types used.

Bead Types dialog

The Bead Types dialog allows you to define bead types for use by various of the mesoscale modeling applications. Bead types are uniquely defined by their names, which are case sensitive. Bead types can be [created](#), [edited](#) and [deleted](#) using this dialog.

When a new project is created, there are no known bead types, these must be introduced to a project, either by importing a document that contains beads of the desired types (and specifications), or by explicitly creating them using this dialog.

The grid lists all the bead types currently defined in the project (or filtered according to the checkbox below) together with the color used by default for display of beads of that type. One extra row is listed in the grid, to allow the entry of a new bead type.

Bead Type: Lists, in alphabetical order, the bead types, either all those known to the project or those used by the current document, depending on the setting of the *Filter by current document* checkbox.

Color: This shows the colors assigned to each of the bead types. The color is not editable within this dialog. The color assigned to a specific bead type can be changed using the [Bead Type Properties](#) dialog.


Filter by current document: When checked, the set of bead types that is displayed in the grid will be restricted to just those that are used in the current document. Default = [unchecked](#).

Properties...: Opens the [Bead Type Properties](#) dialog, which allows properties of the selected bead type to be modified.

Note: If the [Bead Type Properties](#) dialog is open, selecting a different row of the Bead Types grid will change the bead type whose properties are shown.

Help: Displays the Help topic in a browser.

Access methods

Menu	Build Build Mesostructure Bead Types
Toolbar	

Bead Type Properties dialog

The Bead Type Properties dialog shows the properties for a specific bead type and allows you to modify them.

Properties for: The string here reports the bead for which the following properties will be defined.

The bead type is chosen according to the selection in the [Bead Types](#) dialog, any change to which will automatically be reflected in the bead type shown here.

Mass: Specify the mass, in amu, for this bead type.

Radius: Specify the radius, in Å, for this bead type.

Note: Not all applications of bead documents acknowledge the Radius property.

Color: Specify the color for this bead type. When the bead type is created using the [Bead Types](#) dialog, a color is automatically generated from the name, but potentially altered to avoid color clashes with existing bead types. Clicking the color chooser allows you to modify the color for the bead type.


Defaults...: Opens the [Bead Type Defaults](#) dialog.

Reset: Resets the properties of the specified bead type to those shown in the [Bead Type Defaults](#) dialog. The color is not reset using the *Reset* button.

Note: Changing the properties of a bead type immediately changes the attributes of any beads that are defined using this bead type.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build / Build Mesostructure / Bead Types / Properties...</i>
Toolbar	 / <i>Properties...</i>

Bead Type Defaults dialog

The Bead Type Defaults dialog allows you to alter the property values that will be used when creating bead types using the [Bead Types](#) dialog, or when using the *Reset* button on the [Bead Type Properties](#) dialog.


Mass: Specify the mass, in amu.

Radius: Specify the radius, in Å.

Note: Changing default property values in this dialog will not effect the property values for existing bead types. To use new default property values for existing bead types, select the desired bead type in the [Bead Types](#) dialog, and then click the *Reset* button on the [Bead Type Properties](#) dialog.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build / Build Mesostructure / Bead Types / Properties... / Defaults...</i>
Toolbar	 / <i>Properties / Defaults...</i>

Coarse Grain dialog

The Coarse Grain dialog allows you to convert a molecular structure composed of atoms and bonds to a mesoscale structure containing beads and connectors. Groups of atoms are replaced by beads, with all identical groups being represented by the same bead type.

Method

This section controls the method(s) of grouping atoms used in the conversion of a molecular structure to a mesoscale structure. At least one of these methods must be selected.

Note: The methods the Coarse Grainer uses to group the atoms are applied according to the order: [Patterns](#), [Motion groups](#), [Subunits](#). Atoms that have been grouped with one method are not available for subsequent methods.

Patterns: When [checked](#), fragments in the selected Study Table document, will be used to define the groups of atoms that are converted to beads. Select a study table containing one or more fragment documents in column A from the dropdown list to define the patterns to be used for conversion to a mesoscale structure.

Motion groups: When [checked](#), motion groups in the active 3D Atomistic document are used to define the atom groupings during conversion to beads.

More...: Opens the Motion Groups dialog.

Subunits: When [checked](#), subunits (such as repeat units generated by the [Polymer Builder](#)) are used to define the groups of atoms converted to beads.

Preview as motion groups: Creates motion groups around the fragments, repeat units, or groups of atoms that will be converted to beads. Atoms that cannot be coarse grained with the chosen matching method can be selected.

Bead typing

This section controls the definition of structures as specific bead types in the generated mesoscale structure.

Typing document: The Study Table document in the current project containing the groups of atoms and their associated bead type names.

Create: Creates a new Study Table document collecting all unique structures in the input document corresponding to beads according to the selected matching method. The Study Table document will be opened and selected in the *Typing document* dropdown list. New typing documents are named *Bead Typing.std*.

Create | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Note: If a Study Table document named *Bead Typing.std* already exists, the new typing document will be called *Bead Typing (2).std* and so on. Previous typing documents will not be overwritten.

Update: Adds to the selected Study Table document all unique structures in the input document corresponding to beads according to the selected matching method. The Study Table document will be opened and remain selected in the *Typing document* dropdown list.

Update | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Coarse grain document

This section controls the initiation of the Coarse Grainer and its behavior regarding the typing document when a molecular structure is converted to a mesoscale structure.

Build: Converts the structure in the current 3D Atomistic document from atoms to beads according to the selected matching method(s). The specified *Typing document* is used to generate the desired bead types and a new structure document called *<seedname> CG.xsd* is created and opened.


Build | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Automatically update typing document: When checked the specified typing document will be updated with any structures corresponding to new bead types found during the coarse graining or a new typing document will be created if one was not specified. If no typing document has been specified, the building of a mesoscale structure can only proceed if the *Automatically update typing document* checkbox is checked. Default = [checked](#).

Note: If the selected *Typing document* contains inappropriate groups of atoms and the *Automatically update typing document* checkbox is unchecked the Coarse Grainer will fail.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build Build Mesostructure Coarse Grain</i>
Toolbar	

Build Mesomolecule dialog

The Build Mesomolecule dialog allows you to create one or more chains of beads, either in a new document or added onto specified locations within an existing mesoscale bead structure.

The *Repeat unit definition* grid lists all the components currently defined for building a new chain of beads. One extra row is listed in the grid, to allow the entry of a new component.

The only type of component currently available is a single bead. This is specified by the *bead type*. Any bead types to be used must be defined in the project using the [Bead Types](#) dialog, before being used in the Mesomolecule Builder.

Component Name: Displays the names of the components that currently specify the composition of a single repeat unit within a chain. The last empty cell can be used to specify a new component from a dropdown list of known bead types.

Number: Specify the number of each component that will be used within a single repeat unit. Default = 1.

Randomize order within repeat unit: When [unchecked](#), each repeat unit will be built in an identical manner. When [checked](#), indicates that the bead type of each repeat unit will be randomly changed.

Number of repeat units: Specify the number of repeat units that are desired in each chain. Default = 1.

Add to branch points: If enabled and checked, the *Build* operation will add chains onto specified beads in the document that currently has focus. If disabled or unchecked, the *Build* operation will create a new document containing a single chain according to the specifications above.

The *Add to branch points* checkbox is enabled only if the document currently in focus already contains some mesoscale beads.


More...: Opens the [Mesomolecule Branches](#) dialog. This allows detailed specification of where and how branches will be added. This button will be enabled only if *Add to branch points* checkbox is checked.

Build: Creates one or more mesoscale molecule chains according to the specifications on this and the [Mesomolecule Branches](#) dialog.

If the *Add to branch points* checkbox is enabled and checked, new chains will be added to the document currently in focus. If it is disabled or unchecked, a new document will be created, containing a single chain according to the specification on this dialog.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build Build Mesostructure Mesomolecule</i>
Toolbar	

Mesomolecule Branches dialog

The Mesomolecule Branches dialog allows you to specify how branches will be added to an existing document containing beads. The options specified on this dialog will be applied if the *Add to branch points* checkbox on the [Build Mesomolecule](#) dialog is enabled and checked when the *Build* button is pressed.

Branch from selected beads: If this option is chosen, new chains will be generated and attached to any beads that are selected in the document in focus.

Branch from terminal beads: If this option is chosen, new chains will be generated and attached to any terminal beads within the current selection, or, if nothing is selected within the document in focus, any terminal beads within the document.

Number of generations: If the *Branch from terminal beads* option is chosen, the process of adding chains can be repeated, with another set of new chains being added to the terminal ends of each of the newly created chains. The *Number of generations* value controls how many times the addition process is performed. Default = 1.


Number of branches to attach: Specify the number of chains that will be added at each branch point (either explicitly defined if *Branch from selected beads* is chosen or implicitly defined if *Branch from terminal beads* is chosen).

Note: If the *Randomize order within repeat unit* checkbox on the [Build Mesomolecule](#) dialog is checked, then each repeat unit in each chain will be randomized independently. So different chains contain different orders of component beads, although the total composition will be the same.

Note: The geometry used to add new branches is somewhat arbitrary, using the Cartesian framework. It is designed to give a reasonable schematic view of the resulting structure rather than a properly minimized structure. If the geometry of newly added branches is not appropriate, a significantly different orientation can be obtained by rotating the seed structure before adding the new branches.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build Build Mesostructure Mesomolecule More...</i>
Toolbar	 / <i>More...</i>

Build Mesostructure Template dialog

The Build Mesostructure Template dialog allows you to design a template for a complex mesoscale structure by assembling abstract "formers" representing the type, shape, and position of mesoscale phases and objects. Templates can be packed with specific mixtures of mesoscale molecules and the resulting 3D Atomistic document used as the input for a Mesocite calculation.


The Build Mesostructure Template dialog contains the following tabs:

- [Add Formers](#): Allows you to define a mesoscale simulation cell and to construct variously shaped formers within that system.
- [Fillers](#): Allows you to specify filler materials representing abstract phase types and assign them to the formers you have created.
- [Formers](#): Allows you to select, rename, and control the space ownership and visibility of formers.

Note: The settings of the Mesostructure Template Builder can be reset on the Settings Organizer dialog using the Mesostructure Builder entry in the Visualizer category.

Help: Displays the Help topic for the current tab.

Access methods

Menu	<i>Build Build Mesostructure Mesostructure Template</i>
Toolbar	

Add Formers tab

The *Add Formers* tab on the Build Mesostructure Template dialog allows you to define a mesoscale simulation cell and to construct variously shaped formers within that system.

Note: When using templates as input to the [Build Mesostructure dialog](#) dimensions will be interpreted in Å.

Tip: Once formers have been added to a system, they can be moved using the tools on the 3D Movement toolbar and their properties can be edited using the Properties Explorer.

Former type: Select the type of child former to be added to the currently active 3D Mesoscale document from the dropdown list. Available options are:

- [System](#)
- [Droplet](#)
- [Rod](#)
- [Slab](#)
- [Shell](#)
- [Tube](#)

Note: If a system former has not yet been defined in the currently active 3D Mesoscale document, the only option available from the *Former type* dropdown list will be [System](#). Once a system former has been added to the document, the child formers - [Droplet](#), [Rod](#), [Slab](#), [Shell](#), and [Tube](#) - will become available, but the [System](#) option will be removed.

Extent (X Y Z): Specifies the dimensions of the system former that will be created.

Note: The *Extent (X Y Z)* controls are only visible if the *Former type* is set to [System](#).

Radius: Specify the radius of a droplet or rod former.

Note: The *Radius* control is only visible if the *Former type* is set to [Droplet](#) or [Rod](#).

Depth: Specify the depth of a slab former.

Note: The *Depth* control is only visible if the *Former type* is set to [Slab](#).

Inner radius:, Outer radius: Specify the radii of a shell or tube former.

Note: The *Inner radius* and *Outer radius* controls are only visible if the *Former type* is set to [Shell](#) or [Tube](#).

Orientation: Select the alignment of a rod or slab former with respect to the axes of the system former from the dropdown list. Available options are:

- [Along X](#)
- [Along Y](#)
- [Along Z](#)

Note: The *Orientation* control is only visible if the *Former type* is set to [Rod](#), [Slab](#) or [Tube](#).

Coordinates: Select the coordinate system to be used to define a child former from the dropdown list. Available options are:

- [Fractional](#) (default)
- [Cartesian](#)

Position (a b c) or (x y z): Specify the position of the center of the child former in the *Position (a b c)* or *Position (x y z)* text boxes, as appropriate according to the option selected for the *Coordinates* system.

Note: The *Coordinates* control and the related text boxes are only visible if the *Former type* is set to [Droplet](#), [Rod](#), [Slab](#), [Shell](#) or [Tube](#).

Enable surface packing: Specify whether or not the filler components should be packed on the surface of the former.

Packing surface: Specify on which former surface the filler components should be packed when surface packing is enabled.

Note: The *Packing surface:* control is only visible if the former has more than one surface, i.e. if the *Former type* is set to [Slab](#), [Shell](#) or [Tube](#).

Filler: Select an existing filler in the current document from the dropdown list or specify a name for a new filler.


Note: If you do not wish to specify a filler at this stage, delete the name in the *Filler* text box and leave it blank. Fillers can be assigned subsequently using the [Fillers](#) tab. If you specify a new filler in the *Filler* text box, then the new filler will be added to the list on the *Fillers* tab. Filler names are case insensitive and all the fillers defined in a document must have unique names.

Build: Generates a new mesostructure template when the *Former type* is set to [System](#). New child formers can then be added to this parent system.

Add: Inserts the selected former into the currently active 3D Mesoscale document according to the specified settings. If a new filler name was specified, a filler with that name will be created, assigned a color, and added to the new former. If an existing filler was selected, that filler will be assigned to the new former.

Note: It is possible to specify or move child formers such that they extend beyond the boundaries of the simulation cell, as defined by the system former. Although the Materials Visualizer will render and display regions that fall outside the simulation cell in order to help you visualize the system you are creating, you should note that when the template is packed with mesoscale molecules, only the volume enclosed by the system former will be filled.

Access methods

Menu	<i>Build Build Mesostructure Mesostructure Template Add Formers</i>
Toolbar	 / <i>Add Formers</i>

Fillers tab

The *Fillers* tab allows you to specify filler materials representing abstract phase types and assign them to the formers you have created.

All the fillers presently defined for the currently active 3D Mesoscale document are listed at the top of the dialog. Click on an entry in the list to select the corresponding filler.

The name of each filler is displayed next to the icon. If no name has been assigned to the filler, the default name *Filler x* is displayed, where *x* is an integer that depends on the number of unnamed fillers in the currently active document. To rename a filler, click on the existing or default name and enter the new name. Filler names are case insensitive and all the fillers defined in a document must have unique names.


You can delete fillers from the currently active document by selecting the relevant item in the list and pressing the DELETE key.

Add: Inserts a new undefined filler into the list at the top of the dialog.

Color: Displays the name of the currently selected filler, if one has been assigned. Select a color to be used to represent the selected filler in the currently active 3D Mesoscale document. Clicking on the color control provides access the Microsoft Common Color Control for color selection. The spin controls allow the brightness to be adjusted.

Assign to formers: Fills the currently selected former(s) in the active 3D Mesoscale document with the specified filler. The name of the selected filler and the number of formers that are currently selected in the active document are shown next to the button. When no formers are selected, the default selection is to fill all the formers in the active document.

Access methods

Menu	<i>Build Build Mesostructure Mesostructure Template Fillers</i>
Toolbar	 / <i>Fillers</i>

Formers tab

The *Formers* tab on the Build Mesostructure Template dialog allows you to select, rename, and control the space ownership and visibility of formers.

All the formers presently defined for the currently active 3D Mesoscale document are listed at the top of the dialog. Click on an entry in the list to select the corresponding former in the active document. Check or uncheck the checkboxes to control the visibility of the formers. The name of each former is displayed next to the visibility control checkbox. If no name has been assigned to the former, a default name is

displayed. To rename a former, click on the existing or default name, then enter the new name. You can delete formers from the current document by selecting the relevant item in the list and pressing the DELETE key.



Move to top of tree: Moves the selected child former to the top of the hierarchy.



Move up one level: Moves the selected child former up one place in the hierarchy.



Move down one level: Moves the selected child former down one place in the hierarchy.




Move to bottom of tree: Moves the selected child former to the bottom of the hierarchy.

Note: The system former cannot be moved, it must remain at the root of the hierarchy.

Tip: A point in the simulation cell is owned by one former only, there is no shared ownership of space. Where two formers intersect, the former that is lower in the hierarchy will take ownership of the intersection space. So, for example, to simulate a vesicle by placing a larger droplet around a smaller droplet, it is necessary to ensure that the larger outer droplet is above the smaller inner droplet in the hierarchy so that the inner droplet will assume control over the intersection between the two, which is effectively the entire volume of the inner droplet. If the outer droplet is lower in the hierarchy, it will claim the space of the inner droplet as well, making the inner droplet completely redundant.

Access methods

Menu	<i>Build Build Mesostructure Mesostructure Template Formers</i>
Toolbar	 / <i>Formers</i>

Build Mesostructure dialog

The Build Mesostructure dialog allows you to create a structured mesoscale system from a template constructed using the [Mesostructure Template Builder](#). This can then be used as an input for a Mesocite mesoscale dynamics or DPD calculation.

The Build Mesostructure dialog contains the following tabs:


- [Components](#): Allows you to relate specific mesoscale molecules to the fillers defined in the input system and to specify their relative amounts.
- [Options](#): Allows you to select the content of the output document and details of the molecule conformations.

Build: Generates a new 3D Atomistic Document containing the constructed mesostructure.

Build | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Help: Displays the Help topic for the current tab.

Access methods

Menu	<i>Build Build Mesostructure Mesostructure</i>
Toolbar	

Components tab

The *Components* tab allows you to relate specific mesoscale molecules to the fillers defined in the input system and to specify their relative amounts and proportions in the grid.

Filler: Lists all the fillers defined in the active 3D Mesoscale document.

Mesoscale Molecule: Allows you to select the mesoscale molecule to assign to the filler from a document in the current project.

Relative Amount: Defines the proportion of beads in the filler that belong to this type of molecule.

Note: The relative amount of a type of molecule refers to the mass contained in molecules of this type, not to the number of molecules in the system. For example, if two molecule types [A 4 B](#) and [S 1](#) have relative amounts 1 and 3, it means that $\frac{1}{4}$ of all mass in the system will be in molecules of type [A 4 B 4](#), and $\frac{3}{4}$ in molecules of type [S 1](#).

Fraction: Displays the mass fraction of this type of molecule in the filler.


Note: The *Fraction* is calculated from the *relative amounts*, such that the sum of all fractions equals 1. Fractions cannot be edited, but will update if a change to any of the relative amounts is made.

Density: The target bead mass per unit volume. Default = [1 g/cm³](#).

Note: Using the default values for bead type mass (100 amu) and bead type diameter (8 Å), a density of [1 g/cm³](#) corresponds to 3 beads per cubic diameter, which is the reduced density most often used in DPD type calculations.

Note: It is possible to specify or move child formers in a mesostructure template such that they extend beyond the boundaries of the simulation cell, as defined by the system former. Although the Materials Visualizer will render and display regions that fall outside the simulation cell in order to help you visualize the system, you should note that when the template is packed with mesoscale molecules, only the volume enclosed by the system former will be filled.

Access methods

Menu	Build Build Mesostructure Mesostructure Components
Toolbar	 / Components

Options tab

The *Options* tab allows you to select the content of the output document and details of the molecule conformations.

Content: Select the type of object that will be created. Available options are [Molecules](#) (default), [Molecules + Fields](#), and [Fields](#).

Field resolution: If fields are output from the build process, their resolution can be set. The resolution is the distance between two grid points in Å. Default = [16 Å](#).

Randomize conformations: Allows the conformations of the mesoscale molecules to be random or ordered when they are added to the template.


Units: Specify the type of units to use, options are [Physical](#) and [Reduced](#). Physical units are the usual units based on Å and atomic mass units. For the [Reduced](#) option the unit of length is normally taken as the interaction cutoff radius, r_c , and the unit of mass as the mass of one bead.

More...: Opens the [Reduced Units](#) dialog, this is only available when the *Units* are set to [Reduced](#).

More...: Opens the [Bead Packing Options](#) dialog.

Note: The length scale of the output bead structure is the same as the input document containing the formers. Both length scales are in Å. The output is suitable for Mesocite simulations.

Access methods

Menu	<i>Build Build Mesostructure Options</i>
Toolbar	 / Options

Bead Packing Options dialog

The *Bead Packing Options* dialog allows you to define, select, and remove tags on selected beads to control the way those beads are packed in the template. Beads defined as heads will be packed on the surface of formers, if those surfaces are enabled for surface packing. By defining tail beads the orientation of the molecule can be controlled. Molecules are packed onto the surface such that the head to tail direction is normal to the surface. To use this dialog you must open a document containing a mesomolecule, and select the beads that are to be tagged as head or tail.

Bead tag: Defines whether the bead is a [Head](#) or a [Tail](#).

Create: Adds a [Head](#) or a [Tail](#) tag to the selected bead.


Select: Selects all the [Head](#) or [Tail](#) tags in the active document.

Delete: Removes the [Head](#) or [Tail](#) tags from the active document.

Note: The tag sets are only used to control surface packing and will not be present in the output document.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build Build Mesostructure Mesostructure Packing More...</i>
Toolbar	 / Packing More...

Reduced Units dialog

The Reduced Units dialog allows you to set up the unit scaling to be used for building mesostructures when reduced units are used.


Length scale: The unit of length in the reduced unit system. This should normally be equal to the interaction cutoff radius r_c , which is comparable to the diameter of a bead. Default = 8 Å.

Mass scale: The unit of mass in the reduced unit system. This should normally be equal to the mass of a bead. Default = 100 amu.

Note: These options are only available if *Units* is set to [Reduced](#) on the [Options](#) tab of the Build Mesostructure dialog.

Help: Displays the Help topic in a browser.

Access methods

Menu	<i>Build / Build Mesostructure / Options / More...</i>
Toolbar	 / <i>Options / More...</i>

Crystal Builder

Crystals are of particular importance for Materials modeling. Many materials of interest are crystalline or polycrystalline. For modeling purposes, this is very advantageous: even though the macroscopic dimensions of a crystalline material may be extremely large compared to interatomic distances, it is usually sufficient to consider the contents of only a single repeat unit (unit cell) of the material for the simulation - its interactions with the virtually infinite number of equivalent cells can then be described with efficient mathematical techniques.

There are, of course, many important materials that are not crystalline - liquids, amorphous systems, etc. For such systems, it is often still advantageous to introduce artificial so-called periodic boundary conditions: a box of the material is periodically repeated in two or three space dimensions. Effectively, this creates an artificial 'supercrystal' of the material.

Why is such a supercrystal more appropriate for treating amorphous/liquid systems than just treating a small finite section of the system? Suppose simulation technique limit simulations to systems with no more than, say, 1000 molecules in it. If these molecules are placed into a box, 1000 molecules corresponds to a fairly small box size of, say, $50 \times 50 \times 50 \text{ \AA}^3$. In a simulation, boundary effects at the surface of this box would need to be considered, and it might be difficult to extract bulk properties from such a simulation for comparison with experiment. If, instead, this box is periodically and infinitely repeated in all three space dimensions, such boundary effects are removed, and the simulation provides a much better approximation of bulk effects. The Amorphous Cell module, for example, works on such supercells.

Another important aspect of crystalline materials is that they often exhibit symmetry beyond pure translational periodicity. Certain translations, reflections, and rotations often map the atomic arrangement onto itself (see the Symmetry section). The Materials Visualizer automatically retains symmetry relationships between objects and ensures the correct symmetrization of properties such as atomic charge, element type, etc. (see Symmetry in Materials Studio).

Crystals are important when performing atomistic simulations of large or infinite systems. The Materials Visualizer provides a broad range of support for the description of crystals. For example, you have control over which subset among the infinite number of atoms in a crystal are visible at any given time. You can also change the symmetry or periodicity of crystals, and you can [build a crystal](#) from its atomic constituents. Finally, it is also possible to [remove the periodicity](#) from a crystalline model and to look at a part of the crystal in isolation.

Tasks in Crystal Builder

Materials Studio offers a number of features for building, manipulating and viewing crystal structures. The following sections discuss how to use the crystal tools in the Materials Visualizer in more detail.

Building and rebuilding crystals

Materials Studio provides a number of tools which allow you to construct a crystal from its constituent parts and modify existing crystal structures. This functionality is essential for a variety of tasks:

1. You may wish to perform a simulation on a crystal system which is only available in a format which cannot be imported directly into Materials Studio, for example hard copy. You can do this by entering the appropriate [space group](#), [specifying the lattice parameters](#), and then [adding atoms](#) to the structure.
2. Alternatively, you may wish to modify an existing crystal structure in preparation for a particular simulation. For example, by adding solvent molecules to polymer crystals or dopant to a zeolite structure.
3. You may wish to perform quantum mechanical modeling of a surface. Many of the tools used in quantum mechanical modeling are unable to handle 2D periodic structures. In such situations it may be necessary to make the surface into a crystal by adding a region of vacuum (which removes surface surface interactions) and repeating the surface units periodically.
4. You might also wish to isolate parts of a crystal structure by [removing its periodicity](#).
5. Finally, for many simulations it is necessary to convert a given crystal structure to a lower symmetry description. For example, in a molecular dynamics simulation of a crystal the atoms will move away from their equilibrium positions and the intermediate positions will not satisfy the symmetry constraints of the space group.

Since Materials Studio strictly imposes symmetry relationships, it is necessary to [create a supercell](#) or [convert to P1 symmetry](#) before attempting such a simulation.

In Materials Studio, a crystal is defined by the following:

- The lattice, which defines the periodicity of the crystal.
- A set of original objects, for example atoms and bonds. This set of objects must contain the atoms for at least one asymmetric unit but it may contain as many atoms as you wish.
- A set of symmetry operations, which together with the lattice vectors specify how any object in the infinite crystal is related to the original objects. If any of the original objects are related to each other by symmetry or periodicity, duplicates are removed.

Note: The original objects are the objects visible before you build the crystal using the [Build Crystal](#) dialog. To view the original objects for any given crystal, simply [unbuild](#) the crystal.

Building a crystal

To build a crystal structure from scratch

1. Create an empty 3D model document.
2. Select the [space group](#).
3. Specify the [lattice parameters](#) a , b , c , α , β and γ .
4. Build the crystal lattice using the *Build* button on the [Build Crystal](#) dialog.
5. [Add](#) asymmetric unit atoms at the appropriate positions. Typically, atom positions are supplied in fractional coordinates.

Additionally, you can set the following options using the [Options](#) tab on the [Build Crystal](#) dialog:

- the tolerance used to test for proximity to special positions.
- whether or not to automatically calculate bonds after building the crystal.
- how the lattice is aligned with regard to the global Cartesian axes.

To add atoms or molecular fragments to an existing crystal structure, either

[Add](#) atoms to the crystal structure one by one

or

1. Create an empty 3D model document.
2. Sketch or import the molecular fragment you wish to add.
3. Select and copy the fragment.
4. Open the document containing the crystal structure you want to modify.
5. Paste the fragment into the crystal structure.
6. One symmetry copy of the fragment will be selected. If you accidentally deselect the fragment, you can select it again with the Select Fragment option in the shortcut menu or by using the Undo operation.
7. Move the fragment to the desired position in the crystal lattice using the mouse.

To build a crystal from molecular fragments, either

1. Create an empty 3D model document.
2. Open the [Build Crystal](#) dialog.
3. Select the [space group](#).
4. Specify the [lattice parameters](#) a , b , c , α , β and γ .
5. Click the *Build* button.
6. Add molecular fragments as described above.

or

1. Create an empty 3D model document.
2. Sketch or import the molecular fragment with which you wish to build the crystal.
3. Open the [Build Crystal](#) dialog.
4. Select the [space group](#).
5. Specify the [lattice parameters](#).
6. Click the *Build* button.
7. Select the required fragments using the Select Fragment option in the shortcut menu.
8. Move the fragments to the desired positions in the crystal lattice using the mouse.

To build a crystal from a surface

1. Import the surface with which you wish to build the crystal.

Note: You can create surfaces in Materials Studio either by building them from their constituent parts using the [surface builder](#) or by [cleaving](#) them from an existing crystal structure.

2. Open the [Build Vacuum Slab Crystal](#) dialog.
3. Specify the *Vacuum orientation*, the *Vacuum thickness* and the surface *Slab position* for the crystal on the [Vacuum Slab](#) tab.
4. Click the *Build* button.

Additionally, you can set the following options using the [Options](#) tab on the Build Vacuum Slab Crystal dialog:

- whether and how the lattice is aligned with regard to the global Cartesian axes
- whether or not any plane group symmetry present in the surface is translated into space group symmetry in the crystal

To build a layered crystal

1. Select *Build* / *Build Layers* from the menu bar to open the [Build Layers](#) dialog.
2. On the [Define Layers](#) tab, specify a *Source document* for each layer that you want to include in the crystal, using the document chooser.

Note: Only *periodic* 3D model documents (crystals or surfaces) can be used as *Source documents*. If you select a nonperiodic 3D model document from the document chooser an error message will be generated.

3. If desired, supply a new *Set name* for each layer.
4. Check the *Build layered structure as a crystal* option.
5. Click the *Build* button to create a new crystal structure from the layers you specified.

Additionally, you can set the following options using the [Layer details](#), [Matching](#), and [Options](#) tabs on the [Build Layers](#) dialog:

- The thickness of the *Vacuum* region between the layers.
- The cleave rule that is used to extract a given layer from the corresponding source crystal.
- The relative orientation of a given layer in the new layered crystal.
- The fractional offset of a given layer in the **a** and **b** directions of the new layered crystal lattice.
- The lattice parameters of the new layered crystal.
- How the lattice parameters of crystal layers are adjusted during the *Build* operation.
- The orientation of the new layered crystal lattice.
- The maximum percentage by which the lattice parameters of the layers are allowed to vary from the chosen values.
- Whether the relevant layer parameters should be set to values suitable for the *Confined Shear* task in Forcite.

Rebuilding a crystal

If necessary you can remove symmetry from a structure and rebuild a crystal with different lattice parameters and different space group settings. Rebuilding a crystal is a fairly uncommon operation and is necessary only for simulations which have to be carried out with a variety of different space group settings.

Note: When rebuild is used information about the current crystal is lost. Therefore it should only be used when the current space group is incorrect.

Tip: To modify the lattice parameters for the current crystal without changing the space group, use the Lattice Parameters dialog accessible from the Symmetry toolbar.

Rebuilding a crystal involves removing all symmetry from the structure, leaving just the [original objects](#), then rebuilding it by repeating the build process.

To rebuild a crystal

1. Open the [Rebuild Crystal](#) dialog.
2. Select the new space group from the [Space Group](#) tab.
3. If necessary, specify new lattice parameters on the [Lattice Parameters](#) tab.
4. Click the *Rebuild* button.

Tip: If you use the *Apply* button to build a crystal, the [Build Crystal](#) dialog remains open and is renamed *Rebuild Crystal*. The *Build* button is also renamed *Rebuild*.

Selecting a space group

Space groups represent symmetry in crystal structures. There are 230 different space groups, although each one can be described in a number of different ways (settings) according to the particular choice of lattice.

Before [building or rebuilding a crystal](#), you must choose the space group to be used.

To select a space group

1. Open the [Build Crystal](#) dialog. If the current document is a crystal, the dialog will be called *Rebuild Crystal*.
2. Select the [Space Group](#) tab and enter the name of the space group (in either Hermann Mauguin or Schoenflies notation), the space group number from International Tables or choose from the list of numbers and short names in the *Enter group* dropdown list. You can use *List groups* to filter the group list by crystal system.
3. Select the required space group setting from the *Option* dropdown list. If you select a new space group, the setting which matches the previous space group as closely as possible is selected. Otherwise, the standard setting for the specified space group is given.
4. Review the *Space group information* and the list of *Operators* for the space group you have chosen.

Tip: To view symmetry and space group information for the current structure without making changes use the Show Symmetry dialog accessible from the Symmetry toolbar.

Specifying the crystal lattice

Lattice parameters determine the length and relative orientation of the lattice vectors in space. In crystals, these lattice vectors determine how each atom is repeated periodically. Any translation of an atom by an integer multiple of a lattice vector or vectors results in that atom reaching the position of another atom in the crystal.

Before [building or rebuilding a crystal](#), you must specify the lattice parameters to be used.

To specify the lattice parameters

1. Open the [Build Crystal](#) dialog. If the current document is a crystal, the dialog will be called *Rebuild Crystal*.
2. Select the [Lattice Parameters](#) tab.
3. Specify the unit cell *Lengths*: a , b and c , either by typing values in the text boxes or by using the spin controls.
4. Specify the unit cell *Angles*: α , β and γ , either by typing values in the text boxes or by using the spin controls.
5. If the space group has already been set and the lattice parameters are constrained to certain values by the specified space group symmetry, these constraints are shown and in most cases the appropriate text boxes are disabled.

Note: If the lattice parameters are set and subsequently a different space group is selected, the constraints may change. In such cases you should always check the lattice parameters before building or rebuilding the crystal.

Note: When rebuild is used information about the current crystal is lost. Therefore it should only be used when the current space group is incorrect.

Tip: To modify the lattice parameters for the current crystal without changing the space group, use the Lattice Parameters dialog accessible from the Symmetry toolbar.


Adding atoms to crystals

In certain circumstances you may wish to add atoms to a crystal:

1. You may wish to perform a simulation on a crystal system which is only available in a format which cannot be imported directly into Materials Studio, for example hard copy. You can do this by entering the appropriate [space group](#), [specifying the lattice parameters](#), and then adding atoms to the structure.
2. Alternatively, you may wish to modify an existing crystal structure in preparation for a particular simulation. For example, by adding solvent molecules to polymer crystals or dopant to a zeolite structure.
3. You can also use the *Add Atoms* option to build nonperiodic structures.

To add an atom to a crystal



1. Choose *Build | Add Atoms* from the menu bar, or click the *Add Atoms* button on the *Atoms & Bonds* toolbar to display the [Add Atoms](#) dialog.
2. Select the element type of the atom to be added from the list of common and recently used elements on the [Atoms](#) tab or click the  button to open the Periodic Table dialog.
3. Select the *Coordinate system* on the [Options](#) tab. Your selection also determines which labels appear on the coordinate text boxes on the *Atoms* tab. If you choose [Fractional](#), the labels a, b, and c are used. If you choose [Cartesian](#), x, y, and z are used.

Note: If the current system is nonperiodic, the *Coordinate system* list is disabled and Cartesian values are used.

4. On the [Atoms](#) tab, provide a *Name* for the atom to be added. Alternatively, leave the text box blank and a default name will be used.
5. Enter coordinate values for the atom to be added.
6. If necessary, alter the default values for the Oxidation State and Occupancy of the new atom.
7. Select the type of Temperature Factors to be used.
8. Click the *Add* button to add the new atom.

To enable calculation of bonds and to test for proximity

1. Check *Test for bonds as atoms are created* on the [Options](#) tab. When you click the *Add* button bonds will be created to other atoms in the structure, if the criteria defined in the Bond Calculation dialog are met.
2. Check *Warn if atoms are closer than*. When you click the *Add* button a proximity test is performed. Before each atom is added its position relative to other atoms in the structure is determined. If any other atoms in the structure, or any symmetry copies of the new atom are found to be closer than the specified value, a warning is displayed. You can then cancel the *Add* action.

Adding atoms to structures

Atom positioning is particularly useful for entering published crystal structures or inorganic structures, as well as adding ions to an existing structure. It may also be used in conjunction with nonperiodic structures.

You can build structures by adding atoms to the current model one at a time at specified coordinate positions. Atom positions can be described using Cartesian coordinates, surface fractional coordinates, or crystal fractional coordinates.

Several atom properties can be specified including oxidation state, occupancy and atomic temperature factors. There are also options to add bonding using the current bonding criteria, and to test for proximity to other atoms and symmetry copies of the new atom.

To add an atom

1. Choose *Build | Add Atoms* from the menu bar to open the Add Atoms dialog.
2. On the *Atoms* tab select an *Element* from the list common and recently used elements.
3. On the *Options* tab select a *Coordinate system*. This selection also sets the labels of the coordinate text boxes on the *Atoms* tab. If you choose Fractional, the labels are set to A, B, and C (crystal) or U, V, W (surface). Cartesian (labels X, Y, Z) is the other possible value. If the system in scope is nonperiodic the list is disabled and the labels are set to Cartesian values.
4. *Name* the atom to be added and enter coordinate values and other properties.
5. Click the *Add* button to add the new atom

To add bonds and test for proximity

1. On the *Options* tab check *Test for bonds as atoms are created*. When you click the *Add* button the new atom will be bonded to other atoms in the model if the criteria defined in bonding preferences module are met.
2. Check *Warn if atoms are closer than*. When you click the *Add* button the value in the adjacent text box is used to test for proximity is made before each atom is added. If any other atoms in the model, or any symmetry copies of the new atom are found to be too close, a warning is displayed, and you can cancel the Add action. If the chosen position corresponds to a special position of the crystal, the other symmetry operations which give the same special position are ignored.

Building crystals from surfaces

The [Build Vacuum Slab Crystal](#) dialog enables you to build a crystal from a surface. The crystal is constructed by repeating the surface in a given direction using a repeat distance which is greater than the surface thickness. This introduces a region of vacuum between the surface units.

To build a crystal from a surface

1. Create an empty 3D model document.
2. Import the surface with which you wish to build the crystal.

Note: You can create surfaces in Materials Studio either by building them from their constituent parts using the [surface builder](#) or by [cleaving](#) them from an existing crystal structure.

3. Open the [Build Vacuum Slab Crystal](#) dialog.
4. Specify the *Vacuum orientation*, the *Vacuum thickness* and the surface *Slab position* for the crystal on the [Vacuum Slab](#) tab.
5. Click the *Build* button.

Additionally, you can set the following options using the [Options](#) tab on the [Build Vacuum Slab Crystal](#) dialog:

- whether and how the lattice is aligned with regard to the global Cartesian axes
- whether or not any plane group symmetry present in the surface is translated into space group symmetry in the crystal

Building a layered crystal

The [Build Layers](#) dialog enables you to build a layered crystal from two or three source crystals or surfaces.

To build a layered crystal

1. Select *Build* / *Build Layers* from the menu bar to open the [Build Layers](#) dialog.
2. On the [Define Layers](#) tab, specify a *Source document* for each layer that you want to include in the crystal, using the document chooser.

Note: Only *periodic* 3D model documents (crystals or surfaces) can be used as *Source documents*. If you select a nonperiodic 3D model document from the document chooser an error message will be generated.

3. If desired, supply a new *Set name* for each layer.
4. Check the *Build layered structure as a crystal* option.
5. Click the *Build* button to create a new crystal structure from the layers you specified.

Additionally, you can set the following options using the [Layer details](#), [Matching](#) and [Options](#) tabs on the [Build Layers](#) dialog:

- The thickness of the *Vacuum* region between the layers.
- The cleave rule that is used to extract a given layer from the corresponding source crystal.
- The relative orientation of a given layer in the new layered crystal.
- The fractional offset of a given layer in the **a** and **b** directions of the new layered crystal lattice.
- The lattice parameters of the new layered crystal.
- How the lattice parameters of crystal layers are adjusted during the *Build* operation.
- The orientation of the new layered crystal lattice.
- The maximum percentage by which the lattice parameters of the layers are allowed to vary from the chosen values.
- Whether the relevant layer parameters should be set to values suitable for the *Confined Shear* task in Forcite.

The layered structure is built up with layer 1 at the bottom, then layer 2 and (if it is present) layer 3 at the top. The layers are aligned **a** with **a** and **b** with **b** and stacked along **c**.

Tip: If you want to align the axes of the layers differently, use the Redefine Lattice dialog to transform the layer lattices before building the layered structure.

The length of the **c** lattice vector is determined by adding together the **c** lattice vectors (or the surface thicknesses) of each layer (after they have been adjusted) and the *Vacuum* regions for each layer.

Note: When the layered structure is built, all symmetry relationships within the layer unit cells are removed. As a result the new crystal lattice will have P1 symmetry.

Tip: Any symmetry present in the layered crystal structure can be reimposed using the Find Symmetry tool.

Rebuilding a crystal

Rebuilding a crystal first of all involves removing all symmetry from the structure, leaving just those objects that formed the asymmetric unit, as well as any that were marked as original objects in the original build.

Once you have removed symmetry you can rebuild it by repeating the build process.

Warning. The rebuild action loses much information about the current structure and should only be used when a structure currently has an incorrect space group specification.

To rebuild a crystal

1. Choose *Build | Crystal | Rebuild Crystal* from the menu bar to open the Rebuild Crystal dialog
2. On the *Space Groups* tab enter the new space group values.
3. On the *Lattice Parameters* tab make any necessary changes
4. Click the *Rebuild* button

Tip: If you use the *Apply* button to build a crystal, the dialog remains open and the label on the *Build* button changes to *Rebuild*.

Removing crystal periodicity

Materials Studio allows you to interact easily with infinite crystals. You can perform almost all operations directly on the infinite lattice and you can control which parts of the structure are visible.

As a result, it is not usually necessary to remove the crystal periodicity to perform common operations such as editing and structural modifications. However, in certain situations it may be desirable to isolate parts of a crystal structure by removing periodicity. For example, you may wish to construct a finite cluster of a crystalline material, or you may wish to compare the energy of an isolated molecule with the energy of the molecular crystal.

There are two ways in which you can remove symmetry from a crystal:

- By cutting a finite section, or a [nonperiodic superstructure](#), out of the lattice.
- By [unbuilding](#) the crystal, leaving the [original objects](#) that were used to build it.

Making a nonperiodic superstructure

The *Nonperiodic Superstructure* command allows you to generate a finite cluster from an infinite crystal or surface. This command is useful if, for example, the simulation technique you wish to use cannot handle periodicity, or if you are interested in the interactions between two finite particles.

Note: The *Nonperiodic Superstructure* command is enabled only if the current structure is a crystal or a surface.

To make a nonperiodic superstructure

1. Select *Build | Symmetry | Nonperiodic Superstructure* from the menu bar.
2. The command removes all symmetry and periodicity relationships within the structure, and deletes all atoms except those that are currently visible.
3. Other objects, such as bonds, molecules and monitors will persist, if the atoms to which they refer remain.

Unbuilding a crystal

The *Unbuild Crystal* command allows you to return to the [original objects](#) from which a crystal was built. This command is useful if, for example, you wish to compare the energy of an asymmetric unit molecule with the energy of the molecular crystal.

Note: The *Unbuild Crystal* command is enabled only if the current structure is a crystal.

To unbuild a crystal

1. Select *Build | Symmetry | Unbuild Crystal* from the menu bar.
2. The command removes all symmetry and periodicity within the structure, and deletes all objects except those that formed the asymmetric unit, along with any others that were the [original objects](#) used to build the crystal.
3. A record of the current symmetry definition is attached to the unbuild crystal, so that you can [rebuild](#) it later, with the same parameters.

Redefining crystal symmetry

For many simulations it is necessary to convert a given crystal structure to a lower symmetry description. For example, in a molecular dynamics simulation of a crystal the atoms will move away from their equilibrium positions and the intermediate positions will not satisfy the symmetry constraints of the space group.

Since Materials Studio strictly imposes symmetry relationships, it is necessary to [create a supercell](#) or [convert to P1 symmetry](#) before attempting such a simulation.

It may also be useful to define a supercell or reduce the symmetry in systems with interstitial or substitutional disorder such as doped semiconductors, incommensurate structures under lattice distortions or magnetic structures such as antiferromagnets.

For other, computationally demanding, simulations it is desirable to use the smallest possible unit cell, while retaining all of the symmetry of the crystal lattice.

If the space group of the crystal includes a centering operation (A-, B-, C-, Face-, Body- or Rhombohedral centering), the lattice can be redefined such that a smaller unit cell, without centering is generated. These are known as primitive unit cells. Primitive unit cells usually have less convenient shapes.

The process of converting a centered cell into a primitive cell is known as [converting to a primitive representation](#). The inverse operation, of converting a primitive unit cell back into a centered cell, is known as [converting to a conventional representation](#).

Unlike the *Create Supercell* and *Make P1* operations, conversion to a primitive representation and back to a conventional representation does not result in the loss of any symmetry information.

Creating a crystal supercell

A crystal supercell is obtained if, instead of defining the unit cell as the smallest possible repeat unit in the crystal, a larger repeat unit is used. You can specify the structure of a supercell by defining its range in unit cells. The new structure will have P1 symmetry and lattice vectors that are multiples of the original unit cell vectors. The concept of a supercell is essential for studying lattice vibrations and to build up structures with substitutional, interstitial or magnetic disorder.

When a supercell is created, no objects are destroyed. However, the symmetry relationships between objects are redefined. As a result, certain atoms which were previously related by translational symmetry are no longer symmetry related and can move independently.

To create a crystal supercell

1. Select *Build | Symmetry | Supercell* from the menu bar to open the Supercell dialog.
2. Enter values for the *Supercell range* in the **a**, **b** and **c** directions.
3. Create the crystal supercell using the *Create Supercell* button.

Redefining a crystal lattice

To redefine a crystal lattice

1. Select *Build | Symmetry | Redefine Lattice* from the menu bar to open the Redefine Lattice dialog.
2. On the New lattice tab, specify the new lattice vectors **a**, **b** and **c** in terms of the existing ones. For example, to transpose the **a** and **b** axes type: **a: 0 1 0, b: -1 0 0, c: 0 0 1**.

Note: It is necessary to invert one of the axes in order to generate a right-handed set of lattice vectors.

3. If required, check the *Reorient structure after redefinition* checkbox on the Options tab to reorient the new lattice so that it matches the *Orientation standard* of the original.
4. Click the *Redefine* button.

Checks are made on the new lattice vectors to ensure that they:

- are lattice vectors of the current lattice
- are independent of each other
- form a right-handed set

If the lattice vectors you specified are acceptable, the lattice is redefined. If the new lattice vectors fail any of the checks, the *Redefine* operation is canceled and you are prompted to specify alternative vectors.

Note: Redefining the lattice removes all symmetry relationships within the unit cell. As a result the new structure will have P1 symmetry.

Converting a crystal to P1 symmetry

Crystals with P1 symmetry only possess translational periodicity.

For many simulations, it is necessary to convert a higher symmetry crystal structure to P1 symmetry. For example, during a molecular dynamics simulation when the atoms will move away from their equilibrium positions and the intermediate positions will not satisfy the symmetry constraints of the higher symmetry space group.

The *Make P1* command removes all symmetry from the current structure and maintains only its translational periodicity.

Note: The *Make P1* command is enabled only if the current structure is a crystal with greater than P1 symmetry.

To convert to P1 symmetry

1. Select *Build | Symmetry | Make P1* from the menu bar.
2. The command removes all symmetry relationships present in the crystal lattice but maintains the periodic relationships.
3. No objects are destroyed by *Make P1* and any nonsymmetric or nonperiodic object properties are retained.

Converting a crystal to primitive representation

For certain, computationally demanding, simulations it is desirable to use the smallest possible unit cell, while retaining all of the symmetry of the crystal lattice.

To do this it is necessary to redefine the crystal symmetry in terms of a primitive lattice. A primitive lattice is one which has only a single lattice point in each unit cell.

The *Primitive Cell* command converts the current crystal lattice into its primitive representation.

Note: The *Primitive Cell* command is enabled only if the current structure is a crystal with a nonprimitive lattice.

Note: If the current crystal is trigonal or hexagonal, this command is renamed [Rhombohedral Representation](#).

To convert a crystal to primitive representation

1. Select *Build | Symmetry | Primitive Cell* from the menu bar.
2. The command replaces the current lattice vectors with the vectors of the primitive form of the lattice and changes the space group description accordingly.
3. All of the objects in the crystal are redefined according to the new symmetry.
4. No objects are destroyed by *Primitive cell* and all object properties are retained.
5. In chemical terms, the redefined structure is identical to the initial structure.

Converting a crystal to conventional representation

For certain, computationally demanding, simulations it is desirable to [redefine crystal symmetry](#) in terms of a primitive lattice.

However, in most cases it is simpler to work with crystal structures that have symmetry defined in terms of a conventional lattice. The conventional representation of a crystal structure often includes a centering operation and usually has a simpler unit cell shape.

The *Conventional Cell* command converts a primitive crystal lattice into its conventional representation.

Note: The *Conventional Cell* command is enabled only if the current structure is a crystal with a nonconventional primitive lattice.

Note: If the current crystal is trigonal or hexagonal, this command is renamed [Hexagonal Representation](#).

Since the same primitive lattice can usually be converted into many different settings of a conventional lattice, the *Conventional Cell* command often involves a choice about the conventional setting to use. Typically, *Conventional Cell* will convert a primitive lattice into the standard setting of the space group to which the crystal belongs. However, if the primitive lattice was obtained from a nonstandard setting of a conventional lattice using the *Primitive Cell* command, *Conventional Cell* will convert it back into the original nonstandard setting.

Converting a crystal to conventional representation

1. Select *Build | Symmetry | Conventional Cell* from the menu bar.
2. The command replaces the current lattice vectors with the vectors of the appropriate conventional setting of the lattice (as described above) and changes the space group description accordingly.
3. All of the objects in the crystal are redefined according to the new symmetry.
4. No objects are destroyed by *Conventional Cell* and all object properties are retained.
5. In chemical terms, the redefined structure is identical to the initial structure.

Converting a crystal to rhombohedral representation

The *Rhombohedral Representation* command is a special form of the more general *Primitive Cell* command, that applies only to crystals in trigonal or hexagonal space groups.

The functionality is the same as the [conversion to primitive representation](#), except that in trigonal and hexagonal systems the primitive form of the space group corresponds to the rhombohedral setting, as given in International Tables.

Note: The *Rhombohedral Representation* command is enabled only if the current structure is a trigonal or hexagonal crystal with a nonrhombohedral lattice.

To convert a crystal to rhombohedral representation

1. Select *Build | Symmetry | Rhombohedral Representation* from the menu bar.
2. The command replaces the current lattice vectors with the vectors of the rhombohedral form of the lattice and changes the space group description accordingly.
3. All of the objects in the crystal are redefined according to the new symmetry.
4. No objects are destroyed by *Rhombohedral Representation* and all object properties are retained.
5. In chemical terms, the redefined structure is identical to the initial structure.

Converting a crystal to hexagonal representation

The *Hexagonal Representation* command is a special form of the more general *Conventional Cell* command, that applies only to crystals in trigonal or hexagonal space groups.

The functionality is the same as the [conversion to conventional representation](#), except that in trigonal and hexagonal systems the conventional form of the space group corresponds to the hexagonal setting, as given in International Tables.

Note: The *Hexagonal Representation* command is enabled only if the current structure is a trigonal or hexagonal crystal with a nonconventional rhombohedral lattice.

To convert a crystal to hexagonal representation

1. Select *Build | Symmetry | Hexagonal Representation* from the menu bar.
2. The command replaces the current lattice vectors with the vectors of the appropriate hexagonal setting of the lattice and changes the space group description accordingly.
3. All of the objects in the crystal are redefined according to the new symmetry.
4. No objects are destroyed by *Hexagonal Representation* and all object properties are retained.
5. In chemical terms, the redefined structure is identical to the initial structure.

Dialogs in Crystal Builder

Most crystal building and modification tasks can be performed using dialogs which can be accessed from the Build menu.

Build Crystal dialog

The Build Crystal dialog allows you to [construct a crystal](#) from its constituent parts. It also allows you to specify the symmetry and lattice parameters of the crystal.

Note: If the current 3D model document is a crystal or a structure on which an [Unbuild](#) operation has been performed, this dialog is named Rebuild Crystal. The Rebuild Crystal dialog can be used to modify the symmetry and lattice parameters of an existing crystal or to rebuild a crystal structure following an *Unbuild* operation.

When the Build Crystal dialog is open, you will not be able to work with other parts of Materials Studio. The Build Crystal dialog contains the following tabs:

- [Space Group](#): Allows you to specify the space group used to build or rebuild the crystal.
- [Lattice Parameters](#): Allows you to specify values of the lattice parameters for the crystal you wish to build.
- [Options](#): Allows you to set build options and specify the orientation standard for the crystal you wish to build.

Build: Creates a crystal from the current active 3D model, using the parameters specified in the [Space Group](#) and [Lattice Parameters](#) tabs and closes the dialog.

Note: If the current active 3D model is a crystal, the button is labeled *Rebuild*.

Rebuild: Unbuilds the current crystal, then creates a new crystal using the parameters specified in the [Space Group](#) and [Lattice Parameters](#) tabs and closes the dialog.

Build or Rebuild | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Note: When rebuild is used information about the current crystal is lost. Therefore it should only be used when the space group is incorrect.

Tip: To modify the lattice parameters for the current crystal without changing the space group, use the Lattice Parameters dialog accessible from the Symmetry toolbar.

Apply: Creates (or recreates) a crystal from the current active 3D model, using the parameters specified in the [Space Group](#) and [Lattice Parameters](#) tabs without closing the dialog.

Note: If the current active 3D model is not a crystal, after *Apply* is used the *Build Crystal* dialog is renamed *Rebuild Crystal* and the *Build* button is renamed *Rebuild*.

Cancel: Closes the dialog without making any changes.

Help: Displays the Help topic for the current tab.

Access methods

Menu	<i>Build Crystals Build/Rebuild Crystal...</i>
------	--

Space Group tab

Note: The *Space Group* tab may appear in a dialog named either *Build Crystal* or *Rebuild Crystal* depending on whether or not the crystal has already been built.

The *Space Group* tab allows you to specify the space group used to build or rebuild the crystal. Initially, the space group of the current 3D model document is displayed. If no space group is defined for the current 3D model, default values are shown.

Enter group: Specify the space group to be used.

Type the required space group symbol or the corresponding space group number from International Tables in the text box. Alternatively, choose the required space group from the dropdown list.

Note: The space groups listed on the *Enter group* dropdown vary depending on which *List groups* option is selected.

List groups: Specify which space groups should be listed on the *Enter group* dropdown.

Select one of the seven crystal systems; [Triclinic](#), [Monoclinic](#), [Orthorhombic](#), [Tetragonal](#), [Trigonal](#), [Hexagonal](#) or [Cubic](#) to restrict the list to space groups from that crystal system.

Alternatively, select [All](#) to list space groups from all crystal systems.

Option: Select the required space group settings from the dropdown list. The standard settings for the current space group are used by default.

Space group information: Displays general information about the selected space group.

Operators: Displays brief details of the symmetry operators for the selected space group.

Details...: Provides access to the Show Symmetry dialog which contains further details about the symmetry operators for the selected space group.

Note: When opened in this way, the Show Symmetry dialog closes when the *Build Crystal* dialog is closed.

Access methods

Menu	<i>Build Crystals Build/Rebuild Crystal... Space Group</i>
------	--

Lattice Parameters tab

Note: The *Lattice Parameters* tab may appear in a dialog named either *Build Crystal* or *Rebuild Crystal* depending on whether or not a crystal has already been built.

Lattice parameters determine the length and relative orientation of the lattice vectors in space. In crystals and surfaces, these lattice vectors determine how each atom is repeated periodically. Any translation of an atom by an integer multiple of a lattice vector or vectors results in that atom reaching the position of another atom in the crystal.

The *Lattice Parameters* tab allows you to specify values of the lattice parameters for the crystal you wish to build.

Lattice type: Displays the lattice type of the selected space group.

If the lattice parameters are constrained to certain values by the specified space group symmetry, these constraints are shown and the appropriate text boxes are disabled.

Lengths: Specify the values of the lattice parameters a , b and c , which determine the lengths of the lattice vectors.

The spin controls beside each text box change the current values in increments of 0.2 Å.

Angles: Specify the values of the lattice parameters α , β and γ , which determine the relative orientations of the lattice vectors.

The spin controls beside each text box change the current values in increments of 1.0 degree.

Access methods

Menu *Build | Crystals | Build/Rebuild Crystal... | Lattice Parameters*

Options tab

Note: The *Options* tab may appear in a dialog named either *Build Crystal* or *Rebuild Crystal* depending on whether or not a crystal has already been built.

The *Options* tab allows you to set build options and specify the orientation standard for the crystal you wish to build. Use this tab to:

- specify the tolerance used to test for proximity to special positions
- select whether or not to calculate bonds automatically after building the crystal
- specify how the lattice is aligned with respect to the global Cartesian axes

Check for atoms on special positions: When checked indicates that atoms will be tested for their proximity to any special positions and for overspecification using the tolerance specified in the associated text box.

The tolerance value is the maximum distance allowed between two points for them to be considered as belonging to the same lattice site.

Note: This option is disabled if *Ignore special positions* is checked.

Ignore special positions: When checked indicates that atoms will not be tested for their proximity to special positions or for overspecification. All possible symmetry equivalents will be generated for each atom in the structure.

Calculate bonding after build: When checked indicates that a bond calculation will be performed after the crystal has been built using the values specified on the Bond Calculation dialog.

Orientation standard: Specify the alignment of the crystal lattice with respect to the Cartesian axes and the rule used for subsequent alterations.

Note: The *Cerius²* standard is C along Z , B in YZ plane and the *Insight II* standard is A along X , B in XY plane.

Access methods

Menu *Build | Crystals | Build/Rebuild Crystal... | Options*

Build Vacuum Slab Crystal dialog

The Build Vacuum Slab Crystal dialog enables you to build a crystal from a surface. The crystal is constructed by repeating the surface in a given direction using a repeat distance which is greater than the surface thickness. This introduces a region of vacuum (or slab) between the surface units.

Note: This dialog is enabled only if the currently active 3D model document contains a surface.

The Build Vacuum Slab Crystal dialog contains the following tabs:

- **Vacuum Slab:** Allows you to specify the orientation and position of the surface within the crystal and the thickness of the vacuum region.
- **Options:** Allows you to control how the orientation and space group symmetry of the crystal will be determined.

Build: Creates a slab crystal from the surface in the current active 3D model document using the parameters specified in the *Vacuum Slab* and *Options* tabs.

Build | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Cancel: Closes the dialog without building a vacuum slab.

Help: Displays the Help topic for the current tab.

Access methods

Menu *Build | Crystals | Build Vacuum Slab...*

Vacuum Slab tab

The *Vacuum Slab* tab allows you to specify the orientation and position of the surface within the crystal and the thickness of the vacuum region.

Vacuum orientation: Select the crystal direction in which the vacuum region will be oriented from the dropdown list.

The *u* and *v* axes of the surface are mapped on to the remaining two crystal axes such that they form a right-handed set. The crystal is built in a direction normal to the vacuum region plane.

Vacuum thickness: Specify the thickness of the vacuum region. This is added to the thickness of the surface to give the repeat distance of the crystal or the *Crystal thickness*.

Crystal thickness: Alternatively, specify the repeat distance of the crystal. The thickness of the surface is subtracted from this value to give the *Vacuum thickness*.

Note: The thickness of the surface is determined by finding the maximum distances between the surface atoms and the plane of the surface.

Note: The *Vacuum thickness* and the *Crystal thickness* are related, therefore a change in the value of one will cause a corresponding change in the value of the other.

Slab position: Specify the position of the surface within the crystal.

The default value of **0.0** positions the surface at the bottom of the crystal cell with the vacuum region entirely above it. A positive value less than the *Vacuum thickness* positions the surface within the crystal cell with portions of the vacuum above and below it.

Access methods

Menu *Build | Crystals | Build Vacuum Slab... | Vacuum Slab*

Options tab

The *Options* tab allows you to control how the orientation and space group symmetry of the slab crystal will be determined.

Crystal orientation standard: Specify the rule used to align the crystal with respect to the Cartesian axes.

Note: The *Cerius²* standard is *C* along *Z*, *B* in *YZ* plane and the *Insight II* standard is *A* along *X*, *B* in *XY* plane.

Re-orient structure after update: When checked indicates that the whole crystal will be reoriented to match the *Crystal orientation standard* after it has been built. Default = [unchecked](#).

Transfer surface symmetry to crystal: When checked indicates that any plane group symmetry present in the surface will be translated into space group symmetry in the crystal. Default = [checked](#).

Note: If *Transfer surface symmetry to crystal* is unchecked, no symmetry translation will be performed and the resulting crystal will have P1 symmetry.

Access methods

Menu	Select <i>Build</i> <i>Crystals</i> <i>Build Vacuum Slab...</i> <i>Options</i>
------	--

Add Atoms dialog

The Add Atoms dialog allows you to add atoms to a structure and specify their oxidation state, occupancy, and atomic temperature factors. There are also options to add bonding using the current bonding criteria and to test for proximity to other atoms and symmetry copies of the new atom.


The Add Atoms dialog contains the following tabs:

- [Atoms](#): Allows you to add atoms to a structure.
- [Options](#): Allows you to set additional options for the atom to be added.

Add: Adds the selected atom to the current structure. If the structure is a surface or a crystal, appropriate symmetry and periodic copies will be generated.


Help: Displays the Help topic for the current tab.

Access methods

Menu	<i>Build</i> <i>Add Atoms</i>
Toolbar	

Atoms tab

The *Atoms* tab allows you to add atoms to a structure.

Element: Select the element type of the atom to be added from the dropdown list or use the  button to access the Periodic Table dialog.

Name: Provide a name for the atom to be added.

Oxidation state: Specify the oxidation state of the atom to be added. Default = [0](#).

Occupancy: Specify the occupancy value for the atom to be added. Default = [1.0](#) (normal occupancy).

Temperature factors: Select the type of temperature factor that will be defined for the atom to be added. Available options are:

- [None](#)
- [Isotropic](#)
- [Anisotropic](#)


Specify appropriate values for the temperature factors in the corresponding text boxes, as follows:

Option	Format	Number of values
Isotropic	U, B	1
Anisotropic	Uij, Bij, β_{ij}	6

a, b, c / u, v, w / x, y, z: Specify the position of the new atom in either fractional or Cartesian coordinates.

Note: The labels that appear depend on whether fractional or Cartesian coordinates have been selected from *Coordinate system* on the [Options](#) tab and on whether the system is a crystal, a surface or nonperiodic.

Access methods

Menu	<i>Build Add Atoms Atoms</i>
Toolbar	 / <i>Atoms</i>

Options tab

The *Options* tab allows you to set additional options for the atom to be added.


Test for bonds as atoms are created: When checked, indicates bonds between the new atom and other atoms in the structure will be generated automatically according to the criteria defined in the Bond Calculation dialog.

Warn if atoms are closer than: When checked, indicates that a proximity test will be performed before the new atom is added, using the tolerance value specified. If other atoms in the structure, or symmetry copies of the new atom are found to be within the tolerance, a warning is displayed and you can cancel the *Add* operation.

Note: If the requested coordinate for the new atom is identical to that of an existing atom, an error will be displayed and the operation will fail.

Coordinate system: Specify the coordinate system to be used. The labels for the coordinate text boxes on the [Atoms](#) tab will vary accordingly. Default = [Fractional](#) for crystals and surfaces; [Cartesian](#) for other types of structures.

Access methods

Menu	<i>Build Add Atoms Options</i>
Toolbar	 / <i>Options</i>

Surface Builder

In Materials Studio, the term *surface* denotes any 2D periodic system. Support for surfaces is similar to the support for [crystals](#) - periodicity is built in at a fundamental level and symmetry relationships are strictly enforced.

Surfaces are used to model a variety of systems and physical phenomena. For example, the study of surfaces is crucial in understanding particle shape (morphology) of crystals, and interactions on surfaces play a vital role for catalytic processes.

Tasks in Surface Builder

Materials Studio offers a number of features for building, manipulating and viewing surfaces.

The following sections discuss how to use the surface tools in the Materials Visualizer in more detail.

Building and rebuilding surfaces

Materials Studio provides a number of tools which allow you to construct surfaces from their constituent parts and modify existing surfaces. This functionality is essential for a variety of tasks:

1. You may wish to perform a simulation on a surface system which is only available in a format which cannot be imported directly into Materials Studio, for example hard copy. You can do this by entering the appropriate [plane group](#), [specifying the lattice parameters](#), and then [adding atoms](#) to the structure.
2. You may wish to modify an existing surface in preparation for a particular simulation, for example, by adding solvent molecules to a surface in order to model surface solvent interactions.
3. Alternatively, you may wish to cleave a crystal to expose a particular crystal surface.
4. You might also wish to isolate parts of a surface by [removing its periodicity](#).
5. Finally, for many simulations it is necessary to convert a given surface to a lower symmetry description. For example, in a molecular dynamics simulation of a surface, the atoms will move away from their equilibrium positions and the intermediate positions will not satisfy the symmetry constraints of the plane group. Because Materials Studio strictly imposes symmetry relationships, it is necessary to [create a supercell](#) or [convert to p1 symmetry](#) before attempting such a simulation.

In Materials Studio, a surface is defined by the following:

- The lattice, which defines the periodicity of the surface.
- A set of original objects, for example atoms and bonds. This set of objects must contain the atoms for at least one asymmetric unit but it may contain as many atoms as you wish.
- A set of symmetry operations, which together with the lattice vectors specify how any object in the infinite surface is related to the original objects. If any of the original objects are related to each other by symmetry or periodicity, duplicates are removed.

Note: The original objects are the objects visible before you build the surface using the [Build Surface](#) dialog. To view the original objects for any given surface, simply [unbuild](#) the surface.

Building a surface

To build a surface from scratch

1. Create an empty 3D model document.
2. Open the [Build Surface](#) dialog.
3. Select the [plane group](#).
4. Specify the [lattice parameters](#) u , v and ϑ .
5. Click the *Build* button.
6. [Add](#) asymmetric unit atoms at the appropriate positions. Typically, atom positions are supplied in fractional coordinates.

Additionally, you can set the following options using the [Options](#) tab on the [Build Surface](#) dialog:

- The tolerance used to test for proximity to special positions
- Whether to automatically calculate bonds after building the surface
- How the lattice is aligned with regard to the global Cartesian axes

To add atoms or molecular fragments to an existing surface, either

[Add](#) atoms to the surface one at a time

or

1. Create an empty 3D model document.
2. Sketch or import the molecular fragment you wish to add.
3. Select and copy the fragment.
4. Open the document containing the surface you want to modify.
5. Paste the fragment into the surface.
6. One symmetry copy of the fragment will be selected. If you accidentally deselect the fragment, you can select it again with the Select Fragment option in the shortcut menu or by using the Undo operation.
7. Move the fragment to the desired position in the surface lattice using the mouse.

To build a surface from molecular fragments, either

1. Create an empty 3D model document.
2. Open the [Build Surface](#) dialog.
3. Select the [plane group](#).
4. Specify the [lattice parameters](#) u , v and ϑ .
5. Click the *Build* button.
6. Add molecular fragments as described above.

or

1. Create an empty 3D model document.
2. Sketch or import the molecular fragment with which you wish to build the surface.
3. Open the [Build Surface](#) dialog.
4. Select the [plane group](#).
5. Specify the [lattice parameters](#).
6. Click the *Build* button.
7. Select the required fragments using the Select Fragment option in the shortcut menu.
8. Move the fragments to the desired positions in the surface lattice using the mouse.

To cleave a surface from a crystal

1. Import the source crystal from which you wish to cleave the surface.

Note: You can create crystals in Materials Studio by [building](#) them from their constituent parts using either the [Build Crystal](#) or the [Build Vacuum Slab Crystal](#) dialogs.

2. On the [Cleave Surface](#) dialog, choose the [Surface Box](#) tab and specify the Miller indices of the *Cleave plane*, the *Thickness* and position of the *Top* of the new surface relative to the origin.
3. Select a *Cleave rule* and specify how dangling bonds created by the cleave operation are handled.
4. Click the *Cleave* button.

Additionally, you can set the following options using the [Surface Mesh](#) and [Options](#) tabs on the [Cleave Surface](#) dialog:

- Surface vectors *U* and *V* for the *Cleave plane*. These must be lattice vectors of the crystal being cleaved and they must lie in the *Cleave plane*.

Note: If a *Cleave plane* is specified on the *Surface Box* dialog suitable *Surface vectors* will be generated automatically.

- The origin of the surface lattice in fractional coordinates of the source crystal.
- Whether and how the surface lattice is aligned with regard to the global Cartesian axes.
- Whether and how the current cleave slab is superimposed on the source crystal in the Visualizer.
- Whether the source crystal is automatically recleaved when the cleave parameters are changed.

To build a layered surface or interface

1. Select *Build | Build Layers* from the menu bar to open the [Build Layers](#) dialog.
2. On the [Define Layers](#) tab, specify a *Source document* for each layer that you want to include in the surface, using the document chooser.

Note: Only *periodic* 3D model documents (crystals or surfaces) can be used as *Source documents*. If you select a nonperiodic 3D model document from the document chooser an error message will be generated.

3. If desired, supply a new *Set name* for each layer.
4. Check the *Build layered structure as a surface* option.
5. Click the *Build* button to create a new surface structure from the layers you specified.

Additionally, you can set the following options using the [Layer details](#), [Matching](#) and [Options](#) tabs on the [Build Layers](#) dialog:

- The thickness of the *Vacuum* region between the layers.
- The cleave rule that is used to extract a given layer from the corresponding source crystal.
- The relative orientation of a given layer in the new layered surface.
- The fractional offset of a given layer in the *u* and *v* directions of the new layered surface lattice.
- The lattice parameters of the new layered surface.
- How the lattice parameters of crystal layers are adjusted during the *Build* operation.
- The orientation of the new layered surface lattice.
- The maximum percentage by which the lattice parameters of the layers are allowed to vary from the chosen values.

Rebuilding a surface

If necessary you can remove symmetry from a surface and rebuild it with different lattice parameters and different plane group settings. Rebuilding a surface is a fairly uncommon operation and is necessary only for simulations which have to be carried out with a variety of different plane group settings.

Note: When rebuild is used information about the current surface is lost. Therefore it should only be used when the current plane group is incorrect.

Tip: To modify the lattice parameters for the current surface without changing the plane group, use the Lattice Parameters dialog accessible from the Symmetry toolbar.

Rebuilding a surface involves removing all symmetry from the structure, leaving just the [original objects](#), then rebuilding it by repeating the build process.

To rebuild a surface

1. Open the [Rebuild Surface](#) dialog.
2. Select the new plane group from the [Plane Group](#) tab.
3. If necessary, specify new lattice parameters on the [Lattice Parameters](#) tab.
4. Click the *Rebuild* button.

Tip: If you use the *Apply* button to build a surface, the [Build Surface](#) dialog remains open and is renamed *Rebuild Surface*. The *Build* button is also renamed *Rebuild*.

Selecting a plane group

Plane groups represent symmetry in surfaces. There are 17 different plane groups.

Before [building or rebuilding a surface](#), you must choose the plane group to be used.

To select a plane group

1. Open the [Build Surface](#) dialog. If the current document is a surface, the dialog will be called *Rebuild Surface*.
2. Select the [Plane Group](#) tab and enter the name of the plane group, the plane group number from International Tables or choose from the list of numbers and names in *Enter group* dropdown list.
3. The *Details* button provides further information about the symmetry operators for the selected plane group.

Tip: To view symmetry and plane group information for the current structure without making changes use the Show Symmetry dialog accessible from the Symmetry toolbar.

Specifying the surface lattice

Lattice parameters determine the length and relative orientation of the lattice vectors in space. In surfaces, these lattice vectors determine how each atom is repeated periodically. Any translation of an atom by an integer multiple of a lattice vector or vectors results in that atom reaching the position of another atom in the surface.

Before [building or rebuilding a surface](#), you must specify the lattice parameters to be used.

To specify the lattice parameters

1. Open the [Build Surface](#) dialog. If the current document is a surface, the dialog will be called *Rebuild Surface*.
2. Select the [Lattice Parameters](#) tab.
3. Specify the unit cell *Lengths*: u and v , either by typing values in the text boxes or by using the spin controls.
4. Specify the unit cell *Angle*: ϑ either by typing value in the text box or by using the spin controls.
5. If the plane group has already been set and the lattice parameters are constrained to certain values by the specified plane group symmetry, these constraints are shown and in most cases the appropriate text boxes are disabled.

Note: If the lattice parameters are set and subsequently a different plane group is selected, the constraints may change. In such cases you should always check the lattice parameters before building or rebuilding the surface.

Note: When rebuild is used information about the current surface is lost. Therefore it should only be used when the current plane group is incorrect.

Tip: To modify the lattice parameters for the current surface without changing the plane group, use the Lattice Parameters dialog accessible from the Symmetry toolbar.


Adding atoms to surfaces

In certain circumstances you may wish to add atoms to a surface. For example, you may wish to add solvent molecules to a surface, in order to understand the interactions between them.

There are several ways of doing this. You can copy and paste a molecular fragment into the document containing the surface.

Alternatively, if you wish to place atoms at a well-defined positions on the surface, you can use the *Add Atoms* option.

To add an atom to a surface

1. Choose *Build / Add Atoms* from the menu bar, or click the *Add Atoms* button on the Atoms & Bonds toolbar to display the [Add Atoms](#) dialog.
2. Select the element type of the atom to be added from the list of common and recently used elements on the [Atoms](#) tab or click the  button to open the Periodic Table dialog.
3. Select the *Coordinate system* on the [Options](#) tab. Your selection also determines which labels appear on the coordinate text boxes on the *Atoms* tab. If you choose [Fractional](#), the labels u , v and w are used. If you choose [Cartesian](#), x , y , and z are used.

Note: If the current system is nonperiodic, the *Coordinate system* list is disabled and Cartesian values are used.

4. On the [Atoms](#) tab, provide a *Name* for the atom to be added. Alternatively, leave the text box blank and a default name will be used.
5. Enter coordinate values for the atom to be added.
6. If necessary, alter the default values for the Oxidation State and Occupancy of the new atom.

7. Select the type of Temperature Factors to be used.
8. Click the *Add* button to add the new atom.

To enable calculation of bonds and to test for proximity

1. Check *Test for bonds as atoms are created* on the [Options](#) tab. When you click the *Add* button bonds will be created to other atoms in the structure, if the criteria defined in the Bond Calculation dialog are met.
2. Check *Warn if atoms are closer than*. When you click the *Add* button a proximity test is performed. Before each atom is added its position relative to other atoms in the structure is determined. If any other atoms in the structure, or any symmetry copies of the new atom are found to be closer than the specified value, a warning is displayed. You can then cancel the *Add* action.

Cleaving surfaces from crystals

The [Cleave Surface](#) dialog allows you to cleave a crystal in a specified direction to create a surface.

To cleave a surface from a crystal

1. Create an empty 3D model document.
2. Import the source crystal from which you wish to cleave the surface.

Note: You can create crystals in Materials Studio by [building](#) them from their constituent parts using either the [Build Crystal](#) or the [Build Vacuum Slab Crystal](#) dialogs.

3. On the [Surface Box](#) tab of the [Cleave Surface](#) dialog, specify the Miller indices of the *Cleave plane*, the *Thickness* and position of the *Top* of the new surface relative to the origin.
4. Select a *Cleave rule* and specify how dangling bonds created by the cleave operation are handled.

Note: The capping algorithm replaces the atom at the end of any bond cut by the cleave operation with an atom of the element type specified. Any bonds between capping atoms are removed. However, capping atoms may still have multiple bonds if, for example, two atoms in the surface were bound to the same atom just outside it. As a consequence, a chemically meaningful surface is not always generated.

Tip: Often a more chemically sensible surface can be obtained by cleaving the crystal without capping and adding hydrogen atoms subsequently, using the Adjust Hydrogen tool on the Sketch toolbar.

5. Click the *Cleave* button.

Additionally, you can set the following options using the [Surface Mesh](#) and [Options](#) tabs on the [Cleave Surface](#) dialog:

- Surface vectors *U* and *V* for the *Cleave plane*. These must be lattice vectors of the crystal being cleaved and they must lie in the *Cleave plane*.

Note: If a *Cleave plane* is specified on the *Surface Box* dialog suitable *Surface vectors* will be generated automatically.

- The origin of the surface lattice in fractional coordinates of the source crystal.
- Whether and how the surface lattice is aligned with regard to the global Cartesian axes.
- Whether and how the current cleave slab is superimposed on the source crystal in the Visualizer.
- Whether the source crystal is automatically recleaved when the cleave parameters are changed.

Building a layered surface or interface

The [Build Layers](#) dialog enables you to build a layered surface or interface from two or three source crystals or surfaces.

To build a layered surface or interface

1. Select *Build* / *Build Layers* from the menu bar to open the [Build Layers](#) dialog.
2. On the [Define Layers](#) tab, specify a *Source document* for each layer that you want to include in the surface, using the document chooser.

Note: Only *periodic* 3D model documents (crystals or surfaces) can be used as *Source documents*. If you select a nonperiodic 3D model document from the document chooser an error message will be generated.

3. If desired, supply a new *Set name* for each layer.
4. Check the *Build layered structure as a surface* option.
5. Click the *Build* button to create a new surface structure from the layers you specified.

Additionally, you can set the following options using the [Layer details](#), [Matching](#) and [Options](#) tabs on the [Build Layers](#) dialog:

- The thickness of the *Vacuum* region between the layers.
- The cleave rule that is used to extract a given layer from the corresponding source crystal.
- The relative orientation of a given layer in the new layered surface.
- The fractional offset of a given layer in the **u** and **v** directions of the new layered surface lattice.
- The lattice parameters of the new layered surface.
- How the lattice parameters of crystal layers are adjusted during the *Build* operation.
- The orientation of the new layered surface lattice.
- The maximum percentage by which the lattice parameters of the layers are allowed to vary from the chosen values.

The layered structure is built up with layer 1 at the bottom, then layer 2 and (if it is present) layer 3 at the top. The layers are aligned **u** with **u** and **v** with **v** and stacked along the normal to the **uv** plane.

Tip: If you want to align the axes of the layers differently, use the Redefine Lattice dialog to transform the layer lattices before building the layered structure.

Note: When the layered structure is built, all symmetry relationships within the layer unit cells are removed. As a result the new surface lattice will have p1 symmetry.

Removing surface periodicity

Materials Studio allows you to interact easily with infinite surfaces. You can perform almost all operations directly on the infinite lattice and you can control which parts of the structure are visible.

As a result, it is not usually necessary to remove the surface periodicity to perform common operations such as editing and structural modifications. However, in certain situations it may be desirable to isolate parts of a surface by removing periodicity. For example, you may wish to compare the energy of an isolated molecule with the energy of the molecular surface.

There are two ways in which you can remove symmetry from a surface:

- By cutting a finite section, or a [nonperiodic superstructure](#), out of the lattice.
- By [unbuilding](#) the surface, leaving the [original objects](#) that were used to build it.

Making a nonperiodic superstructure

If you wish to construct a finite-size cluster from a material, for example because your simulation technique cannot handle periodicity, or because you are interested in the interaction of two finite-sized particles, you can use the *Nonperiodic Superstructure* command. It is enabled if the current structure is a single surface.

To make a nonperiodic superstructure

Choose *Build | Symmetry | Nonperiodic Superstructure* from the menu bar. The command removes all symmetry and periodicity relationships within the structure, and deletes all atoms except those currently visible. The command is thus closely coupled to atom visibility.

Other objects such as bonds, molecules and monitors will persist, depending on the status of the atoms referred to. The resulting structure will be nonperiodic with no symmetry.

Unbuilding a surface

The *Unbuild Surface* command allows you to return to the [original objects](#) from which a surface was built. This command is useful if, for example, you wish to compare the energy of an asymmetric unit molecule with the energy of the molecular surface.

Note: The *Unbuild Surface* command is enabled only if the current structure is a surface.

To unbuild a surface

1. Select *Build | Symmetry | Unbuild Surface* from the menu bar.
2. The command removes all symmetry and periodicity within the structure, and deletes all objects except those that formed the asymmetric unit, along with any others that were the [original objects](#) used to build the surface.
3. A record of the current symmetry definition is attached to the unbuild surface, so that you can [rebuild](#) it later, with the same parameters.

Redefining surface symmetry

For many simulations it is necessary to convert a given surface to a lower symmetry description. For example, in a molecular dynamics simulation of a surface the atoms will move away from their equilibrium positions and the intermediate positions will not satisfy the symmetry constraints of the plane group.

Since Materials Studio strictly imposes symmetry relationships, it is necessary to [create a surface supercell](#) or [convert to p1 symmetry](#) before attempting such a simulation.

It may also be useful to define a supercell or reduce the symmetry in systems with interstitial or substitutional disorder such as doped semiconductors, incommensurate structures under lattice distortions or magnetic structures such as antiferromagnets.

For other, computationally demanding, simulations it is desirable to use the smallest possible unit cell, while retaining all of the symmetry of the surface lattice.

If the plane group of the surface includes a centering operation (C-centering), the lattice can be redefined such that a smaller unit cell, without centering is generated. These are known as primitive unit cells. Primitive unit cells usually have less convenient shapes.

The process of converting a centered cell into a primitive cell is known as [converting to a primitive representation](#). The inverse operation, of converting a primitive unit cell back into a centered cell, is known as [converting to a conventional representation](#).

Unlike the *Create Supercell* and *Make p1* operations, conversion to a primitive representation and back to a conventional representation does not result in the loss of any symmetry information.

Creating a surface supercell

A surface supercell is obtained if, instead of defining the unit cell as the smallest possible repeat unit in the surface, a larger repeat unit is used. You can specify the structure of a supercell by defining its range in unit cells. The new structure will have p1 symmetry and lattice vectors that are multiples of the original unit cell vectors. The concept of a supercell is essential for studying lattice vibrations and to build up structures with substitutional, interstitial or magnetic disorder.

When a supercell is created, no objects are destroyed. However, the symmetry relationships between objects are redefined. As a result, certain atoms which were previously related by translational symmetry are no longer symmetry related and can move independently.

To create a surface supercell

1. Select *Build | Symmetry | Supercell* from the menu bar to open the Supercell dialog.
2. Enter values for the *Supercell range* in the **u** and **v** directions.
3. Click the *Create Supercell* button.

Redefining a surface lattice

To redefine a surface lattice

1. Select *Build | Symmetry | Redefine Lattice* from the menu bar to open the Redefine Lattice dialog.
2. On the New lattice tab, specify the new lattice vectors **u** and **v** in terms of the existing ones. For example, to transpose the **u** and **v** axes type:
u: 0 1, v: 1 0
3. If required, check the *Reorient structure after redefinition* checkbox on the Options tab to reorient the new lattice so that it matches the *Orientation standard* of the original.
4. Click the *Redefine* button.

Checks are made on the new lattice vectors to ensure that they:

- are lattice vectors of the current lattice
- are independent of each other

If the lattice vectors you specified are acceptable, the lattice is redefined. If the new lattice vectors fail any of the checks, the *Redefine* operation is canceled and you are prompted to specify alternative vectors.

Note: Redefining the lattice removes all symmetry relationships within the unit cell. As a result the new structure will have p1 symmetry.

Converting a surface to p1 symmetry

Surfaces with p1 symmetry only possess translational periodicity.

For many simulations, it is necessary to convert a higher symmetry surface to p1 symmetry. For example, during a molecular dynamics simulation when the atoms will move away from their equilibrium positions and the intermediate positions will not satisfy the symmetry constraints of the higher symmetry plane group.

The *Make p1* command removes all symmetry from the current structure and maintains only its translational periodicity.

Note: The *Make p1* command is enabled only if the current structure is a surface with greater than p1 symmetry.

To convert to p1 symmetry

1. Select *Build | Symmetry | Make p1* from the menu bar.
2. The command removes all symmetry relationships present in the surface lattice but maintains any periodic relationships.
3. No objects are destroyed by *Make p1* and any nonsymmetric or nonperiodic object properties are retained.

Converting a surface to primitive representation

For certain, computationally demanding, simulations it is desirable to use the smallest possible unit cell, while retaining all of the symmetry of the surface lattice.

To do this it is necessary to redefine the surface symmetry in terms of a primitive lattice. A primitive lattice is one which has only a single lattice point in each unit cell.

The *Primitive Cell* command converts the current surface lattice into its primitive representation.

Note: The *Primitive Cell* command is enabled only if the current structure is a surface with a nonprimitive lattice.

To convert a surface to primitive representation

1. Select *Build | Symmetry | Primitive Cell* from the menu bar.
2. The command replaces the current lattice vectors with the vectors of the primitive form of the lattice and changes the plane group description accordingly.
3. All of the objects in the surface are redefined according to the new symmetry.
4. No objects are destroyed by *Primitive Cell* and all object properties are retained.
5. In chemical terms, the redefined structure is identical to the initial structure.

Converting a surface to conventional representation

For certain, computationally demanding, simulations it is desirable to [redefine surface symmetry](#) in terms of a primitive lattice.

However, in most cases it is simpler to work with surfaces that have symmetry defined in terms of a conventional lattice. The conventional representation of a surface structure may include a centering operation and usually has a simpler unit cell shape.

The *Conventional Cell* command converts a primitive surface lattice into its conventional representation.

Note: The *Conventional Cell* command is enabled only if the current structure is a surface with a nonconventional primitive lattice.

Since the same primitive lattice can sometimes be converted into different settings of a conventional lattice, the *Conventional Cell* command may involve a choice about the conventional setting to use. Typically, *Conventional Cell* will convert a primitive lattice into the standard setting of the plane group to which the surface belongs. However, if the primitive lattice was obtained from a nonstandard setting of a conventional lattice using the *Primitive Cell* command, *Conventional Cell* will convert it back into the original nonstandard setting.

Converting a surface to conventional representation

1. Select *Build | Symmetry | Conventional Cell* from the menu bar.
2. The command replaces the current lattice vectors with the vectors of the appropriate conventional setting of the lattice (as described above) and changes the plane group description accordingly.
3. All of the objects in the surface are redefined according to the new symmetry.
4. No objects are destroyed by *Conventional cell* and all object properties are retained.
5. In chemical terms, the redefined structure is identical to the initial structure.

Dialogs in Surface Builder

Most surface building and modification tasks can be performed using dialogs which can be accessed from the Build menu.

The following sections describe each of these dialogs in more detail.

Build Surface dialog

The Build Surface dialog allows you to [construct a surface](#) from its constituent parts. It also allows you to specify the symmetry and lattice parameters of the surface.

Note: If the current 3D model document is a surface or a structure on which an [Unbuild](#) operation has been performed, this dialog is named Rebuild Surface. The Rebuild Surface dialog can be used to modify the symmetry and lattice parameters of an existing surface or to rebuild a surface following an *Unbuild* operation.

When the Build Surface dialog is open, you will not be able to work with other parts of Materials Studio.

The Build Surface dialog contains the following tabs:

- [Plane Group](#): Allows you to specify the plane group used to build or rebuild the surface.
- [Lattice Parameters](#): Allows you to specify values of the lattice parameters for the surface you wish to build.
- [Options](#): Allows you to set build options and specify the orientation standard for the surface you wish to build.

Build: Creates a surface from the current active 3D model, using the parameters specified in the [Plane Group](#) and [Lattice Parameters](#) tabs and closes the dialog.

Note: If the current active 3D model is a surface, the button is labeled *Rebuild*.

Rebuild: Unbuilds the current surface, then creates a new surface using the parameters specified in the [Plane Group](#) and [Lattice Parameters](#) tabs and closes the dialog.

Build or Rebuild | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Note: When rebuild is used information about the current surface is lost. Therefore it should only be used when the plane group is incorrect.

Tip: To modify the lattice parameters for the current surface without changing the plane group, use the Lattice Parameters dialog accessible from the Symmetry toolbar.

Apply: Creates (or recreates) a surface from the current active 3D model, using the parameters specified in the [Plane Group](#) and [Lattice Parameters](#) tabs without closing the dialog.

Note: If the current active 3D model is not a surface, after *Apply* is used the *Build Surface* dialog is renamed *Rebuild Surface* and the *Build* button is renamed *Rebuild*.

Cancel: Closes the dialog without creating a surface.

Help: Displays the Help topic for the current tab.

Access methods

Menu *Build | Surfaces | Build/Rebuild Surface...*

Plane Group tab

Note: The *Plane Group* tab may appear in a dialog named either *Build Surface* or *Rebuild Surface* depending on whether or not a surface has already been built.

The *Plane Group* tab allows you to specify the plane group used to build or rebuild the surface. Initially, the plane group of the current 3D model document is displayed. If no plane group is defined for the current 3D model, default values are shown.

Enter group: Specify the plane group to be used.

Type the required plane group symbol or the corresponding plane group number from International Tables in the text box. Alternatively, choose the required plane group from the dropdown list.

Below the *Enter group* text box, general information about the selected plane group is displayed.

Details...: Provides access to the Show Symmetry dialog which contains further details about the symmetry operators for the selected plane group.

Note: When opened in this way, the Show Symmetry dialog closes if the *Build Surface* dialog is closed.

Access methods

Menu *Build | Surfaces | Build/Rebuild Surface... | Plane Group*

Lattice Parameters tab

Note: The *Lattice Parameters* tab may appear in a dialog named either *Build Surface* or *Rebuild Surface* depending on whether or not a surface has already been built.

Lattice parameters determine the length and relative orientation of the lattice vectors in space. In crystals and surfaces, these lattice vectors determine how each atom is repeated periodically.

The *Lattice Parameters* tab allows you to specify values of the lattice parameters for the surface you wish to build.

Lattice type: Displays the lattice type of the selected plane group.

If the lattice parameters are constrained to certain values by the specified plane group symmetry, these constraints are shown and the appropriate text boxes are disabled.

Lengths: Specify values of the lattice parameters u and v which determine the lengths of the lattice vectors.

The spin controls beside each text box change the current values by 0.2 Å.

Angle: Specify the value of the lattice parameter ϑ , which determines the relative orientation of the lattice vectors.

The spin control beside the text box changes the current value by 1.0 degree.

Access methods

Menu *Build | Surfaces | Build/Rebuild Surface... | Lattice Parameters*

Options tab

Note: The *Options* tab may appear in a dialog named either *Build Surface* or *Rebuild Surface* depending on whether or not a surface has already been built.

The *Options* tab allows you to set build options and specify the orientation standard for the surface you wish to build. Use this tab to:

- specify the tolerance used to test for proximity to special positions
- select whether or not to calculate bonds automatically after building the surface
- specify how the lattice is aligned with respect to the global Cartesian axes

Check for atoms on special positions: When checked indicates that atoms will be tested for their proximity to any special positions and for overspecification using the tolerance specified in the associated text box.

The tolerance value is the maximum distance allowed between two points for them to be considered as belonging to the same lattice site.

Note: This option is disabled if *Ignore special positions* is checked.

Ignore special positions: When checked indicates that atoms will not be tested for their proximity to special positions or for overspecification. All possible symmetry equivalents will be generated for each atom in the structure.

Calculate bonding after build: When checked indicates that a bond calculation will be performed after the surface has been built using the values specified on the Bond Calculation dialog.

Orientation standard: Specify the alignment of the lattice with respect to the Cartesian axes and the rule used for subsequent alterations.

Access methods

Menu *Build | Surfaces | Build/Rebuild Surface... | Options*

Cleave Surface dialog

The Cleave Surface dialog allows you to cut a crystal in a specified plane to create a surface.

Note: If the current 3D model document is a cleaved surface, this dialog is named Recleave Surface.

The Cleave Surface dialog contains the following tabs:

- [Surface Box](#): Allows you to specify the cleave plane and related parameters.
- [Surface Mesh](#): Allows you to specify surface vectors and related parameters.
- [Options](#): Allows you to set additional options for the cleaved surface.

Cleave: Performs a cleave operation on the current active 3D model using the parameters specified in the *Surface Box*, *Surface Mesh* and *Options* tabs.

A new document containing the cleaved surface is created and becomes the active document.

The *Cleave Surface* dialog remains open and is renamed *Recreate Surface*.

Recreate: Performs a cleave operation on the source crystal using the parameters specified in the *Surface Box*, *Surface Mesh* and *Options* tabs and replaces the contents of the current active document with the new surface generated.

Cleave or Recreate | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Note: By default, the *Recreate* operation is carried out automatically whenever changes are made to the cleave slab parameters or any other parameters that affect the cleave surface. To switch off this feature, uncheck *Automatically recreate when parameters change* on the [Options](#) tab.

Help: Displays the Help topic for the current tab.

Access methods

Menu *Build | Surfaces | Cleave/Recreate Surface...*

Surface Box tab

Note: The *Surface Box* tab may appear in a dialog named either *Cleave Surface* or *Recreate Surface*, depending on whether or not the currently active 3D model document contains a cleaved surface.

Cleave plane (h k l): Specify the cleave plane in terms of the Miller indices h, k, and l, separated by spaces.

Note: The *Cleave plane* field specifies a plane in which the surface vectors (on the *Surface Mesh* tab) must lie. If you wish to allow arbitrary surface vectors, you can leave the *Cleave plane* field blank.

Top: Specify the distance from the origin to the top of the cleave slab as either:

- *Fractional* - a fraction of the d-spacing of the Miller family of the cleave plane
- *Angstrom* - an absolute value in Å

The spin controls between the *Fractional* and *Angstrom* text boxes move the cleave slab up and down incrementally in a direction normal to the cleave plane. The size of the increment used is variable and is chosen such that the next extra set of atoms or beads is included in the cleave slab with each move.

Thickness: Specify the thickness of the cleave slab either as a fraction of the d-spacing or in Å.

The spin controls between the *Fractional* and *Angstrom* text boxes change the thickness of the cleave slab incrementally. The size of the increment used is variable and is chosen such that the next extra set of atoms or beads is included in the cleave slab with each move.

Note: The *Fractional* fields for *Top* and *Thickness* are left blank if an arbitrary cleave plane is specified.


Cleave rule: Select the cleave rule to be used to determine whether an atom or bead is included in the cleave slab from the dropdown list. Available options are:

- [Default](#) - atoms or beads are included in the cleave slab if the center of geometry of the connected fragment to which they belong lies within the cleave slab boundary. Adjustments are made for atoms or beads belonging to infinitely connected networks.
- [Atomic](#) - atoms or beads are included in the cleave slab if they lie within the cleave slab boundary.

Cap bonds on: Specify how dangling bonds created by the cleave operation are handled.

Select which faces of the cleave slab should have dangling bonds capped from the first dropdown list. Available options are:

- Neither
- Top
- Bottom
- Both

Specify the element to be used as the capping atom in the second text box. A list of common elements is provided in the dropdown list. Alternatively, click the  button to select an element from the entire periodic table.

Note: The capping algorithm applies only to bonds between atoms. Any connections between beads that are cut by the cleave operation will be removed.

Tip: Often, a more chemically sensible surface can be obtained by cleaving the crystal without capping and then adding hydrogen atoms subsequently, using the Adjust Hydrogen tool on the *Sketch* toolbar.

Note: When changes are made to the cleave slab parameters or any other parameters that affect the cleave surface, a *ReCleave* operation is carried out automatically, by default. To switch off this feature, uncheck the *Automatically recleave when parameters change* checkbox on the [Options](#) tab.

Access methods

Menu *Build | Surfaces | Cleave/ReCleave Surface... | Surface Box*

Surface Mesh tab

Note: The *Surface Mesh* tab may appear in a dialog named either *Cleave Surface* or *ReCleave Surface*, depending on whether or not the currently active 3D model document contains a cleaved surface.

Surface vectors: Specify surface vectors, U and V , for the cleave plane, separated by spaces.

The surface vectors used must be lattice vectors of the crystal being cleaved and they must lie in the cleave plane. If an arbitrary cleave plane is specified, the surface vectors may be any lattice vectors of the crystal.

Note: If a cleave plane is specified on the *Surface Box* tab, suitable surface vectors will be generated automatically.

Mesh lengths: Displays the lengths of the mesh defined by the specified surface vectors.

Angle: Displays the angle of the mesh defined by the specified surface vectors.

Box volume / cell volume: Displays the ratio of the cleave slab volume to the source crystal unit cell volume.

Origin (a b c): Specify the position of the origin of the surface lattice in fractional coordinates of the source crystal.

Move along: Adjust the position of the origin of the surface lattice as required. The two spin controls move the origin in the directions of the U and V vectors, respectively.

Step size: Specify the size and units of the increment used by the *Move along* spin controls.

Note: When changes are made to the cleave slab parameters or any other parameters that affect the cleave surface, a *Recleave* operation is carried out automatically, by default. To switch off this feature, uncheck the *Automatically recleave when parameters change* checkbox on the [Options](#) tab.

Access methods

Menu [Build](#) | [Surfaces](#) | [Cleave/ReCleave Surface...](#) | [Surface Mesh](#)

Options tab

Note: The *Options* tab may appear in a dialog named either *Cleave Surface* or *Recleave Surface* depending on whether or not the current active 3D model document contains a cleaved surface.

Orientation standard: Specify the rule used to align the surface lattice with respect to the Cartesian axes.

Reset orientation and origin after cleave: When checked indicates that the surface lattice will be reoriented to match the *Orientation standard* and moved so that its origin coincides with the Cartesian origin after the cleave operation. Default = [unchecked](#).

Note: If *Reset orientation after cleave* is unchecked the surface lattice will not be reoriented and therefore the atoms in it will have Cartesian coordinates identical to those of the corresponding atoms in the source crystal.

Display surface box in crystal: When checked indicates that the outline of the current cleave slab will be superimposed on the source crystal in the Visualizer. Default = [checked](#).

The color of the outline can be changed using the Microsoft Common Color Control adjacent to the checkbox. The spin controls allow you to adjust the brightness.

Automatically re-cleave when parameters change: When checked indicates that any change to the cleave slab parameters or any other parameters that affect the cleave surface will cause an automatic recleave. Default = [checked](#).

Note: This checkbox is enabled only when the *Cleave Surface* dialog is in Recleave mode.

Access methods

Menu [Build](#) | [Surfaces](#) | [Cleave/ReCleave Surface...](#) | [Options](#)

Layer Builder

In Materials Studio, a *layered structure* is 2D or 3D periodic structure that has been created by superimposing a number of component layers. In a 2D layered structure, the component layers are generated by making copies of an existing surface, or by cleaving a surface from an existing crystal. In a 3D layered structure, the component layers are generated in a similar manner but they are converted into a [slab](#) before being combined with the other component layers. Vacuum regions separate the component layers, which are identified in the resultant model using sets.

The following topics provide more details on the layer building functionality in Materials Studio:

- [Building a layered crystal](#)
- [Building a layered surface or interface](#)

The following tutorial examples illustrate some applications of the Layer Builder:

- Twinning in silicon - illustrates how to create an interface between two different surfaces of a crystal.

Building a layered crystal

The [Build Layers](#) dialog enables you to build a layered crystal from two or three source crystals or surfaces.

To build a layered crystal

1. Select *Build* / *Build Layers* from the menu bar to open the [Build Layers](#) dialog.
2. On the [Define Layers](#) tab, specify a *Source document* for each layer that you want to include in the crystal, using the document chooser.

Note: Only *periodic* 3D model documents (crystals or surfaces) can be used as *Source documents*. If you select a nonperiodic 3D model document from the document chooser an error message will be generated.

3. If desired, supply a new *Set name* for each layer.
4. Check the *Build layered structure as a crystal* option.
5. Click the *Build* button to create a new crystal structure from the layers you specified.

Additionally, you can set the following options using the [Layer details](#), [Matching](#) and [Options](#) tabs on the [Build Layers](#) dialog:

- The thickness of the *Vacuum* region between the layers.
- The cleave rule that is used to extract a given layer from the corresponding source crystal.
- The relative orientation of a given layer in the new layered crystal.
- The fractional offset of a given layer in the **a** and **b** directions of the new layered crystal lattice.
- The lattice parameters of the new layered crystal.
- How the lattice parameters of crystal layers are adjusted during the *Build* operation.
- The orientation of the new layered crystal lattice.
- The maximum percentage by which the lattice parameters of the layers are allowed to vary from the chosen values.
- Whether the relevant layer parameters should be set to values suitable for the *Confined Shear* task in Forcite.

The layered structure is built up with layer 1 at the bottom, then layer 2 and (if it is present) layer 3 at the top. The layers are aligned **a** with **a** and **b** with **b** and stacked along **c**.

Tip: If you want to align the axes of the layers differently, use the Redefine Lattice dialog to transform the layer lattices before building the layered structure.

The length of the **c** lattice vector is determined by adding together the **c** lattice vectors (or the surface thicknesses) of each layer (after they have been adjusted) and the *Vacuum* regions for each layer.

Note: When the layered structure is built, all symmetry relationships within the layer unit cells are removed. As a result the new crystal lattice will have P1 symmetry.

Tip: Any symmetry present in the layered crystal structure can be reimposed using the Find Symmetry tool.

Building a layered surface or interface

The [Build Layers](#) dialog enables you to build a layered surface or interface from two or three source crystals or surfaces.

To build a layered surface or interface

1. Select *Build* / *Build Layers* from the menu bar to open the [Build Layers](#) dialog.
2. On the [Define Layers](#) tab, specify a *Source document* for each layer that you want to include in the surface, using the document chooser.

Note: Only *periodic* 3D model documents (crystals or surfaces) can be used as *Source documents*. If you select a nonperiodic 3D model document from the document chooser an error message will be generated.

3. If desired, supply a new *Set name* for each layer.
4. Check the *Build layered structure as a surface* option.
5. Click the *Build* button to create a new surface structure from the layers you specified.

Additionally, you can set the following options using the [Layer details](#), [Matching](#) and [Options](#) tabs on the [Build Layers](#) dialog:

- The thickness of the *Vacuum* region between the layers.
- The cleave rule that is used to extract a given layer from the corresponding source crystal.
- The relative orientation of a given layer in the new layered surface.
- The fractional offset of a given layer in the **u** and **v** directions of the new layered surface lattice.
- The lattice parameters of the new layered surface.
- How the lattice parameters of crystal layers are adjusted during the *Build* operation.
- The orientation of the new layered surface lattice.
- The maximum percentage by which the lattice parameters of the layers are allowed to vary from the chosen values.

The layered structure is built up with layer 1 at the bottom, then layer 2 and (if it is present) layer 3 at the top. The layers are aligned **u** with **u** and **v** with **v** and stacked along the normal to the **uv** plane.

Tip: If you want to align the axes of the layers differently, use the Redefine Lattice dialog to transform the layer lattices before building the layered structure.

Note: When the layered structure is built, all symmetry relationships within the layer unit cells are removed. As a result the new surface lattice will have p1 symmetry.

Build Layers dialog

The Build Layers dialog allows you to construct a layered structure from two or three periodic 3D model documents.

The Build Layers dialog contains the following tabs:

- **Define Layers:** Allows you to specify the periodic 3D model documents that will be used to build the layered structure and to name the layers.
- **Layer Details:** Allows you to specify how each layer will be added to the new layered structure.
- **Matching:** Allows you to select the lattice parameters of the new layered structure from those of the source models or their average and also to control what happens when the lattices of the source models do not match.
- **Options:** Allows you to set additional build options and specify the orientation of the new layered structure.

Build: Creates a layered structure from the periodic 3D model documents specified in the [Define Layers](#) tab using the parameters specified in the [Layer details](#), [Matching](#), and [Options](#) tabs.

Build | Copy Script: Converts the current settings to a script and copies the script to the clipboard. Refer to the Generating scripts topic for more information on using sections of scripts generated from a dialog.

Note: If, after adjustment, the lattice parameters of any of the layers vary from the lattice parameters of the new layered structure by more than the tolerance specified in the [Options](#) tab, a warning will be generated and you will be asked to choose whether to proceed with the *Build* operation.

Layered structures are built up with layer 1 at the bottom, then layer 2 and (if it is present) layer 3 at the top. The layers are aligned **a** with **a** (or **u** with **u**) and **b** with **b** (or **v** with **v**) and they are stacked along **c** (or the normal to the **uv** plane).

Tip: If you want to align the axes of the layers differently, use the Redefine Lattice dialog to transform the layer lattices before building the layered structure.

If a surface is to be generated from the layered structure, layer 1 will be below the plane of the surface, layer 2 will be above it and (if it is present) layer 3 will be placed on top of layer 2.

If a crystal is to be generated from the layered structure, the contributions made by each layer to the length of the **c** lattice vector are determined according to the nature of the source models and the cleave rules used:

- Layers generated from crystals using the [Atomic](#) cleave rule contribute the thickness of **c** lattice vector of the source crystal to the overall thickness.
- Layers generated from crystals using the [Default](#) cleave rule contribute the thickness of the extracted surface (modified to account for the van der Waals radii of the exposed atoms) to the overall thickness.
- Layers generated from surfaces contribute the surface thickness to the overall thickness.

The **c** lattice vector of the new layered crystal is given by adding together the contributions from each layer (after they have been adjusted) and the *Vacuum* regions for each layer.

Note: When the layered structure is built, all symmetry relationships within the layer unit cells are removed. As a result the new structure will have P1 (crystal) or p1 (surface) symmetry.

Tip: Any symmetry present in layered *crystal* structures can be reimposed using the Find Symmetry tool.

Help: Displays the Help topic for the current tab.

Access methods

Menu *Build | Build Layers*

Define Layers tab

The *Define Layers* tab allows you to specify the periodic 3D model documents that will be used to build the layered structure. It also enables you to give the layers names.

Source documents: Specify the source documents that will be used to build the layered structure with the document chooser. The document chooser filters the files in the current project so that only currently loaded 3D model documents are displayed.

Note: Only *periodic* 3D model documents (crystals or surfaces) can be used as source documents. If you select a nonperiodic 3D model document from the document chooser an error message will be generated.

Set names: Specify the set names that will be used to identify each of the layers in the new 3D model document.

Tip: Use the options available on the Edit Sets dialog to differentiate the layers in the new 3D model document.

Build layered structure as a crystal: When checked, indicates that the layered structure will be repeated periodically to generate a [crystal](#).

Build layered structure as a surface: When checked, indicates that only one layered structure will be created to generate a [surface](#).

Access methods

Menu *Build | Build Layers | Define Layers*

Layer Details tab

The *Layer Details* tab on the Build Layers dialog allows you to specify how each layer will be added to the new layered structure.

Vacuum: Specify the thickness of the vacuum region that will be added above the given layer. The *Vacuum* for layer 1 will be the spacing between layers 1 and 2 and so on.

If a crystal is to be generated from the layered structure, the *Vacuum* for the top layer will be the spacing between one layered structure and the next, in the *c* direction.

If a surface is to be generated from the layered structure, the *Vacuum* textbox for the top layer will be disabled.

Cleave: Select the cleave rule that will be used to extract the given layer from the corresponding crystal source model. The options available are:

- **Default:** Cuts the crystal in the planes $c = 0.0$ and $c = 1.0$ and includes connected atoms if the centers of mass of the molecules to which they belong lie within the cleaved layer.
- **Atomic:** Cuts the crystal in the planes $c = 0.0$ and $c = 1.0$ and does not include any connected atoms.

Tip: If you want more control over how your crystal source model is cleaved, use the [Cleave Surface](#) dialog to generate a surface, before building the layered structure.

Note: If the source model for the given layer is a surface, the *Cleave* control will be disabled.

Note: The cleave rule selected affects the contribution made by each layer to the overall thickness of any layered crystal structure generated. See the description of the *Build* button for more details.

Flip: Specify the relative orientation of the given layer in the new layered structure. The options available are:

- **No:** The given layer will be in the same orientation relative to the lattice vectors as it was in the source model.
- **A:** The given layer will be rotated by 180 degrees about its **a** axis (if the source model is a crystal) or its **u** axis (if the source model is a surface) axis, thus transforming the lattice vectors from **a, b, c** to **a, -b, -c** (crystal) or **u, v** to **u, -v** (surface).
- **B:** The given layer will be rotated by 180 degrees about its **b** axis (if the source model is a crystal) or its **v** axis (if the source model is a surface) axis, thus transforming the lattice vectors from **a, b, c** to **-a, b, -c** (crystal) or **u, v** to **-u, v** (surface).

Note: Use of the *Flip* option will change the angle used for lattice matching from γ to $180-\gamma$ (if the layered structure is to be a crystal) or from ϑ to $180-\vartheta$ (if the layered structure is to be a surface).

Origin offset: Specify the fractional offset of the given layer in the **a** and **b** directions of the new crystal lattice or the **u** and **v** directions of the new surface lattice.

Access methods

Menu | [Build](#) | [Build Layers](#) | [Layer details](#)

Matching tab

The *Matching* tab allows you to select the lattice parameters of the new layered structure from those of the source models or their average. It also allows you to control what happens when the lattices of the source models do not match.

Choose new lattice parameters: Select the desired set of lattice parameters from the list displayed.

The lattice parameters which define the **ab** (crystal) or **uv** (surface) planes of each of the layers are shown, together with the average values.

Adjustment of component layers: Choose how the lattice parameters of *crystal* layers will be adjusted during the *Build* operation.

Note: This option is enabled only if the selected layer comes from a crystal source document.

The choices available are:

- [Keep constant volume](#) - Adjustments to the lattice parameters will be made by varying the thickness of the layer while keeping the volume (and thus the density) fixed.
- [Keep constant thickness](#) - Adjustments to the lattice parameters will be made by varying the volume (and thus the density) while keeping the thickness of the layer fixed.

Note: The fractional coordinates of the atoms in crystal layers remain fixed during the adjustments.

Access methods

Menu [Build](#) | [Build Layers](#) | [Matching](#)

Options tab

The *Options* tab allows you to set additional build options and specify the orientation of the new layered structure.

Lattice Orientation: Select the orientation of the new layered structure. The available options are:

- [Use orientation from layer 1](#)
- [Use orientation from layer 2](#)
- [Use orientation from layer 3](#)

Warn if lattice mis-match is greater than: Specify the maximum percentage by which the lattice parameters of the layers are allowed to vary from the chosen values. Default = [5.0%](#).

Note: If any of the lattice vectors vary from the chosen values by more than this amount, a warning message will be generated when the *Build* operation is performed.

Configure for confined shear use: When checked, indicates that relevant layer parameters will be set to values suitable for the *Confined Shear* task in Forcite. Specifically:

- The *Set names* for layers 1, 2 and 3 will be set to [BOTTOMWALL](#), [FLUID](#), and [TOPWALL](#), respectively and the controls disabled.
- The output structure type will be set to crystal and the control disabled.
- The *Vacuum* for layers 1 and 2 will be set to [0.0](#) and the controls disabled.
- The *Vacuum* for layer 3 will be set to [20.0](#).
- The cleave rule for all layers will be set to [Atomic](#) and the controls disabled.

Note: If you are creating a layered structure for use in a *Confined Shear* calculation it is recommended that the central [FLUID](#) layer is created with the Amorphous Cell module, using the *Cell Type* [Confined Layer](#).

Note: If the *Configure for confined shear use* checkbox is subsequently unchecked, all of the controls will be reenabled but the values of the parameters will remain unchanged.

Access methods

Menu [Build](#) | [Build Layers](#) | [Options](#)