



**ENTERPRISE ARCHITECT**

Série de Guides d'Utilisateur

# **Guide de Modélisation Logicielle - Fondamentaux**

Author: Sparx Systems & Stephen Maguire

Date: 7/11/2024

Version: 17.0

CRÉÉ AVEC  **ENTERPRISE  
ARCHITECT**

# Table des Matières

Introduction	7
Développement intégré	9
Aperçu Fonctionnalité	11
Générer Code Source	13
Générer une classe unique	16
Générer un groupe de classes	17
Générer un Paquetage	18
Mettre à jour le contenu Paquetage	20
Synchroniser Modèle et le code	22
Namespaces	23
Importation du code source	24
Projets d'importation	26
Importer le code source	29
Notes sur l'importation du code source	30
Script d'importation de ressources	32
Importer une structure de répertoire	34
Importer un module binaire	36
Classes non trouvées lors de l'importation	37
Modification du code source	38
Langues prises en charge	41
Configurer les associations de fichiers	42
Comparer les éditeurs	43
Barre d'outils Éditeur de Code	44
Éditeur de Code Menu Contexte	47
Créer un cas d'utilisation pour la méthode	50
Éditeur de Code Fonctions	52
Détails de la fonction	53
Intelli-sens	56
Rechercher et remplacer	58
Rechercher dans les fichiers	61
Rechercher un fichier	64
Recherche Intelli-sense	66
Raccourcis clavier Éditeur de Code	69
Motifs d'application ( Modèle + Code)	73
Intégration MDG et Ingénierie de Code	75
Interaction et expérience utilisateur	76
Démarrage	77
Modèles de flux d'interaction IFML	78
Boîte à outils de wireframe Android	80
Boîte à outils de wireframe pour iPhone/tablette Apple	88
Boîte à outils de wireframe Windows Phone	96
Boîte à outils de wireframe Dialogue	103
Boîte à outils de wireframe de page Web	114
Génération de code Modèle Comportementale	117
Génération de code - Diagrammes d'activités	120
Génération de code - Diagrammes d'interaction	122
Génération de code – Statemachines	123

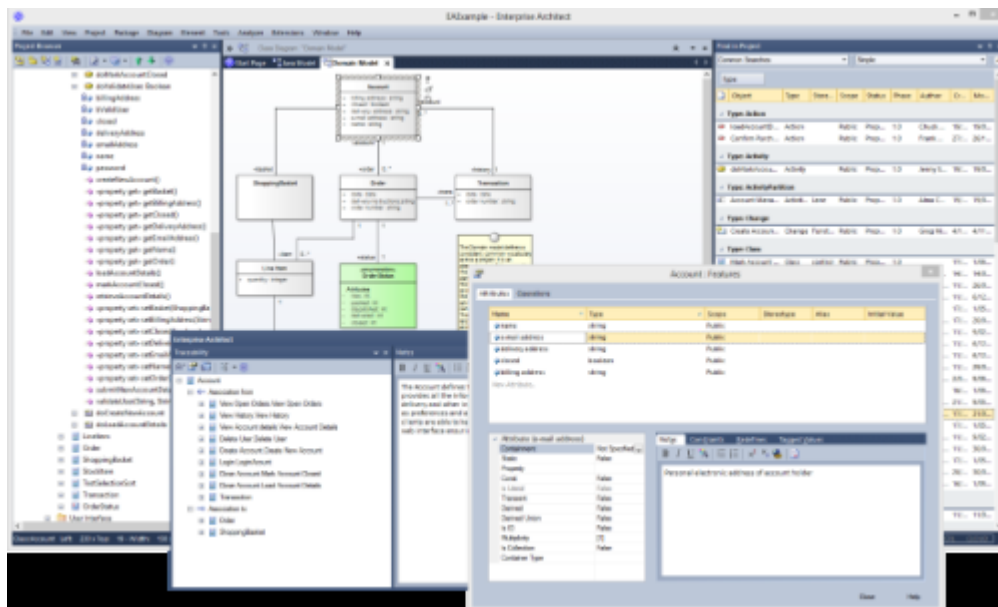
Statemachine héritée Gabarits .....	127
Code Java généré à partir d' Statemachine héritée Gabarit .....	129
Modélisation Statemachine pour les HDL .....	135
Boîtes de dialogue Interface Utilisateur Win32 .....	137
Modélisation des dialogues UI .....	139
Importer Dialogue unique à partir d'un fichier RC .....	141
Importer toutes les boîtes de dialogue à partir du fichier RC .....	142
Exporter Dialogue vers un fichier RC .....	143
Concevoir un nouveau Dialogue .....	144
Motifs de Gang of Four (GoF) .....	147
Paramètres de configuration .....	149
Options d'ingénierie du code source .....	150
Options de génération de code .....	152
Types de composants d'importation .....	154
Options du code source .....	155
Options - Éditeurs de Code .....	157
Propriétés de la langue de l'éditeur .....	159
Options - Durée de vie Object .....	161
Options - Attribut/Opérations .....	162
Conventions Modélisation .....	164
Conventions ActionScript .....	166
Conventions Ada 2012 .....	168
Conventions C .....	171
Programmation orientée Object en C .....	174
Conventions C# .....	176
Conventions C++ .....	180
Conventions C++ gérées .....	183
Conventions C++/CLI .....	184
Conventions Delphi .....	186
Conventions Java .....	188
Conventions d'AspectJ .....	190
Conventions PHP .....	191
Conventions Python .....	193
Conventions SystemC .....	194
Conventions VB.NET .....	196
Conventions Verilog .....	199
Conventions VHDL .....	201
Conventions de Visual Basic .....	204
Options de langue .....	206
Options ActionScript - Utilisateur .....	208
Options ActionScript - Modèle .....	209
Options Ada 2012 - Utilisateur .....	210
Options Ada 2012 - Modèle .....	211
Options ArcGIS - Utilisateur .....	212
Options ArcGIS - Modèle .....	213
Options C - Utilisateur .....	214
Options C - Modèle .....	215
Options C# - Utilisateur .....	217
Options C# - Modèle .....	218
Options C++ - Utilisateur .....	219
Options C++ - Modèle .....	220

Options Delphi - Utilisateur .....	222
Options Delphi - Modèle .....	223
Propriétés Delphi .....	224
Options Java - Utilisateur .....	225
Options Java - Modèle .....	226
Options MySQL - Utilisateur .....	228
Options MySQL - Modèle .....	229
Options PHP - Utilisateur .....	230
Options PHP - Modèle .....	231
Options Python - Utilisateur .....	232
Options Python - Modèle .....	233
Options SystemC - Utilisateur .....	234
Options SystemC - Modèle .....	235
Options Teradata - Utilisateur .....	236
Options Teradata - Modèle .....	237
Options VB.NET - Utilisateur .....	238
Options VB.NET - Modèle .....	239
Options Verilog - Utilisateur .....	240
Options Verilog - Modèle .....	241
Options VHDL - Utilisateur .....	242
Options VHDL - Modèle .....	243
Options Visual Basic - Utilisateur .....	244
Options Visual Basic - Modèle .....	245
Options linguistiques MDG Technologie .....	246
Options de réinitialisation .....	247
Classes de collection d'ensembles .....	248
Exemple d'utilisation des classes de collection .....	250
Chemins locaux .....	253
Dialogue sur les chemins locaux .....	254
Macros de langage .....	256
Développement de langages de programmation .....	258
Cadre de code Gabarit .....	260
Code Gabarit Personnalisation .....	261
Coder et transformer Gabarits .....	262
Base Gabarits .....	264
Génération et transformation de code d'exportation Gabarits .....	267
Génération et transformation de code d'importation Gabarits .....	268
Synchroniser le code .....	269
Synchroniser les sections existantes .....	271
Ajouter de nouvelles sections .....	272
Ajouter de nouvelles Fonctionnalités et éléments .....	273
L'éditeur Code Gabarit .....	274
Créer un nouveau Gabarit personnalisé .....	276
Syntaxe du code Gabarit .....	277
Texte littéral .....	278
Variables .....	279
Macros .....	282
Macros de substitution Gabarit .....	284
Macros de substitution de champs .....	286
Exemples de substitution .....	287
Macros de substitution de champs d'attributs .....	289

Macros de substitution de champs de classe	291
Option de génération de code Macros de substitution de champ	294
Macros de substitution de champs de connecteur	299
Macros de substitution de champs de contraintes	303
Macros de substitution de champs d'effort	304
Macros de substitution de champs de fichiers	305
Macros de substitution de champs d'importation de fichiers	306
Macros de substitution de champs de liens	308
Macros de substitution de champs de fichiers liés	310
Macros de substitution de champs métriques	311
Macros de substitution de champs d'opération	312
Macros de substitution de champs de Paquetage	314
Macros de substitution de champs de paramètres	315
Macros de substitution de champs problématiques	316
Macros de substitution de champs d'exigences	317
Macros de substitution de champs de ressources	318
Macros de substitution de champs de risque	319
Macros de substitution de champs de scénario	320
Macros de substitution Valeur Étiquetée	321
Macros de substitution de paramètres Gabarit	323
Macros de substitution de champs Test	324
Macros de fonctions	325
Contrôle des macros	331
Liste des macros	332
Branchement des macros	334
Macros de synchronisation	336
La macro d'instructions de traitement (PI)	337
Macros de génération de code pour Statemachines Exécutables	338
Macros de génération de code EASL	349
Collections EASL	352
Propriétés EASL	356
Appeler Gabarits depuis Gabarits	363
L'éditeur Code Gabarit dans le développement des OMD	364
Créer Gabarits personnalisés	365
Personnaliser Gabarits de base	367
Ajouter de nouveaux Gabarits stéréotypés	368
Remplacer Gabarits par défaut	370
Cadre grammatical	371
Grammaire Syntaxe	372
Instructions de grammaire	374
Règles de grammaire	375
Termes de grammaire	376
Commandes de grammaire	377
Nœuds AST	379
Édition des grammaires	387
Analyse des résultats AST	389
Profilage de l'analyse grammaticale	390
Éditeur de macros	391
Exemples de grammaires	392



# Introduction



Enterprise Architect in addition to its business model simulation, coding and visual execution analysis capability is a fully fledged Integrated Development Environment. This allows Software Engineers, Programmers and other technical staff to develop, maintain and round trip engineer source code in a wide range of programming languages with all the facilities that developers using other IDEs such as Eclipse and Visual Studio would expect and much more.

Round trip code engineering is supported by the Code Template Framework and the Grammar framework allowing any programming language to be incorporated. These frameworks provide a mechanism for developers to create support for additional languages or to configure the way that round trip engineering works out to the box.

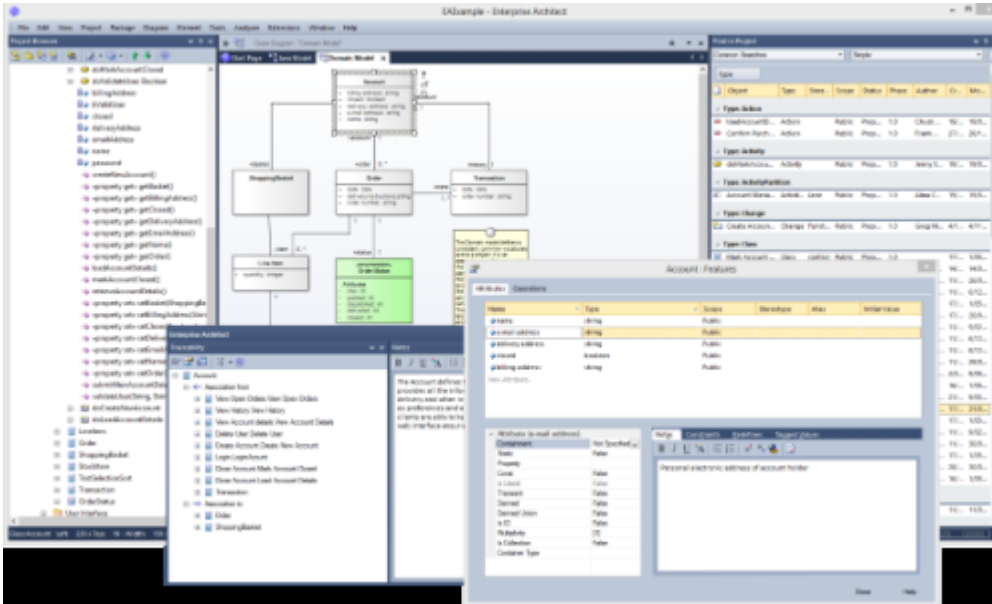
In addition to the IDE there is support for the design, visualization and resource generation for Win32 screens.

There is also an integrated Database Engineering tool that connects to live databases and allows database schemas to be imported and exported including support for incremental changes. XML schemas can also be developed and managed using the handy Schema Composer allowing user defined Schemas to be generated that are compliant with industry standard schemas such as NIEM.





# Développement intégré



Enterprise Architect fournit un ensemble inégalé d'outils et fonctionnalités pour l'ingénieur logiciel, afin de l'aider à créer des systèmes logiciels robustes et sans erreur. L'ingénieur peut commencer par définir l'architecture et s'assurer qu'elle correspond aux exigences et aux spécifications. Les modèles neutres sur le plan technologique peuvent être transformés pour cibler une gamme complète de langages de programmation. L'environnement de développement piloté par Modèle s'adapte à diverses technologies.

## Fonctionnalités

### Outils de développement

- Développement piloté par Modèle avec les meilleurs outils UML de leur catégorie
- Générer et rétro-ingénierie du code
- Personnaliser la génération de code avec gabarits
- Scripts d'Analyseur pour gérer vos applications
- Éditeurs de code pour créer la base de code
- Débugueurs pour enquêter sur le comportement
- Des profilers pour visualiser le comportement
- Analyseurs pour enregistrer le comportement
- Testpoints pour la validation des contrats de programmation
- Intégration avec JUnit et NUnit
- Intégration Eclipse ou Visual Studio si nécessaire

### Traçabilité

Traçabilité en un coup d'œil des généralisations, réalisations, associations, dépendances et plus encore. Personnalisez les vues de relation. Naviguez facilement entre les éléments associés dans le modèle.

### Usage

Parcourez rapidement l'utilisation des éléments dans tous diagrammes. Effectuez des recherches d'éléments efficaces à l'aide de requêtes sophistiquées.

### Langues populaires

- C/ C++

- Java
- Famille Microsoft .NET
- ADA
- Python
- Perl
- PHP

**Boîtes à outils** Des boîtes à outils sont fournies pour une vaste gamme de technologies modélisation et de langages de programmation.

**Motifs d'application** Enterprise Architect fournit des projets de démarrage complets, y compris des informations sur le modèle, du code et des scripts de construction, pour plusieurs types d'applications de base.

# Aperçu Fonctionnalité

L'ingénierie de code avec Enterprise Architect englobe largement divers processus pour la conception, la génération et la transformation du code à partir de votre modèle UML .

## Fonctionnalités

### Ingénierie de code pilotée par Modèle

- Génération de code source et rétro-ingénierie pour de nombreux langages populaires, notamment C++, C# , Java, Delphi, VB.Net, Visual Basic, ActionScript, Python et PHP
- Un éditeur de code source « surlignage de syntaxe » intégré
- gabarits de génération de code, qui vous permettent de personnaliser le code source généré selon les spécifications de votre entreprise

### Des transformations pour un développement rapide

- Transformations Model Driven Architecture (MDA) avancée à l'aide gabarits de transformation
- Transformations intégrées pour DDL, C# , Java, EJB et XSD
- Un Modèle indépendant de la plate-forme peut être utilisé pour générer et synchroniser plusieurs modèles spécifiques à la plate-forme, offrant ainsi une augmentation significative de la productivité
- diagramme de transformation XSL, boîte à outils, éditeur et débogueur.

### Analyse d'Exécution Visuelle / Débogage, Vérification et Visualisation

- Exécuter des scripts de build, de test, de débogage, exécuter et de déploiement
- Intégrer le développement et modélisation UML au développement et à la compilation des sources
- Générer des classes de test NUnit et JUnit à partir de classes sources à l'aide de transformations MDA
- Intégrer le processus de test directement dans l'IDE Enterprise Architect
- Déboguer les applications .NET , Mono, Java et Microsoft Native (C, C++ et Visual Basic)
- Concevoir et exécuter des suites Test basées sur les principes de la programmation par contrat
- Débogage de la feuille de style XSL

### Modélisation de base de données

Enterprise Architect vous permet de :

- Reverse engineering à partir de nombreux SGBD populaires, notamment SQL Server, My SQL, Access, PostgreSQL et Oracle
- Modèle tableaux de base de données, des colonnes, des clés, foreign keys et des relations complexes à l'aide UML et d'un profil modélisation de données intégré
- Générer des scripts DDL pour créer des structures de base de données cibles

### Ingénierie de la technologie XML

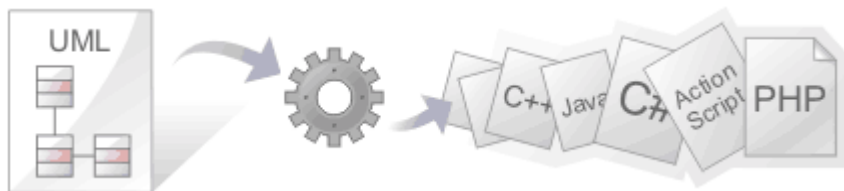
Enterprise Architect vous permet de modéliser, d'effectuer l'ingénierie directe et inverse rapidement sur deux technologies XML W3C clés :

- Schéma XML (XSD)
- Langage de définition de service Web (WSDL)

support de XSD et WSDL est essentielle au développement d'une Service Oriented Architecture (SOA) complète, et le couplage d' UML 2.5 et XML fournit le mécanisme naturel pour la mise en œuvre d'artefacts SOA basés sur XML au sein

d'une organisation.

## Générer Code Source



La génération de code source est le processus de création de code de programmation à partir d'un modèle UML . Cette approche présente de nombreux avantages, car les Paquetages , classes et interfaces de code source sont automatiquement créés et élaborés avec des variables et des méthodes.

Enterprise Architect peut également générer du code à partir d'un certain nombre de modèles comportementaux, notamment diagrammes Statemachine , Séquence et Activity. Il existe un mécanisme gabarit extrêmement flexible qui permet à l'ingénieur de personnaliser entièrement la manière dont le code source est généré, y compris les en-têtes de commentaires dans les méthodes et les classes de collection utilisées.

D'un point de vue technique et qualité, l'avantage le plus convaincant de cette approche est que les modèles UML et donc l'architecture et la conception sont synchronisés avec le code de programmation. Un chemin traçable ininterrompu peut être créé à partir des objectifs, des moteurs commerciaux et des exigences des parties prenantes jusqu'aux méthodes du code de programmation.

### Facilités

Facilité	Description
Langues	<p>Enterprise Architect supporte la génération de code dans chacun de ces langages logiciels :</p> <ul style="list-style-type: none"> <li>• Scénario Action</li> <li>• Ada</li> <li>• ArcGIS</li> <li>• C</li> <li>• C# (pour .NET 1.1, .NET 2.0 et .NET 4.0)</li> <li>• C++ (standard, plus extensions C++ gérées .NET )</li> <li>• Delphes</li> <li>• Java (y compris Java 1.5, Aspects et Génériques)</li> <li>• JavaScript</li> <li>• mFQL</li> <li>• MySQL</li> <li>• PHP</li> <li>• Python</li> <li>• Teradata SQL</li> <li>• Visual Basic</li> <li>• Visual Basic .NET</li> <li>• Script de flux de travail</li> </ul> <p>Vous pouvez également générer du code Hardware Definition Language dans ces langages :</p> <ul style="list-style-type: none"> <li>• VHDL</li> <li>• Verilog</li> </ul>

	<ul style="list-style-type: none"> <li>• SystèmeC</li> </ul>
Éléments	<p>Le code est généré à partir d'éléments de modèle de classe ou d'interface. Vous devez donc créer les éléments de classe et d'interface requis pour la génération. Tous les autres types d'éléments contribuant au code (tels que Statemachines ou les activités) doivent être des éléments enfants d'une classe.</p> <p>Ajoutez des attributs (qui deviennent des variables) et des opérations (qui deviennent des méthodes). Les contraintes et les réceptions sont également prises en charge dans le code.</p>
Paramètres	<p>Avant de générer du code, vous devez vous assurer que les paramètres par défaut pour la génération de code correspondent à vos besoins ; configurez les valeurs par défaut pour qu'elles correspondent à la langue et aux préférences requises.</p> <p>Les préférences que vous pouvez définir incluent les constructeurs et destructeurs par défaut, les méthodes pour les interfaces et les options Unicode pour les langues créées.</p> <p>Des langages tels que Java supportent les « espaces de noms » et peuvent être configurés pour spécifier une racine d'espace de noms.</p> <p>En plus des paramètres par défaut pour la génération de code, Enterprise Architect facilite la définition d'options de génération spécifiques pour chacun des langages pris en charge.</p>
Cadre de code Gabarit	<p>Le Code Gabarit Framework (CTF) vous permet de personnaliser la manière dont Enterprise Architect génère le code source et permet également la génération de langages qui ne sont pas spécifiquement pris en charge par Enterprise Architect .</p>
Chemins locaux	<p>Les noms de chemin locaux vous permettent de remplacer les noms de répertoire par étiquettes .</p>
Code Comportementale	<p>Vous pouvez également générer du code logiciel à partir de trois paradigmes de modélisation comportementale UML :</p> <ul style="list-style-type: none"> <li>• diagrammes d'interaction ( Séquence )</li> <li>• diagrammes d'activité</li> <li>• diagrammes Statemachine (en utilisant Legacy Statemachine Gabarits dans les opérations de génération de code sous « Tâches »)</li> <li>• diagrammes Statemachine (utilisant un artefact Statemachine Exécutable )</li> </ul>
Génération de code en direct	<p>Dans le menu déroulant « Développer &gt; Code source &gt; Options », vous avez la possibilité de mettre à jour votre code source instantanément lorsque vous apportez des modifications à votre modèle.</p>
Tâches	<p>Lorsque vous générez du code, vous effectuez une ou plusieurs de ces tâches :</p> <ul style="list-style-type: none"> <li>• Générer une classe unique</li> <li>• Générer un groupe de classes</li> <li>• Générer un Paquetage</li> <li>• Mettre à jour le contenu Paquetage</li> </ul>

## Notes

- La plupart des outils fournis par Enterprise Architect pour l'ingénierie et le débogage du code sont disponibles dans

les éditions Professional et supérieures d' Enterprise Architect ; la génération de code Comportementale est disponible dans les éditions Unified et Ultimate


- Lorsque la sécurité est activée, vous avez besoin des autorisations d'accès « Générer Code Source et le DDL » et « Rétro-ingénierie à partir du DDL et du code source »

## Générer une classe unique

Avant de générer du code pour une seule classe, vous :

- Compléter la conception de l'élément de modèle (Classe ou Interface)
- Créer des connecteurs d'héritage vers les parents et des associations vers d'autres classes utilisées
- Créez des connecteurs d'héritage vers les interfaces que votre classe implémente ; le système fournit une option pour générer des stubs de fonction pour toutes les méthodes d'interface qu'une classe implémente

### Générer du code pour une seule Classe

Étape	Action
1	Ouvrez le diagramme contenant la classe ou l'interface pour laquelle générer du code.
2	<p>Cliquez sur la classe ou l'interface requise et sélectionnez l'option de ruban « Développer &gt; Code source &gt; Générer &gt; Générer un élément unique » ou appuyez sur F11.</p> <p>La dialogue « Générer du code » s'affiche, grâce à laquelle vous pouvez contrôler comment et où votre code source est généré.</p>
3	<p>Dans le champ « Chemin », cliquez sur le bouton  et sélectionnez un nom de chemin vers lequel votre code source doit être généré.</p>
4	<p>Dans le champ « Langue cible », cliquez sur la flèche déroulante et sélectionnez la langue à générer ; cela devient l'option permanente pour cette classe, donc modifiez-la à nouveau si vous n'effectuez qu'un seul passage dans une autre langue.</p>
5	<p>Cliquez sur le bouton Avancé.</p> <p>La dialogue « Options Object » s'affiche, fournissant des sous-ensembles des pages d'options « Ingénierie du code source » et de langage de code dans la dialogue « Préférences ».</p>
6	<p>Définissez les options personnalisées (pour cette classe uniquement), puis cliquez sur le bouton Fermer pour revenir à la dialogue « Générer le code ».</p>
7	<p>Dans les champs « Import(s) / En-tête(s) », saisissez des instructions d'importation, des #includes ou d'autres informations d'en-tête.</p> <p>Note que dans le cas de Visual Basic, ces informations sont ignorées ; dans le cas de Java, les deux zones de texte d'importation sont fusionnées ; et dans le cas de C++, la première zone de texte d'importation est placée dans le fichier d'en-tête et la seconde dans le fichier corps (.cpp).</p>
8	<p>Cliquez sur le bouton Générer pour créer le code source.</p>
9	<p>Une fois terminé, cliquez sur le bouton Vue pour voir ce qui a été généré.</p> <p>Note que vous devez d'abord configurer votre visualiseur/éditeur par défaut pour chaque type de langue ; vous pouvez également configurer l'éditeur par défaut sur la page « Éditeurs de Code » de la fenêtre Préférences (' Démarrer &gt; Application &gt; Préférences &gt; Préférences &gt; Source Code Engineering &gt; Éditeurs de Code ').</p>



# Générer un groupe de classes

En plus de pouvoir générer du code pour une classe individuelle, vous pouvez également sélectionner un groupe de classes pour la génération de code par lots. Lorsque vous faites cela, vous acceptez toutes les options de génération de code par défaut pour chaque classe de l'ensemble.

## Générer un groupe de classe

Étape	Détail
1	Sélectionnez un groupe de classes et/ou d'interfaces dans un diagramme .
2	<p>Cliquez sur un élément du groupe et sélectionnez l'option de ruban « Développer &gt; Code source &gt; Générer &gt; Générer les éléments sélectionnés » (ou appuyez sur Maj+F11).</p> <p>Si aucun code n'existe pour les éléments sélectionnés, la dialogue « Enregistrer sous » s'affiche, dans laquelle vous spécifiez le chemin d'accès et le nom de chaque fichier de code ; saisissez ces informations et cliquez sur le bouton Enregistrer.</p>
3	<p>La dialogue « Génération par lots » s'affiche, indiquant l'état du processus au fur et à mesure de son exécution (le processus peut être trop rapide pour voir cette dialogue ).</p> <p>Si le code existe déjà pour les éléments de classe sélectionnés et que des modifications ont été apportées au nom ou à la structure de la classe, la boîte de dialogue « Synchroniser l'élément &lt; nom paquetage &gt;.&lt; nom de l'élément&gt; » peut également dialogue ; cette dialogue permet de synchroniser le modèle et le code.</p>

## Notes

- Si l'un des éléments sélectionnés n'est pas une classe ou une interface, l'option de génération de code n'est pas disponible

## Générer un Paquetage

En plus de générer du code source à partir de classes uniques et de groupes de classes, vous pouvez générer du code à partir d'un Paquetage . Cette fonctionnalité fournit des options pour générer de manière récursive du code à partir de Paquetages enfants et générer automatiquement des structures de répertoires basées sur la hiérarchie Paquetage . Cela vous aide à générer du code pour toute une branche de votre modèle de projet en une seule étape.

### Accéder

Ruban	Développer > Code Source > Générer > Générer Tout
Raccourcis Clavier	Ctrl+Alt+K

### Générer du code à partir d'un Paquetage , sur la dialogue Générer le Code Source Paquetage

Étape	Action
1	<p>Dans le champ « Synchroniser », cliquez sur la flèche déroulante et sélectionnez l'option de synchronisation appropriée :</p> <ul style="list-style-type: none"> <li>« Synchroniser le modèle et le code » : le code des classes avec des fichiers existants est synchronisé avec ce fichier ; le code des classes sans fichier existant est généré vers le fichier cible affiché</li> <li>« Écraser le code » : tous les fichiers cibles sélectionnés sont écrasés (générés vers l'avant)</li> <li>« Ne pas générer » : Générer du code uniquement pour les classes sélectionnées qui n'ont pas de fichier existant ; toutes les autres classes sont ignorées</li> </ul>
2	<p>Mettez en surbrillance les classes pour lesquelles générer du code ; laissez non sélectionnées celles pour lesquelles vous ne souhaitez pas générer de code.</p> <p>Si vous souhaitez afficher plus d'informations dans la disposition , vous pouvez redimensionner le dialogue et ses colonnes.</p>
3	<p>Pour qu'Enterprise Architect génère automatiquement des répertoires et des noms de fichiers en fonction de la hiérarchie Paquetage , cochez la case « Générer automatiquement des fichiers » ; cela active le champ « Répertoire racine », dans lequel vous sélectionnez un répertoire racine sous lequel les répertoires sources doivent être générés.</p> <p>Par défaut, la fonctionnalité « Générer automatiquement des fichiers » ignore tous les chemins de fichiers déjà associés à une classe ; vous pouvez modifier ce comportement en cochant également la case « Conserver les chemins de fichiers existants ».</p>
4	<p>Pour inclure le code de tous les sous-packages dans la sortie, cochez la case 'Inclure Paquetages Enfants' .</p>
5	<p>Cliquez sur le bouton Générer pour commencer à générer du code.</p> <p>Au fur et à mesure de la génération du code, Enterprise Architect affiche des messages de progression. Si une classe nécessite un nom de fichier de sortie, le système vous promps à en saisir un au moment opportun (en supposant que l'option Générer automatiquement les fichiers n'est pas sélectionnée). Par</p>

exemple, si les classes sélectionnées incluent des classes partielles, une prompt s'affiche pour vous demander de saisir le nom de fichier dans lequel générer le code pour la deuxième classe partielle.

## Plus d'informations sur le dialogue

Option	Action
Paquetage de racines	Vérifiez le nom du Paquetage pour lequel le code doit être généré.
Synchroniser	Sélectionnez les options qui spécifient comment les fichiers existants doivent être régénérés.
Générer automatiquement des fichiers	Spécifiez si Enterprise Architect doit générer automatiquement des noms de fichiers et des répertoires, en fonction de la hiérarchie Paquetage .
Répertoire racine	Si Générer automatiquement les fichiers est sélectionnée, affichez le chemin sous lequel les structures de répertoire générées sont créées.
Conserver les chemins de fichiers existants	Si Générer automatiquement les fichiers est sélectionnée, indiquez si vous souhaitez utiliser les chemins de fichiers existants associés aux classes. Si l' Générer automatiquement les fichiers n'est pas sélectionnée, Enterprise Architect génère automatiquement le code de classe vers les chemins déterminés, que les fichiers sources soient déjà associés aux classes ou non.
Inclure tous Paquetages enfants	Générez également du code pour toutes les classes dans tous les sous-packages du Paquetage cible dans la liste. Cette option facilite la génération récursive de code pour un Paquetage donné et ses sous-Packages.
Sélectionner les objets à Générer	Répertoriez toutes les classes disponibles pour la génération de code sous les Paquetages cibles ; seul le code des classes sélectionnées (mises en surbrillance) est généré. Les classes sont répertoriées avec leur fichier source cible.
Sélectionner tout	Marquez toutes les classes de la liste comme sélectionnées.
Ne rien sélectionner	Marquer toutes les classes de la liste comme non sélectionnées.
Générer	Démarrer la génération du code pour toutes les classes sélectionnées.
Annuler	Quittez la dialogue ' Générer Paquetage Source Code ' ; aucun code de classe n'est généré.

## Mettre à jour le contenu Paquetage

Outre la génération et l'importation de code, Enterprise Architect offre la possibilité de synchroniser le modèle et le code source, en créant un modèle qui représente les dernières modifications du code source et vice versa. Vous pouvez utiliser soit le modèle comme source, soit le code comme source.

Le comportement et les actions de la synchronisation dépendent des paramètres que vous avez sélectionnés dans la page « Attributs et opérations » de la dialogue « Préférences ». En utilisant ces paramètres, vous pouvez soit protéger, soit supprimer automatiquement les informations du modèle qui ne sont pas présentes dans le code, et prompt une décision sur fonctionnalités du code qui ne sont pas présentes dans le modèle. Dans ces deux exemples, les cases à cocher appropriées ont été sélectionnées pour une protection maximale des données :

- Vous avez généré du code source, mais vous avez apporté des modifications ultérieures au modèle ; lorsque vous générez à nouveau du code, Enterprise Architect ajoute tous les nouveaux attributs ou méthodes au code source existant, laissant intact ce qui existe déjà, ce qui signifie que les développeurs peuvent travailler sur le code source, puis générer des méthodes supplémentaires selon les besoins à partir du modèle, sans que leur code ne soit écrasé ou détruit
- Vous avez peut-être apporté des modifications à un fichier de code source, mais le modèle contient notes et des caractéristiques détaillées que vous ne souhaitez pas perdre ; en synchronisant le code source dans le modèle, vous importez des attributs et des méthodes supplémentaires, mais ne modifiez pas les autres éléments du modèle.

Grâce aux méthodes de synchronisation, il est simple de maintenir le code source et les éléments du modèle à jour et synchronisés.

### Accéder

Ruban	Développer > Code source > Synchroniser > Synchroniser Paquetage
-------	--

### Synchronisez le contenu Paquetage avec le code source

Champ/Bouton	Action
Type de mise à jour	Sélectionnez le bouton radio pour effectuer une ingénierie directe ou une ingénierie inverse des classes Paquetage , selon le cas.
Inclure paquetages enfants dans la génération	Cochez la case pour inclure Paquetages enfants dans la synchronisation.
OK	<p>Cliquez sur le bouton pour démarrer la synchronisation.</p> <p>Enterprise Architect utilise les noms de répertoire spécifiés lors de la première importation/génération de la source du projet et met à jour le modèle ou le code source en fonction de l'option choisie. Si :</p> <ul style="list-style-type: none"> <li>• Exécution de la synchronisation vers l'avant AND</li> <li>• Il existe des différences entre le modèle et le code AND</li> <li>• La case à cocher « Lors de la synchronisation en avant, prompt la suppression fonctionnalités de code qui ne sont pas dans le modèle » est sélectionnée dans la boîte de dialogue « Options - Attributs et opérations »</li> </ul> <p>Ensuite, la dialogue « Synchroniser l'élément &lt; nom paquetage &gt;.&lt; nom de l'élément&gt; » s'affiche.</p>

	Dans le cas contraire, aucune autre action n'est requise.
--	---

## Notes

- La synchronisation du code ne modifie pas les corps des méthodes ; le code comportemental ne peut pas être synchronisé et la génération de code ne fonctionne que lors de la génération de l'intégralité du fichier
- Dans les éditions Corporate , Unified et Ultimate d' Enterprise Architect , si la sécurité est activée, vous devez disposer de l'autorisation « Générer Code Source et DDL » pour synchroniser le code source avec les éléments du modèle.

## Synchroniser Modèle et le code

Vous pouvez soit :

- Synchroniser le code d'un Paquetage de Classes avec le modèle dans la fenêtre Navigateur , ou
- Régénérer le code à partir d'un lot de classes dans le modèle

Dans de tels processus, il peut y avoir des éléments dans le code qui ne sont pas présents dans le modèle.

Si vous souhaitez piéger ces éléments et les résoudre manuellement, cochez la case « Lors de la synchronisation directe, prompt la suppression fonctionnalités de code non présentes dans le modèle » dans la boîte de dialogue « Options - Attributes et opérations », de sorte que la dialogue « Synchroniser l'élément < nom paquetage >.< nom de l'élément > » s'affiche, fournissant des options pour répondre à chaque élément.

### Synchroniser Items

Bouton	Détail
Sélectionner tout	Mettez en surbrillance et sélectionnez tous les éléments dans la colonne Fonctionnalité .
Tout Effacer	Désélectionnez et supprimez la surbrillance de tous les éléments de la colonne Fonctionnalité .
Supprimer	Marquez les fonctionnalités de code sélectionnées à supprimer du code (la valeur dans la colonne Action change en Supprimer).
Réaffecter	Marquez les fonctionnalités de code sélectionnées à réaffecter aux éléments du modèle. Cela n'est possible que lorsqu'un élément de modèle approprié est présent et n'est pas déjà défini dans le code. La dialogue Sélectionner la Fonctionnalité de classe correspondante s'affiche. Vous pouvez alors sélectionner la classe à laquelle réaffecter la fonctionnalité . Cliquez sur le bouton OK pour marquer la fonctionnalité à réaffecter.
Ignorer	Marquez les éléments de code sélectionnés non présents dans le modèle à ignorer complètement (valeur par défaut ; la valeur dans la colonne Action reste ou passe à <none> ).
Réinitialiser par défaut	Réinitialisez les éléments sélectionnés sur Ignorer (la valeur dans la colonne Action passe à <none> ).
OK	Apportez les modifications attribuées aux éléments et fermez le dialogue .

# Namespaces

Les langages tels que Java supportent les structures de Paquetage ou les espaces de noms. Dans Enterprise Architect vous pouvez spécifier un Paquetage comme racine d'espace de noms, ce qui indique où commence la structure d'espace de noms de votre modèle de classe ; tous Paquetages subordonnés situés sous une racine d'espace de noms formeront la hiérarchie d'espace de noms pour les classes et les interfaces contenues.

Pour définir un Paquetage comme racine d'espace de noms, cliquez sur le Paquetage dans la fenêtre Navigateur et sélectionnez l'option de ruban « Développer > Code source > Options > Définir comme racine Namespace ». L'icône Paquetage dans la fenêtre Navigateur change pour afficher un coin coloré indiquant que ce Paquetage est une racine d'espace de noms.



Le code source Java généré, par exemple, ajoutera automatiquement une déclaration de Paquetage au début du fichier généré, indiquant l'emplacement de la classe dans la hiérarchie de Paquetage sous la racine de l'espace de noms.

Pour effacer une racine d'espace de noms existante, cliquez sur la racine d'espace de noms de Paquetage dans la fenêtre Navigateur et décochez l'option de ruban « Développer > Code source > Options > Définir comme racine Namespace ».

Pour afficher une liste d'espaces de noms, sélectionnez l'option de ruban « Paramètres > Données de référence > Paramètres > Racines Namespace » ; la dialogue « Namespaces » s'affiche. Si vous double-cliquez sur un espace de noms dans la liste, le Paquetage est mis en surbrillance dans la fenêtre Navigateur ; vous pouvez également cliquer-droit sur l'espace de noms et sélectionner l'option « Localiser Paquetage dans Navigateur ».

Vous pouvez également effacer la racine de l'espace de noms sélectionné en sélectionnant l'option « Effacer l'attribut Namespace ».

Pour omettre un Paquetage subordonné d'une définition d'espace de noms, sélectionnez l'option de ruban « Développer > Code source > Options > Supprimer Namespace » ; pour inclure à nouveau le Paquetage dans l'espace de noms, désélectionnez l'option de ruban.

## Notes

- Lors de la génération de code, tout nom de Paquetage contenant des caractères d'espacement est automatiquement traité comme une racine d'espace de noms.

## Importation du code source



La possibilité de visualiser simultanément le code de programmation et les modèles dont il est dérivé apporte de la clarté à la conception d'un système. L'une des fonctionnalités pratiques d'ingénierie de code d'Enterprise Architect est la possibilité de procéder à une rétro-ingénierie du code source dans un modèle UML. Une large gamme de langages de programmation sont pris en charge et il existe des options qui régissent la manière dont les modèles sont générés. Une fois le code dans le modèle, il est possible de le maintenir synchronisé avec le modèle, que les modifications aient été apportées directement dans le code ou dans le modèle lui-même. Les structures de code sont mappées dans leurs représentations UML ; par exemple, une classe Java est mappée dans un élément de classe UML, les variables sont définies comme des attributs, les méthodes modélisées comme des opérations et les interactions entre les classes Java représentées par les connecteurs appropriés.

La représentation du code de programmation sous forme de constructions de modèles vous aide à mieux comprendre la structure du code et la manière dont il implémente la conception, architecture et les exigences, et finalement comment il fournit la valeur commerciale.

Il est important de noter que si un système n'est pas bien conçu, la simple importation de la source dans Enterprise Architect ne le transforme pas en un modèle UML facilement compréhensible. Lorsque vous travaillez avec un système mal conçu, il est utile d'évaluer le code en unités gérables en examinant les Paquetages ou éléments de modèle individuels générés à partir du code ; par exemple, en faisant glisser une classe spécifique d'intérêt sur un diagramme, puis en utilisant l'option « Insérer des éléments associés » à un niveau pour déterminer les relations immédiates entre cette classe et d'autres classes. À partir de ce point, il est possible de créer des cas d'utilisation qui identifient l'interaction entre les classes de code source, offrant ainsi une vue d'ensemble du fonctionnement de l'application.

Plusieurs options guident la manière dont le code est rétroconçu, notamment si les commentaires sont importés dans notes et comment ils sont formatés, comment les méthodes de propriété sont reconnues et si des relations de dépendance sont créées pour les types de retour d'opération et de paramètre.

## Propriété des droits d'auteur

Les situations qui se prêtent généralement à la rétro-ingénierie ont tendance à fonctionner sur du code source qui :

- Vous avez déjà développé
- Fait partie d'une bibliothèque tierce pour laquelle vous avez obtenu l'autorisation d'utiliser
- Fait partie d'un cadre que votre organisation utilise
- Est développé quotidiennement par vos développeurs

Si vous examinez un code dont vous ou votre organisation n'êtes pas propriétaire ou pour lequel vous n'avez pas l'autorisation spécifique de copier et de modifier, vous devez vous assurer que vous comprenez et respectez les restrictions de droits d'auteur sur ce code avant de commencer le processus de rétro-ingénierie.

## Langues supportées pour Rétro-ingénierie

Langue
Scénario Action



Ada 2012 (éditions Unified et Ultimate )
C
C#
C++
CORBA IDL ( MDG Technologie )
Delphes
Java
PHP
Python
SystemC (éditions Unified et Ultimate )
Verilog (éditions Unified et Ultimate )
VHDL (éditions Unified et Ultimate )
Visual Basic
Visual Basic .NET

## Notes

- Rétro-ingénierie est supporté dans les éditions Professional , Corporate , Unified et Ultimate d' Enterprise Architect
- Si la sécurité est activée, vous devez disposer de l'autorisation « Reverse engineering à partir de DDL et du code source » pour effectuer la rétro-ingénierie du code source et synchroniser les éléments du modèle avec le code
- À l'aide d' Enterprise Architect , vous pouvez également importer certains types de fichiers binaires, tels que les fichiers Java .jar et les fichiers .NET PE.
- Rétro-ingénierie d'autres langages est actuellement disponible grâce à l'utilisation de MDG Technologies répertoriées sur les pages MDG Technologie du site Web Sparx Systems

# Projets d'importation

Enterprise Architect support d'importer des projets logiciels créés dans Visual Studio, Mono, Eclipse et NetBeans. L'importation et le travail sur des projets dans Enterprise Architect présentent de nombreux avantages, notamment l'accès immédiat aux outils modélisation et fonctionnalités de gestion réputés d' Enterprise Architect , mais également l'accès aux outils de développement tels que la simulation, le débogage et le profilage.

## Accéder

Ruban	Développer > Code source > Solutions > Importer un <type de projet>
-------	---

## Importer une solution Visual Studio

Cette option vous permet d'importer un ou plusieurs projets à partir d'un fichier de solution Visual Studio existant ou d'une instance en cours d'exécution de Visual Studio. L' assistant génère un modèle de classe pour chacun des projets et les Scripts d'Analyseur appropriés pour chaque configuration de Visual Studio.

## Importer une solution mono

Cette option vous permet d'importer des projets Mono à partir d'un fichier solution. La dialogue qui s'affiche est la même que la dialogue « Importation de Visual Studio », mais vous pouvez choisir de cibler Linux ou Windows . L' assistant génère un modèle de classe pour chacun des projets et les configure pour le débogage. Les Scripts d'Analyseur générés font référence à msbuild pour construire les projets.

## Importer un projet Eclipse

L' Assistant Eclipse peut effectuer une rétro-ingénierie d'un projet Java décrit par son fichier Eclipse .project et sa compilation ANT. La fonctionnalité produira un modèle de classe UML et Scripts d'Analyseur pour chacune des cibles ANT que vous sélectionnez. Le processus générera également un script pour chaque protocole de débogage que vous sélectionnez via l' Assistant . Vous aurez le choix entre JDWP (Java Débogueur Wire Protocol), adapté aux serveurs, et JVMTI (Java Virtual Machine Tools Interface), adapté aux applications Java autonomes. Ces scripts doivent être utilisés pour déboguer le projet dans Enterprise Architect .

## Importer un projet NetBeans

L' Assistant NetBeans peut effectuer une rétro-ingénierie d'un projet Java décrit par un fichier de projet XML NetBeans et une compilation ANT. L' Assistant crée un modèle de classe UML du projet et Scripts d'Analyseur pour chacune des cibles ANT que vous sélectionnez. Le processus génère également un script pour chaque protocole de débogage que vous sélectionnez via l' Assistant . Ces scripts doivent être utilisés pour déboguer le projet dans Enterprise Architect . Vous aurez le choix entre JDWP (Java Débogueur Wire Protocol), adapté aux serveurs, et JVMTI (Java Virtual Machine Tools Interface), qui convient aux applications Java autonomes.

## Options d'importation

Lorsque vous choisissez d'importer une solution Visual Studio ou Mono, la dialogue « Importer une solution Visual Studio » s'affiche. Remplissez les champs comme indiqué dans ce tableau .

Lorsque vous choisissez d'importer une solution Eclipse ou Netbeans, l'écran de démarrage Assistant approprié s'affiche. Parcourez les écrans en suivant les prompts s'affichent sur chaque écran.

Option	Description
<liste des projets>	Après avoir sélectionné le fichier de solution, les projets de la solution sont répertoriés dans le panneau. Sélectionnez les projets à importer par l' Assistant . Vous pouvez utiliser le bouton Tous pour sélectionner tous les projets et le bouton Aucun pour effacer la sélection de projets.
Sélectionner le fichier de solution	Recherchez et sélectionnez le fichier de solution à partir duquel effectuer l'importation. Les fichiers de solution Mono et les fichiers de solution Visual Studio ont une extension de fichier .sln.
Effectuer un Exécuter à Sec	Sélectionnez cette option pour effectuer l'importation en tant exécuter à sec, afin de vérifier les éventuelles erreurs dans le processus ou la sortie avant de répéter l'importation pour modifier le contenu du modèle. Cliquez sur le bouton Journal Vue pour vérifier le log de l'importation.
Créer Paquetage par fichier	Sélectionnez cette option pour effectuer l'importation avec une granularité plus fine, en créant un Paquetage distinct pour chaque fichier.
Importer	Cliquez sur ce bouton pour démarrer le processus d'importation.
Demande de définitions de macros manquantes	Non applicable aux importations de Mono Solution. Pour les projets C++ dans Visual Studio, l'analyseur peut rencontrer des macros non reconnues. Si vous sélectionnez cette option, vous serez averti lorsqu'un tel événement se produit et aurez la possibilité de définir la macro. Si vous ne sélectionnez pas cette option, le modèle de classe résultant peut manquer de certains éléments.
Créer Diagramme pour chaque Paquetage	Lorsque cette option est sélectionnée, un diagramme de classe est créé, représentant le modèle de classe pour chaque Paquetage . Le résultat est un modèle plus grand mais plus coloré. La désélection de cette option entraînera l'omission de la création diagramme et l' exécuter de l'importation sera plus rapide.
Générer Scripts d'Analyseur	Pour les solutions Visual Studio, la sélection de cette option générera Scripts d'Analyseur pour chaque configuration de projet en plus des scripts pour chaque configuration de solution. Les scripts permettront de construire et de déboguer le(s) programme(s) décrit(s) par la solution immédiatement après la fin de l'importation. Cochez la case « Windows » ; si vous ne sélectionnez pas cette option, aucune fonctionnalités Analyseur d'Exécution ne sera configurée. Pour les solutions Mono, cette option vous permet de cibler Linux ou Windows . Si vous sélectionnez Linux, il est supposé que la machine sur laquelle s'exécute Enterprise Architect est Linux, que la plate-forme (Java ou Mono) y est installée et que les programmes compilés exécuter sous Linux.
Projet de démarrage	Lorsque cette option est sélectionnée, le script de ce projet deviendra le modèle par défaut. Les outils de débogage, le ruban Exécuter et les boutons de la barre d'outils cibleront automatiquement ce programme.



## Importer le code source

Vous pouvez importer du code source dans votre modèle Enterprise Architect pour procéder à la rétro-ingénierie d'un module. Au fur et à mesure de l'importation, Enterprise Architect fournit des informations sur la progression. Une fois tous les fichiers importés, Enterprise Architect effectue une deuxième passe pour résoudre les associations et les relations d'héritage entre les classes importées.

### Procédure - Importer le code source

Étape	Action
1	Dans la fenêtre Navigateur , sélectionnez (ou ajoutez) un diagramme dans lequel importer les classes.
2	<p> Cliquez sur l'arrière-plan diagramme et soit :</p> <ul style="list-style-type: none"><li> Sélectionnez l'option de ruban « Développer &gt; Code source &gt; Fichiers » et cliquez sur la langue appropriée, ou</li><li> Si la barre d'outils de génération de code s'affiche, cliquez sur la flèche déroulante « Importer » et sélectionnez la langue à importer</li></ul> <p>La liste des langues inclura toutes les langues personnalisées pour lesquelles vous avez créé des structures de modèle.</p>
3	Dans le navigateur de fichiers qui apparaît, recherchez et sélectionnez un ou plusieurs fichiers de code source à importer.
4	Cliquez sur le bouton Ouvrir pour démarrer le processus d'importation.

## Notes sur l'importation du code source

Vous pouvez importer du code dans votre projet Enterprise Architect, dans une gamme de langages de programmation. Enterprise Architect supporte la plupart des constructions et des mots-clés pour chaque langage de programmation. Vous sélectionnez le type de fichier source approprié pour le langage, comme code source à importer.

Si vous avez besoin support une fonctionnalité particulière pour laquelle vous estimez qu'elle manque, veuillez contacter Sparx Systems.

### Notes

- Lors de la rétro-ingénierie d'attributs avec des substitutions de paramètres (attributs basés sur des modèles) :
  - Si une classe avec des définitions de paramètres gabarit appropriées est trouvée, un connecteur d'association est créé et ses substitutions de paramètres sont configurées
  - Un connecteur d'association est également créé si une entrée correspondante est définie comme une classe de collection ou dans l'option « Classes de collection supplémentaires » (pour C#, C++ et Java) ; pour un exemple, voir *Exemple Utilisation des classes de collection*

### notes sur le langage de programmation

Langue	Notes
ActionScript	Type de fichier source approprié : fichier .as code.
C	Type de fichier source approprié : fichiers d'en-tête .h et/ou fichiers .c. Lorsque vous sélectionnez un fichier d'en-tête, Enterprise Architect recherche automatiquement le fichier d'implémentation .c correspondant à importer, en fonction des options d'extension et du chemin de recherche spécifiés dans les options C. Enterprise Architect ne développe pas les macros qui ont été utilisées, celles-ci doivent être ajoutées à la liste interne des macros de langage.
C++	Type de fichier source approprié : fichier d'en-tête .h. Enterprise Architect recherche automatiquement le fichier d'implémentation .cpp en fonction de l'extension et du chemin de recherche définis dans les options C++ ; lorsqu'il trouve le fichier d'implémentation, il peut l'utiliser pour résoudre les noms de paramètres et notes de méthode si nécessaire. Lors de l'importation de code source C++, Enterprise Architect ignore les déclarations de pointeur de fonction. Pour les importer dans votre modèle, vous pouvez créer un typedef pour définir un type de pointeur de fonction, puis déclarer des pointeurs de fonction en utilisant ce type ; les pointeurs de fonction déclarés de cette manière sont importés en tant qu'attributs du type de pointeur de fonction. Enterprise Architect ne développe pas les macros qui ont été utilisées ; celles-ci doivent être ajoutées à la liste interne des macros de langage.
C#	Type de fichier source approprié : .cs.
Delphes	Type de fichier source approprié : .pas.

<p>Java</p>	<p>Type de fichier source approprié : .java. Enterprise Architect supporte les extensions du langage AspectJ.</p> <div data-bbox="520 304 1043 734" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <pre> «aspect» ThingObserving - observers: Vector = new Vector() + addObserver(Thing, Thing) : void + removeObserver(Thing, ThingObserver) : void - updateObserver(Thing, ThingObserver) : void «advice» + after(Thing) : void   changes(t) «pointcut» - changes(Thing) : void   target(t) &amp;&amp; call(Void Thing.set*(int))                     </pre> </div> <p>Les aspects sont modélisés à l'aide de classes avec le stéréotype aspect ; ces aspects peuvent alors contenir des attributs et des méthodes comme pour une classe normale.</p> <p>Si un attribut ou une opération intertype est requis, vous pouvez ajouter une étiquette 'className' dont la valeur est le nom de la classe à laquelle elle appartient.</p> <p>Les points de coupure sont définis comme des opérations avec le stéréotype &lt;&lt;pointcut&gt;&gt; et peuvent se produire dans n'importe quelle classe, interface ou aspect Java ; les détails du point de coupure sont inclus dans le champ « comportement » de la méthode.</p> <p>Un conseil est défini comme une opération avec le stéréotype &lt;&lt;conseil&gt;&gt; ; le point de coupure sur lequel ce conseil opère se trouve dans le champ « comportement » et agit comme une partie de la signature unique de la méthode. afterAdvice peut également avoir l'une des Valeur Étiquetées revenant ou jetant.</p>
<p>PHP</p>	<p>Type de fichier source approprié : .php, .php4 ou .inc. La syntaxe de condition imbriquée si est activée.</p>
<p>Python</p>	<p>Type de fichier source approprié : .py.</p>
<p>Visual Basic</p>	<p>Type de fichier source approprié : fichier de classe .cls.</p>
<p>Visual Basic .NET</p>	<p>Type de fichier source approprié : fichier de classe .vb.</p>


## Script d'importation de ressources

Enterprise Architect supporte l'importation et l'exportation de Scripts de ressources Microsoft Windows (sous forme de fichiers .rc), qui contiennent les définitions dialogue Win32® (celles avec le stéréotype « win32Dialog ») pour l'interface utilisateur graphique d'une application. Les ressources Dialogue sont importées et exportées pour une langue spécifique, en utilisant par défaut les paramètres régionaux du système informatique actuel.

### Accéder

Ruban	Développer > Code source > Fichiers > Importer un script de ressources
Raccourcis Clavier	F7 (synchroniser l'élément avec le code)


### Importer des ressources dialogue à partir d'un fichier .rc

Option	Action
Fichier de ressources	Cliquez sur le bouton  et localisez le fichier .rc pour importer les éléments de l'écran.
ID de la ressource	Soit: <ul style="list-style-type: none"> <li>Laissez la valeur par défaut « Tous » pour importer tous les éléments de l'écran à partir du fichier, ou</li> <li>Cliquez sur la flèche déroulante et sélectionnez l' ID d'écran d'une dialogue spécifique à importer</li> </ul>
Langue	Cliquez sur la flèche déroulante et sélectionnez la version linguistique (par exemple, Anglais - States Unis) du le dialogue à importer.
Importer	Cliquez sur ce bouton pour importer les écrans du fichier de ressources. La progression de l'importation est indiquée dans le champ situé sous le champ « Langue ».

### Exporter un dialogue vers un fichier .rc

Option	Action
ID d'écran	Valeur par défaut de l' ID Win32UI Valeur Étiquetée de l'élément d'écran sélectionné. (Si le dialogue n'a pas cet ID , ouvrez la page 'Win32UI' de la dialogue ' Propriétés ' de l'élément et fournissez une valeur pour l' étiquette ID .)



Fichier de ressources	<p>Cliquez sur le bouton  et recherchez le fichier .rc dans lequel exporter les éléments de l'écran.</p> <p>Si l'élément a été précédemment importé, ce champ correspond par défaut au fichier source.</p>
Langue	<p>Cliquez sur la flèche déroulante et sélectionnez la version linguistique (par exemple, Anglais - States Unis) du dialogue exporté.</p>
Exporter	<p>Cliquez sur ce bouton pour exporter les écrans du fichier de ressources.</p> <p>La progression de l'exportation est indiquée dans le champ situé sous le champ « Langue ».</p>

## Notes

- Les nouvelles boîtes de dialogue sont exportées vers un fichier .rc existant
- Lors d'une exportation vers un fichier .rc existant, aucune boîte de dialogue n'est jamais supprimée du fichier, même lorsqu'elles sont supprimées du modèle
- Lors d'une importation, aucune boîte de dialogue n'est supprimée du modèle, même si elle est omise du fichier .rc d'origine

## Importer une structure de répertoire

Vous pouvez importer tous les fichiers sources dans une structure de répertoire complète, ce qui vous permet d'importer ou de synchroniser plusieurs fichiers dans une arborescence de répertoires en un seul passage.

Enterprise Architect crée les Paquetages et diagrammes nécessaires pendant le processus d'importation.

### Accéder

Ruban	Développer > Code source > Fichiers > Importer le répertoire source
Raccourcis Clavier	Ctrl+Maj+U

### Importer une structure de répertoire à l'aide de la dialogue « Importer le répertoire source »

Champ	Action
Répertoire racine	Type ou recherchez le nom du répertoire à importer.
Type source	Type ou sélectionnez dans la liste déroulante la langue de codage des fichiers à importer dans le répertoire source.
Déposer	Type ou sélectionnez dans la liste déroulante les extensions de fichier à inclure dans l'importation. Utilisez un « ; » pour séparer les valeurs.
Effectuer un Exécuter à Sec	Si vous souhaitez effectuer l'importation en tant exécuter à sec lorsque vous cliquez sur le bouton OK , cochez cette case. Une fois le traitement terminé, cliquez sur le bouton Vue Log pour vérifier le résultat prévu du processus.
Traiter les sous-répertoires de manière récursive	Si vous souhaitez inclure le contenu des sous-répertoires dans le processus d'importation, cochez cette case.
Importer des composants depuis	Si vous souhaitez importer des fichiers supplémentaires (comme décrit dans la dialogue « Importer les types de composants »), cochez cette case. Vous complétez ensuite l' prompt pour spécifier la provenance des composants.
Ne pas importer de membres privés	Si vous souhaitez exclure les membres privés du modèle lors de l'importation de bibliothèques, cochez cette case.
Demande de définitions de macros manquantes	Lors de l'importation, l'analyseur peut rencontrer des macros non reconnues. Si vous cochez cette case, vous serez averti lorsqu'un tel événement se produit et vous aurez la possibilité de définir la macro. Si vous ne cochez pas cette option, la structure Paquetage résultante peut manquer de certains éléments.
Structure Paquetage	Sélectionnez le bouton radio approprié pour créer un Paquetage pour chaque répertoire, chaque espace de noms ou chaque fichier ; cela peut être restreint en

	fonction du type de source sélectionné.
Créer Diagramme pour chaque Paquetage	Cochez cette case pour créer un diagramme dans chaque Paquetage créé lors de l'import. Cliquez sur le bouton Options pour identifier les fonctionnalités des éléments à inclure dans les diagrammes .
Synchronisation	<p>Sélectionnez le bouton radio approprié pour synchroniser les classes existantes ou écraser les classes existantes.</p> <p>Si une classe de modèle est trouvée qui correspond à celle du code :</p> <ul style="list-style-type: none"><li>• « Synchroniser » met à jour la classe du modèle pour inclure les détails de celle du code, ce qui préserve les informations non représentées dans le code, telles que l'emplacement des classes dans diagrammes</li><li>• « Overwrite » supprime la classe du modèle et en génère une nouvelle à partir du code ; les informations supplémentaires ne sont pas conservées.</li></ul> <p>Si l'option « Utiliser les horodatages » est sélectionnée, la représentation avec l'horodatage le plus récent (modèle ou code) aura la priorité.</p>
Supprimer les classes non trouvées dans le code	<p>Sélectionnez le bouton radio approprié pour spécifier comment gérer les classes de modèle existantes qui ne sont pas présentes dans le code importé.</p> <ul style="list-style-type: none"><li>• « Ne jamais supprimer » conserve toutes les classes existantes dans le modèle.</li><li>• « Invite à l'action » vous permet de réviser les classes individuellement</li><li>• « Toujours supprimer » supprime du modèle toute classe qui n'est pas présente dans le code importé.</li></ul>
OK	Cliquez sur ce bouton pour démarrer l'importation.

# Importer un module binaire

Enterprise Architect vous permet de procéder à la rétro-ingénierie de certains types de modules binaires.

## Accéder

Ruban	Développer > Code source > Fichiers > Importer un module binaire
-------	--

## Utiliser

Actuellement, les types autorisés sont :

- Archive Java (.jar)
- Fichier PE .NET (.exe, .dll) - Les fichiers DLL et EXE natifs Windows ne sont pas pris en charge, seuls les fichiers PE contenant des données d'assemblage .NET
- Fichier de langue intermédiaire (.il)

Enterprise Architect crée les Paquetages et diagrammes nécessaires pendant le processus d'importation ; la sélection de la case à cocher « Ne pas importer les membres privés » exclut les membres privés des bibliothèques de l'importation dans le modèle.

Lors de l'importation de fichiers .NET , vous pouvez importer via la réflexion ou via le désassemblage, ou laisser le système sélectionner la meilleure méthode - cela peut entraîner l'utilisation des deux types.

L'importateur basé sur la réflexion s'appuie sur un programme .NET et nécessite l'installation de l'environnement d'exécution .NET .

L'importateur basé sur le désassembleur s'appuie sur un programme Windows natif appelé Ildasm.exe, qui est un outil fourni avec le SDK MS .NET ; le SDK peut être téléchargé à partir du site Web de Microsoft.

Un choix de méthodes d'importation est disponible car certains fichiers ne sont pas compatibles avec la réflexion (comme mscorlib.dll) et ne peuvent être ouverts qu'à l'aide du désassembleur ; cependant, l'importateur basé sur la réflexion est généralement beaucoup plus rapide.

Vous pouvez également configurer :

- S'il faut synchroniser ou écraser les classes existantes lorsqu'elles sont trouvées ; si une classe de modèle est trouvée correspondant à celle du fichier :
  - Synchroniser les mises à jour de la classe modèle pour inclure les détails de celle du fichier, qui conserve les informations non représentées dans le fichier, telles que l'emplacement des classes dans diagrammes
  - Overwrite supprime la classe du modèle et en génère une nouvelle à partir du fichier, ce qui supprime et ne remplace pas les informations complémentaires
- Faut-il créer un diagramme pour chaque Paquetage
- Ce qui est affiché sur diagrammes créés par l'importation

## Classes non trouvées lors de l'importation

Lors de la rétro-ingénierie de votre code, il peut arriver que des classes soient délibérément supprimées de votre code source.

La fonctionnalité « Importer le répertoire source » garde une trace des classes avec lesquelles elle s'attend à se synchroniser et, dans la dialogue « Importer la structure du répertoire », fournit des options sur la façon de gérer les classes qui n'ont pas été trouvées.

Vous pouvez sélectionner l'option appropriée pour Enterprise Architect, à la fin de l'importation, ignore les classes manquantes, les supprime automatiquement ou vous prompt à les gérer.

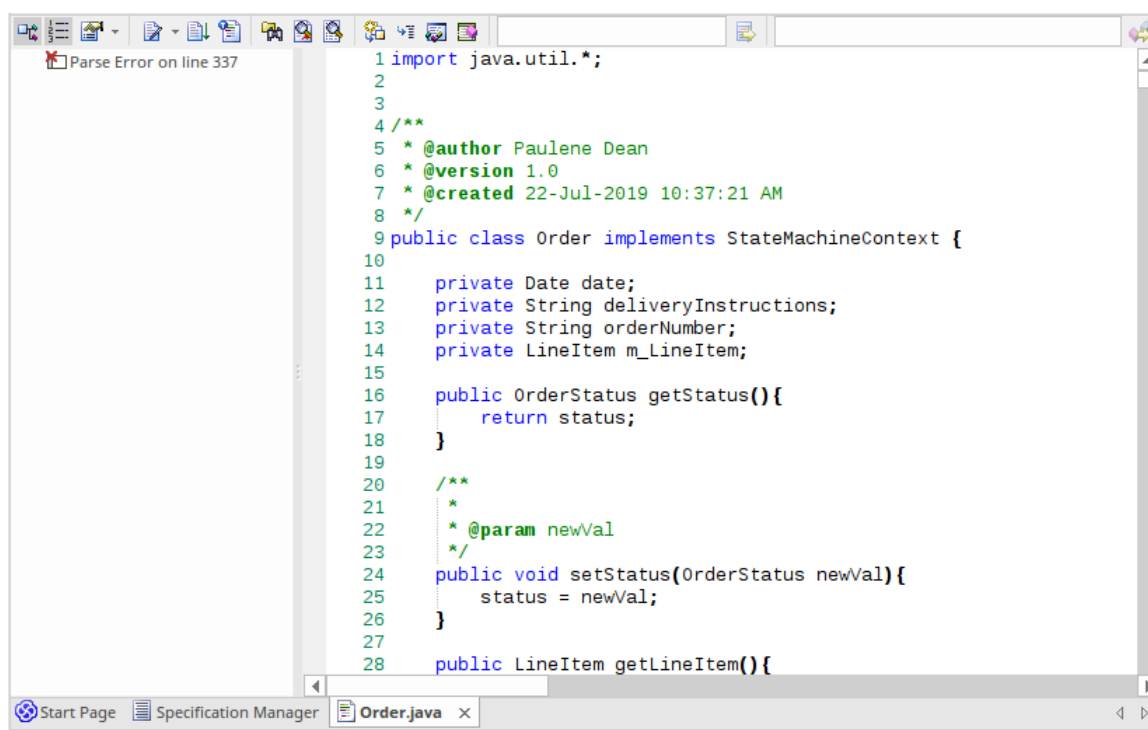
Dans la dialogue « Importer la structure du répertoire », si vous sélectionnez le bouton radio « Demander Action » pour révision manuellement les classes manquantes, une dialogue s'affiche dans laquelle vous spécifiez la gestion de chaque classe manquante dans le code importé.

Par défaut, toutes les classes sont marquées pour suppression ; pour conserver une ou plusieurs classes, sélectionnez-les et cliquez sur le bouton Ignorer.

## Modification du code source

Enterprise Architect contient un éditeur de code source riche en fonctionnalités qui vous permet de visualiser, de modifier et de gérer votre code source directement dans l'outil. Une fois le code source généré pour une ou plusieurs classes, il peut être visualisé dans cet environnement d'édition flexible. Le fait de voir le code dans le contexte des modèles UML dont il est dérivé apporte de la clarté au code et aux modèles, et comble le fossé entre la conception et la mise en œuvre qui a historiquement introduit des erreurs dans les systèmes logiciels.

L'Éditeur de Code source est complet, avec une arborescence de structure pour une navigation facile des attributs, propriétés et méthodes. Les numéros de ligne peuvent être affichés et les options de surbrillance de la syntaxe peuvent être configurées. De nombreuses fonctionnalités que les ingénieurs logiciels connaissent dans leur IDE préféré, telles qu'Intelli-sense et la saisie semi-automatique du code, sont incluses dans l'éditeur. Il existe de nombreuses fonctionnalités supplémentaires, telles que l'enregistrement de macros qui facilite la gestion du code source dans Enterprise Architect. Il existe également de nombreuses options de gestion du code, disponibles via le menu contextuel de l'éditeur de code, la barre d'outils et les touches de fonction.



```

1 import java.util.*;
2
3
4 /**
5  * @author Paulene Dean
6  * @version 1.0
7  * @created 22-Jul-2019 10:37:21 AM
8  */
9 public class Order implements StateMachineContext {
10
11     private Date date;
12     private String deliveryInstructions;
13     private String orderNumber;
14     private LineItem m_LineItem;
15
16     public OrderStatus getStatus(){
17         return status;
18     }
19
20     /**
21     *
22     * @param newVal
23     */
24     public void setStatus(OrderStatus newVal){
25         status = newVal;
26     }
27
28     public LineItem getLineItem(){

```

Pour la plupart des langages de programmation, un seul fichier est créé à partir d'une classe UML, mais dans le cas de C++, les classes d'en-tête et d'implémentation sont créées et l'éditeur de code source affiche ces fichiers dans des onglets séparés.

Un certain nombre d'options modifient le fonctionnement de l'éditeur de code source ; elles peuvent être modifiées à l'aide de la dialogue « Préférences » disponible dans le ruban Démarrer :

' Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > Éditeurs de Code '

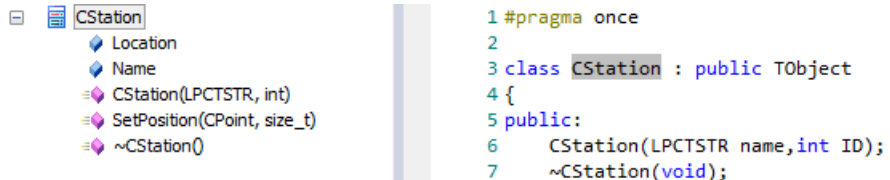
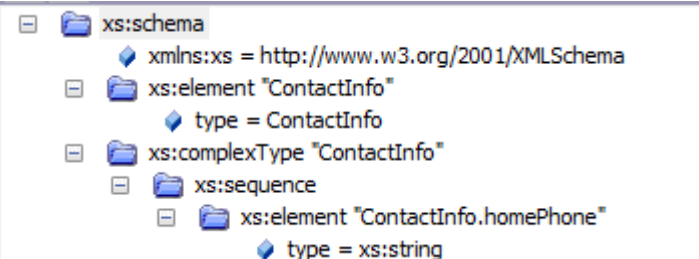
Il existe des variantes de l'Éditeur de Code source, avec différentes méthodes d'accès. Les variantes sont présentées dans la rubrique *Comparer les éditeurs* .

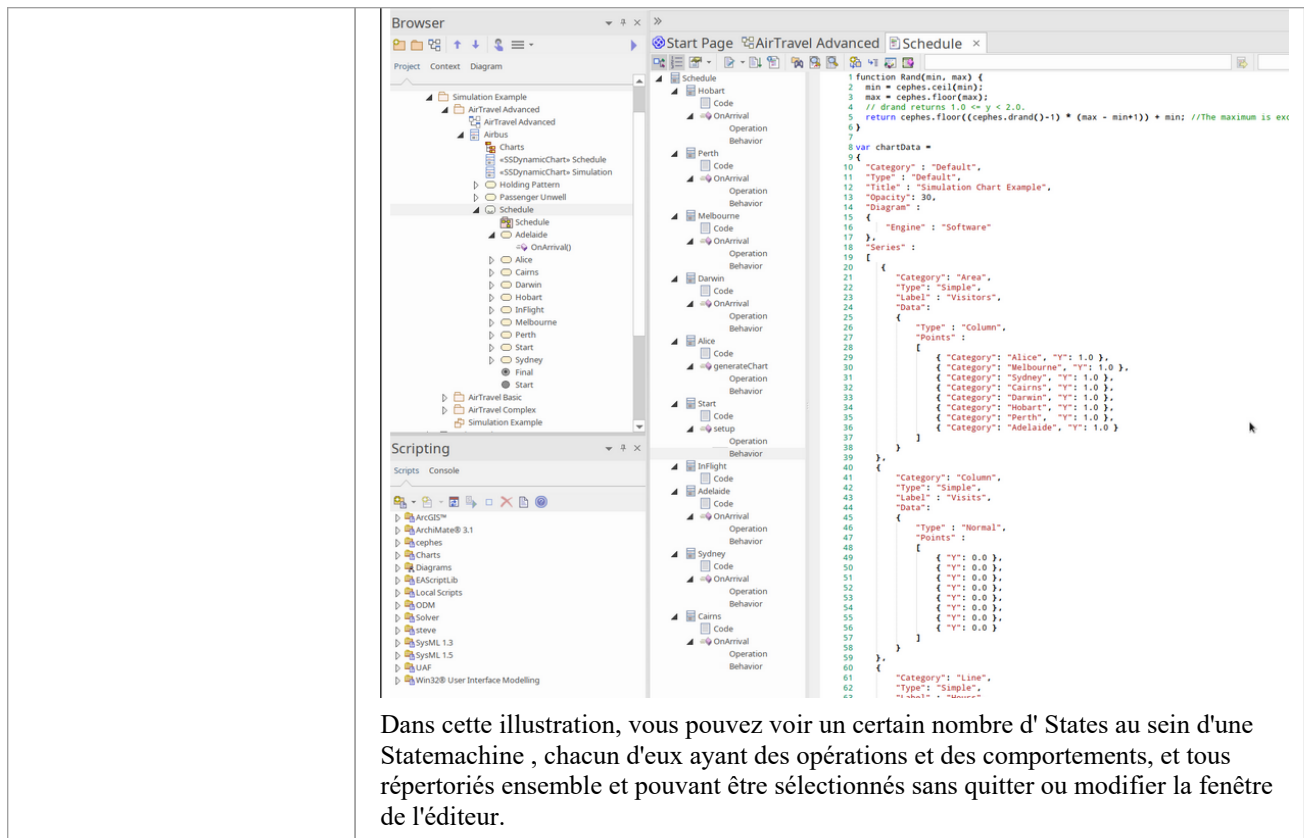
### Accéder

Ruban	Exécuter > Source > Modifier > Ouvrir le fichier source (fichier externe) ou Exécuter > Source > Modifier > Modifier la source de l'élément (pour un fichier source existant) ou
-------	---

	<p>Exécuter &gt; Source &gt; Modifier &gt; Modifier le nouveau fichier source ou                  Conception &gt; Élément &gt; Comportement ou                  Développer &gt; Code source &gt; Comportement</p>
Raccourcis Clavier	<p>F12 ou Ctrl+E (pour le code existant pour les éléments du modèle)                  Ctrl+Alt+O (pour localiser les fichiers externes)</p>

## Facilités

Facilité	Description
Éditeur de code source	<p>Par défaut, l'éditeur de code source est défini sur :</p> <ul style="list-style-type: none"> <li>Analyser tous les fichiers ouverts et afficher une arborescence des résultats</li> <li>Afficher les numéros de ligne</li> </ul>  <p>Si vous modifiez un fichier XML, l'arborescence de la structure reflète l'ordre et la structure exacts du document.</p> 
Structure arborescente	<p>L'arborescence de la structure des fichiers est disponible pour les fichiers de langues prises en charge, tels que C++, C# , Java et XML. L'arborescence peut être utile pour parcourir rapidement le contenu, de la même manière qu'un tableau de contenu le ferait pour d'autres documents.</p>
Comportements Simulation	<p>Si vous éditez les comportements des éléments d'un diagramme Statemachine ou d'activités, l' Éditeur de Code vous permet de lister et d'éditer les comportements de tous les éléments du diagramme ensemble, en utilisant une arborescence de structure.</p>



Dans cette illustration, vous pouvez voir un certain nombre d' States au sein d'une Statemachine , chacun d'eux ayant des opérations et des comportements, et tous répertoriés ensemble et pouvant être sélectionnés sans quitter ou modifier la fenêtre de l'éditeur.

## Notes

- Pour la plupart des éléments sélectionnés, vous pouvez utiliser les touches F12 ou Ctrl+E pour afficher le code source.
- Lorsque vous sélectionnez un élément pour afficher le code source, si l'élément n'a pas de fichier de génération (c'est-à-dire que le code n'a pas été ou ne peut pas être généré, comme pour un cas d'utilisation), Enterprise Architect vérifie si l'élément a un lien vers une opération ou un attribut d'un autre élément - si un tel lien existe et que cet autre élément a du code source, le code de cet élément s'affiche
- Vous pouvez également localiser le répertoire contenant un fichier source qui a été créé ou importé dans Enterprise Architect et le modifier ou ses fichiers associés à l'aide d'un éditeur externe tel que le Bloc-notes ou Visual Studio ; cliquez sur l'élément dans la fenêtre Navigateur et appuyez sur Ctrl+Alt+Y



## Langues prises en charge


Les Éditeurs de Code Source peuvent afficher du code dans un large éventail de langages, comme indiqué ici. Pour chaque langage, l'éditeur met en évidence - en texte coloré - la syntaxe standard du code.

- Ada (.ada, .ads, .adb)
- ActionScript (.as)
- Document BPEL (.bpel)
- C++ (.h, .hh, .hpp, .c, .cpp, .cxx)
- C# (.cs)
- Langage Query structuré DDL (.sql)
- Delphes/Pascal (.pas)
- Fichiers de différences/ Patch (.diff, .patch)
- Définition Type de document (.dtd)
- Fichiers batch DOS (.bat)
- Scripts de commande DOS (.cmd)
- HTML (.html)
- Langage de définition d'interface (.idl, .odl)
- Java (.java)
- JavaScript (.javascript)
- JScript (.js)
- Grammaire de la forme Backus-Naur modifiée (.mbnf)
- PHP (.php, .php4, .inc)
- Python (.py)
- Langage de balisage généralisé standard (.sgml)
- SystèmeC (.sc)
- Visual Basic 6 (.bas)
- VB.NET (.vb)
- VBScript (.vbs)
- Verilog (.v)
- Langage de description du matériel VHSIC (.vhdl)
- Configuration des ressources de Visual Studio (.rc)
- XML (langage de balisage extensible) (.xml)
- XSD (définition de schéma XML)
- XSL (langage de feuille de style XML)

## Configurer les associations de fichiers

Si vous êtes un utilisateur Windows®, vous pouvez configurer Enterprise Architect comme gestionnaire de documents par défaut pour vos fichiers sources de langue.

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > Éditeurs de Code : Configurer les associations de fichiers Enterprise Architect 
-------	--

### Actes

Pour chaque type de fichier que vous souhaitez ouvrir dans Enterprise Architect, cliquez sur la case à cocher à gauche du nom du type de fichier. Après avoir sélectionné tous les types de documents souhaités, cliquez sur le bouton Enregistrer.

Après cela, cliquez sur n'importe quel fichier correspondant dans l'Explorateur Windows® pour l'ouvrir dans Enterprise Architect.

### Notes

- Vous pouvez modifier les programmes par défaut, ou les documents gérés par eux, directement via l'option « Programmes par défaut » dans le Panneau de configuration Windows®.

# Comparer les éditeurs

Enterprise Architect propose quatre principales variantes d'éditeur de code, disponibles via plusieurs chemins d'accès. Les options d'accès les plus directes sont identifiées dans ces descriptions.

Les trois premières variantes d'éditeur de code répertoriées ont le même format d'affichage, la même barre d'outils d'options, les mêmes options de menu contextuel et les mêmes touches de fonction internes. Elles diffèrent par leur méthode d'accès et leur mécanisme d'affichage.

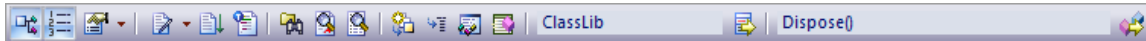
## Variantes de l'éditeur

Variante	Détails
Code source Vue	<p>F12 Ctrl+E</p> <p>Menu contextuel de la classe   « Code source Vue »</p> <p>Description : Affiche le code sur un onglet du Diagramme Vue ; l'étiquette de l'onglet affiche le nom du fichier et l'extension (comme .java) ; encore une fois, pour C++, il existe deux onglets pour les fichiers d'en-tête et d'implémentation.</p> <p>Vous pouvez afficher le code source d'autres classes sur des onglets supplémentaires, en resélectionnant l'option de menu/les touches de la classe suivante.</p>
Fenêtre de code source (ancrable)	<p>Alt+7</p> <p>« Exécuter &gt; Source &gt; Modifier &gt; Ouvrir le fichier source »</p> <p>Description : Affiche le contenu du fichier source d'une classe sélectionnée (sauf si le langage est C++, lorsque la fenêtre affiche un onglet pour le fichier d'en-tête et un onglet pour le fichier d'implémentation).</p> <p>Si vous sélectionnez une classe différente, la fenêtre change pour afficher le code de la nouvelle classe (à moins que la première classe n'appelle la seconde, auquel cas la fenêtre défile jusqu'au code de la seconde classe).</p>
Éditeur interne, code source externe	<p>Ctrl+Alt+O</p> <p>Option du ruban « Exécuter &gt; Source &gt; Modifier &gt; Ouvrir le fichier source »</p> <p>Description : utilisez cette option si vous avez l'intention de modifier du code externe, des fichiers XML ou DDL (c'est-à-dire du code non importé ou généré dans Enterprise Architect).</p> <p>Affiche un navigateur externe, puis ouvre le fichier de code sélectionné spécifique sous forme d'onglet du Diagramme Vue (pour C++, pas deux fichiers de code) ; sinon, cela est identique à l'option F12.</p>
Éditeur externe, code source interne ou externe	<p>Ctrl+Alt+Y</p> <p>Menu contextuel de la classe   Répertoire Open Source</p> <p>Description : affiche un navigateur de fichiers externe, ouvert sur le répertoire contenant les fichiers sources de la classe sélectionnée ; vous pouvez ouvrir les fichiers dans le Bloc-notes, Visual Studio ou d'autres outils que vous pourriez avoir sur votre système.</p>

# Barre d'outils Éditeur de Code

Lorsque vous réviser le code d'une partie de votre modèle dans l'éditeur de code source, vous pouvez accéder à un large éventail de fonctions d'affichage et d'édition à partir de la barre d'outils de l'éditeur.

## Barre d'outils Éditeur de Code



## Options de la barre d'outils

**Structure arborescente** Cliquez sur cette icône pour afficher ou masquer le panneau de hiérarchie des éléments (le panneau de gauche de l'éditeur de code source).

**Numéros de ligne** Cliquez sur cette icône pour afficher ou masquer les numéros de ligne par rapport aux lignes de code.

**Propriétés d'ingénierie du code source** Cliquez sur la flèche déroulante pour afficher un menu d'options permettant de sélectionner les pages individuelles « Ingénierie du code source » de la boîte de dialogue « Préférences », à partir desquelles vous pouvez configurer les options d'affichage et de comportement pour l'ingénierie du code source :

- Langue
- Options de mise en évidence de la syntaxe
- Éditeur de Code Options
- Options d'ingénierie de code
- Raccourcis clavier Éditeur de Code

**Fonctions de l'éditeur** Cliquez sur la flèche déroulante pour afficher un menu donnant accès à une gamme de fonctions d'édition de code :

- Ouvrir le fichier correspondant (Ctrl+Maj+O) - ouvre l'en-tête ou le fichier d'implémentation associé au fichier actuellement ouvert
- Accéder à l'accolade correspondante (Ctrl+E) - pour une accolade ouvrante ou fermante sélectionnée, met en surbrillance l'accolade fermante ou ouvrante correspondante dans la paire
- Aller à la ligne (Ctrl+G) - affiche une dialogue dans laquelle vous sélectionnez le numéro de la ligne à mettre en surbrillance ; cliquez sur le bouton OK pour déplacer le curseur sur cette ligne
- Historique du curseur précédent (Ctrl+-) - le visualiseur de code source conserve un historique des 50 positions de curseur précédentes, créant un enregistrement lorsque le curseur est déplacé soit de plus de 10 lignes de sa position précédente, soit lors d'une opération de recherche et de remplacement ; l'option de menu déplace le curseur vers la position dans l'enregistrement d'historique du curseur immédiatement précédent
- Historique du curseur suivant (Ctrl+Maj+-) - si vous êtes passé à une position de curseur antérieure, cette option déplace le curseur vers la position dans l'enregistrement d'historique du curseur immédiatement suivant
- Rechercher (Ctrl+F) - affiche une dialogue dans laquelle vous définissez une string de texte et des options de recherche pour localiser cette string de texte

dans le code

- Remplacer (Ctrl+R) - affiche une dialogue dans laquelle vous définissez une string de texte et des options de recherche pour localiser cette string de texte dans le code et la remplacer par une autre string de texte ; le dialogue propose des options pour localiser et remplacer chaque occurrence selon votre choix, ou pour remplacer toutes les occurrences immédiatement
- Mettre en surbrillance les mots correspondants - (Ctrl+3) Active ou désactive la mise en surbrillance des mots correspondants lors d'une opération de recherche ; par défaut, cette option est activée
- Enregistrer une macro - enregistre vos prochaines frappes de touches pour les sauvegarder sous forme de macro
- Arrête d'Enregistrer et Enregistrer la macro - arrête l'enregistrement des frappes et affiche la dialogue « Enregistrer la macro » dans laquelle vous spécifiez un nom pour la macro
- Lire la macro - affiche la dialogue « Ouvrir une macro » à partir de laquelle vous sélectionnez et exécutez une macro enregistrée, pour répéter les frappes de touches enregistrées
- Basculer le commentaire de ligne (Ctrl+Maj+C) - commente (//) ou rétablit le code pour chaque ligne complète dans laquelle le texte est mis en surbrillance
- Basculer le commentaire de flux (Ctrl+Maj+X) - insère un commentaire de flux (/ \* \*/) à la position du curseur (commente uniquement les caractères et les lignes mis en surbrillance) ou rétablit le texte commenté sous forme de code
- Activer/désactiver les caractères d'espacement (Ctrl+Maj+W) - affiche ou masque les caractères d'espacement : --> (espace de tabulation) et . (espace de caractère)
- Basculer les caractères de fin de ligne (Ctrl+Maj+L) - affiche ou masque les caractères de fin de ligne : CR (retour chariot) et LF (saut de ligne)
- Basculer la synchronisation de l'arborescence - sélectionne automatiquement l'élément de l'arborescence lorsque le contexte change dans l'éditeur de code
- Ouvrir le dossier contenant - ouvre le navigateur de fichiers dans le dossier contenant le fichier de code ; vous pouvez ouvrir d'autres fichiers dans votre éditeur externe par défaut à des fins de comparaison et de travail parallèle

**Enregistrer la source et resynchroniser la classe**

Cliquez sur cette icône pour enregistrer le code source et resynchroniser le code et la classe dans le modèle.

**Code Gabarits**

Cliquez sur cette icône pour accéder à l'éditeur Gabarits de code, pour éditer ou créer gabarits de code pour la génération de code.

**Rechercher dans Projet Navigateur**

Pour une ligne de code sélectionnée, cliquez sur cette icône pour mettre en surbrillance la structure correspondante dans la fenêtre Navigateur . S'il existe plusieurs possibilités, la dialogue « Correspondances possibles » s'affiche, répertoriant les occurrences de la structure parmi lesquelles vous pouvez sélectionner celle qui vous intéresse.

**Rechercher dans les fichiers**

Cliquez sur cette icône pour rechercher le nom object sélectionné dans les fichiers associés et afficher les résultats de la recherche dans la fenêtre Recherche de fichiers. Vous pouvez affiner et actualiser la recherche en spécifiant des critères dans la barre d'outils de la fenêtre Rechercher dans les fichiers.

**Recherche dans Modèle**

Cliquez sur cette icône pour rechercher le texte sélectionné dans tout le modèle et afficher les résultats de la recherche dans la vue Rechercher dans Projet .

**Accéder à la déclaration**

Cliquez sur cette icône pour localiser la déclaration d'un symbole dans le code source.

<b>Aller à la définition</b>	Cliquez sur cette icône pour localiser la définition d'un symbole dans le code source (applicable aux langages tels que C++ et Delphi, où les symboles sont déclarés et définis dans des fichiers séparés).
<b>Liste de saisie semi-automatique</b>	Cliquez sur cette icône pour afficher la liste d'autocomplétion des valeurs possibles ; double-cliquez sur une valeur pour la sélectionner.
<b>Informations sur les paramètres</b>	Lorsque le curseur se trouve entre les parenthèses de la liste des paramètres d'une opération, cliquez sur cette icône pour afficher la signature de l'opération, mettant en évidence le paramètre actuel.
<b>Rechercher la classe actuelle dans la fenêtre Navigateur</b>	Cliquez sur cette icône pour afficher le nom de la classe actuellement sélectionnée dans le code et mettez en surbrillance ce nom dans la fenêtre Navigateur ; s'il existe plusieurs possibilités, la dialogue « Correspondances possibles » s'affiche, répertoriant les occurrences de la classe parmi lesquelles vous pouvez sélectionner celle requise.
<b>Trouver un membre</b>	Cliquez sur cette icône pour afficher le nom de l'attribut ou de la méthode actuellement sélectionné dans le code, et mettez en surbrillance ce nom dans la fenêtre Navigateur ; s'il existe plusieurs possibilités, la dialogue « Correspondances possibles » s'affiche, répertoriant les occurrences de la fonctionnalité parmi lesquelles vous pouvez sélectionner celle requise.

## Notes

- L'option « Enregistrer la macro » désactive Intelli-sense pendant l'enregistrement de la macro
- Vous pouvez attribuer des touches pour exécuter la macro, au lieu d'utiliser la liste déroulante de la barre d'outils et dialogue « Ouvrir la macro »

# Éditeur de Code Menu Contexte

Lorsque vous travaillez sur un fichier avec un éditeur de code, vous pouvez effectuer un certain nombre d'opérations de recherche et d'édition de code pour réviser le contenu du fichier. Ces options sont disponibles via le menu contextuel de l'éditeur et peuvent varier en fonction de l'éditeur de code que vous utilisez.

## Accéder

Menu Contexte	Cliquez-droit sur la string de texte du code sur laquelle vous travaillez
---------------	---

## Options

**Accéder à la déclaration** Localisez et mettez en surbrillance la déclaration d'un symbole dans le code source.

**Aller à la définition** Localisez et mettez en évidence la définition d'un symbole dans le code source (applicable aux langages tels que C++ et Delphi, où les symboles sont déclarés et définis à des endroits séparés).

**Ouvrir dans l'éditeur de grammaire** Ouvre une vue qui vous permet d'examiner ou de valider le code à l'aide de la grammaire appropriée.

**Synchroniser l'arbre avec l'éditeur** Recherche et affiche l'élément courant (méthode par exemple) dans l'arborescence de la structure.

**Synchronisation automatique de l'arbre et de l'éditeur** Une fois sélectionné, l'arborescence de la structure affichera automatiquement l'élément en cours de traitement dans l'éditeur.

**Validation du schéma XML** Permet de valider un schéma XML.

**Rechercher « < string > »** Afficher un sous-menu fournissant des options permettant de localiser la string de texte sélectionnée dans une plage d'emplacements.

- 'Rechercher dans Projet Navigateur' - Mettre en surbrillance l' object contenant le texte sélectionné dans la fenêtre Navigateur
- « Rechercher dans les fichiers ouverts » : recherchez la string de texte sélectionnée dans les fichiers ouverts associés et affichez les résultats de la recherche dans la fenêtre Rechercher dans les fichiers ; vous pouvez affiner et actualiser la recherche en spécifiant des critères dans la barre d'outils de la fenêtre Rechercher dans les fichiers
- « Rechercher dans les fichiers » : recherchez la string de texte sélectionnée dans tous les fichiers associés (fermés ou ouverts) et affichez les résultats de la recherche dans la fenêtre Rechercher dans les fichiers ; vous pouvez affiner et actualiser la recherche en spécifiant des critères dans la fenêtre Rechercher dans Barre d'outils de la fenêtre Fichiers (raccourci clavier : F12)
- 'Rechercher dans Modèle' - Effectuez une recherche par 'Nom d'élément' dans la recherche Modèle facilité et affichez les résultats dans l'onglet Recherche Modèle

	<ul style="list-style-type: none"> <li>• 'Rechercher dans Scripts ' - (Disponible lorsque vous travaillez dans l' Éditeur de Script ) Ouvrez la fenêtre Rechercher dans les fichiers, définissez le champ 'Chemin de recherche' sur 'Rechercher dans Scripts ' et le champ 'Texte de recherche' sur le texte sélectionné, puis recherchez tous les scripts pour la string de texte et affichez les résultats de la recherche ; vous pouvez affiner et actualiser la recherche en spécifiant des critères dans la barre d'outils de la fenêtre Rechercher dans les fichiers</li> <li>• 'EA User Guide' - Afficher la description de l'élément de code dans le <i>Guide d'Utilisateur d'Enterprise Architect</i></li> <li>• 'Google' - Afficher les résultats d'une recherche Google sur le texte</li> <li>• « MSDN » – Afficher les résultats d'une recherche sur le texte dans le réseau des développeurs Microsoft (MSDN)</li> <li>• 'Sun Java SE' - Affichage des résultats d'une recherche sur le texte dans la facilité 'Sun Search' de Sun Microsystems</li> <li>• « Wikipédia » : affiche n'importe quelle entrée sur l' objet sur le site Web de Wikipédia</li> <li>• 'Koders' - Affiche les résultats d'une recherche de la string de texte sur Koders.com</li> </ul>
<b>Recherche Intelli-sense</b>	<p>Effectuez une recherche sur la string spécifiée à l'aide du service Code Miner ou de la bibliothèque spécifiée dans le script Analyzer actuel. Les résultats s'affichent dans l'onglet « Code Miner » de la fenêtre Rechercher dans les fichiers.</p> <p>Raccourci clavier : Maj+F12</p>
<b>Régler Débogueur sur Ligne</b>	<p>(Si le débogueur est en cours d'exécution et a atteint un point d'arrêt.) Déplacez le point d'exécution vers la ligne en cours. Vérifiez que vous n'ignorez aucun code ou déclaration qui affecte la section suivante du code en cours de débogage.</p>
<b>Variable d'affichage</b>	<p>(Si le débogueur est en cours d'exécution.) Ouvrez la fenêtre Variables locales et mettez en surbrillance la variable locale pour le point actuel dans le code.</p>
<b>Afficher dans la visionneuse String</b>	<p>Affichez le contenu complet d'une string variable dans le visualiseur String .</p>
<b>Créer un cas d'utilisation pour '&lt; string &gt;'</b>	<p>Affichez la dialogue « Créer un cas d'utilisation pour la méthode », à travers laquelle vous créez un cas d'utilisation pour la méthode contenant la string de texte.</p>
<b>Point d'Arrêt</b>	<p>Affiche un sous-menu d'options permettant de créer un marqueur d'enregistrement sur la ligne de code sélectionnée. Les marqueurs d'enregistrement que vous pouvez ajouter sont les suivants :</p> <ul style="list-style-type: none"> <li>• Point d'Arrêt</li> <li>• Démarrer Enregistrement Marqueur</li> <li>• Fin de Enregistrement Marqueur</li> <li>• Marqueur de capture automatique de pile</li> <li>• Méthode d'enregistrement automatique Marqueur</li> <li>• Point de Trace</li> </ul>
<b>Testpoints</b>	<p>Options d'affichage pour ajouter un nouveau Testpoint , afficher le gestionnaire Testpoints (fenêtre Testpoints ) ou modifier un Testpoint existant si un ou plusieurs sont déjà définis à l'emplacement sélectionné.</p> <p>(Les sous-options dépendent du type de fichier de code que vous réviser .)</p>
<b>Validation XML</b>	<p>Permet de vérifier la conformité d'un document XML avec ses propres références de schéma ou à l'aide d'un schéma spécifié par l'utilisateur ; soit un fichier de</p>



schéma local, soit une URL.

**Ouvrir (Fermer) l'IME** Ouvrez (ou fermez) l'éditeur de méthode de saisie pour pouvoir saisir du texte dans une langue étrangère sélectionnée, comme le japonais. Vous pouvez définir la langue du clavier à l'aide du Panneau de configuration Windows - Options régionales et linguistiques.

**Copier le lien hypertexte de position** Copie la position du curseur sous forme de lien hypertexte pouvant être collé dans les éditeurs Rich Notes, comme un message dans l'onglet « Chat » de la fenêtre Chat & Mail. Utilisez simplement l'option de menu contextuel « Coller » dans le message et spécifiez le texte du lien.  
Le lecteur peut cliquer sur le lien pour ouvrir le fichier source et déplacer le curseur à la position du curseur sélectionnée dans le fichier.

**Copier le texte du lien hypertexte** Copie la string de texte sélectionnée sous forme de lien hypertexte pouvant être collé dans les éditeurs Rich Notes, comme un message dans l'onglet « Chat » de la fenêtre Chat & Mail. Utilisez simplement l'option de menu contextuel « Coller » dans le message.  
Le lecteur peut cliquer sur le lien pour ouvrir le fichier source et déplacer le curseur sur la première occurrence de cette string de texte dans le fichier.

**Numéros de ligne** ( Éditeur de Script uniquement.) Affichez ou masquez les numéros de ligne de code sur le côté gauche de l'écran de l'éditeur.

**Annuler**  
**Couper**  
**Copie**  
**Coller**  
**Supprimer**  
**Sélectionner tout**

Ces six options fournissent des fonctions simples pour éditer le code.

## Notes

- Les options dans la moitié inférieure du sous-menu « Rechercher <string> » (après « Rechercher dans Scripts ») sont configurables ; vous pouvez ajouter de nouveaux outils de recherche ou supprimer des outils existants en modifiant le fichier searchProviders.xml dans le dossier Sparx Systems > EA > Config - ce fichier est au format de document de description OpenSearch

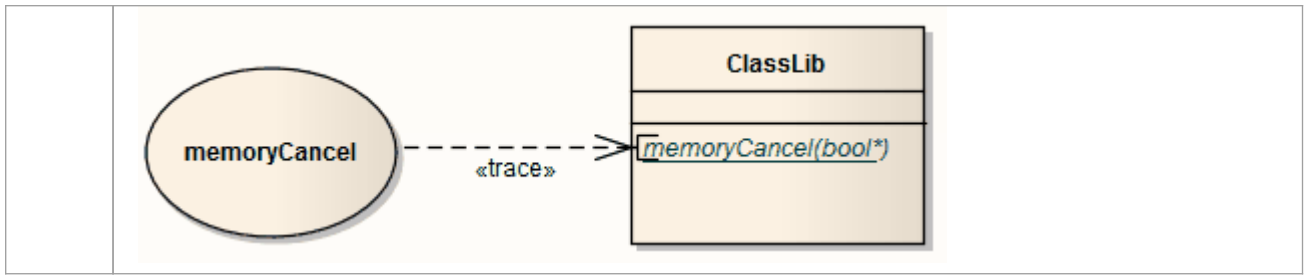
## Créer un cas d'utilisation pour la méthode

À l'aide du menu contextuel de l'éditeur de code, vous pouvez créer un élément de cas d'utilisation pour une méthode que vous sélectionnez dans le code. Vous pouvez également :

- Lier le cas d'utilisation directement à la méthode
- Ajoutez la classe parent à un diagramme (si elle n'est pas déjà dans le diagramme sélectionné) et/ou ajoutez l'élément Cas d'utilisation au diagramme
- Bloc d'affichage de tous les attributs ou méthodes qui ne sont pas également les cibles des liens fonctionnalité

### Créer un cas d'utilisation pour une méthode, via l'éditeur de code

Étape	Action
1	(Si vous souhaitez représenter le cas d'utilisation et son lien avec la méthode dans un diagramme ), cliquez sur le nom diagramme dans la fenêtre Navigateur .
2	Dans l'éditeur de code, cliquez-droit sur le nom de la méthode ou sur n'importe quelle partie du corps de la méthode et sélectionnez l'option « Créer une méthode pour <methodname> ». La dialogue « Créer un cas d'utilisation pour la méthode » s'affiche.
3	La fonction de base de cette dialogue est de créer un cas d'utilisation pour la méthode sélectionnée : <ul style="list-style-type: none"> <li>• Si c'est tout ce qui est requis, cliquez sur le bouton OK ; l'élément Use Case est créé dans la fenêtre Navigateur , dans le même Paquetage que la classe parent de la méthode et avec le même nom que la méthode</li> <li>• Si vous souhaitez rendre la relation tangible, continuez la procédure</li> </ul>
4	Pour créer un connecteur Trace reliant le cas d'utilisation à la méthode, cochez la case « Lier le cas d'utilisation à la méthode ».
5	Pour ajouter la classe parent de la méthode au diagramme , si elle n'est pas déjà là, cochez la case « Ajouter une classe au Diagramme ».
6	Pour ajouter le cas d'utilisation nouvellement créé au diagramme , cochez la case « Ajouter un cas d'utilisation au Diagramme » ; cela affichera maintenant le cas d'utilisation, la classe et le connecteur Trace sur le diagramme .
7	Pour afficher uniquement les fonctionnalités (attributs et méthodes) de la classe parent qui sont les cibles des relations « lien vers fonctionnalité », cochez la case « Afficher uniquement fonctionnalités liées dans la classe ». La classe peut contenir n'importe quel nombre d'attributs et de méthodes, mais ceux sans relation « lien vers fonctionnalité » sont masqués.
8	Cliquez sur le bouton OK pour créer et représenter le cas d'utilisation et la relation ; si vous avez sélectionné toutes les options, le diagramme contient maintenant des éléments liés ressemblant à cette illustration :



# Éditeur de Code Fonctions

L' Éditeur de Code commun fournit une variété de fonctions pour aider au processus d'édition de code, notamment :

- Mise en évidence de la syntaxe
- Signets
- Historique du curseur
- Correspondance des accolades
- Indentation automatique
- Commenter les sélections
- Guides de portée
- Zoom
- Sélection de ligne
- Intelli-sens
- Rechercher et remplacer
- Rechercher dans les fichiers

Une gamme de ces fonctions est disponible via des combinaisons de touches du clavier et/ou des options de menu contextuel.

Vous pouvez personnaliser plusieurs fonctionnalités de l' Éditeur de Code en définissant des propriétés dans les fichiers de configuration Éditeur de Code ; par exemple, par défaut, la ligne contenant le curseur est toujours surlignée, mais vous pouvez désactiver la surbrillance.

# Détails de la fonction

## Éditeur de Code Fonctions

Fonction	Description
Mise en évidence de la syntaxe	<p>L' Éditeur de Code met en évidence - en texte coloré - la syntaxe de code standard de tous les formats de fichiers de langage pris en charge par Enterprise Architect</p> <pre> 1 #pragma once 2 #include "afxwin.h" 3 #include "afxcmn.h" 4 5 6 // CToolBox dialog 7 8 class CToolBox : public CDialog 9 { 10     DECLARE_DYNAMIC(CToolBox) 11     CRect m_rect; 12     int m_offset; </pre> <p>Vous pouvez définir comment l' Éditeur de Code implémente la coloration syntaxique pour chaque langage, via la page ' Éditeurs de Code ' de la dialogue 'Préférences'.</p>
Signets	<p>Les signets indiquent une ligne intéressante dans le document ; vous pouvez les activer et les désactiver pour une ligne particulière en appuyant sur Ctrl+F2.</p> <p>De plus, vous pouvez appuyer sur F2 et Maj+F2 pour accéder au signet suivant ou précédent dans le document.</p> <p>Pour effacer tous les signets dans le fichier de code, appuyez sur Ctrl+Maj+F2.</p>
Historique du curseur	<p>L' Éditeur de Code Control conserve un historique des 50 positions précédentes du curseur ; une entrée dans la liste d'historique est créée lorsque :</p> <ul style="list-style-type: none"> <li>Le curseur est déplacé de plus de 10 lignes par rapport à sa position précédente</li> <li>Le curseur est déplacé lors d'une opération de recherche/remplacement</li> </ul> <p>Vous pouvez accéder à un point antérieur dans l'historique du curseur en appuyant sur Ctrl+-, et à un point ultérieur en appuyant sur Ctrl+Maj+-.</p>
Correspondance des accolades	<p>Lorsque vous placez le curseur sur une accolade ou un crochet, l' Éditeur de Code met en surbrillance son partenaire correspondant ; vous pouvez ensuite naviguer jusqu'à l'accolade correspondante en appuyant sur Ctrl+E.</p> <pre> 28     function ProtectedFunctionTest: boolean; 29     procedure ProtectedProcedureTest(a: WideString); </pre>
Indentation automatique	<p>Pour chaque langage prise en charge, l' Éditeur de Code ajuste l'indentation d'une nouvelle ligne en fonction de la présence d'instructions de contrôle ou de jetons de bloc de portée dans les lignes menant à la position du curseur.</p>

	<pre> 358 { 359     for(size_t t = 0; t &lt; Stations.size(); t++) 360     { 361         if(Stations[t]-&gt;Location == loc) 362             return Stations[t]; 363     } 364     return NULL; 365 }         </pre> <p>Les niveaux de retrait sont indiqués par des lignes horizontales pâles.</p> <p>Vous pouvez également mettre en retrait manuellement les lignes et les blocs de code sélectionnés en appuyant sur la touche Tab ; pour annuler l'indentation du code sélectionné, appuyez sur Maj+Tab.</p>
<p>Commenter les sélections</p>	<p>Pour les langages qui support les commentaires, l' Éditeur de Code peut commenter des sélections entières de code.</p> <p>L' Éditeur de Code reconnaît deux types de commentaires :</p> <ul style="list-style-type: none"> <li>• Commentaire de ligne - des lignes entières sont commentées dès le début (par exemple : // Ceci est un commentaire)</li> <li>• Commentaires de flux - les sections d'une ligne sont commentées à partir d'un point de départ spécifié jusqu'à un point de fin spécifié (par exemple : /* Ceci est un commentaire */)</li> </ul> <p>Vous pouvez basculer les commentaires sur la ligne ou la sélection actuelle en appuyant sur :</p> <ul style="list-style-type: none"> <li>• Ctrl+Maj+C pour les commentaires de ligne, ou</li> <li>• Ctrl+Maj+X pour les commentaires de flux</li> </ul>
<p>Guides de portée</p>	<p>Si le curseur est placé sur un marqueur d'indentation, l' Éditeur de Code effectue un « retour en arrière » pour trouver la ligne qui a démarré la portée à ce niveau d'indentation ; si la ligne est trouvée et est actuellement à l'écran, elle est surlignée en bleu clair.</p> <pre> -- 93     // If there were any answers, then return a packet, if not then just return null 94     // to indicate the server has no response 95     if ( answers.size() &gt; 0 ) 96     { 97         DNSPacket responsePacket = Helpers.createResponsePacket( answers, this.theS 98         responsePacket.queryID = receivedPacket.queryID; 99 100         return responsePacket; 101     }         </pre> <p>Alternativement, si la ligne est hors écran, une info-bulle s'affiche indiquant le numéro de la ligne et son contenu :</p> <pre> 93     // If there were any answers, then return a packet, if not then just return null 94     // to indicate the server has no response 95     if ( answers.size() &gt; 0 ) 96     { 97         Line 73: private DNSPacket processQuery(DNSPacket receivedPacket) 98         DNSPacket responsePacket = Helpers.createResponsePacket( answers, this.theS 99         responsePacket.queryID = receivedPacket.queryID; 100         return responsePacket; 101     }         </pre>
<p>Zoom</p>	<p>Vous pouvez zoomer et dézoomer sur le contenu de l' Éditeur de Code en utilisant :</p> <ul style="list-style-type: none"> <li>• Ctrl+clavier + et</li> <li>• Ctrl+clavier -</li> </ul> <p>Le zoom peut être rétabli à 100 % en utilisant Ctrl+clavier /.</p>

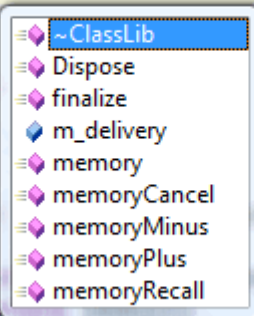
Sélection de ligne	<p>Si vous souhaitez déplacer le curseur sur une ligne de code spécifique, appuyez sur Ctrl+G et, en réponse à l'invite, saisissez le numéro de ligne.</p> <p>Appuyez sur le bouton OK ; l'éditeur affiche la ligne de code spécifiée avec le curseur à gauche.</p>
--------------------	---

## Intelli-sens

Intelli-sense est une fonctionnalité qui permet de choisir les éléments et les valeurs de code à mesure que vous tapez. Tous les éditeurs de code n'utilisent pas Intelli-sense ; par exemple, Intelli-sense est désactivé lorsque vous enregistrez une macro dans la Visionneuse de code source .

Intelli-sense vous fournit une assistance contextuelle via des listes de saisie semi-automatique, des info-bulles et des informations de survol de la souris.

### Facilités

Facilité	Description
Liste de saisie semi-automatique	<p>Une liste de saisie semi-automatique fournit une liste de saisies semi-automatiques possibles pour le texte actuel ; la liste est automatiquement appelée lorsque vous entrez un jeton d'accès (tel qu'un point ou un pointeur d'accès) après un objet ou un type contenant des membres.</p> <pre> 57     public void memoryRecall () 58     { 59         this. 60     } 61 62     public 63 64     } 65 66     public 67     { 68         in 69         re 70     } 71 </pre>  <p>Vous pouvez également invoquer la liste de saisie semi-automatique manuellement en appuyant sur Ctrl+Espace ; l'Éditeur de Code recherche alors les correspondances pour le mot menant au point d'invocation.</p> <p>Sélectionnez un élément dans la liste et appuyez sur la touche Entrée ou sur la touche Tab pour insérer l'élément dans le code ; pour fermer la liste de saisie semi-automatique, appuyez sur Échap.</p>
Conseils d'appel	<p>Les info-bulles affichent la signature de la méthode actuelle lorsque vous saisissez le jeton de la liste de paramètres (par exemple, en ouvrant une parenthèse) ; si la méthode est surchargée, l'info-bulle affiche des flèches que vous pouvez utiliser pour naviguer dans les différentes signatures de méthode</p>



	<pre> 20      //PostDraw Adornments 21      //Stereotyped Static Adornments 22          //Add Stakeholder's STAKE 23      setpenwidth(); 24          // Add a th SetPenWidth(int penwidth) 25      startpath(); 26          moveto(25,37); 27          lineto(25,52); 28      endpath(); 29      strokepath(); 30          //Add tip </pre>
<p>Informations sur le survol de la souris</p>	<p>Vous pouvez afficher la documentation de support pour les éléments de code (par exemple, les attributs et les méthodes) en passant le curseur sur l'élément en question.</p> <pre> 11      dockable = "none"; 12      string 13      / Dock elements together. Tagged V 14      / Valid Values: none, standard 15      //PreDraw Derived Attribute I </pre>

# Rechercher et remplacer

Chacun des éditeurs de code d' Enterprise Architect facilite la recherche et le remplacement de termes dans l'éditeur, via la dialogue « Rechercher et remplacer ».

## Accéder

Raccourcis Clavier	<p>Mettez en surbrillance la string de texte requise et appuyez sur :</p> <ul style="list-style-type: none"> <li>• Ctrl+F pour les contrôles de recherche uniquement, ou</li> <li>• Ctrl+R pour les commandes de recherche et de remplacement</li> </ul> <p>Dans chaque cas, le champ « Rechercher » est renseigné avec le texte actuellement sélectionné dans l'éditeur. Si aucun texte n'est sélectionné dans l'éditeur, le champ « Rechercher » est renseigné avec le mot se trouvant à la position actuelle du curseur. Si aucun mot n'existe à la position actuelle du curseur, le dernier terme recherché est utilisé.</p>
--------------------	--

## Opérations de base – Commandes

Commande	Action
Trouver la suite	Recherchez et mettez en surbrillance l'instance suivante (par rapport à la position actuelle du curseur) du texte spécifié dans le champ « Rechercher ».
Remplacer	Remplacez l'instance actuelle du texte spécifié dans le champ « Rechercher » par le texte spécifié dans le champ « Remplacer par », puis recherchez et mettez en surbrillance l'instance suivante (par rapport à la position actuelle du curseur) du texte spécifié dans le champ « Rechercher ».
Remplacer tout	Remplacez automatiquement toutes les instances du texte spécifié dans le champ « Rechercher » par le texte spécifié dans le champ « Remplacer par ».

## Opérations de base – Options

Option	Action
Étui à allumettes	Spécifiez que la casse de chaque caractère de la string de texte dans le champ « Rechercher » est significative lors de la recherche de correspondances dans le code.
Faire correspondre le mot entier	Spécifiez que la string de texte dans le champ « Rechercher » est un mot complet et ne doit pas correspondre à des instances du texte qui font partie d'une string plus longue. Par exemple, les recherches pour ARE ne doivent pas correspondre à ces lettres

	dans les instances des mots AREA ou ARENA.
Rechercher vers le haut	Effectuez la recherche à partir de la position actuelle du curseur jusqu'au début du fichier, plutôt que dans la direction par défaut de la position actuelle du curseur jusqu'à la fin du fichier.
Utiliser des expressions régulières	Évaluez des séquences de caractères spécifiques dans les champs « Rechercher » et « Remplacer par » en tant qu'expressions régulières.

## Concepts

Concept	Description
Expressions régulières	<p>Une expression régulière est une définition formelle d'un Motif de recherche, qui peut être utilisée pour faire correspondre des caractères, des mots ou motifs de caractères spécifiques.</p> <p>Par souci de simplicité, le mécanisme « rechercher et remplacer » de Éditeur de Code ne supporte qu'un sous-ensemble de la grammaire standard des expressions régulières.</p> <p>Le texte dans les champs « Rechercher » et « Remplacer par » n'est interprété comme une expression régulière que si la case à cocher « Utiliser les expressions régulières » est sélectionnée dans la boîte dialogue « Rechercher et remplacer ».</p>
Métaséquences	<p>Si la case à cocher « Utiliser les expressions régulières » est sélectionnée, la plupart des caractères du champ « Rechercher » sont traités comme des littéraux (c'est-à-dire qu'ils ne correspondent qu'à eux-mêmes).</p> <p>Les exceptions sont appelées métaséquences ; chaque métaséquence reconnue dans la dialogue « Rechercher et remplacer » Éditeur de Code est décrite dans ce tableau :</p> <ul style="list-style-type: none"> <li>• \&lt; - Indique que le texte est le début d'un mot ; par exemple : \<cat <i="" correspond="" à="">catastrophe et <i>cataclysm</i> , mais n'est pas <i>concaténé</i></cat></li> <li>• \&gt; - Indique que le texte est la fin d'un mot ; par exemple : hat\&gt; correspond à <i>that</i> et <i>chat</i> , mais pas à <i>hate</i></li> <li>• (...) - Indique des caractères uniques alternatifs qui peuvent être mis en correspondance - les caractères peuvent être spécifiques (chr) ou dans une plage alphabétique ou numérique (am) ; par exemple : (hc) at correspond à <i>hat</i> et <i>cat</i> mais pas à <i>bat</i> , et (am) Class correspond à n'importe quel nom dans la plage <i>aClass-mClass</i></li> <li>• (^...) - Indique des caractères uniques alternatifs qui doivent être exclus d'une correspondance - les caractères peuvent être spécifiques (^chr) ou dans une plage alphabétique ou numérique (^am) ; par exemple : (^hc) at correspond à <i>rat</i> et <i>bat</i> , mais <i>hat</i> et <i>cat</i> sont exclus, et (^am) Class correspond à n'importe quel nom dans la plage <i>nClass</i> à <i>zClass</i> , mais <i>aClass</i> à <i>mClass</i> sont exclus</li> <li>• ^ - Correspond au début d'une ligne</li> <li>• \$ - Correspond à la fin d'une ligne</li> <li>• * - Correspond au caractère précédent (ou au jeu de caractères) 0 fois ou plus ; par exemple : ba*t correspond à <i>bt</i> , <i>bat</i> , <i>baat</i> , <i>baaat</i> et ainsi de suite, et b(ea)*t correspond à <i>bt</i> , <i>bet</i> , <i>bat</i> , <i>beat</i> , <i>beet</i> , <i>baat</i> et ainsi de suite</li> <li>• + - Correspond au caractère précédent (ou au jeu de caractères) 1 ou plusieurs fois ; par exemple : ba+t correspond à <i>bat</i> , <i>baat</i> et <i>baaat</i> mais pas à <i>bt</i> , et</li> </ul>

	<p>b(ea) +t correspond à <i>bet</i> , <i>bat</i> , <i>beat</i> , <i>beet</i> et <i>baat</i> mais pas à <i>bt</i></p> <p>Si une métaséquence à caractère unique est précédée d'une barre oblique inverse (\), elle est traitée comme un caractère littéral : c\(\at\) correspond à c(at) car les crochets sont traités littéralement.</p> <p>Lorsque la case à cocher « Utiliser les expressions régulières » est sélectionnée, un menu d'aide à la métaséquence est disponible à droite des champs « Rechercher » et « Remplacer par » ; la sélection d'une métaséquence dans ce menu insère la métaséquence dans le champ, remplaçant ou enveloppant le texte actuellement sélectionné selon le cas.</p>
Régions marquées	<p>Lors de la « recherche et du remplacement » avec des expressions régulières, jusqu'à neuf sections du terme d'origine peuvent être substituées dans le terme de remplacement.</p> <p>Les métaséquences \"(' et \") indiquent le début et la fin d'une région balisée ; la section du texte correspondant qui se trouve dans la région balisée peut être incluse dans le texte de remplacement avec la métaséquence \"n' (où <i>n</i> est le numéro de la région balisée entre 1 et 9).</p> <p>Par exemple:</p> <p>Trouver : <i>les choses de</i> \((A-Za-z) +\)</p> <p>Remplacer par <i>des éléments appartenant à</i> \1</p> <p>Texte original : <i>Ce sont toutes les affaires de Michael .</i></p> <p>Texte remplacé : <i>Ce sont tous les éléments qui appartiennent à Michael.</i></p>

## Rechercher dans les fichiers

Les recherches de texte de fichier sont proposées par la fenêtre Rechercher dans les fichiers et depuis les Éditeurs de Code , pour rechercher des fichiers à partir de noms de données et de structures. Ces fichiers peuvent être des fichiers de code externes, des fichiers de code que vous avez déjà ouverts dans Enterprise Architect , des scripts de modèles internes ou le sous-système d'aide.

L'onglet « Recherche de fichiers » conserve un historique des chemins de fichiers que vous avez explorés, vous aidant à revenir rapidement aux dossiers fréquemment utilisés dans votre système de fichiers. Vous pouvez également sélectionner une string de recherche précédemment utilisée, si vous devez répéter une recherche plusieurs fois. Lorsque vous recherchez des fichiers de code, vous pouvez également limiter la recherche à des fichiers de types spécifiques, en sélectionnant les extensions de fichier, et en incluant uniquement le dossier sélectionné ou tous ses sous-dossiers. Une autre facilité utile consiste à pouvoir choisir d'afficher les résultats de la recherche sous la forme d'une liste de toutes les instances de la string , ou d'une liste de fichiers contenant la string avec les instances regroupées sous le fichier dans lequel elles se trouvent.

Pour toutes les recherches, vous pouvez préciser que la recherche doit être sensible à la casse et/ou faire correspondre la string de recherche à des mots complets.

### Accéder

Ruban	Explorer > Rechercher > Fichiers Exécuter > Source > Rechercher Exécuter > Source > Modifier > Rechercher dans les fichiers
Menu Contexte	Cliquez-droit sur le texte sélectionné   Rechercher <texte sélectionné>   Rechercher dans les fichiers
Raccourcis Clavier	F12, Ctrl+Maj+Alt+F

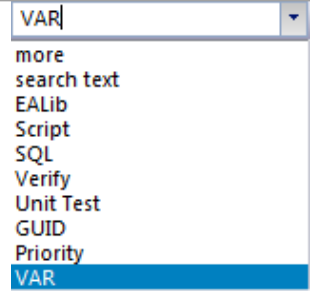
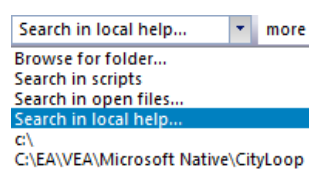
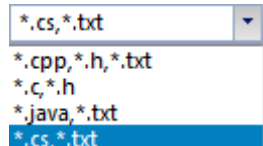




### Barre d'outils de recherche






Vous pouvez utiliser les options de la barre d'outils dans la fenêtre Rechercher dans les fichiers pour contrôler l'opération de recherche. L'état de chaque bouton persiste dans le temps pour toujours refléter vos critères de recherche précédents.



### Options

Option	Action
	Le champ « Texte à rechercher ». Type la string de texte à rechercher. Tout texte que vous saisissez est automatiquement enregistré dans la liste déroulante, jusqu'à un maximum de dix chaînes ; le texte ajouté après cela remplace la string de texte la plus ancienne de la liste. Vous pouvez cliquer sur la flèche déroulante et sélectionner l'une de ces chaînes de texte enregistrées, si vous préférez.

	
	<p>Le champ « Chemin de recherche ». Spécifiez le dossier dans lequel effectuer la recherche ou le type de recherche.</p> <p>Vous pouvez saisir le chemin du dossier à rechercher directement dans la zone de texte, ou cliquer sur la flèche déroulante et sélectionner « Parcourir le dossier » pour effectuer une recherche à l'aide de la dialogue « Parcourir le dossier ».</p> <p>Tous les chemins que vous saisissez sont automatiquement enregistrés dans la liste déroulante, jusqu'à un maximum de dix chemins ; les chemins ajoutés après cela remplacent le chemin le plus ancien de la liste. Vous pouvez sélectionner l'un de ces chemins enregistrés si vous préférez.</p> <p>Outre « Rechercher un dossier », il existe trois autres options fixes dans la liste déroulante :</p> <ul style="list-style-type: none"> <li>• « Rechercher dans les scripts », qui recherche les scripts locaux et définis par l'utilisateur dans la fenêtre Scriptant</li> <li>• « Rechercher dans les fichiers ouverts », qui limite la recherche aux fichiers que vous avez ouverts dans Enterprise Architect</li> <li>• « Rechercher dans l'aide locale », qui recherche les fichiers d'aide locaux qui ont été installés à partir du site Web Sparx Systems ; les résultats répertorient les rubriques d'aide contenant le terme de recherche, ainsi que le numéro de ligne et la ligne dans laquelle le texte apparaît</li> </ul> <p>Ces options désactivent la liste déroulante « Rechercher les types de fichiers ».</p>
	<p>Le champ « Rechercher les types de fichiers ». Cliquez sur la flèche déroulante et sélectionnez les types de fichiers (extensions de fichiers) à rechercher.</p>
	<p>Cliquez sur cette icône pour lancer la recherche.</p> <p>Pendant la recherche, tous les autres boutons de la barre d'outils sont désactivés. Vous pouvez annuler la recherche à tout moment en cliquant à nouveau sur le bouton Rechercher.</p> <p>Si vous changez l'un de ces boutons à bascule, vous devez exécuter à nouveau la recherche pour modifier la sortie.</p>
	<p>Cliquez sur cette icône pour activer ou désactiver la sensibilité à la casse de la recherche. Le message d'info-bulle identifie le paramètre actuel.</p>
	<p>Cliquez sur cette icône pour basculer entre la recherche de n'importe quelle correspondance et la recherche des seules correspondances qui forment un mot entier. Le message d'info-bulle identifie le paramètre actuel.</p>
	<p>Cliquez sur cette icône pour basculer entre la limitation de la recherche à un seul chemin et l'inclusion de tous les sous-dossiers sous ce chemin. Le message d'info-bulle identifie le paramètre actuel.</p>

	<p>Cliquez sur cette icône pour sélectionner le format de présentation des résultats de la recherche ; vous avez deux options :</p> <ul style="list-style-type: none"><li>• Liste Vue - (comme indiqué) chaque ligne de résultat se compose du chemin d'accès au fichier et du numéro de ligne, suivis du texte de la ligne ; plusieurs lignes d'un fichier sont répertoriées sous forme d'entrées distinctes</li><li>• Vue arborescente - () chaque ligne de résultat se compose du chemin d'accès au fichier qui correspond aux critères de recherche et du nombre de lignes correspondant au texte de recherche dans ce fichier ; vous pouvez développer l'entrée pour afficher le numéro de ligne et le texte de chaque ligne</li></ul>
	<p>Cliquez sur cette icône pour ajouter un nouvel onglet de recherche. Vous pouvez créer jusqu'à quatre nouveaux onglets de recherche. Les recherches peuvent également exécuter simultanément.</p>
	<p>Cliquez sur cette icône pour effacer les résultats.</p>
	<p>Si nécessaire, cliquez sur cette icône pour supprimer toutes les entrées dans les listes déroulantes Chemin de recherche, Texte de recherche et Types de fichiers de recherche.</p>

## Rechercher un fichier

L'onglet « Rechercher un fichier » de la fenêtre Rechercher dans les fichiers fournit un outil qui peut vous aider à trouver des fichiers plus rapidement. L'onglet agit comme un explorateur de système de fichiers et offre une alternative rapide à la dialogue d'ouverture de fichier habituelle. Les recherches de fichiers sont rapides et simples, vous permettant de rechercher les fichiers qui vous intéressent sans perdre votre flux de travail actuel. L'affichage peut être basculé entre la vue rapport et la vue liste.

### Accéder

Ruban	Explorer > Rechercher > Fichiers > Rechercher un fichier
Raccourcis Clavier	Ctrl+Maj+Alt+F

### Barre d'outils

La barre d'outils propose un filtre de recherche et une zone de liste déroulante de navigation dans les dossiers. La barre d'outils propose des options permettant de mémoriser les emplacements de recherche et d'alternier entre les vues de liste et de rapport.



### Options

	Cliquez pour accéder au dossier parent.
	Le contrôle de filtrage vous permet d'exclure les fichiers qui ne correspondent pas aux critères que vous saisissez. Le symbole générique * est automatiquement ajouté au texte, il n'est donc pas nécessaire de l'ajouter vous-même. Pour rechercher tous les fichiers qui contiennent le terme « jvm », saisissez simplement « jvm ». Pour trouver des images .png contenant le terme « red », vous pouvez saisir *red*.png. Appuyez sur la touche Entrée pour mettre à jour les résultats.
	Entrez le chemin d'un répertoire et appuyez sur la touche Entrée pour afficher les fichiers à cet emplacement Utilisez la liste déroulante pour sélectionner les emplacements marqués d'un signet pour le modèle actuel. Les emplacements peuvent être gérés à l'aide du menu de la barre d'outils.
	Permet de gérer les emplacements affichés dans le combo de répertoires. <ul style="list-style-type: none"> <li>Mémoriser le chemin - stocke la valeur actuelle du champ « Répertoire » de sorte que, lorsque vous revenez ultérieurement à la fenêtre Rechercher dans les fichiers, le champ « Répertoire » utilise par défaut cette valeur (s'il s'agit de la seule valeur « mémorisée ») ou propose la valeur dans la liste déroulante</li> </ul>



	<ul style="list-style-type: none"> <li>• Oublier le chemin - efface la valeur actuelle de la mémoire afin qu'elle ne soit pas proposée comme valeur possible pour le champ « Répertoire »</li> <li>• Mémoriser le filtre - stocke la valeur actuelle dans le champ « Filtre » de sorte que lorsque vous revenez ultérieurement à la fenêtre Rechercher dans les fichiers, le champ « Filtre » utilise par défaut cette valeur</li> <li>• Oublier le filtre - supprime la valeur du champ « Filtre » de la mémoire afin qu'elle ne soit pas placée dans le champ la prochaine fois que vous accédez à la fenêtre</li> </ul>
	<p>Dans cette vue, la liste affiche les colonnes « Nom », « Date de modification », « Type » et « Taille ».</p> <p>Les colonnes peuvent être triées par ordre croissant ou décroissant. Cliquez une troisième fois sur la colonne pour supprimer l'ordre de tri.</p>
	<p>La vue de liste supprime les colonnes et est pratique lorsqu'un dossier contient de nombreux fichiers.</p>

## Raccourcis Clavier

	Ensembles se concentrent sur le contrôle du filtre.
	Accède au dossier parent.
	Accède au dossier parent.
	Si un dossier est sélectionné, ouvre le dossier, sinon ouvre les fichiers sélectionnés.

## Recherche Intelli-sense

Les fonctionnalités Intelli-sense d' Enterprise Architect sont développées à l'aide de l'outil Code Miner de Sparx Systems . Code Miner offre un accès rapide et complet aux informations d'une base de code existante. Le système offre un accès complet à tous les aspects du code source d'origine, soit « à la volée », comme on pourrait le faire avec un éditeur de code, soit sous forme de résultats de recherche produits par des requêtes écrites dans le langage mFQL Code Miner .

### Accéder

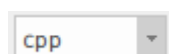
Dans la fenêtre Rechercher dans les fichiers, cliquez sur l'onglet « Code Miner ».

Ruban	Explorer > Rechercher > Fichiers
Raccourcis Clavier	Ctrl+Maj+Alt+F

### Le contrôle Code Miner

Ce contrôle présente une interface permettant d'effectuer des requêtes sur plusieurs bases de code à la fois. Les bases de code qu'il utilise sont des bases de données construites à l'aide de l'outil Code Miner d' Enterprise Architect . Ces bases de données forment une bibliothèque, qui peut également être partagée lorsqu'elle est déployée en tant que service. Les requêtes pouvant être exécuter sont répertoriées et sélectionnées à l'aide de la barre d'outils, ce qui permet un accès facile au code source des requêtes, pour l'édition et la composition. Les requêtes n'ont pas besoin d'être compilées ; elles sont visualisées, modifiées et enregistrées comme n'importe quel fichier de code source. Les requêtes qui prennent un seul paramètre peuvent utiliser n'importe quelle sélection dans un éditeur de code ouvert. L'interface supporte la saisie manuelle des paramètres pour les requêtes qui prennent plusieurs arguments.

Le premier contrôle de la barre d'outils répertorie les espaces de noms disponibles. La sélection d'un espace de noms limite les requêtes affichées à celles qui se trouvent dans cet espace de noms.



Le contrôle suivant fournit une liste déroulante de toutes les requêtes dans le fichier de requête pour l'espace de noms sélectionné.



Le troisième contrôle est une zone de liste déroulante d'édition. Par défaut, un seul paramètre de requête est extrait du texte sélectionné dans un éditeur de code ouvert, mais vous pouvez également saisir le(s) paramètre(s) directement dans ce champ. Les paramètres multiples doivent être séparés par des virgules. Ceci est suivi du bouton Rechercher pour exécuter la requête. Les requêtes peuvent être modifiées à tout moment à l'aide du bouton Modifier à côté du bouton Rechercher.



Le panneau « Résultat » est un contrôle arborescent qui répertorie les résultats de la requête regroupés par fichier.

**Result**

```
▶ e:\java\jdk-1.8.0_91\src\com\sun\org\apache\xerces\internal\util\domutil.java
▶ e:\java\jdk-1.8.0_91\src\java\util\stream\pipelinehelper.java
▶ e:\java\jdk-1.8.0_91\src\java\util\vector.java
▶ e:\java\jdk-1.8.0_91\src\javax\swing\defaultlistmodel.java
```

## Bibliothèques Code Miner

Les bibliothèques Code Miner sont un ensemble de bases de données qui peuvent être utilisées par les fournisseurs Enterprise Architect Intelli-sense pour obtenir et interroger des informations sur plusieurs bases de code. Chaque base de données est créée à partir du répertoire de code source racine d'une base de code, à l'aide d'une grammaire spécialisée adaptée à son langage (C++, Java ou C#).

Les bibliothèques sont créées, mises à jour, supprimées ou ajoutées dans l'Éditeur de Script Analyseur. Un scénario typique d'utilisation de cette fonctionnalité serait de créer une base de données pour un projet de développement et des bases de données supplémentaires pour les frameworks référencés par le projet. Votre base de données de développement peut être mise à jour fréquemment au fur et à mesure que les modifications de code s'accumulent, tandis que les frameworks statiques seront mis à jour moins souvent. Les bibliothèques peuvent être recherchées de manière similaire à l'outil « Recherche de fichiers », mais Code Miner offre des capacités de recherche avancées grâce à son langage mFQL.

- Plusieurs domaines/cadres peuvent être recherchés à la fois
- Une requête peut être exécuter en une fraction du temps requis pour une recherche de fichier
- Les requêtes peuvent être codées pour faciliter la recherche de critères complexes
- Les requêtes peuvent prendre plusieurs paramètres
- Tous les fichiers sont indexés sur la base de constructions UML équivalentes, permettant des recherches intelligentes produisant des résultats significatifs dans un cadre modélisation

## Fichiers Query Code Miner

Les requêtes Code Miner sont conservées dans un fichier de code source unique qui doit avoir l'extension .mFQL. Un ensemble de requêtes de base est fourni avec chaque installation Enterprise Architect ; celles-ci peuvent être situées dans le sous-répertoire config\codeminer. Ce fichier de requête doit être nommé par défaut dans tout script Analyzer que vous modifiez.

Avant de modifier une requête, il est conseillé de copier ce fichier dans un emplacement de travail et de nommer la copie dans tout script Analyzer que vous utilisez. De cette façon, vous disposerez toujours d'un fichier de référence auquel vous référer.

Les requêtes sont mieux considérées comme des fonctions écrites dans le langage mFQL. En tant que telles, elles ont des noms uniques, peuvent être qualifiées par un espace de noms unique et peuvent spécifier des paramètres. Le fichier fournit les requêtes répertoriées dans la barre d'outils du contrôle Intelli-sense. Chaque fois que des modifications apportées à un fichier de requête sont enregistrées, les requêtes répertoriées dans la zone de liste déroulante de la barre d'outils de recherche sont mises à jour en conséquence. Cette image est un exemple de requête simple écrite en mFQL.

```
188 .....
189 namespace java
190 {
191 //
192 // Find all references
193 //
194 query::findByName($param1)
195 {
196     distinct(GetByValue( $param1 +))
197 }
198
199 query::findMethodByName($name)
200 {
201     move( 1, "METHOD", intersect( GetByNode("NAME"), GetByValue( $name ) ) )
202 }
203
204 query::findMethodCall($name)
205 {
206     filter( "METHOD_ACCESS", intersect(GetByNode("NAME"), GetByValue( $name ) ) )
207 }
208
```

# Raccourcis clavier Éditeur de Code

## Clés

Clé	Description
Ctrl+G	Déplacer le curseur vers une ligne spécifiée
↓	Déplacer le curseur d'une ligne vers le bas
Maj+↓	Étendre la sélection vers le bas d'une ligne
Ctrl+↓	Faites défiler une ligne vers le bas
Alt+Maj+↓	Étendre la sélection rectangulaire d'une ligne vers le bas
↑	Déplacer le curseur d'une ligne vers le haut
Maj+↑	Étendre la sélection d'une ligne vers le haut
Ctrl+↑	Faire défiler une ligne vers le haut
Alt+Maj+↑	Étendre la sélection rectangulaire d'une ligne vers le haut
Ctrl+(	Déplacer le curseur d'un paragraphe vers le haut
Ctrl+Maj+(	Étendre la sélection jusqu'à un paragraphe
Ctrl+)	Déplacer le curseur d'un paragraphe vers le bas
Ctrl+Maj+)	Étendre la sélection d'un paragraphe vers le bas
←	Déplacer le curseur d'un caractère vers la gauche
Maj+←	Étendre la sélection d'un caractère vers la gauche
Ctrl+←	Déplacer le curseur d'un mot vers la gauche
Ctrl+Maj+←	Étendre la sélection d'un mot vers la gauche
Alt+Maj+←	Étendre la sélection rectangulaire d'un caractère vers la gauche
→	Déplacer le curseur d'un caractère vers la droite.
Maj+→	Étendre la sélection d'un caractère vers la droite
Ctrl+→	Déplacer le curseur d'un mot vers la droite
Ctrl+Maj+→	Étendre la sélection d'un mot à droite

Alt+Maj+→	Étendre la sélection rectangulaire d'un caractère vers la droite
Ctrl+←	Déplacer le curseur d'une partie du mot vers la gauche
Ctrl+Maj+←	Étendre la sélection à gauche d'une partie du mot
Ctrl+→	Déplacer le curseur vers la droite d'une partie du mot
Ctrl+Maj+→	Étendre la sélection à droite d'une partie du mot
Maison	Déplacer le curseur au début de la ligne actuelle
Maj+Accueil	Étendre la sélection jusqu'au début de la ligne actuelle
Ctrl+Accueil	Déplacer le curseur au début du document
Ctrl+Maj+Accueil	Étendre la sélection jusqu'au début du document
Alt+Accueil	Déplacer le curseur au début absolu de la ligne
Alt+Maj+Accueil	Étendre la sélection rectangulaire jusqu'au début de la ligne
Fin	Déplacer le curseur à la fin de la ligne actuelle
Maj+Fin	Étendre la sélection jusqu'à la fin de la ligne actuelle
Ctrl+Fin	Déplacer le curseur à la fin du document
Ctrl+Maj+Fin	Étendre la sélection jusqu'à la fin du document
Alt+Fin	Déplacer le curseur jusqu'à la fin absolue de la ligne
Alt+Maj+Fin	Étendre la sélection rectangulaire jusqu'à la fin de la ligne
Page précédente	Déplacer le curseur vers le haut d'une page
Maj+Page précédente	Étendre la sélection jusqu'à une page
Alt+Maj+Page précédente	Étendre la sélection rectangulaire sur une page
Page suivante	Déplacer le curseur vers le bas d'une page
Maj+Page suivante	Étendre la sélection vers le bas d'une page
Alt+Maj+Page suivante	Étendre la sélection rectangulaire sur une page
Supprimer	Supprimer le caractère à droite du curseur
Maj+Suppr	Sélection de coupe

Ctrl+Suppr	Supprimer le mot à droite du curseur
Ctrl+Maj+Suppr	Supprimer jusqu'à la fin de la ligne
Insérer	Activer/désactiver la frappe
Maj+Insertion	Coller
Ctrl+Insertion	Sélection de copie
Retour arrière	Supprimer le caractère à gauche du curseur
Maj+Retour arrière	Supprimer le caractère à gauche du curseur
Ctrl+Retour arrière	Supprimer le mot à gauche du curseur
Ctrl+Maj+Retour arrière	Supprimer du début de la ligne jusqu'au curseur
Alt+Retour arrière	Annuler supprimer
Languette	Indenter le curseur d'une tabulation
Ctrl+Maj+I	Indenter le curseur d'une tabulation
Maj+Tab	Annuler l'indentation du curseur d'une tabulation
Ctrl+clavier (+)	Agrandir
Ctrl+clavier(-)	Zoom arrière
Ctrl+clavier (/)	Restaurer le zoom
Ctrl+Z	Annuler
Ctrl+Y	Refaire
Ctrl+X	Sélection de coupe
Ctrl+C	Sélection de copie
Ctrl+V	Coller
Ctrl+L	Ligne de coupe
Ctrl+T	Transposer la ligne
Ctrl+Maj+T	Copier la ligne
Ctrl+A	Sélectionner le document entier
Ctrl+D	Sélection en double

Ctrl+U	Convertir la sélection en minuscules
Ctrl+Maj+U	Convertir la sélection en majuscules
Ctrl+E	Déplacer le curseur vers l'accolade correspondante
Ctrl+Maj+E	Étendre la sélection à l'accolade correspondante
Ctrl+Maj+C	Basculer la ligne de commentaire sur la sélection
Ctrl+Maj+X	Basculer le commentaire du flux sur la sélection.
Ctrl+F2	Activer/désactiver le signet
F2	Aller au signet suivant
Maj+F2	Aller au signet précédent
Ctrl+Maj+F2	Effacer tous les signets du fichier actuel
Ctrl+Maj+W	Activer/désactiver les caractères d'espacement
Ctrl+Maj+L	Basculer les caractères EOL
Ctrl+Espace	Invoquer la saisie semi-automatique.
Ctrl+-	Revenir en arrière dans l'historique du curseur
Ctrl+Maj+-	Avancer dans l'historique du curseur
F12	Démarrer /Annuler la recherche de mot-clé dans le(s) fichier(s).
Ctrl+F	Rechercher du texte
Ctrl+R	Remplacer le texte

## Notes

- En plus de ces touches, vous pouvez attribuer des combinaisons de touches (Ctrl+Alt+<n>) à des macros que vous définissez dans la Source Éditeur de Code



# Motifs d'application ( Modèle + Code)

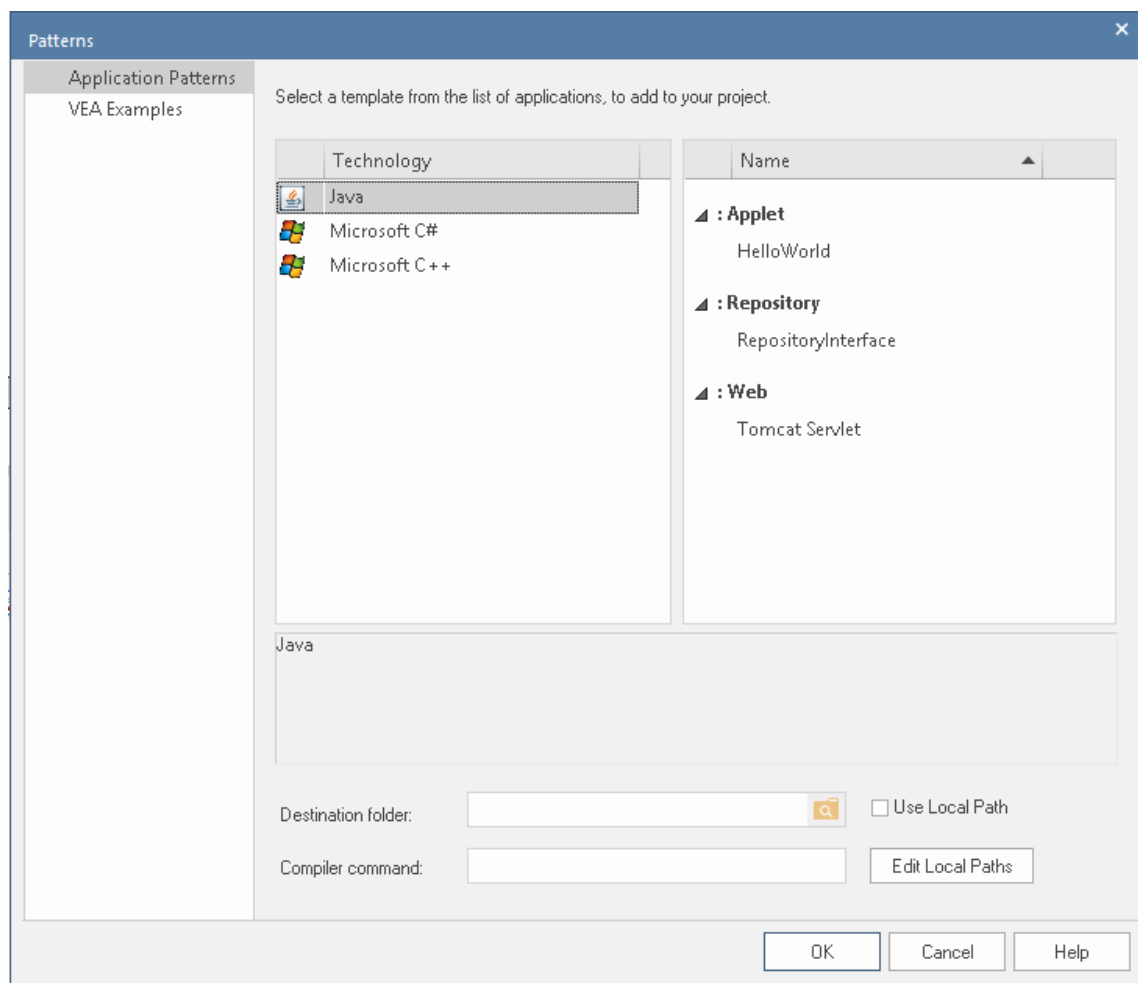
Pour vous permettre de démarrer un projet basé sur du code le plus rapidement possible, Enterprise Architect vous aide à générer des projets de démarrage comprenant des informations sur le modèle, du code et des scripts de construction pour l'un des nombreux types d'applications de base. Motifs incluent :

- Applications Windows MFC
- Programmes Java
- Services Web ASP.NET

## Accéder

Ruban	Développer > Code source > Créer à partir de Motif > Motifs d'application
-------	---

## Générer des modèles



Option	Action

Technologie	Sélectionnez la technologie appropriée.
Nom	Affiche les Motifs d'application disponibles pour la technologie sélectionnée ; sélectionnez le Motif requis à importer.
<description>	Affiche une description du Motif sélectionné.
Dossier de destination	Recherchez et sélectionnez le répertoire dans lequel charger le code source de l'application.
Utiliser le chemin local	Activez la sélection d'un chemin local existant pour placer le code source sous ; modifiez le champ « Dossier de destination » en une sélection déroulante.
Commande Compilateur	Affiche le chemin de commande du compilateur par défaut pour la technologie sélectionnée ; vous devez : <ul style="list-style-type: none"> <li>• Confirmez que le compilateur peut être trouvé à ce chemin, ou</li> <li>• Modifier le chemin d'accès à l'emplacement du compilateur</li> </ul>
Modifier les chemins locaux	De nombreux Motifs d'application spécifient leur compilateur à l'aide d'un chemin local. La première fois que vous utilisez un Motif vous devez cliquer sur ce bouton pour vous assurer que le chemin local pointe vers le bon emplacement. La dialogue « Chemins locaux » s'affiche.

## Notes

- Si nécessaire, vous pouvez publier Motifs d'application personnalisés en ajoutant des fichiers au répertoire *AppPatterns* dans lequel Enterprise Architect est installé ; les répertoires de niveau supérieur sont répertoriés comme Technologies et peuvent contenir un fichier d'icône pour personnaliser l'icône affichée pour la technologie. Les répertoires situés en dessous sont définis comme des groupes dans la liste Motifs ; les Motifs sont identifiés par la présence de quatre fichiers portant un nom correspondant : un fichier zip (.zip), un fichier XMI (.xml), un fichier de configuration (.cfg) et une icône facultative (.ico)
- Le fichier de configuration supporte ces champs :
  - [fournisseur], [langue], [plateforme], [url], [description], [version] - tous affichés dans la <description> champ
  - [xmireootpaths] - le chemin racine du code source dans le XMI exporté ; il est remplacé par le dossier de destination sélectionné lorsque l'utilisateur applique le Motif d'application

## Intégration MDG et Ingénierie de Code

MDG Integration for Eclipse et MDG Integration for Visual Studio sont des produits qui vous aident à créer et à maintenir vos modèles UML directement dans ces deux environnements de développement intégrés populaires, à l'aide de la fenêtre Enterprise Architect Navigateur . Les modèles peuvent être générés en code source à l'aide du moteur gabarit riche et flexible qui donne à l'ingénieur un contrôle complet sur la manière dont le code est généré. Le code source existant peut également être rétroconçu et synchronisé avec les modèles UML . Une fois l'intégration installée, l'IDE deviendra une plate-forme modélisation riche en fonctionnalités, économisant du temps et des efforts et réduisant le risque d'erreur en reliant la gestion des exigences, Architecture et la conception à l'ingénierie du code source.

Une documentation riche et expressive peut être générée automatiquement dans une large gamme de formats, notamment DOCX, PDF et HTML. La documentation peut inclure diagrammes d'exigences, de conception et architecture ainsi que des descriptions de code source, mettant le code source en contexte.

Vous pouvez acheter MDG Integration for Eclipse <sup>TM</sup> et MDG Integration for Visual Studio <sup>TM</sup> ou télécharger les éditions d'essai à partir du site Web Sparx Systems .

# Interaction et expérience utilisateur

## Visualisez les interfaces utilisateur pour les smartphones, les tablettes et les ordinateurs portables à l'aide de wireframes

Les pages de la boîte à outils Wireframe proposent une large gamme d'icônes que vous pouvez utiliser dans modélisation filaire pour représenter l'apparence d'un périphérique à un moment particulier de l'exécution d'une application. Les périphériques que vous pouvez modéliser incluent :

- Téléphones et tablettes Android
- iPhones et tablettes Apple
- Téléphones Windows
- Boîtes de dialogue à l'écran
- Pages Web - pour modéliser le fonctionnement des pages Web

## Accéder

Sur la boîte à outils Diagramme , cliquez sur  pour afficher la dialogue ' Trouvez Item de Boîte à Outils ' et spécifiez ' Wireframing ', ' Android ', ' Apple ', ' Dialogue ', ' Windows Phone ' ou ' Webpage '.

Ruban	Conception > Diagramme > Boîte à outils
Raccourcis Clavier	Ctrl+Maj+3

## Notes


- Chacune des boîtes à outils Diagramme de wireframing fournit un ou plusieurs Motifs que vous pouvez faire glisser sur un diagramme pour illustrer ce que vous pourriez réaliser ou pour servir de base au modèle que vous développez.
- La boîte dialogue « Propriétés » pour les éléments Wireframe s'ouvre automatiquement sur un onglet « Wireframe » de niveau supérieur sur lequel vous pouvez modifier directement le rendu de l'élément, ou sur un onglet « Valeur Étiquetés » de deuxième niveau si vous définissez le rendu de l'élément en modifiant le XML pour les propriétés de ce type d'élément
- Certains des éléments que vous créez à partir des pages de la boîte à outils « Wireframe » sont correctement rendus après avoir modifié les Valeur Étiquetés qui définissent leurs caractéristiques
- Au fur et à mesure que vous développez vos diagrammes d'interface, vous pouvez établir les positions et disposition des éléments à main levée en faisant glisser et en « déplaçant » les éléments, ou imposer une certaine régularité à l'aide des options diagramme « Snap To Grid »

# Démarrage

Pour commencer avec Interaction utilisateur et expertise dans Enterprise Architect , sélectionnez d'abord la perspective Wireframe

## Sélection de la perspective

Enterprise Architect divise les nombreuses fonctionnalités de l'outil en Perspectives , ce qui vous permet de vous concentrer sur une tâche spécifique et de travailler avec les outils dont vous avez besoin sans être distrait par d'autres fonctionnalités . Pour travailler avec les fonctionnalités Modélisation de l'expérience utilisateur, vous devez d'abord sélectionner cette perspective :

 <nom de la perspective> > Conception UX > Wireframes

Le paramétrage de la Perspective garantit que les diagrammes Wireframe, leurs boîtes à outils et autres fonctionnalités de la Perspective seront disponibles par défaut.

## Exemple Diagramme

Un exemple de diagramme fournit une introduction visuelle au sujet et vous permet de voir certains des éléments et connecteurs importants qui sont créés pour spécifier ou décrire la manière dont un wireframe, dialogue ou une page Web est défini, notamment : les téléphones et tablettes Android, Apple et Windows , les boîtes de dialogue et les pages Web avec des boutons, des zones de liste déroulante et des boutons radio.

## Modèles de flux d'interaction IFML

Le langage Modélisation de flux d'interaction (IFML) fournit aux architectes système, aux ingénieurs logiciels et aux développeurs de logiciels des outils qui supportent la description indépendante de la plate-forme des interfaces utilisateur graphiques pour les applications accessibles ou déployées sur des systèmes tels que les ordinateurs de bureau, les ordinateurs portables, les assistants numériques personnels, les téléphones mobiles et les tablettes. Le langage a été développé par l' Object Management Group ; la spécification IFML (version 1.0. Février 2015) est disponible sur le site Web de l'OMG.

Vous pouvez accéder au profil UML de l'OMG pour IFML dans Enterprise Architect .

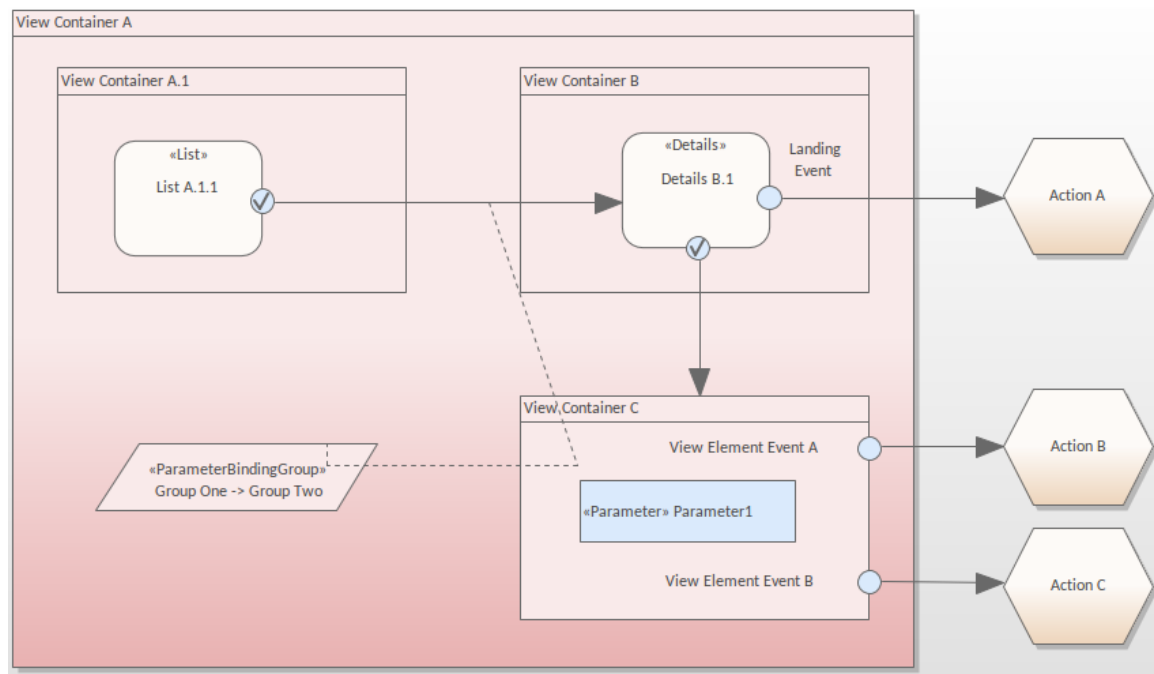
### IFML dans Enterprise Architect

Dans Enterprise Architect vous pouvez modéliser les flux d'interaction des applications rapidement et simplement grâce à l'utilisation de la MDG Technologie intégrée à l'installateur Enterprise Architect . Les facilités IFML sont fournies sous la forme de :

- Onze motifs de modèles IFML pour la saisie d'informations, l'interaction et le wireframe, les recherches et les applications de bureau, disponibles sur la page 'Interaction Flow IFML' du Constructeur de Modèle .
- Deux types diagramme IFML - diagramme IFML et diagramme IFML Domain Modèle - accessibles via la page à onglet ' Constructeur de diagramme ' de la dialogue Constructeur de Modèle
- Pages « Concepts essentiels », « Noyau » et « Extensions » de l'IFML dans la boîte à outils Diagramme

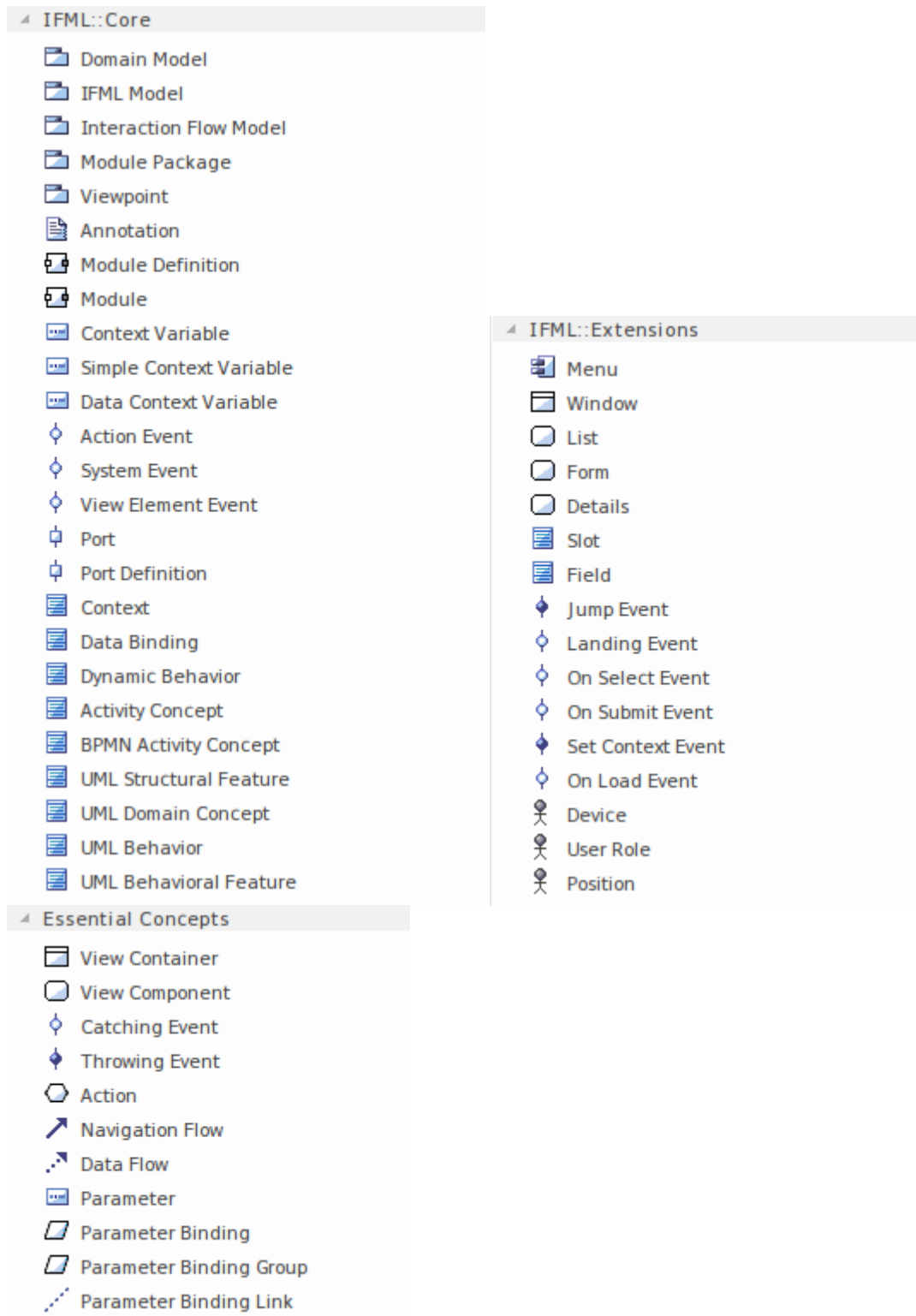
Vous pouvez, si vous le souhaitez, faire de la technologie IFML votre technologie par défaut en la sélectionnant dans la fenêtre MDG Technologies et en cliquant sur le bouton Définir Actif .

### Exemple Diagramme



### Pages de la boîte à outils IFML

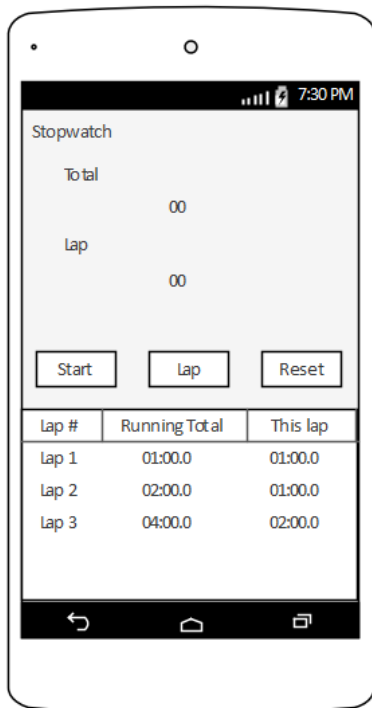
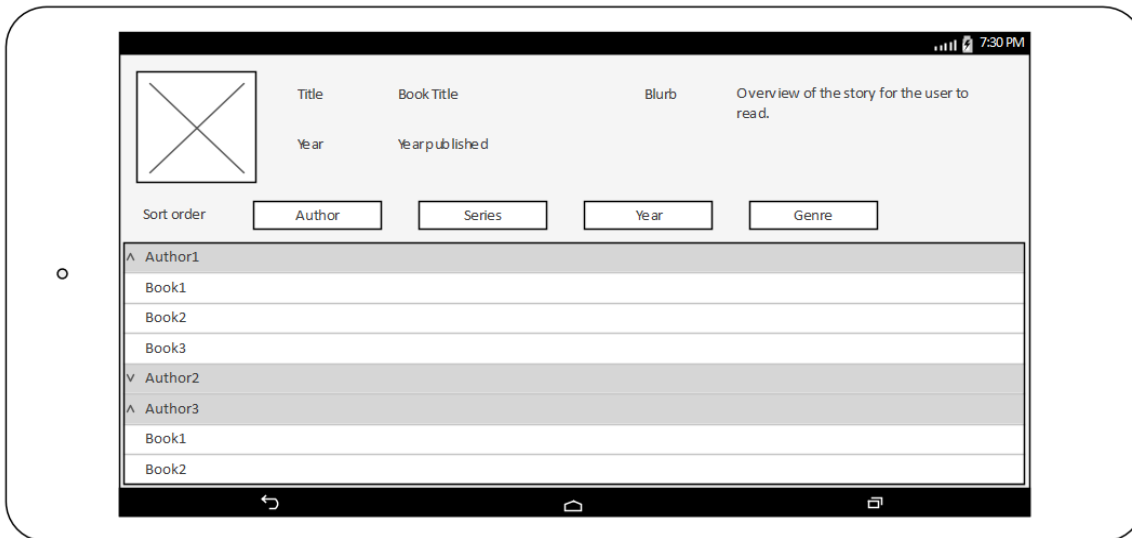
Les objets définis par la Spécification IFML peuvent être créés dans votre modèle à l'aide des icônes de ces pages de la boîte à outils Diagramme :




Certains objets de l'IFML Spécification sont configurés par Valeur Étiquetées sur le type d'élément principal. Par exemple, l'élément Vue Container peut être défini comme conteneur Vue par défaut, Landmark ou XOR en définissant la Valeur Étiquetée appropriée sur l'élément sur « True ».

# Boîte à outils de wireframe Android

Les pages 'Android Wireframing' de la boîte à outils Diagramme fournissent les gabarits pour modélisation l'apparence physique d'une tablette ou d'un téléphone Android à un état donné d'exécution d'une application. Elles fournissent également Motifs pour générer une structure de modèle standard pour chaque appareil Android.



## Accéder

Sur la Boîte à outils Diagramme , cliquez sur  pour afficher la dialogue ' Trouvez Item de Boîte à Outils ' et spécifiez 'Wireframing' ou 'Android'.

Ruban	Conception > Diagramme > Boîte à outils
-------	---






Raccourcis Clavier	Ctrl+Maj+3
--------------------	------------



## Types d'écran Android

Item	Description
Téléphone Android	<p>Génère un cadre pour le visage du téléphone Android que vous modélisation . Une prompt s'affiche pour vous permettre de spécifier l'orientation portrait ou paysage.</p> <p>Les contrôles enfants seront contenus dans la zone de l'écran.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Boutons de menu - cliquez sur le bouton déroulant et sélectionnez « Afficher » ou « Masquer » la barre de boutons de menu en bas de l'écran</li> <li>• Barre de notification - cliquez sur le bouton déroulant et sélectionnez « Afficher » ou « Masquer » la barre de notification en haut de l'écran</li> </ul>
Tablette Android	<p>Génère un cadre pour la tablette Android que vous modélisation . Une prompt s'affiche pour vous permettre de spécifier l'orientation portrait ou paysage.</p> <p>Les contrôles enfants seront contenus dans la zone de l'écran.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Boutons de menu - cliquez sur le bouton déroulant et sélectionnez « Afficher » ou « Masquer » la barre de boutons de menu en bas de l'écran</li> <li>• Barre de notification - cliquez sur le bouton déroulant et sélectionnez « Afficher » ou « Masquer » la barre de notification en haut de l'écran</li> </ul>
Espace Client	<p>Génère un élément de cadre qui représente la zone client de l'appareil.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Style de Bordure - cliquez sur la flèche déroulante et choisissez de rendre la bordure sous forme de ligne continue ou de ligne pointillée, ou de ne pas avoir de bordure du tout (Aucune)</li> <li>• Barre de défilementH - cliquez sur la flèche déroulante et choisissez de placer une barre de défilement horizontale en haut ou en bas de la zone client, ou de ne pas avoir de barre de défilement horizontale ( &lt;none&gt; )</li> <li>• Barre de défilement V - cliquez sur la flèche déroulante et choisissez de placer une barre de défilement verticale à gauche ou à droite de la zone client, ou de ne pas avoir de barre de défilement verticale ( &lt;none&gt; )</li> </ul>

## Composite

Item	Description
Vue de la liste extensible	<p>Génère un élément qui représente une liste groupée à deux niveaux qui peut être développée pour afficher un ou les deux niveaux d'éléments.</p> <p>Double-cliquez sur l'élément pour afficher la dialogue « Propriétés » sur la page « Wireframe », qui affiche le nœud racine (le nom de l'élément de liste extensible),</p>

	<p>les éléments d'en-tête de groupe (directement sous le nom de l'élément) et les éléments de groupe (sous les en-têtes de groupe).</p> <ul style="list-style-type: none"> <li>• Pour ajouter un nouvel élément, cliquez sur le niveau supérieur (le nom de l'élément ou les noms des titres de groupe) et sur le bouton Ajouter, et en réponse à l' prompt saisissez le nom de l'élément, ce qui ajoute le nom au bas de la section de la liste dans laquelle vous l'avez créé. Pour positionner l'élément sur la liste, cliquez dessus dans la liste et cliquez sur les boutons  </li> <li>• Pour supprimer un élément de la liste, cliquez dessus et cliquez sur le bouton Supprimer ; l'élément est immédiatement supprimé de la liste</li> </ul> <p>Cliquez sur le nom Item (l'élément de liste de premier niveau directement sous le nom de l'élément).</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le nom de ce groupe d'éléments</li> <li>• Développé - cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher les sous-éléments de cet élément ou Faux pour masquer les sous-éléments</li> </ul> <p>Cliquez sur le nom d'un sous-élément (élément de liste de deuxième niveau) directement sous le nom d'un Item . Les nœuds de sous-élément n'ont pas de nœud enfant.</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le nom du sous-élément</li> <li>• Sélectionné - cliquez sur la flèche déroulante et sélectionnez Vrai pour mettre en surbrillance l'élément comme sélectionné ou Faux pour omettre la surbrillance</li> </ul>
Tableau	<p>Génère un élément Tableau avec des colonnes, des lignes et des cellules étiquetées. Double-cliquez sur le tableau pour faire apparaître la dialogue « Propriétés » de l'élément sur la page « Wireframe », qui fournit les facilités d'édition du tableau (ajout, renommage et suppression de colonnes et de lignes, modification de la largeur des colonnes et édition du texte des cellules) via des options et des boutons du menu contextuel. Note que l'éditeur ne fournit pas une image réelle de l'apparence du tableau à l'écran.</p> <ul style="list-style-type: none"> <li>• Tracer des lignes - cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher les lignes horizontales et verticales entre les cellules tableau , ou Faux pour masquer les lignes</li> <li>• Mettre en surbrillance les en-têtes - cliquez sur la flèche déroulante et sélectionnez Vrai pour mettre en surbrillance l'en-tête de chaque colonne, ou Faux pour laisser les colonnes tableau d'une couleur uniforme</li> </ul>
Onglet Hôte	<p>Génère un élément de contrôle d'onglets sur le diagramme . Vous pouvez nommer les onglets et les marquer comme sélectionnés. Cependant, les éléments enfants ne « basculeront » pas lors du changement d'onglets (c'est-à-dire que définir un onglet différent comme sélectionné affichera toujours les mêmes éléments enfants dans l'espace des onglets).</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Onglets - cliquez sur la flèche déroulante et cliquez sur l'onglet à mettre en surbrillance comme sélectionné Vous pouvez nommer les onglets et en ajouter d'autres à la liste en modifiant la liste « Valeurs : » dans les notes Valeur Étiquetée</li> </ul> <p>Si vous réduisez la taille de l'élément de sorte que tous les onglets ne puissent pas être affichés, une icône de défilement (  ) s'affiche automatiquement dans le coin supérieur droit de l'élément.</p>
Liste simple	Génère une liste déroulante contenant une liste d'éléments sans sous-éléments.

	<p>Double-cliquez sur l'élément pour afficher la dialogue « Propriétés » sur la page « Wireframe ».</p> <ul style="list-style-type: none"> <li>• Pour modifier un élément en un texte différent, remplacez le texte « Élément&lt;n&gt; » dans le champ « Nom »</li> <li>• Mettez en surbrillance l'élément comme sélectionné, dans le champ « Sélectionné », cliquez sur la flèche déroulante et sélectionnez Vrai</li> <li>• Ajoutez un autre élément, cliquez sur le nom de l'élément et sur le bouton Ajouter, puis saisissez le nom de l'élément</li> <li>• Supprimez un élément de la liste, cliquez dessus et cliquez sur le bouton Supprimer</li> <li>• Déplacez un élément vers une position différente dans la liste, cliquez dessus et cliquez sur l'un des boutons   selon le cas</li> </ul>
<p>Liste à 2 voies</p>	<p>Génère une liste déroulante comme pour une liste simple, mais chaque nom d'élément est en gras et peut avoir une description sous le nom de l'élément.</p> <p>Double-cliquez sur l'élément pour afficher la dialogue « Propriétés » sur la page « Wireframe ».</p> <ul style="list-style-type: none"> <li>• Pour ajouter la description, cliquez sur le nom de l'article et, dans le champ « Texte », saisissez le texte de la description</li> </ul> <p>Les autres options sont les mêmes que pour la liste simple.</p>
<p>Liste de contrôle</p>	<p>Génère une liste déroulante comme pour une liste simple, mais chaque nom d'élément est accompagné d'un contour coché à sa droite. Pour les éléments sélectionnés, le contour est rempli.</p> <p>Double-cliquez sur l'élément pour afficher la dialogue « Propriétés » sur la page « Wireframe ».</p> <ul style="list-style-type: none"> <li>• Pour définir une coche sur sélectionnée (remplie), cliquez sur la flèche déroulante « Cochée » et cliquez sur Vrai ; pour modifier la coche sur non sélectionnée (contour), définissez le champ sur Faux</li> </ul> <p>Les autres options sont les mêmes que pour la liste simple.</p>
<p>Liste à choix unique</p>	<p>Génère une liste déroulante comme pour une liste simple, mais chaque nom d'élément possède un contour de bouton radio à sa droite. Pour un élément sélectionné, le contour est rempli.</p> <p>Double-cliquez sur l'élément pour afficher la dialogue « Propriétés » sur la page « Wireframe ».</p> <ul style="list-style-type: none"> <li>• Pour définir un bouton radio sur sélectionné (rempli), cliquez sur la flèche déroulante du champ « Sélectionné » et cliquez sur Vrai ; pour modifier le bouton radio sur non sélectionné (contour), définissez le champ sur Faux</li> </ul> <p>Tous les éléments peuvent être définis sur False (non sélectionnés), mais si plusieurs éléments sont définis sur True (sélectionnés), seul l'élément le plus bas de la liste s'affiche comme sélectionné</p> <p>Les autres options sont les mêmes que pour la liste simple.</p>


## Widgets de formulaire

Item	Description
	Génère un élément représentant un commutateur Android simple. Le commutateur

Changer	<p>peut avoir deux états (par exemple, activé et désactivé) et une étiquette tirée du nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>States - cliquez sur la flèche déroulante et sélectionnez la position du commutateur représentée par l'élément Vous pouvez modifier le champ « Valeurs : » dans les Notes Valeur Étiquetée pour modifier le texte des valeurs d'état Vous pouvez ajouter plus de deux valeurs, mais vous ne pouvez basculer le champ « States » qu'entre les deux premières valeurs de la liste ; les autres valeurs sont ignorées si elles sont sélectionnées</li> </ul>
Contrôle de notation	<p>Génère un élément représentant une bande de notation par étoiles. L'élément affiche toujours cinq étoiles et le nombre d'étoiles remplies indique la note.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Évaluation - cliquez sur la flèche déroulante et sélectionnez le nombre d'étoiles à afficher remplies (la note)</li> </ul> <p>Vous ne pouvez redimensionner l'élément qu'horizontalement ; la dimension verticale s'ajuste automatiquement pour toujours représenter cinq étoiles de forme uniforme.</p>
Bouton bascule	<p>Génère un élément représentant un commutateur unicellulaire sans étiquette (le nom de l'élément n'est pas affiché).</p> <p>Vous pouvez modifier l'état représenté de la même manière que pour l'élément « Switch ».</p>
Barre de progression (grande)	<p>Génère un élément représentant l'icône de progression circulaire d'Android.</p>
Barre de progression (horizontale)	<p>Génère un élément représentant la progression d'un processus, par défaut à 30 % d'avancement.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Progression - saisissez le pourcentage d'avancement de la progression à représenter sur l'élément</li> </ul>
Bouton	<p>Génère un élément Button étiqueté simple, le texte de l'étiquette étant le nom de l'élément.</p>
Bouton radio	<p>Génère un élément de bouton radio étiqueté, le texte de l'étiquette étant le nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>State - cliquez sur la flèche déroulante et sélectionnez l'état à représenter - « Sélectionné » (rempli) ou « Non sélectionné » (contour)</li> </ul>
Case à cocher	<p>Génère un élément de case à cocher étiqueté, le texte de l'étiquette étant le nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>State - cliquez sur la flèche déroulante et sélectionnez l'état à représenter - « Coché » (icône de coche) ou « Non coché » (contour de la case)</li> </ul>
Barre de recherche	<p>Génère un élément représentant la progression de la lecture d'un fichier audio ou vidéo.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Progression - saisissez le pourcentage de progression à représenter sur</li> </ul>

	l'élément
Clavier	<p>Génère un élément qui représente un clavier pour les applications Android.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Type - bascule l'image entre un clavier texte (QWERTY) et un clavier numérique</li> </ul>
Fileur	<p>Génère un élément représentant la version Android d'une zone de liste déroulante.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Développé - cliquez sur la flèche déroulante et sélectionnez Vrai pour représenter la zone de liste déroulante utilisée, affichant toutes les valeurs, ou Faux pour représenter la zone de liste déroulante avec une seule valeur sélectionnée</li> <li>Valeurs - cliquez sur la flèche déroulante et sélectionnez la valeur à représenter comme actuellement sélectionnée dans la zone de liste déroulante Vous pouvez modifier le texte des valeurs et en ajouter d'autres à la liste en modifiant la liste « Valeurs : » dans notes de la Valeur Étiquetée</li> </ul>

## Champs de texte

Item	Description
Texte brut	<p>Génère un élément de texte sans bordure, le texte étant le nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Aligner le texte - cliquez sur la flèche déroulante et choisissez d'aligner le texte à gauche, au centre ou à droite du cadre de l'élément</li> <li>Multiligne - cliquez sur la flèche déroulante et sélectionnez Vrai pour permettre au texte de s'enrouler sur plusieurs lignes (augmentant automatiquement la profondeur de l'élément), ou Faux pour afficher uniquement le texte qui tient sur une ligne dans la largeur de l'élément actuel</li> </ul>
Texte multiligne	<p>Génère un élément de texte sans bordure, mais qui peut contenir du texte sur plusieurs lignes avec un formatage HTML de base.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Aligner le texte - cliquez sur la flèche déroulante et choisissez d'aligner le texte à gauche, au centre ou à droite du cadre de l'élément</li> <li>Texte - cliquez sur le bouton  pour éditer l'écran Note Valeur Étiquetée, sur lequel vous pouvez créer le texte à représenter sur le diagramme ; cela peut utiliser étiquettes de formatage HTML telles que &lt;b&gt; &lt;/b&gt; pour gras ou &lt;u&gt; &lt;/u&gt; pour souligné - tous les formats HTML ne sont pas pris en charge</li> </ul>

## Image\_Média

Item	Description

Image	Génère un espace réservé pour indiquer où une image sera placée sur le téléphone ou la tablette. Vous pouvez afficher une image réelle en attribuant une image alternative à l'élément.
Lecteur vidéo	Génère un élément qui représente un contrôle de lecteur vidéo sur le téléphone ou la tablette.
Lecteur audio	Génère un élément qui représente un contrôle de lecteur audio sur le téléphone ou la tablette.

## Heure\_date

Item	Description
Calendrier	<p>Génère un élément représentant un calendrier (l'image par défaut), affichant le mois, le jour et l'année en cours en fonction de la date du système.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Format de date - cliquez sur la flèche déroulante et sélectionnez le format de date à afficher si les spinners sont affichés : <ul style="list-style-type: none"> <li>- j/m/a</li> <li>- l/j/a</li> <li>- a/m/j (impossible si le calendrier est également affiché)</li> </ul> </li> <li>ShowCalendar - la valeur par défaut est True pour afficher le calendrier ; si les spinners sont également affichés, le calendrier forcera l'affichage du spinner à afficher l'année à l'extrémité droite et remplacera le spinner de l'année. Si défini sur False, les spinners du jour, du mois et de l'année s'affichent quelle que soit la valeur « ShowSpinners »</li> <li>Les Spinners de Show <ul style="list-style-type: none"> <li>- Si défini sur True AND « ShowCalendar » est True, affiche le spinner contrôles pour le jour et le mois (le calendrier agit comme une année marqueur)</li> <li>- Si défini sur False AND « ShowCalendar » est True, aucun spinner n'est montré avec le calendrier</li> <li>- Si « ShowCalendar » est défini sur False, le jour, le mois et l'année les spinners sont automatiquement affichés dans le format défini dans 'DateFormat', affichant la date du système</li> </ul> </li> </ul> <p>Vous ne pouvez pas redimensionner cet élément.</p>
Sélecteur de date	<p>Génère un élément qui représente un ensemble de spinners affichant la date du jour, dérivée de la date du système.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Date - cliquez sur la flèche déroulante et sélectionnez une date différente dans le calendrier ; si la date actuelle n'est pas celle d'aujourd'hui, vous pouvez la réinitialiser à la date d'aujourd'hui en cliquant sur le bouton Aujourd'hui ou en effaçant simplement le champ valeur « Date »</li> <li>DateFormat - comme pour l'élément Calendrier</li> <li>ShowCalendar - la valeur par défaut est False pour masquer le calendrier ; sinon, comme pour l'élément Calendar</li> <li>ShowSpinners - comme pour l'élément Calendrier</li> </ul>

	Vous ne pouvez pas redimensionner cet élément.
Sélecteur de temps	<p>Génère un élément qui représente une paire de spinners indiquant l'heure actuelle, en heures et minutes, dérivée de l'horloge système.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Vue 24 heures - cliquez sur la flèche déroulante et définissez sur Vrai pour afficher l'heure au format 24 heures, ou sur Faux pour afficher l'heure au format 12 heures avec AM ou PM à droite de l'heure</li> <li>• Heure - remplacez les heures, les minutes et le paramètre AM/PM pour afficher une heure autre que l'heure système (vous ne pouvez régler l'heure qu'au format 12 heures) ; pour revenir à l'heure système, remplacez le champ par « 12:00 AM »</li> </ul> <p>Vous ne pouvez pas redimensionner cet élément.</p>
Horloge	<p>Génère un élément qui représente un cadran d'horloge analogique avec des aiguilles pour les heures et les minutes et sans chiffres, affichant l'heure système. Vous pouvez modifier le rendu en un affichage numérique.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Vue 24 heures sur 24 - si vous définissez le « Type » sur « Numérique », cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher l'heure au format 24 heures ou Faux pour afficher l'heure au format 12 heures avec AM ou PM à droite, selon le cas</li> <li>• Heure - remplacez les heures, les minutes et le paramètre AM/PM pour afficher une heure autre que l'heure système (vous ne pouvez régler l'heure qu'au format 12 heures) ; pour revenir à l'heure système, remplacez le champ par 12:00 AM</li> <li>• Type - cliquez sur la flèche déroulante et sélectionnez « Numérique » pour afficher l'heure sous forme d'affichage numérique ou « Analogique » pour afficher l'heure sous forme de cadran d'horloge</li> </ul> <p>Vous pouvez redimensionner l'élément au format « Analogique », mais pas au format « Numérique ».</p>

## Motifs Android


Item	Description
Téléphone Android / Tablette Android	Ces Motifs créent des exemples de modèles des deux configurations de produits Android. Vous pouvez les utiliser comme exemples de conception des éléments, comme composants de base d'un modèle plus vaste ou comme point de départ pour développer un modèle plus détaillé d'un ou plusieurs produits.

## Boîte à outils de wireframe pour iPhone/tablette Apple

Les pages de la boîte à outils Diagramme « Apple Wireframing » fournissent les gabarits pour modélisation l'apparence physique d'un iPhone ou d'une tablette Apple à un état donné d'exécution d'une application. Elles fournissent également un certain nombre de Motifs pour générer des structures de modèles pour différentes versions de l'iPhone ou de l'iPad.



### Accéder

Sur la Boîte à outils Diagramme, cliquez sur  pour afficher la dialogue 'Trouvez Item de Boîte à Outils' et spécifiez 'Wireframing' ou 'Apple'.

Ruban	Conception > Diagramme > Boîte à outils
-------	---



Raccourcis Clavier	Ctrl+Maj+3
--------------------	------------

## Types d'écran Apple

Item	Description
iPad Air, iPad Mini, iPhone 4s, iPhone 5c, iPhone 5s, iPhone 6, iPhone 6 Plus	<p>Ces icônes génèrent chacune un cadre pour le type d'appareil que vous avez sélectionné. Une prompt s'affiche pour vous permettre de spécifier l'orientation portrait ou paysage. (L'illustration principale montre un cadre paysage pour iPhone 5s et un cadre portrait pour iPhone 4s.)</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• ShowStatusBar - cliquez sur le bouton déroulant et sélectionnez « Afficher » ou « Masquer » l'image de la barre d'état sur l'élément</li> </ul>

## Contrôles

Item	Description
Case à cocher	<p>Génère un élément de case à cocher étiqueté, le texte de l'étiquette étant le nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• State - cliquez sur la flèche déroulante et sélectionnez l'état à représenter - « Coché » (icône de coche) ou « Non coché » (contour de la case)</li> </ul>
Bouton radio	<p>Génère un élément de bouton radio étiqueté, le texte de l'étiquette étant le nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• State - cliquez sur la flèche déroulante et sélectionnez l'état à représenter - « Sélectionné » (rempli) ou « Non sélectionné » (contour)</li> </ul>
Zone de liste déroulante	<p>Génère un élément représentant une zone de liste déroulante.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• DropDownState - cliquez sur la flèche déroulante et sélectionnez « Ouvrir » pour représenter la zone de liste déroulante utilisée, affichant toutes les valeurs, ou « Fermé » pour représenter la zone de liste déroulante avec une seule valeur sélectionnée</li> <li>• Items - cliquez sur la flèche déroulante et sélectionnez l'élément à représenter comme étant actuellement sélectionné dans la zone de liste déroulante et, si la liste des éléments est développée, mis en surbrillance dans la liste Vous pouvez modifier le texte des éléments et ajouter ou supprimer des éléments dans la liste en modifiant la liste « Valeurs : » dans les notes Valeur Étiquetée</li> </ul>
Étiquette	<p>Génère un élément de texte d'étiquette. Le nom de l'élément correspond au texte de l'étiquette.</p>

	<p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Aligner le texte - définit l'alignement du texte à gauche, au centre ou à droite</li> <li>Multiligne - définit l'étiquette pour afficher le texte sur plusieurs lignes</li> </ul>
Liste	<p>Génère un élément de zone de liste.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Items - cliquez sur la flèche déroulante et sélectionnez l'élément à afficher en surbrillance dans la liste Vous pouvez ajouter, supprimer et renommer des éléments en modifiant la liste « Valeurs : » dans les Notes Valeur Étiquetés .</li> <li>Type de liste - cliquez sur la flèche déroulante et sélectionnez l'affichage de la liste sous la forme « Simple », « Numérotée » ou « À puces »</li> </ul>
Tableau	<p>Génère un élément Tableau avec des colonnes, des lignes et des cellules étiquetées.</p> <p>Double-cliquez sur le tableau pour faire apparaître la dialogue « Propriétés » de l'élément sur la page « Wireframe », qui fournit les facilités d'édition du tableau (ajout, renommage et suppression de colonnes et de lignes, modification de la largeur des colonnes et édition du texte des cellules). Vous pouvez soit éditer le texte en cliquant dessus, soit en cliquant avec le bouton droit de la souris et en sélectionnant une option. Note que l'éditeur ne fournit pas une image réelle de l'apparence du tableau à l'écran.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Dessiner des lignes - masque (ou affiche) les lignes sur toutes les cellules sous les en-têtes de colonne afin que le tableau ressemble à un élément de liste à la place</li> <li>Mettre en surbrillance les en-têtes - met en surbrillance l'en-tête de chaque colonne afin qu'il soit facilement distinguable</li> <li>Propriétés - affiche le HTML du tableau</li> </ul>
Image	<p>Génère un espace réservé pour indiquer où une image sera placée dans le dialogue .</p> <p>Vous pouvez afficher une image réelle en attribuant une image alternative à l'élément.</p>





## Commandes Apple

Item	Description
Barre d'adresse	<p>Génère un élément de barre d'adresse URL.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Adresse - définit le texte affiché dans la barre d'adresse</li> </ul>
Icône de l'application	<p>Génère un élément App Icône .</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Image - spécifie le nom d'une image conservée dans le gestionnaire d'images, qui s'affiche sous la forme de l'icône de l'application (tapez simplement le nom de l'image tel qu'il est répertorié dans la colonne « Nom » du gestionnaire d'images)</li> <li>Nombre de notifications - indique le nombre de notifications en attente pour</li> </ul>

	cette application ; le nombre est affiché dans le cercle dans le coin supérieur droit de l'élément
Bouton	<p>Génère un élément Button étiqueté. Le texte de l'étiquette correspond au nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Style du bouton - définit la forme de l'élément : <ul style="list-style-type: none"> <li>- « Normal » dessine un rectangle aux bords arrondis</li> <li>- « Précédent » dessine une forme de fanion pointant vers la gauche</li> <li>- « Suivant » dessine une forme de fanion pointant vers la droite</li> </ul> </li> <li>• Alignement du texte - définit l'alignement du texte du bouton dans l'élément ; aligné à gauche, centré ou aligné à droite</li> </ul>
Sélecteur de date/heure	<p>Génère un élément d'affichage de date et d'heure.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Date - définit la date à laquelle l'élément s'affiche ; si aucune date n'est spécifiée, ce sera la date actuelle du système</li> <li>• Étendu - si True, dessine l'élément comme une section de calendrier avec la date et/ou l'heure actuelle sélectionnée (comme défini par la Valeur Étiquetée ' Type ') ; si False, dessine l'élément comme une simple string de texte indiquant la date et/ou l'heure</li> <li>• Heure - définit l'heure affichée sur l'élément ; remplacez les segments d'heure, de minute et AM/PM si nécessaire Si aucune heure n'est spécifiée, l'heure actuelle du système s'affichera (l'heure non spécifiée s'affiche sous la forme 12h00).</li> <li>• Type - dessine l'élément affichant la date, l'heure ou les deux</li> </ul>
Liste des groupes	<p>Génère un élément de liste groupée avec deux niveaux d'entrée.</p> <p>Double-cliquez sur l'élément pour ouvrir la dialogue « Propriétés » de la page « Wireframe ». Utilisez cette page pour composer les entrées de la liste.</p> <p>Cliquez sur le nœud racine (le nom de l'élément).</p> <ul style="list-style-type: none"> <li>• Afficher les groupes - cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher les éléments de la liste de premier niveau ou Faux pour les masquer ; les éléments de la liste de deuxième niveau s'affichent dans les deux cas mais doivent appartenir à un élément de groupe, qu'il soit affiché ou non</li> </ul> <p>Cliquez sur le nœud Groupe (l'élément de liste de premier niveau) directement sous le nœud racine.</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le texte de l'élément de liste de premier niveau (le nom du groupe d'éléments)</li> </ul> <p>Cliquez sur un nœud d'élément (élément de liste de deuxième niveau) directement sous un nœud de groupe. Les nœuds Item n'ont pas de nœud enfant.</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le nom de l'élément de la liste</li> <li>• Texte - saisissez tout texte supplémentaire à afficher (par défaut) sous le nom de l'élément</li> <li>• Texte près du lien - cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher le texte supplémentaire en face du nom de l'élément, où vous pouvez également ajouter un lien, ou Faux pour conserver le texte sous le nom de l'élément</li> <li>• Image - sélectionnez un nom d'image dans la liste déroulante ou saisissez simplement le nom tel qu'il apparaît dans la colonne « Nom » du gestionnaire d'images, pour afficher l'image à gauche du nom de l'élément</li> <li>• Est un lien - cliquez sur la flèche déroulante et sélectionnez Vrai pour indiquer</li> </ul>

	<p>que l'élément est lié à une autre page ou à des informations supplémentaires en affichant un caractère « &gt; » à droite du nom de l'élément ; sélectionnez Faux pour masquer le caractère de lien</p> <ul style="list-style-type: none"> <li>• Lien vers l'image - sélectionnez un nom d'image dans la liste déroulante ou saisissez simplement le nom tel qu'il apparaît dans la colonne « Nom » du gestionnaire d'images, pour afficher l'image (si « Est un lien » est vrai) comme icône de lien au lieu du caractère &gt;</li> <li>• Sélectionné - cliquez sur la flèche déroulante et sélectionnez Vrai pour mettre en surbrillance le nom de l'élément comme sélectionné, ou Faux pour ne pas mettre l'élément en surbrillance</li> </ul>
Clavier	<p>Génère un élément qui représente un clavier.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Type - bascule l'image entre un clavier texte (QWERTY) et un clavier numérique</li> </ul>
Icône de chargement	<p>Génère un élément qui représente l'icône de chargement d'Apple.</p>
Contrôle des pages	<p>Génère un élément rendu sous la forme d'une ligne de cercles gris, indiquant le nombre de pages disponibles et laquelle d'entre elles représente la page actuellement affichée.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Page actuelle - dessine un cercle blanc dans la séquence de cercles gris pour indiquer lequel représente la page actuellement affichée</li> <li>• Pages - le nombre de cercles à dessiner, indiquant le nombre de pages parcourues par ce contrôle (redimensionnez l'élément manuellement pour afficher le nombre de cercles spécifié)</li> </ul>
Barre de recherche	<p>Génère un élément représentant un champ de recherche.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• DrawStyle - bascule l'apparence de l'élément entre la recherche d'application par défaut avec une longue-vue de requête et une recherche Web sur un appareil Apple</li> <li>• Espace réservé - saisissez le texte qui sera affiché dans le contrôle, pour prompt l'utilisateur à saisir le terme de recherche</li> <li>• Invite - saisissez le texte prompt à afficher au-dessus du champ de requête, comme un rappel de ce qu'il faut rechercher ; par exemple, si la recherche consiste à rechercher des films/vidéos enregistrées sur l'appareil, vous pouvez saisir « Rechercher des vidéos » (« DrawStyle » doit être défini par défaut)</li> <li>• Portée - cliquez sur la flèche déroulante et sélectionnez l'emplacement de recherche mis en surbrillance dans la barre de portée (« DrawStyle » doit être défini sur la valeur par défaut et « Afficher la barre de portée » doit être défini sur Vrai)</li> <li>• Afficher les signets - lorsqu'il est défini sur True, un petit symbole de livre sera dessiné sur le côté droit du champ de requête, pour indiquer que ce contrôle de recherche stockera les recherches précédentes et pourra les utiliser à nouveau ('DrawStyle' doit être défini sur la valeur par défaut)</li> <li>• Afficher le bouton Annuler - affiche un bouton Annuler à droite du champ de requête ('DrawStyle' doit être défini sur la valeur par défaut)</li> <li>• Afficher la barre de portée - affiche une ligne d'options de portée pour ce contrôle de recherche, sous le champ de requête ('DrawStyle' doit être défini sur la valeur par défaut)</li> <li>• Afficher les résultats de la recherche - affiche une flèche déroulante sur le côté</li> </ul>

	<p>droit de la zone de recherche, pour indiquer que la requête affichera les résultats de la recherche (remplace l'icône Signet si les deux sont configurées pour s'afficher) (« DrawStyle » doit être défini sur la valeur par défaut)</p> <ul style="list-style-type: none"> <li>Alignement du texte - aligne le texte de l'espace réservé à gauche, à droite ou au centre du champ de requête</li> </ul>
Contrôle des segments	<p>Génère un élément représentant un ensemble d'onglets (par défaut trois onglets). Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Onglets - cliquez sur la flèche déroulante et sélectionnez le numéro de l'onglet à mettre en surbrillance pour indiquer l'onglet actuel ; ouvrez et modifiez les Notes Valeur Étiquetée pour ajouter, supprimer ou renommer des onglets</li> </ul>
Contrôle du spinner	<p>Génère un élément représentant un contrôle spinner avec une liste d'éléments parmi lesquels peuvent être sélectionnés. Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Vérifier Item sélectionné - lorsqu'il est défini sur True, une coche sera dessinée sur le côté gauche de l'élément défini comme sélectionné</li> <li>Items - cliquez sur la flèche déroulante et sélectionnez l'article à indiquer comme sélectionné ; vous pouvez ouvrir et modifier les Notes Valeur Étiquetée pour ajouter, renommer ou supprimer des articles de la liste</li> <li>Bords arrondis - cliquez sur la flèche déroulante et sélectionnez le(s) côté(s) qui présentent des coins arrondis ; vous pouvez utiliser cette propriété pour ancrer plusieurs contrôles de rotation les uns à côté des autres afin de créer des sélections de contrôles de rotation plus complexes, comme une page affichant une sélection pour le pays, State, la ville et la banlieue - définissez : <ul style="list-style-type: none"> <li>Les deux pour définir des coins arrondis des deux côtés, si le spinner le contrôle est tout seul</li> <li>À gauche pour définir des coins arrondis sur le bord gauche et des coins pointus sur le bord droit, si ce spinner est le premier d'une rangée de quais commandes du spinner</li> <li>Droit de définir des coins arrondis sur le bord droit et des coins pointus sur le bord gauche, si ce spinner est le dernier d'une rangée de quais commandes du spinner, ou</li> <li>Aucun pour définir des angles vifs des deux côtés, si ce spinner est amarré entre deux autres</li> </ul> </li> <li>Alignement du texte - cliquez sur la flèche déroulante et sélectionnez où aligner le texte de l'élément - à gauche, à droite ou au centre du contrôle</li> </ul>
Moteur pas à pas	<p>Génère un élément représentant un contrôle de décrémentation/incrémentation (moins/plus).</p>
Changer	<p>Génère un élément représentant un interrupteur marche/arrêt coulissant. Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>DisplayText - bascule entre l'affichage et le masquage des deux valeurs de texte d'état</li> <li>State - bascule entre l'état « activé » (arrière-plan gris avec un cercle blanc à droite et - si DisplayText est True - le texte « activé ») et l'état « désactivé » (arrière-plan blanc avec un cercle blanc à gauche et - si DisplayText est True - le texte « désactivé »)</li> </ul> <p>Les valeurs du texte d'état par défaut sont Activé et Désactivé ; vous pouvez modifier les Notes sur la Valeur Étiquetée pour les remplacer par n'importe quelle autre paire de valeurs (vous pouvez ajouter d'autres valeurs, mais seules les deux premières sont appliquées)</p>

Champ de texte	<p>Génère un champ de texte dans lequel l'utilisateur final peut saisir du texte libre, tel que le nom d'une page de connexion. Le champ contient le texte « TextField » et un cercle barré.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Style de Bordure - cliquez sur la flèche déroulante et sélectionnez le style de bordure pour l'élément : <ul style="list-style-type: none"> <li>- Rectangle arrondi - un rectangle aux coins arrondis</li> <li>- Lunette - un rectangle avec des bords supérieurs et gauches biseautés</li> <li>- Ligne - un rectangle avec des angles vifs et des bords à une seule ligne</li> <li>- Aucun - un rectangle sans bordures</li> </ul> </li> <li>• Alignement du texte - cliquez sur la flèche déroulante et définissez le texte pour qu'il s'aligne à gauche, à droite ou au centre du contrôle</li> </ul>
Titre	<p>Génère un élément qui représente le titre d'une page, le nom de l'élément étant le texte du titre (par exemple, « Paramètres »). L'élément est un rectangle avec un arrière-plan gris et un texte blanc.</p>
Barre d'outils	<p>Génère une image qui représente une barre d'outils avec des icônes par défaut, que vous pouvez ajouter ou remplacer par des images du gestionnaire d'images.</p> <p>Double-cliquez sur l'élément pour afficher la dialogue « Propriétés » sur la page « Wireframe », qui répertorie les icônes affichées dans le cadre de l'élément Barre d'outils.</p> <ul style="list-style-type: none"> <li>• Pour ajouter une nouvelle icône, cliquez sur « Barre d'outils » et sur le bouton Ajouter, et en réponse à l' prompt saisissez le nom de l'icône, ce qui ajoute le nom au bas de la liste ; cliquez sur le nom et dans le champ Propriété « Image », saisissez le nom du fichier d'icône, tel qu'il est affiché dans le panneau « Image chargée » du gestionnaire d'images Vous pouvez ajouter un « Séparateur » (affiché sous la forme  ) à la liste, pour séparer différents groupes d'icônes sur la barre d'outils ; saisissez simplement le nom « Séparateur » dans l' prompt de nom Pour positionner l'icône sur la barre d'outils, cliquez dessus dans la liste et cliquez sur les boutons  </li> <li>• Pour supprimer une icône de la liste, cliquez dessus et cliquez sur le bouton Supprimer ; l'icône est immédiatement supprimée de la liste et, lorsque vous fermez le dialogue , de la barre d'outils</li> </ul>
Barre d'onglets	<p>Génère un élément qui représente une rangée d'onglets représentés par des images.</p> <p>Double-cliquez sur l'élément pour afficher la dialogue « Propriétés » sur la page « Wireframe », qui montre le nœud racine (l'élément TabBar lui-même) et les éléments contenus dans la TabBar (sous forme d'images).</p> <ul style="list-style-type: none"> <li>• Pour ajouter une nouvelle icône, cliquez sur « TabBar » et sur le bouton Ajouter, et en réponse à l' prompt saisissez le nom de l'élément, ce qui ajoute le nom au bas de la liste Pour positionner l'élément sur la TabBar, cliquez dessus dans la liste et cliquez sur les boutons  </li> <li>• Pour supprimer un élément de la liste, cliquez dessus et cliquez sur le bouton Supprimer ; l'élément est immédiatement supprimé de la liste et, lorsque vous fermez le dialogue , de la TabBar</li> </ul> <p>Propriétés du nœud racine :</p> <ul style="list-style-type: none"> <li>• Afficher le texte - affiche (Vrai) ou masque (Faux) les noms des éléments sous forme de texte sous les images des onglets</li> <li>• Couleur d'arrière-plan - cliquez sur la flèche déroulante et sélectionnez la couleur d'arrière-plan de la TabBar</li> </ul>

	<ul style="list-style-type: none"> <li>• Couleur Police sélectionnée - cliquez sur la flèche déroulante et sélectionnez la couleur de police du texte si l'élément est sélectionné</li> <li>• Couleur Police non sélectionnée - cliquez sur la flèche déroulante et sélectionnez la couleur de police du texte si l'élément n'est pas sélectionné</li> </ul> <p>Propriétés du nœud Item :</p> <ul style="list-style-type: none"> <li>• Nom - le nom de l'élément, qui peut être affiché sous l'image dans la barre d'onglets (voir <i>Afficher le texte</i>)</li> <li>• Image - le nom du fichier image à afficher si l'élément n'est pas sélectionné ; le nom du fichier est celui indiqué dans le gestionnaire d'images</li> <li>• Image sélectionnée - le nom du fichier image à afficher si l'élément est sélectionné ; le nom du fichier est celui indiqué dans le gestionnaire d'images</li> </ul> <p>Sélectionnez la page « Général » de la dialogue « Propriétés » de cet élément de la barre d'onglets et cliquez sur l'onglet 'Tags' .</p> <p>Élément Valeur Étiquetée :</p> <ul style="list-style-type: none"> <li>• Onglet sélectionné - Cliquez sur la flèche déroulante et sélectionnez le nom de l'élément à représenter comme sélectionné</li> </ul>
--	--

## Motifs de pommes

Vous pouvez utiliser n'importe lequel de ces Motifs comme exemples de la façon dont les éléments sont utilisés, comme composants de base d'un modèle plus grand ou comme point de départ pour développer un modèle plus détaillé d'un ou plusieurs produits.


Item	Description
Apple iPad Air	Crée un exemple de modèle pour l'Apple iPad Air.
Apple iPad Mini	Crée un exemple de modèle pour l'Apple iPad Mini.
Apple iPhone 4s	Crée un exemple de modèle pour l'Apple iPhone 4s.
Apple iPhone 5c	Crée un exemple de modèle pour l'Apple iPhone 5c.
Apple iPhone 5s	Crée un exemple de modèle pour l'Apple iPhone 5s.
Apple iPhone 6	Crée un exemple de modèle pour l'Apple iPhone 6.
Apple iPhone 6 Plus	Crée un exemple de modèle pour l'Apple iPhone 6 Plus.

# Boîte à outils de wireframe Windows Phone

Les pages de la boîte à outils Diagramme « Windows Phone » fournissent les gabarits pour modélisation de l'apparence physique d'un téléphone Windows 8 .1 à un état donné d'exécution d'une application.



## Accéder

Sur la Boîte à outils Diagramme , cliquez sur  pour afficher la dialogue ' Trouvez Item de Boîte à Outils ' et spécifiez ' Wireframing ' ou ' Windows Phone '.


Ruban	Conception > Diagramme > Boîte à outils
Raccourcis Clavier	Ctrl+Maj+3



## Types d'écran Windows


Item	Description
Téléphone Windows	<p>Génère un cadre avec une zone d'écran pour l'appareil Windows 8 Phone. Une prompt s'affiche dans laquelle vous pouvez spécifier l'orientation portrait ou paysage.</p> <p>Les éléments créés dans la zone de l'écran ne peuvent pas être redimensionnés ou déplacés pour se trouver en dehors des bordures de l'écran. Les éléments créés en dehors du cadre peuvent être déplacés vers et hors du cadre et peuvent être aussi grands que le permettent la vue ou les propriétés de l'élément.</p>

## Texte

Item	Description
Bloc de texte	<p>Génère un élément qui représente le texte dominant tel que les titres et les étiquettes. Le nom de l'élément correspond au texte affiché.</p> <p>Double-cliquez sur l'élément pour ouvrir la dialogue « Propriétés » sur la page « Wireframe ». Elle affiche un ensemble par défaut de six niveaux de styles de titres. Cliquez sur le nom d'un style pour remplir le panneau « Propriétés » à droite de la dialogue , puis cliquez sur la flèche vers le bas pour chaque propriété et sélectionnez les paramètres appropriés pour le style. Si vous préférez, vous pouvez également modifier le nom du style dans la propriété « Nom ».</p> <p>Si nécessaire, vous pouvez ajouter d'autres styles à la liste. Cliquez sur le nom <b>du groupe</b> de styles et sur le bouton Ajouter. Dans l' prompt « Entrer un nom pour l'élément », saisissez un nom pour le style et cliquez sur le bouton OK . Le nouveau style est ajouté à la fin de la liste ; si vous souhaitez le déplacer plus haut dans la liste, cliquez dessus et sur l'icône  . Là encore, vous définissez le style à l'aide du panneau « Propriétés ».</p> <p>Si vous souhaitez supprimer des styles de la liste, cliquez sur le nom du style et sur le bouton Supprimer.</p> <p>Lorsque vous avez défini les styles qui peuvent être utilisés pour ce texte, cliquez sur la page 'Général' de la dialogue ' Propriétés ' et, dans la Valeur Étiquetée ' Type d'en-tête', cliquez sur la flèche déroulante et sélectionnez le style à appliquer au texte de cette zone de texte spécifique.</p>
Zone de texte	<p>Génère un champ de texte simple avec une bordure, dans lequel vous pouvez saisir le texte de votre choix. Le nom de l'élément est le texte affiché et n'est pas renvoyé à la ligne.</p>

## Contrôles



Item	Description
------	-------------

Bouton	<p>Génère une icône rectangulaire représentant un bouton d'écran, avec le nom de l'élément comme texte du bouton.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• State : cliquez sur la flèche déroulante et sélectionnez le bouton État pour représenter : <ul style="list-style-type: none"> <li>- Normal - le bouton non sélectionné</li> <li>- Focalisé - le bouton lorsque le curseur passe dessus</li> <li>- Sélectionné - le bouton lorsqu'il est cliqué</li> <li>- Désactivé - le bouton est grisé lorsqu'il n'est pas disponible</li> </ul> </li> </ul>
Case à cocher	<p>Génère un élément représentant une case à cocher étiquetée, le nom de l'élément étant l'étiquette.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Activé : cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher la case à cocher activée pour la sélection, ou Faux pour afficher la case à cocher désactivée et indisponible</li> <li>• State : cliquez sur la flèche déroulante et sélectionnez « Décoché » pour afficher la case à cocher vide et non sélectionnée, ou « Coché » pour afficher la case à cocher sélectionnée avec une coche à l'intérieur</li> </ul>
Bouton de lien hypertexte	<p>Génère un élément de texte avec le nom de l'élément comme texte souligné affiché, représentant un lien hypertexte sur l'écran.</p> <p>Double-cliquez sur l'élément pour ouvrir la dialogue « Propriétés » sur la page « Wireframe », qui répertorie les trois états d'hyperlien normaux « Lien », « Visité » et « Survol ». Cliquez sur le nom d'un état pour remplir le panneau « Propriétés » à droite de la dialogue , puis cliquez sur la flèche vers le bas pour chaque propriété et sélectionnez les paramètres appropriés pour le style à appliquer à cet état. Si vous préférez, vous pouvez également modifier le nom de l'état dans la propriété « Nom ».</p> <p>Si nécessaire, vous pouvez ajouter d'autres états à la liste. Cliquez sur le nom <b>du groupe</b> d'états et sur le bouton Ajouter. Dans l' prompt « Entrer un nom pour l'élément », saisissez un nom pour l'état et cliquez sur le bouton OK . Le nouvel état est ajouté à la fin de la liste ; si vous souhaitez le déplacer plus haut dans la liste, cliquez dessus et sur l'icône  . Là encore, vous définissez le style à l'aide du panneau « Propriétés ».</p> <p>Si vous souhaitez supprimer des États de la liste, cliquez sur le nom de l'État et sur le bouton Supprimer.</p> <p>Lorsque vous avez défini les états que peut avoir l'hyperlien, cliquez sur la page 'Général' de la dialogue ' Propriétés ' et, dans la zone ' State ' Valeur Étiquetée , cliquez sur la flèche déroulante et sélectionnez l'état dans lequel cet hyperlien doit être représenté.</p>
Image	<p>Génère un objet rectangulaire contenant un « X » pour indiquer l'emplacement d'une image sur l'écran. Aucune propriété n'est à définir.</p>
Bouton radio	<p>Génère un élément représentant un bouton radio étiqueté, le nom de l'élément étant l'étiquette.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Activé : cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher le bouton radio activé pour la sélection, ou Faux pour afficher le bouton radio désactivé et indisponible</li> <li>• State : cliquez sur la flèche déroulante et sélectionnez « Non sélectionné » pour afficher le bouton radio vide, ou « Sélectionné » pour afficher le bouton radio</li> </ul>

	avec un cercle rempli à l'intérieur
--	-------------------------------------


## Carrelage

Les éléments de mosaïque ajoutent à l'écran du téléphone un panneau qui, selon le type, affiche une image et/ou du texte. Le panneau ne peut pas être redimensionné et s'il affiche du texte, celui-ci occupe uniquement la moitié supérieure de l'élément. La quantité de texte affichée est influencée par le type de mosaïque. Vous devrez donc expérimenter avec le type requis pour voir la quantité de texte significatif que vous pouvez afficher.

Item	Description
Collection de tuiles	<p>Ajoute une tuile avec un motif aléatoire, pour représenter une tuile de collection Windows .</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• En-tête : saisissez une string de texte appropriée comme titre de la mosaïque ; le texte s'affiche lorsque « Afficher en arrière » est défini sur Vrai et si le « Type de mosaïque » le supporte</li> <li>• Afficher l'arrière : cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher l'arrière de la tuile au lieu de l'avant ; pour certains types de tuiles, l'arrière ne s'affiche pas quel que soit ce paramètre</li> <li>• Texte : une Valeur Étiquetée dans laquelle vous saisissez le texte à afficher au dos de la tuile ; le format et la police du texte affiché dépendent du 'Tile Type'</li> <li>• Type de tuile : cliquez sur la flèche déroulante et sélectionnez le type de tuile de collection ; cela n'affectera que l'affichage du verso de la tuile, le recto restera toujours le même (consultez la page <i>Web Descriptions Type Gabarit tuiles Windows</i> pour plus d'informations sur les types de tuiles)</li> </ul>
Mosaïque d'image	<p>Ajoute une tuile qui s'affiche initialement sous la forme d'une boîte avec un « X » au centre, mais qui est destinée à afficher une image que vous sélectionnez.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Image : cliquez sur l'icône  et sélectionnez l'image à afficher pour cette mosaïque, dans la dialogue « Gestionnaire d'images »</li> <li>• Texte : saisissez le texte qui sera affiché en blanc en bas de l'image, en fonction du « Type de mosaïque »</li> <li>• Type de mosaïque : cliquez sur la flèche déroulante et sélectionnez le type de mosaïque d'image à afficher ; il s'agira soit d'une image uniquement, soit d'une image avec du texte (consultez la page <i>Web Descriptions Type de Gabarit Windows</i> pour plus d'informations sur les types de mosaïques)</li> </ul>
Tuile Peek	<p>Ajoute une tuile similaire à une tuile d'image, sauf qu'elle peut afficher l'arrière de la tuile pour afficher plus d'informations.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• En-tête : saisissez une string de texte appropriée comme titre de la mosaïque ; le texte s'affiche lorsque « Afficher en arrière » est défini sur Vrai et si le « Type de mosaïque » le supporte</li> <li>• Image : cliquez sur l'icône  et sélectionnez l'image à afficher au recto de cette tuile, à partir de la fenêtre Gestionnaire d'images</li> <li>• Afficher l'arrière : cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher l'arrière de la tuile au lieu de l'avant</li> </ul>

	<ul style="list-style-type: none"> <li>• Texte : une Valeur Étiquetée dans laquelle vous saisissez le texte à afficher au dos de la tuile ; le format et la police du texte affiché dépendent du 'Tile Type '</li> <li>• Type de tuile : cliquez sur la flèche déroulante et sélectionnez le type de tuile Peek à afficher (consultez la page Web <i>Gabarit Type de tuiles Windows</i> pour plus d'informations sur les types de tuiles)</li> </ul>
Mosaïque de texte	<p>Ajoute une mosaïque qui affiche uniquement du texte. Selon le type de mosaïque, vous pouvez afficher une string de texte dans la moitié supérieure du panneau et deux éléments de texte dans le coin inférieur droit du panneau.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Bloc de texte : saisissez une string de texte en deux parties à afficher en bas à droite de la tuile, comprenant une string plus longue qui sera affichée dans une petite police, suivie d'une string plus courte qui sera affichée dans une grande police, les deux chaînes étant séparées par un point-virgule ; la string courte n'affichera que deux caractères dans une tuile carrée, ou jusqu'à 5 caractères dans une tuile large, tandis que la string plus longue peut contenir beaucoup plus de caractères, par exemple : Messages ;16</li> <li>• Texte : saisissez du texte supplémentaire à afficher en haut de la tuile (en fonction du type de tuile), comme une description ou une définition de l' object identifié dans le texte inférieur</li> <li>• Type de tuile : cliquez sur la flèche déroulante et sélectionnez le type de tuile de texte à afficher (consultez la page Web <i>Descriptions Type Gabarit tuiles Windows</i> pour plus d'informations sur les types de tuiles)</li> </ul>

## Commandes Windows Phone

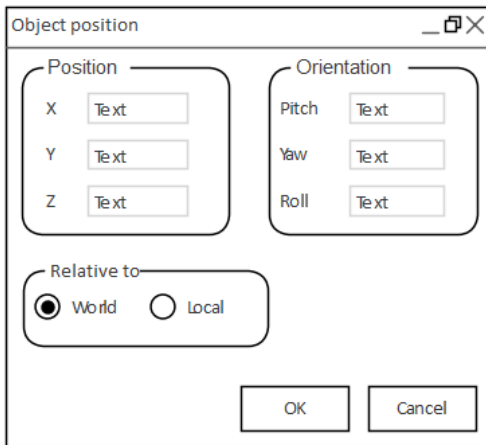
Item	Description
Barre d'applications	<p>Génère un élément qui représente la « barre d'applications Windows », qui s'affiche en bas de l'écran du téléphone pour fournir des commandes supplémentaires. Cela peut inclure jusqu'à un maximum de cinq icônes et six chaînes. Lorsque vous faites glisser l'icône sur le diagramme , vous êtes invité à sélectionner l'orientation « Portrait » ou « Paysage » pour correspondre à l'orientation de l'écran.</p> <p>Double-cliquez sur l'élément pour afficher la dialogue « Propriétés » sur la page « Wireframe », affichant le nom de l'élément en haut comme nœud racine.</p> <p>Cliquez sur le nom de l'élément et, dans le panneau « Propriétés » de droite, cliquez sur la flèche déroulante dans le champ valeur de la propriété « Mode » et sélectionnez :</p> <ul style="list-style-type: none"> <li>• « Mini » : pour représenter la barre d'applications sous la forme d'une fine barre avec uniquement l'icône de menu déroulant (  ) en haut à droite, sans autres icônes ni texte</li> <li>• « Par défaut » : pour représenter la barre d'applications sous la forme d'une barre fine, sauf si elle contient des éléments, auquel cas elle affichera simplement l'icône dans un cercle, sans texte ni nom d'icône</li> <li>• « Étendu » : pour afficher la barre d'applications contenant chaque icône dans un cercle, le nom de l'élément sous le cercle et jusqu'à six chaînes de texte représentant des options de menu supplémentaires</li> </ul> <p>Vous préférerez peut-être définir les propriétés de la barre d'applications après y avoir ajouté des icônes en tant que nœuds enfants. Pour ajouter un nœud enfant, cliquez sur le nœud racine, sur le bouton Ajouter et saisissez le nom de l'icône ou</p>

	<p>object . Dans le panneau de droite, définissez les propriétés du nœud enfant :</p> <ul style="list-style-type: none"> <li>• « Nom » : affiche le nom de l'élément, que vous pouvez modifier si nécessaire ; si la barre d'applications est rendue en mode « Étendu », le nom d'un élément de symbole ou de police sera affiché sous l'icône, tandis que pour un élément de texte, il sera affiché sous et à gauche des icônes dans une liste verticale</li> <li>• « Type » : cliquez sur la flèche déroulante et sélectionnez dans la liste des types d'éléments ; le type que vous sélectionnez déterminera les autres prompts de propriétés affichées : <ul style="list-style-type: none"> <li>- 'SymbolIcon' : affiche l'élément sous forme d'icône de symbole</li> <li>- 'FontIcon' : affiche l'élément sous forme de glyphe, en utilisant une police</li> <li>- 'BitmapIcon' : dessine une image sélectionnée comme icône</li> <li>- « Texte » : (s'applique uniquement en mode « Étendu ») affiche l'élément nom en tant que membre d'une liste verticale ci-dessous et à gauche des icônes ; un maximum de six éléments peuvent être répertoriés à la fois</li> <li>- « Séparateur » : trace une ligne verticale entre les icônes, ce qui compte comme l'un des cinq emplacements disponibles pour les icônes sur la barre d'applications ; Les éléments de séparation n'affichent pas les noms</li> </ul> </li> <li>• « Symbole » : (s'affiche si le « Type » est défini sur « SymbolIcon ») cliquez sur la flèche déroulante et sélectionnez le symbole dans la liste</li> <li>• « FontFamily » : (s'affiche si le « Type » est défini sur « FontIcon ») saisissez le nom de la police avec laquelle dessiner, par exemple « Segoe UI Symbol »</li> <li>• « Glyphe » : (s'affiche si le « Type » est défini sur « FontIcon ») saisissez la valeur hexadécimale du glyphe à dessiner. Par exemple, pour le symbole ©, vous pouvez définir « FontFamily » sur « Arial » et saisir le code hexadécimal « 00A9 » ; les codes de police dans Windows peuvent être trouvés via « Panneau de configuration   Polices   Rechercher un caractère »</li> <li>• « Bitmap » : (s'affiche si le « Type » est défini sur « BitmapIcon ») cliquez sur la flèche déroulante et sélectionnez un bitmap (comme indiqué dans le questionnaire d'images)</li> </ul>
Sélecteur de date	<p>Génère un élément qui représente trois blocs indiquant le jour et la date d'aujourd'hui, le mois et l'année, dérivés de la date du système.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Date - si nécessaire, cliquez sur la flèche déroulante et sélectionnez une date différente dans le calendrier ; si la date affichée n'est pas celle d'aujourd'hui, vous pouvez la réinitialiser à la date d'aujourd'hui en cliquant sur le bouton Aujourd'hui</li> <li>• Format de date - cliquez sur la flèche déroulante et sélectionnez le format de date à afficher : <ul style="list-style-type: none"> <li>- j/m/a</li> <li>- l/j/a</li> <li>- a/m/j</li> </ul> </li> </ul>
Boîte de mot de passe	<p>Génère un élément qui représente un champ de mot de passe à l'écran.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Mot de passe : une string de texte qui représente un mot de passe</li> <li>• Caractère du mot de passe : un caractère qui remplace chaque caractère de la string « Mot de passe » lorsque le mot de passe est masqué (lorsque « Révéler le bouton » ou « Afficher le texte » est défini sur Faux)</li> <li>• Bouton Révéler : s'il est défini sur True (valeur par défaut), dessine un bouton qui affiche la string de texte « Mot de passe » ; s'il est défini sur False, le bouton n'est pas affiché et la string de mot de passe est représentée par une string composée du « Caractère de mot de passe »</li> <li>• Afficher le texte : lorsque « Révéler le bouton » est défini sur Vrai, définir</li> </ul>





	« Afficher le texte » sur Vrai affichera la string de texte « Mot de passe » ; sinon, une string s'affiche composée uniquement du « Caractère du mot de passe »
Barre de progression	Génère un élément représentant une barre d'état « processus en cours », affichant un certain nombre d'étapes « points ».
Anneau de progrès	Génère un élément représentant le cercle de points « traitement en cours » Windows .
Barre de recherche	Génère un élément représentant un champ de recherche Windows , avec l'icône de démarrage de la recherche en forme de « loupe » à la fin. Valeur Étiquetés : <ul style="list-style-type: none"> <li>• Texte d'espace réservé - par défaut, le mot « Rechercher » ; si nécessaire, remplacez-le par une string de texte alternative</li> </ul>
Curseur	Génère un élément représentant un commutateur de commande à glissière, avec le curseur à 50 % de sa largeur. Valeur Étiquetés : <ul style="list-style-type: none"> <li>• Quantité de remplissage - remplacez le champ par une valeur entre 1 et 100, pour définir le pourcentage de l'icône affichée en noir derrière le curseur</li> </ul>
Sélecteur de temps	Génère un élément qui représente deux blocs indiquant l'heure en heures et en minutes, au format 12 heures ou 24 heures. Valeur Étiquetés : <ul style="list-style-type: none"> <li>• Affichage 24 heures - cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher l'heure au format 24 heures ou Faux (valeur par défaut) pour afficher l'heure au format 12 heures</li> <li>• Heure - affiche l'heure en trois sections - heures, minutes et AM/PM ; cliquez sur et remplacez chaque section par la valeur requise pour l'heure</li> </ul>
Interrupteur à bascule	Génère un élément représentant un interrupteur à bascule coulissant avec l'interrupteur à gauche, représentant l'état « arrêt ». Valeur Étiquetés : <ul style="list-style-type: none"> <li>• State - cliquez sur la flèche déroulante et sélectionnez « Activé » pour représenter l'état activé avec le commutateur à droite de l'icône, ou « Désactivé » pour déplacer le commutateur vers la gauche de l'icône afin de représenter l'état désactivé</li> </ul>

## Boîte à outils de wireframe Dialogue

Les pages de la boîte à outils Diagramme « Dialogue Wireframing » fournissent les gabarits pour modélisation de la conception physique, de l'apparence et du fonctionnement d'un dialogue à l'écran. Vous pouvez voir et développer un exemple de modélisation le dialogue en faisant glisser l'icône « Dialogue Motif » sur un diagramme .



### Accéder



Ruban	Design > Diagramme > Toolbox :  > Spécifiez 'Wireframing - Dialogue ' dans la boîte de dialogue ' Trouvez Item de Boîte à Outils '
Raccourcis Clavier	Ctrl+Shift+3 :  > Spécifiez 'Wireframing - Dialogue ' dans la dialogue ' Trouvez Item de Boîte à Outils '
Autre	Vous pouvez afficher ou masquer la boîte à outils Diagramme en cliquant sur les icônes  ou  à l'extrémité gauche de la barre de légende en haut de la Vue Diagramme .

### Types d'écran

Item	Description
Dialogue	<p>Génère un élément qui représente un plan dialogue , avec un titre (le nom de l'élément). Celui-ci comporte plusieurs composants que vous pouvez exposer et définir dans l'élément ' Propriétés ' dialogue , certains en utilisant les Valeur Étiquetés de l'onglet 'Wireframing' de la page 'Général', et d'autres en utilisant la page 'Wireframe' du le dialogue .</p> <p><b>Valeur Étiquetés :</b></p> <ul style="list-style-type: none"> <li>• Bouton Fermer - par défaut, la valeur est Vrai pour afficher une icône « Fermer dialogue » dans le coin supérieur droit du le dialogue ; cliquez sur la flèche déroulante et sélectionnez Faux pour omettre l'icône</li> <li>• Bouton Maximiser - par défaut, la valeur est False pour omettre l'icône</li> </ul>

	<p>« Maximiser dialogue » dans le coin supérieur droit de le dialogue ; cliquez sur la flèche déroulante et sélectionnez True pour afficher l'icône</p> <ul style="list-style-type: none"> <li>• Bouton Réduire - par défaut, la valeur est False pour omettre l'icône « Réduire la boîte dialogue » dans le coin supérieur droit de le dialogue ; cliquez sur la flèche déroulante et sélectionnez True pour afficher l'icône</li> <li>• ScrollbarH - par défaut sur ' &lt;none&gt; ' pour omettre une barre de défilement horizontale dans le dialogue ; cliquez sur la flèche déroulante et sélectionnez « Bas » ou « Haut » pour afficher une barre de défilement dans la position correspondante</li> <li>• ScrollbarV - par défaut sur ' &lt;none&gt; ' pour omettre une barre de défilement verticale dans le dialogue ; cliquez sur la flèche déroulante et sélectionnez « Gauche » ou « Droite » pour afficher une barre de défilement dans la position correspondante</li> <li>• StatusBar - définir l'affichage de la Barre de Statut à l'aide de la page 'Wireframe'</li> <li>• Icône utilisateur - saisissez le nom d'une icône exactement comme indiqué dans le gestionnaire d'images, pour afficher cette icône dans le coin supérieur gauche de le dialogue avant le nom le dialogue (élément)</li> </ul> <p><b>Page de maquette :</b></p> <p>S'affiche par défaut lorsque vous double-cliquez sur l'élément. Utilisez les options pour modifier la Barre de Statut en bas de le dialogue .</p> <p>Cliquez sur le nom de l'élément.</p> <ul style="list-style-type: none"> <li>• Barre de zoom - par défaut sur Vrai pour afficher une barre de zoom à l'extrémité droite de la Barre de Statut ; cliquez sur la flèche déroulante et sélectionnez Faux pour omettre la barre de zoom</li> <li>• Redimensionner la poignée - par défaut sur True pour afficher une icône de redimensionnement (triangle de points) dans le coin inférieur droit de la Barre de Statut ; cliquez sur la flèche déroulante et sélectionnez False pour omettre l'icône</li> </ul> <p>Cliquez sur « Étiquette » : cela définit le premier segment de la barre de progression à l'extrémité gauche de la Barre de Statut .</p> <ul style="list-style-type: none"> <li>• Nom - si nécessaire, remplacez le texte par un autre nom pour la barre de progression ; ce texte n'est pas affiché mais le champ ne peut pas être vide</li> <li>• Texte - si nécessaire, remplacez le texte par défaut par un texte différent à afficher à côté de la barre de progression</li> <li>• Type - par défaut, « Texte » pour afficher la string contenue dans le champ « Texte » ; si nécessaire, cliquez sur la flèche déroulante et sélectionnez :       <ul style="list-style-type: none"> <li>- Barre de progression remplie - pour remplacer le texte par une barre partiellement remplie</li> <li>rectangle (pour représenter une section de la barre de progression avec une partie de traitement terminé), ou</li> <li>- Bloc Progress Bar - pour remplacer le texte par un rectangle contenant blocs (pour représenter une section de la barre de progression avec le traitement en cours</li> <li>action)</li> </ul> </li> </ul> <p>Cliquez sur « Barre de progression » - cela définit un deuxième segment de la barre de progression. Les propriétés sont les mêmes que pour le premier segment, sauf qu'il n'y a pas de « Texte » par défaut et que le « Type » par défaut est « Rempli ».</p> <p>Si vous souhaitez ajouter un autre segment à la barre de progression, cliquez sur le nom de l'élément, cliquez sur le bouton Ajouter et indiquez un nom pour ce segment. Indiquez les valeurs des propriétés « Texte » et « Type » comme précédemment.</p> <p>Si vous souhaitez supprimer des segments de la barre de progression, cliquez sur le</p>
--	--








	<p>nom du segment, puis sur le bouton Supprimer. Si vous supprimez tous les segments ainsi que la « barre de zoom » et la « poignée de redimensionnement », la barre d'état elle-même est supprimée.</p> <p>Vous pouvez modifier la séquence des segments en cliquant sur le nom d'un segment et sur les boutons  .</p>
Espace Client	<p>Génère un élément de cadre qui représente la zone client de l'appareil.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• BorderStyle - cliquez sur la flèche déroulante et choisissez de restituer la bordure sous forme de ligne « pleine » ou de ligne « en pointillés », ou de ne pas avoir de bordure du tout (« Aucune »)</li> <li>• Barre de défilementH - cliquez sur la flèche déroulante et choisissez de placer une barre de défilement horizontale en « Haut » ou en « Bas » de la zone client, ou de ne pas avoir de barre de défilement horizontale (' &lt;none&gt; ')</li> <li>• Barre de défilement V - cliquez sur la flèche déroulante et choisissez de placer une barre de défilement verticale à « gauche » ou à « droite » de la zone client, ou de ne pas avoir de barre de défilement verticale (« &lt;none&gt; »)</li> </ul>



## Contrôles





Item	Description
Bouton	<p>Génère un élément qui représente un bouton simple avec le nom de l'élément comme texte du bouton.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• State - cliquez sur la flèche déroulante et sélectionnez un statut pour le bouton : <ul style="list-style-type: none"> <li>- Normal - rectangle simple, pour un affichage normal où le bouton est juste disponible</li> <li>- Focalisé - une bordure intérieure en surbrillance, indiquant, par exemple, que le bouton est la sélection par défaut</li> <li>- Sélectionné - rectangle rempli, indiquant que le bouton est choisi</li> <li>- Désactivé - texte et bordure pâles, indiquant que le bouton est pas disponible</li> </ul> </li> </ul>
Case à cocher	<p>Génère un élément de case à cocher étiqueté, le texte de l'étiquette étant le nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• State - cliquez sur la flèche déroulante et sélectionnez l'état à représenter - « Coché » (icône de coche) ou « Non coché » (contour de la case)</li> </ul>
Bouton radio	<p>Génère un élément de bouton radio étiqueté, le texte de l'étiquette étant le nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• State - cliquez sur la flèche déroulante et sélectionnez l'état à représenter - « Sélectionné » (rempli) ou « Non sélectionné » (contour)</li> </ul>
Zone de liste déroulante	<p>Génère un élément représentant une zone de liste déroulante.</p> <p>Valeur Étiquetés :</p>






	<ul style="list-style-type: none"> <li>• DropDownState - cliquez sur la flèche déroulante et sélectionnez « Ouvrir » pour représenter la zone de liste déroulante utilisée, affichant toutes les valeurs, ou « Fermé » pour représenter la zone de liste déroulante avec une seule valeur sélectionnée</li> <li>• Items - cliquez sur la flèche déroulante et sélectionnez l'élément à représenter comme étant actuellement sélectionné dans la zone de liste déroulante et, si la liste des éléments est développée, mis en surbrillance dans la liste Vous pouvez modifier le texte des éléments et ajouter ou supprimer des éléments dans la liste en modifiant la liste « Valeurs : » dans les notes Valeur Étiquetée</li> </ul>
Étiquette	<p>Génère un élément de texte d'étiquette, sur lequel le texte d'étiquette est le nom de l'élément.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Aligner le texte - cliquez sur la flèche déroulante et choisissez d'aligner le texte à gauche, au centre ou à droite du cadre de l'élément</li> <li>• Multiligne - cliquez sur la flèche déroulante et sélectionnez Vrai pour permettre au texte de s'enrouler sur plusieurs lignes (augmentant automatiquement la profondeur de l'élément), ou Faux pour afficher uniquement le texte qui tient sur une ligne dans la largeur de l'élément actuel</li> </ul>
Liste	<p>Génère un élément de zone de liste.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Items - cliquez sur la flèche déroulante et sélectionnez l'élément à afficher en surbrillance dans la liste Vous pouvez ajouter, supprimer et renommer des éléments en modifiant la liste « Valeurs : » dans les Notes Valeur Étiquetés .</li> <li>• Type de liste - cliquez sur la flèche déroulante et sélectionnez l'affichage de la liste sous la forme « Simple », « Numérotée » ou « À puces »</li> </ul>
Tableau	<p>Génère un élément Tableau avec des colonnes, des lignes et des cellules étiquetées.</p> <p>Double-cliquez sur le tableau pour afficher la dialogue « Propriétés » de l'élément sur la page « Wireframe », qui fournit les facilités d'édition du tableau (ajout, renommage et suppression de colonnes et de lignes, modification de la largeur des colonnes et édition du texte des cellules) via des options et des boutons du menu contextuel. Note que l'éditeur ne fournit pas une image réelle de l'apparence du tableau à l'écran.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Tracer des lignes - cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher les lignes horizontales et verticales entre les cellules tableau , ou Faux pour masquer les lignes</li> <li>• Mettre en surbrillance les en-têtes - cliquez sur la flèche déroulante et sélectionnez Vrai pour mettre en surbrillance l'en-tête de chaque colonne, ou Faux pour laisser les colonnes tableau d'une couleur uniforme</li> </ul>
Image	<p>Génère un espace réservé pour indiquer où une image sera placée sur le téléphone ou la tablette.</p> <p>Vous pouvez afficher une image réelle en attribuant une image alternative à l'élément.</p>


## Commandes Dialogue





Item	Description
Liste de cases à cocher	<p>Génère un élément représentant une liste de contrôle où chaque élément a une case à cocher sur le côté gauche.</p> <p>Double-cliquez sur le tableau pour afficher la dialogue « Propriétés » de l'élément sur la page « Wireframe », que vous utilisez pour gérer cet élément.</p> <p>Pour chaque élément « Case à cocher », remplissez ces champs :</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le nom ou le texte de l'élément</li> <li>• Coché - cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher une case à cocher cochée pour l'élément, ou Faux pour afficher une case à cocher décochée</li> </ul> <p>Pour ajouter un autre élément à la liste, cliquez sur le nom de l'élément et sur le bouton Ajouter, puis indiquez un nom pour l'élément.</p> <p>Pour supprimer un élément de la liste, cliquez sur l'élément et sur le bouton Supprimer.</p> <p>Vous pouvez modifier la séquence des éléments en cliquant sur le nom d'un élément et sur les boutons  .</p>
Barre de format	<p>Génère un élément simple représentant une barre de formatage de texte.</p> <p>Si vous souhaitez représenter une barre d'outils contenant des icônes que vous avez définies, utilisez l'icône « Barre d'outils ».</p>
Liste Vue	<p>Génère un élément représentant une liste horizontale, rectangulaire ou verticale d'éléments de texte (selon la taille de l'élément) avec ou sans images associées.</p> <p>Double-cliquez sur le tableau pour afficher la dialogue « Propriétés » de l'élément sur la page « Wireframe », que vous pouvez utiliser pour ajouter, supprimer ou modifier les éléments et leurs icônes. Pour chaque élément, remplissez ces champs :</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le nom ou le texte de l'élément ; ce champ ne peut pas être laissé vide</li> <li>• Image - saisissez le nom de l'image ou cliquez sur la flèche déroulante et sélectionnez le nom, comme indiqué dans le gestionnaire d'images</li> <li>• Sélectionné - cliquez sur la flèche déroulante et sélectionnez Vrai pour mettre en surbrillance le nom de l'élément ou Faux pour omettre toute mise en surbrillance ; plusieurs éléments peuvent être mis en surbrillance à la fois</li> </ul> <p>Pour ajouter un autre élément à la liste, cliquez sur le nom de l'élément et sur le bouton Ajouter, puis indiquez un nom pour l'élément.</p> <p>Pour supprimer un élément de la liste, cliquez sur l'élément et sur le bouton Supprimer.</p> <p>Vous pouvez modifier la séquence des éléments en cliquant sur le nom d'un élément et sur les boutons  .</p> <p>Si vous réduisez la taille de l'élément de sorte que tous les éléments ne puissent pas être affichés, une barre de défilement s'affiche automatiquement sur le bord droit de l'élément :</p> 
Barre de Statut	<p>Génère un élément de barre d'état identique à la barre d'état générée automatiquement sur l'élément « Dialogue », sauf que vous pouvez positionner cet</p>

	élément indépendamment le dialogue , selon vos besoins.
Barre d'outils	<p>Génère un élément pour représenter une barre d'outils d'icônes, déjà configurée avec certaines icônes de barre d'outils standard.</p> <p>Double-cliquez sur l'élément pour afficher la page 'Wireframe' de la dialogue ' Propriétés ', qui vous permet d'ajouter, de supprimer ou de modifier les éléments et leurs icônes. Pour chaque élément, remplissez ces champs :</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le nom ou le texte de l'élément ; ce champ ne peut pas être laissé vide</li> <li>• Image - saisissez le nom de l'image de l'icône ou cliquez sur la flèche déroulante et sélectionnez le nom, comme indiqué dans le gestionnaire d'images</li> </ul> <p>Pour ajouter un autre élément à la liste, cliquez sur le nom de l'élément et sur le bouton Ajouter, puis indiquez un nom pour l'élément. Vous pouvez ajouter un ou plusieurs éléments appelés « Séparateur » à la liste, qui s'affichent sous la forme d'un  , pour partitionner les groupes d'icônes associées dans la barre d'outils. Si vous ajoutez une image à cet élément, l'image est remplacée par le  .</p> <p>Pour supprimer un élément de la liste, cliquez sur l'élément et sur le bouton Supprimer.</p> <p>Vous pouvez modifier la séquence des éléments en cliquant sur le nom d'un élément et sur les boutons   .</p>
Lecteur audio	Génère un élément simple pour indiquer un contrôle de lecteur audio.
Calendrier	<p>Génère un élément représentant un calendrier de base, affichant la date du jour dérivée de la date du système.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Date - soit : <ul style="list-style-type: none"> <li>- Remplacez la date affichée dans ce champ ou</li> <li>- Cliquez sur la flèche déroulante pour afficher un calendrier actif page et sélectionnez la date sur celle-ci ; cliquez sur le bouton Aujourd'hui pour réinitialiser la date à la date du système</li> </ul> </li> <li>• Mettre en surbrillance la date - cliquez sur la flèche déroulante et sélectionnez Vrai pour mettre en surbrillance la date définie sur l'icône du calendrier, ou Faux pour omettre la surbrillance</li> </ul>
En-tête	<p>Génère un élément représentant un titre ou un texte d'en-tête dans une dialogue . Le texte lui-même est le nom de l'élément.</p> <p>L'élément peut refléter l'un des nombreux niveaux d'en-tête, chacun avec un style de police différent. Vous spécifiez le niveau d'en-tête à afficher à l'aide de la Valeur Étiquetée 'HeaderType' dans l'élément.</p> <p>Double-cliquez sur l'élément pour afficher la page « Wireframe » de la dialogue « Propriétés », que vous pouvez utiliser pour ajouter, supprimer ou modifier les niveaux et les styles d'en-tête. Pour chaque élément, remplissez ces champs :</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le nom du niveau d'en-tête ; ce champ ne peut pas être laissé vide</li> <li>• Couleur - cliquez sur la flèche déroulante et sélectionnez la couleur appropriée dans la palette</li> <li>• Taille de Police - saisissez la taille de la police ou cliquez sur la flèche déroulante et sélectionnez la taille de la police dans la liste</li> <li>• Police Family - cliquez sur la flèche déroulante et sélectionnez le type de police dans la liste</li> </ul>

	<ul style="list-style-type: none"> <li>• Style Police - cliquez sur la flèche déroulante et sélectionnez le style dans la liste ; sélectionnez vide pour aucun style appliqué</li> <li>• Alignement du texte - cliquez sur la flèche déroulante et choisissez d'aligner le texte à gauche, à droite ou au centre</li> <li>• Décoration de texte - cliquez sur la flèche déroulante et sélectionnez si vous souhaitez afficher un soulignement ou une ligne, ou aucun des deux (vide)</li> </ul> <p>Pour ajouter un autre niveau de titre à la liste, cliquez sur le nom de l'élément et sur le bouton Ajouter, puis indiquez un nom pour le niveau.</p> <p>Pour supprimer un niveau de la liste, cliquez sur l'élément et sur le bouton Supprimer.</p> <p>Vous pouvez modifier la séquence des niveaux de titre en cliquant sur le nom d'un niveau et sur les boutons   .</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• HeaderType - cliquez sur la flèche déroulante et sélectionnez le niveau de titre à afficher</li> </ul>
<p>Lien hypertexte</p>	<p>Génère un élément représentant un lien hypertexte dans l'un des trois états suivants : « Lien », « Visité » et « Survol » (survol par la souris). Le texte du lien hypertexte correspond au nom de l'élément.</p> <p>Double-cliquez sur l'élément pour afficher la page « Wireframe » de la dialogue « Propriétés », qui vous permet d'ajouter, de supprimer ou de modifier les états des liens hypertexte. Pour chaque état, remplissez les champs suivants :</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le nom de l'État ; ce champ ne peut pas être laissé vide</li> <li>• Couleur - cliquez sur la flèche déroulante et sélectionnez la couleur appropriée dans la palette</li> <li>• Taille de Police - saisissez la taille de la police ou cliquez sur la flèche déroulante et sélectionnez la taille de la police dans la liste</li> <li>• Police Family - cliquez sur la flèche déroulante et sélectionnez le type de police dans la liste</li> <li>• Style Police - cliquez sur la flèche déroulante et sélectionnez le style dans la liste ; sélectionnez vide pour aucun style appliqué</li> <li>• Alignement du texte - cliquez sur la flèche déroulante et choisissez d'aligner le texte à gauche, à droite ou au centre</li> <li>• Décoration de texte - cliquez sur la flèche déroulante et sélectionnez si vous souhaitez afficher un soulignement ou une ligne, ou aucun des deux (vide)</li> </ul> <p>Pour ajouter un autre état d'hyperlien à la liste, cliquez sur le nom de l'élément et sur le bouton Ajouter, puis indiquez un nom pour l'état.</p> <p>Pour supprimer un état de la liste, cliquez sur l'élément et sur le bouton Supprimer.</p> <p>Vous pouvez modifier la séquence des états en cliquant sur le nom d'un état et sur les boutons   .</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• State - cliquez sur la flèche déroulante et sélectionnez l'état à représenter sur le diagramme</li> </ul>
<p>Barre de menu</p>	<p>Génère un élément représentant une barre de menu standard en haut de l'écran, initialement avec trois options ('Fichier', 'Edition' et ' Vue ') avec l'option 'Fichier' développée dans un sous-menu.</p> <p>Double-cliquez sur l'élément pour afficher la page « Wireframe » de la dialogue « Propriétés », que vous pouvez utiliser pour ajouter, supprimer ou modifier les options de menu du niveau supérieur, du sous-menu et, si vous préférez, des</p>

	<p>sous-niveaux supplémentaires.</p> <p>Pour chaque option de menu - à n'importe quel niveau - remplissez ces champs :</p> <ul style="list-style-type: none"> <li>• « Nom » : saisissez le nom de cet élément de menu</li> <li>• « Étendu » : cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher le sous-menu de cette option (si elle en a un), ou Faux pour masquer le sous-menu</li> <li>• « En surbrillance » : cliquez sur la flèche déroulante et sélectionnez Vrai pour mettre en surbrillance cette option dans le menu, ou Faux pour la laisser non mise en surbrillance ; si « Étendu » est défini sur Vrai, l'option est automatiquement mise en surbrillance</li> </ul> <p>Pour ajouter une sous-option de menu à n'importe quel niveau, cliquez sur le nom de l'option parente et sur le bouton Ajouter, puis indiquez un nom pour la sous-option. Vous pouvez ajouter un ou plusieurs éléments appelés « Séparateur » à la liste, qui s'affiche sous la forme d'une ligne horizontale sur la liste, pour partitionner des groupes d'options associées dans le menu.</p> <p>Pour supprimer une option de la liste, cliquez sur l'élément et sur le bouton Supprimer.</p> <p>Vous pouvez modifier la séquence d'options en cliquant sur le nom d'une option et sur les boutons  .</p> <p>Pour déplacer une option entre deux niveaux de menu, cliquez sur le nom de l'option et sur les boutons  .</p>
Paragraphe	<p>Génère un élément de texte sans bordure, mais qui peut contenir du texte sur plusieurs lignes avec un formatage HTML de base.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Aligner le texte - cliquez sur la flèche déroulante et choisissez d'aligner le texte à gauche, au centre ou à droite du cadre de l'élément</li> <li>• Texte - cliquez sur le bouton  pour éditer l'écran « Note Valeur Étiquetée », sur lequel vous pouvez créer le texte à représenter sur le diagramme ; celui-ci peut utiliser étiquettes de formatage HTML telles que <code>&lt;b&gt;</code> <code>&lt;/b&gt;</code> pour gras ou <code>&lt;u&gt;</code> <code>&lt;/u&gt;</code> pour souligné - tous les formats HTML ne sont pas pris en charge</li> </ul>
Barre de progression	<p>Génère un élément de barre d'état représentant la progression d'un processus.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Pourcentage de remplissage - la valeur par défaut est 30 % d'achèvement ; saisissez le pourcentage d'achèvement à représenter (le niveau de remplissage de la barre de progression)</li> <li>• Style de remplissage - cliquez sur la flèche déroulante et sélectionnez : <ul style="list-style-type: none"> <li>- « Rempli » pour représenter le pourcentage d'achèvement en tant que barre solide, ou</li> <li>- « Bloc » pour représenter le pourcentage d'achèvement en tant que série de blocs ou de barres verticales</li> </ul> </li> </ul>
Contrôle de notation	<p>Génère un élément représentant une bande de notation par étoiles. L'élément affiche toujours cinq étoiles et le nombre d'étoiles remplies indique la note.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>• Évaluation - cliquez sur la flèche déroulante et sélectionnez le nombre d'étoiles à afficher remplies (la note)</li> </ul> <p>Vous ne pouvez redimensionner l'élément qu'horizontalement ; la dimension verticale s'ajuste automatiquement pour toujours représenter cinq étoiles de forme uniforme.</p>

Barre de défilement - Horizontale	Génère un élément représentant une barre de défilement horizontale.
Barre de défilement - verticale	Génère un élément représentant une barre de défilement horizontale.
Contrôle des onglets	<p>Génère un élément représentant une série d'onglets ou de pages. Vous pouvez nommer les onglets et les marquer comme sélectionnés. Cependant, les éléments enfants ne « basculeront » pas lors du changement d'onglet (c'est-à-dire que définir un onglet différent comme sélectionné affichera toujours les mêmes éléments enfants dans l'espace des onglets).</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>Onglets - cliquez sur la flèche déroulante et cliquez sur l'onglet à mettre en surbrillance comme sélectionné Vous pouvez nommer les onglets et en ajouter d'autres à la liste en modifiant la liste « Valeurs : » dans les notes Valeur Étiquetée</li> </ul> <p>Si vous réduisez la taille de l'élément de sorte que tous les onglets ne puissent pas être affichés, une icône de défilement (  ) s'affiche automatiquement dans le coin supérieur droit de l'élément.</p>
Champ de texte	Génère un élément de texte avec une bordure pâle, le texte étant le nom de l'élément, représentant un champ de saisie de données simple.
Lecteur vidéo	Génère un élément qui représente un contrôle de lecteur vidéo.
Sélecteur de date/heure	<p>Génère un élément qui représente le sélecteur de date/heure Microsoft.</p> <p>Valeur Étiquetés :</p> <ul style="list-style-type: none"> <li>CustomFormat - saisissez un format personnalisé pour tout ou partie des composants jour, date, mois, année, heure, minute et seconde, en utilisant ces codes sensibles à la casse (classés par ordre alphabétique) : <ul style="list-style-type: none"> <li>- d - affiche le jour du mois en utilisant un ou deux chiffres</li> <li>- dd - affiche le jour du mois en utilisant deux chiffres, les chiffres 1 à 9 précédé d'un 0</li> <li>- ddd - affiche le jour de la semaine sous forme de trois caractères abréviation</li> <li>- dddd - affiche le nom du jour en entier</li> <li>- h - affiche l'heure en utilisant un ou deux chiffres, au format 12 heures format d'horloge</li> <li>- hh - affiche l'heure en utilisant deux chiffres, les chiffres 1 à 9 précédés de un 0, au format d'horloge de 12 heures</li> <li>- H - affiche l'heure en utilisant un ou deux chiffres, au format 24 heures format d'horloge</li> <li>- HH - affiche l'heure à l'aide de deux chiffres, les chiffres 1 à 9 précédés de un 0, au format d'horloge de 24 heures</li> <li>- m - affiche les minutes en utilisant un ou deux chiffres</li> <li>- mm - affiche les minutes à l'aide de deux chiffres, chiffres 1 à 9 précédé d'un 0</li> <li>- M - affiche le numéro du mois en utilisant un ou deux chiffres</li> <li>- MM - affiche le numéro du mois à l'aide de deux chiffres, chiffres 1 à 9 précédé d'un 0</li> <li>- MMM - affiche le nom du mois sous forme de trois caractères abréviation</li> <li>- MMMM - affiche le nom du mois en entier</li> <li>- s - affiche les secondes en utilisant un ou deux chiffres</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>- ss - affiche les secondes à l'aide de deux chiffres, chiffres 1 à 9 précédé d'un 0</li> <li>- t - identifie le matin ou l'après-midi avec un seul caractère (A pour (AM, P pour PM)</li> <li>- tt - identifier le matin ou l'après-midi avec les deux caractères abréviation AM ou PM</li> <li>- y - affiche l'année en utilisant un seul chiffre (2022 est affiché comme 2)</li> <li>- yy - affiche l'année en utilisant deux chiffres (2022 s'affiche sous la forme 22)</li> <li>- yyyy - afficher l'année en entier (par exemple, 2022)</li> <li>• Date - remplacez la date ou cliquez sur la flèche déroulante pour afficher un calendrier à partir duquel vous pouvez sélectionner la date à afficher ; la valeur par défaut est la date du jour - si vous la remplacez par une date fixe et souhaitez revenir à la date actuelle (système), cliquez sur le bouton Aujourd'hui</li> <li>• Format - cliquez sur la flèche déroulante et sélectionnez le code du format à utiliser pour afficher la date et l'heure :             <ul style="list-style-type: none"> <li>- Long - le nom complet du jour, la date du jour, le nom complet du mois, l'année entière (par exemple : mercredi 17 février 2022)</li> <li>- Court - le jour du mois en chiffres simples/doubles, le mois sous forme de nombre à deux chiffres, l'année en entier (par exemple : 17/02/2022)</li> <li>- Heure - l'heure, les minutes et les secondes au format 12 heures (pour exemple 12:59:34 PM)</li> <li>- Personnalisé - applique le format personnalisé que vous avez défini dans le 'CustomFormat' Valeur Étiquetée</li> </ul> </li> </ul>
<p>Contrôle des arbres</p>	<p>Génère un élément représentant une hiérarchie ou un arbre de nœuds, avec des lignes brisées reliant les nœuds frères et une zone d'extension (+ ou -) à côté des nœuds qui ont des sous-nœuds.</p> <p>Double-cliquez sur l'élément pour afficher la dialogue « Propriétés » de l'élément sur la page « Wireframe », que vous pouvez utiliser pour ajouter, supprimer ou modifier les nœuds de l'arborescence au niveau supérieur, au sous-niveau et, si vous préférez, aux sous-niveaux supplémentaires.</p> <p>Pour chaque nœud - à n'importe quel niveau - remplissez ces champs :</p> <ul style="list-style-type: none"> <li>• Nom - saisissez le nom de ce nœud</li> <li>• Développé - cliquez sur la flèche déroulante et sélectionnez Vrai pour afficher les nœuds subordonnés de ce nœud (s'il en a), ou Faux pour masquer les nœuds subordonnés</li> <li>• Sélectionné - cliquez sur la flèche déroulante et sélectionnez Vrai pour mettre ce nœud en surbrillance ou Faux pour le laisser non mis en surbrillance</li> </ul> <p>Pour ajouter un sous-nœud à n'importe quel niveau, cliquez sur le nom du nœud parent et sur le bouton Ajouter, puis indiquez un nom pour le sous-nœud.</p> <p>Pour supprimer un nœud de la hiérarchie, cliquez sur le nom du nœud et sur le bouton Supprimer.</p> <p>Vous pouvez modifier la séquence des nœuds en cliquant sur le nom d'un nœud et sur les boutons  .</p> <p>Pour déplacer un nœud entre deux niveaux de la hiérarchie, cliquez sur le nom du nœud et sur les boutons  .</p> <p>Vous pouvez également éditer directement le XML de l'élément dans l'onglet « Wireframe » de la fenêtre Propriétés .</p>
<p>Zone de groupe</p>	<p>Génère un élément représentant une Groupbox, avec le nom de l'élément dans le coin supérieur gauche. Vous pouvez utiliser cet élément pour entourer et regrouper d'autres éléments du le dialogue .</p>

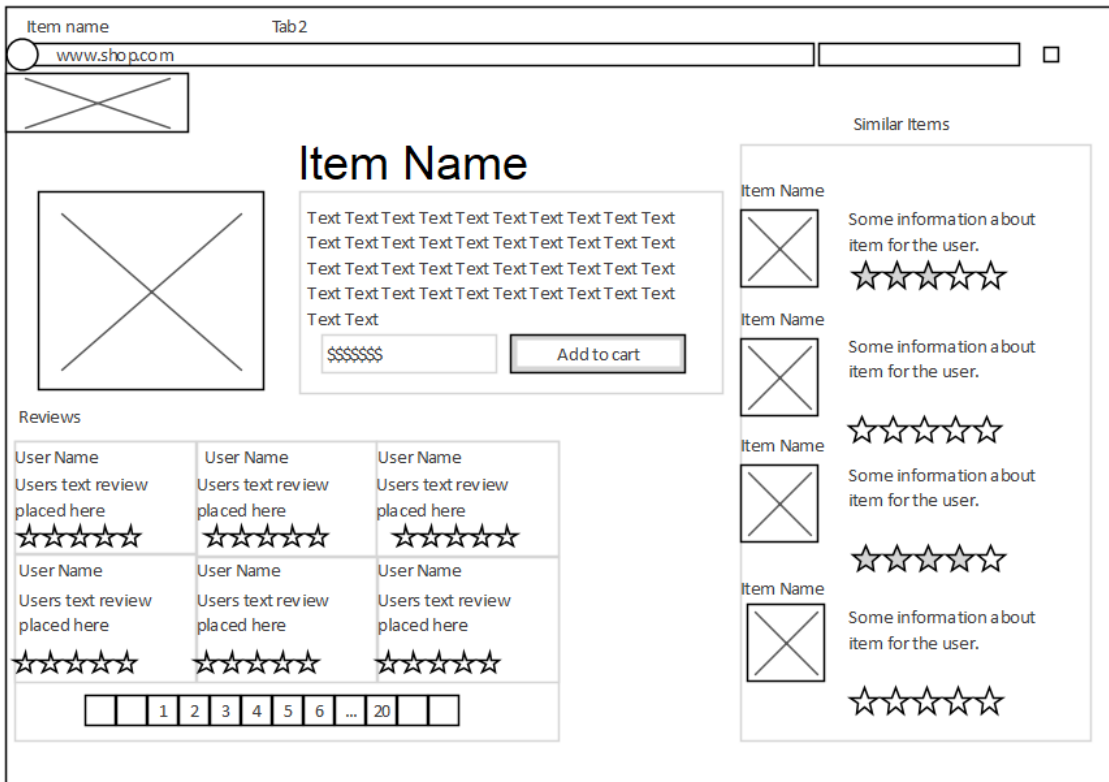


## Motifs


Item	Description
Dialogue	Ce Motif génère une petite dialogue contenant trois panneaux avec des champs de saisie de données et des boutons radio, ainsi que deux boutons, comme illustré au début de cette rubrique. Vous pouvez l'utiliser comme exemple ou comme base pour une conception dialogue similaire.

# Boîte à outils de wireframe de page Web

Les pages de la boîte à outils Diagramme 'Webpage Wireframe' fournissent les gabarits pour modélisation des schémas, des plans ou de la structure d'un site Web, définissant le fonctionnement des pages Web. Vous pouvez voir et développer un exemple de modélisation de l'interface d'une page Web en faisant glisser l'icône 'Webpage Motif' sur un diagramme . Ensuite, à l'aide de la boîte à outils, vous pouvez faire glisser des types d'éléments sur votre diagramme pour ajouter des fonctionnalités clés de l'interface utilisateur.

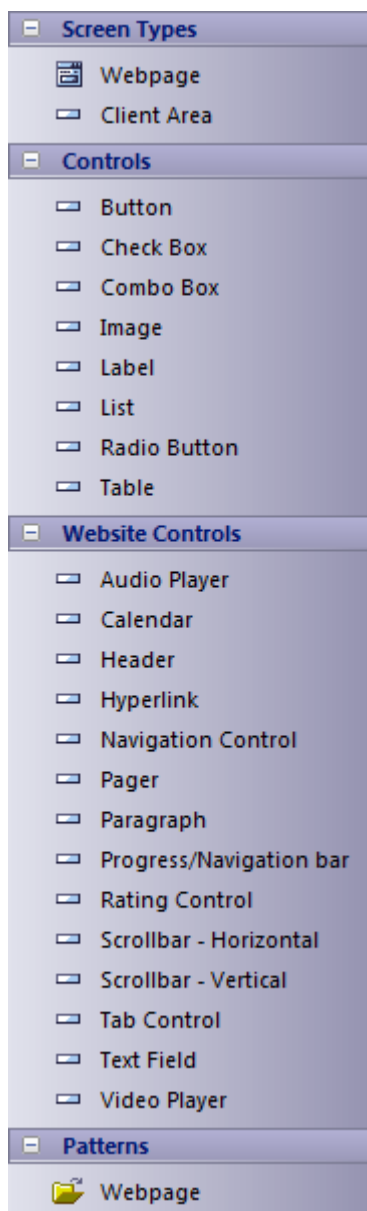


## Accéder


Dans la boîte à outils Diagramme , cliquez sur  pour afficher la dialogue ' Trouvez Item de Boîte à Outils ' et spécifiez 'Wireframing' ou 'Webpage'.


Ruban	Conception > Diagramme > Boîte à outils
Raccourcis Clavier	Ctrl+Maj+3

## Boîte à outils

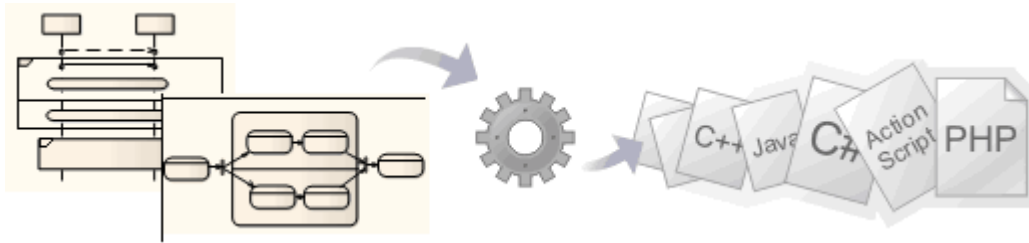


## Fonctionnalités des éléments

Image	Détail
<p>Zone de liste déroulante</p>	<p>La zone de liste déroulante, la liste, la barre de progression/navigation et le contrôle d'onglets fournissent tous des listes de valeurs.</p> <p>Pour étendre une liste :</p> <ul style="list-style-type: none"> <li>• Ouvrez la Dialogue Propriétés (double-clic ou Maj+Entrée)</li> <li>• Sélectionnez l'onglet Wireframing</li> <li>• Sélectionnez la Valeur Étiquetée</li> <li>• Cliquez sur l'icône </li> <li>• et ensuite ajouter, modifier ou supprimer des éléments de la liste « Valeurs ».</li> </ul>

Élément de calendrier	L'élément Calendrier, une fois créé, prend par défaut la date du jour et continue de se mettre à jour chaque jour, sauf si vous définissez une valeur dans la Valeur Étiquetée de l'élément. Si elle est définie, la date reste statique jusqu'à ce qu'elle soit réinitialisée sur « Aujourd'hui » dans la Valeur Étiquetée .
Paragraphe	<p>Dans l'élément Paragraphe, vous pouvez :</p> <ul style="list-style-type: none"> <li>• Définir l'alignement du texte dans la propriété « Aligner le texte »</li> <li>• Modifiez le texte en utilisant l'option [...] dans la propriété 'Texte' (qui est de type &lt;mémo&gt;). Le texte peut utiliser un formatage de texte HTML de base, comme pour d'autres notes formatées dans les éléments.</li> </ul>
Contrôle de navigation	<p>Le contrôle de navigation définit un menu avec, si nécessaire, plusieurs niveaux d'options de sous-menu. Vous pouvez ajouter et supprimer des options à n'importe quel niveau en utilisant l'onglet « Wireframe » de la dialogue « Propriétés ».</p> <p>Chaque option, à n'importe quel niveau, possède les propriétés suivantes :</p> <ul style="list-style-type: none"> <li>• « Nom » : définir le texte de l'option de menu</li> <li>• « Étendu » : indique si l'option affichera ses sous-options (le cas échéant) ; les éléments développés sont toujours mis en surbrillance, quel que soit le paramètre « Étendu »</li> <li>• « En surbrillance » : dessinez l'élément avec un arrière-plan de couleur différente</li> </ul>
Contrôle de progression/navigation	<p>Le contrôle de navigation de progression fournit une progression visuelle à travers les pages.</p> <p>Pour étendre une liste :</p> <ul style="list-style-type: none"> <li>• Ouvrez la Dialogue Propriétés (double-cliquez ou maj+entrée)</li> <li>• Sélectionnez l'onglet Wireframing</li> <li>• Sélectionnez les <i>étapes</i> Valeur Étiquetée</li> <li>• Cliquez sur l'icône </li> <li>• puis ajoutez, modifiez ou supprimez des éléments de la liste « Étapes ».</li> </ul>
Lien hypertexte	<p>Les éléments Hyperlien et En-tête sont également définis dans l'onglet « Wireframe » et possèdent un certain nombre de propriétés de style que vous définissez à l'aide de listes déroulantes simples :</p> <ul style="list-style-type: none"> <li>• Couleur</li> <li>• Taille de Police</li> <li>• Famille Police</li> <li>• Style Police</li> <li>• Alignement du texte</li> <li>• Décoration de texte</li> </ul>

## Génération de code Modèle Comportementale



Les fonctionnalités d'ingénierie système polyvalentes d'Enterprise Architect permettent de générer du code pour les langages de description de logiciels, de systèmes et de matériels directement à partir de modèles comportementaux, tels que diagrammes Statemachine , Séquence (Interaction) et Activity. Les langages pris en charge incluent C(OO), C++, C# , Java, VB.Net, VHDL, Verilog et SystemC.

Le code logiciel peut être généré à partir de diagrammes Statemachine , Séquence et Activity, et des langages de description matérielle à partir de diagrammes Statemachine (en utilisant les gabarits Legacy Statemachine ).

### Accéder

Ruban	Développer > Code Source > Générer
-------	------------------------------------

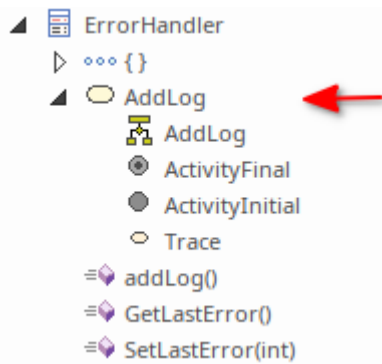
### Spécificités Modèle Comportementale

La génération de code de modèle Comportementale est prise en charge pour les trois principaux types de modèles comportementaux. Cependant, chaque type de modèle comportemental possède ses propres caractéristiques en fonction du type d'élément concerné. Ces rubriques fournissent guidage et des références pour les principaux types d'éléments utilisés.

Type	Description
Activité	Un aperçu des principaux types d'actions et des détails sur leur utilisation dans la génération de code.
Interaction	Détails couvrant l'utilisation des messages et des fragments pour la génération de code des diagrammes d'interaction ( Séquence ).
Statemachines	Détails couvrant les options de définition du code à générer à l'aide States , y compris les comportements - Entrée/Sortie/Do et Transitions dans une Statemachine .

### Structure

La génération de code du modèle Comportementale nécessite principalement que toutes les constructions comportementales soient contenues dans une classe (en tant qu'enfant de cette classe).



Si des constructions comportementales font référence à des éléments externes en dehors du Paquetage actuel, vous devez ajouter un connecteur Import du Paquetage actuel au Paquetage contenant les éléments externes. Pour plus de détails, consultez le connecteur *Import* -type dans la rubrique d'aide *Diagramme Paquetage* .

## Générer du code à partir de diagrammes comportementaux en utilisant le projet EAExample

Étape	Action
1	Ouvrez le fichier EAExample.eap en sélectionnant l'option de ruban ' Démarrer > Aide > Aide > Ouvrir le Modèle d'exemple '.
2	<p>Depuis la fenêtre Navigateur , sélectionnez l'un de ces Paquetages :</p> <p><b>Exemples de langage logiciel :</b></p> <ul style="list-style-type: none"> <li>Exemple Modèle &gt; Ingénierie de Logiciel &gt; Modèle Java avec comportements Générer les classes Account et Order</li> <li>Exemple Modèle &gt; Ingénierie des Systèmes &gt; Modèle d'implémentation &gt; Logiciel &gt; C# Générer la classe DataProcessor</li> <li>Exemple Modèle &gt; Ingénierie des Systèmes &gt; Exemple SysML &gt; Modèle d'implémentation &gt; Logiciel &gt; C++ Générer la classe IO</li> <li>Exemple Modèle &gt; Ingénierie des Systèmes &gt; Exemple SysML &gt; Modèle d'implémentation &gt; Logiciel &gt; Java Générer la classe IO</li> <li>Exemple Modèle &gt; Ingénierie des Systèmes &gt; Exemple SysML &gt; Modèle d'implémentation &gt; Logiciel &gt; VBNet Générer la classe IO</li> </ul> <p><b>Exemples de langage matériel :</b></p> <ul style="list-style-type: none"> <li>Exemple Modèle &gt; Ingénierie des Systèmes &gt; Exemple SysML : Lecteur Audio Portable &gt; Modèle d'implémentation &gt; Matériel &gt; SystemC Générer la classe PlayBack</li> <li>Exemple Modèle &gt; Ingénierie des Systèmes &gt; Exemple SysML : Lecteur Audio Portable &gt; Modèle d'implémentation &gt; Matériel &gt; VHDL Générer la classe PlayBack</li> <li>Exemple Modèle &gt; Ingénierie des Systèmes &gt; Exemple SysML : Lecteur Audio Portable &gt; Modèle d'implémentation &gt; Matériel &gt; Verilog Générer la classe PlayBack</li> </ul>

3	<p>Une fois terminé :</p> <ul style="list-style-type: none"><li>• Sélectionnez la classe qui a été utilisée pour la génération</li><li>• Appuyez sur Ctrl+E pour ouvrir le code source généré.</li></ul> <p>Vous devriez voir des méthodes générées dans le code.</p>
---	---

## Notes

- La génération de code logiciel à partir de modèles comportementaux est disponible dans les éditions Unified et Ultimate d' Enterprise Architect
- La génération de code matériel à partir de modèles Statemachine est disponible dans les éditions Unified et Ultimate d' Enterprise Architect
- Pour C(OO), sur la page « Spécifications C » de la dialogue « Gérer les options Modèle », définissez l'option « Support orientée Object » sur Vrai.  
Voir la rubrique d'aide *Options C - Modèle* .
- La synchronisation du code n'est pas prise en charge pour le code comportemental.

## Génération de code - Diagrammes d'activités

La génération de code à partir de diagrammes d'activité dans une classe nécessite une phase de validation, au cours de laquelle Enterprise Architect utilise l'optimiseur de graphe d'ingénierie système pour analyser le diagramme et le restituer sous forme de différentes constructions à partir desquelles du code peut être généré. Enterprise Architect transforme également les constructions en l'un des différents types d'action (le cas échéant), de manière similaire aux constructions diagramme d'interaction.

### Actes

Action	Description
Actions d'appel (actions d'invocation)	<p>Utilisé pour invoquer des opérations ou des comportements dans un diagramme d'activité ; les deux principales variantes d'appels d'actions prises en charge dans la génération de code comportemental sont :</p> <ul style="list-style-type: none"> <li>• Action CallOperation - utilisée pour invoquer des opérations, qui peuvent être dans la même classe ou dans d'autres classes au sein du même Paquetage ; si vous référencez des opérations d'autres classes au sein du même Paquetage , vous devez avoir une cible à laquelle la demande est transmise</li> <li>• Action CallBehavior - utilisée pour invoquer une autre activité dans un flux d'activités ; l'activité référencée est censée être dans la même classe</li> </ul> <p><b>Arguments</b></p> <p>Les actions d'appel peuvent spécifier des valeurs d'argument correspondant aux paramètres du comportement ou de la fonctionnalité comportementale associée. Vous pouvez ajouter les arguments manuellement ou les créer automatiquement à l'aide du bouton Synchroniser de la dialogue « Arguments ».</p>
Créer un objet action	<p>Utilisé pour désigner une création objet dans le flux d'activité ; vous pouvez définir le Pin de résultat de CreateObjectAction comme l' objet à créer, en utilisant la fenêtre Propriétés de l'élément Action .</p> <p>Le classificateur de CreateObjectAction signifie le classificateur pour lequel une instance doit être créée.</p>
Action de destruction d'objet	<p>Utilisé pour indiquer une suppression objet dans le flux d'activité ; vous pouvez définir le Pin cible de DestroyObjectAction comme l' objet à détruire, en utilisant la fenêtre Propriétés de l'élément Action .</p>
Boucles	<p>L'optimiseur de graphiques d'ingénierie système d' Enterprise Architect est également capable d'analyser et d'identifier les boucles ; une boucle identifiée est rendue en interne sous la forme d'une boucle Action , qui est traduite par les macros de génération de code EASL pour générer le code requis.</p> <p>Vous pouvez avoir une boucle unique, des boucles imbriquées et plusieurs niveaux de boucles imbriquées.</p>
Déclarations conditionnelles	<p>Pour modéliser une instruction conditionnelle, vous utilisez les nœuds Décision / Merge.</p> <p>Alternativement, vous pouvez impliquer des décisions/fusions en interne ; l'optimiseur de graphique attend un nœud de fusion associé pour chaque nœud Décision , afin de faciliter le suivi efficace des différentes branches et l'analyse des constructions de code qu'elles contiennent.</p>



## Notes

- Pour pouvoir générer du code à partir de modèles comportementaux, toutes les constructions comportementales doivent être contenues dans une classe

# Génération de code - Diagrammes d'interaction

Lors de la génération de code à partir de diagrammes d'interaction ( Séquence ) dans une classe, Enterprise Architect applique son optimiseur de graphes d'ingénierie système pour transformer les constructions de classe en paradigmes programmatiques. Les messages et les fragments sont identifiés comme deux des nombreux types d'actions en fonction de leur fonctionnalité, et Enterprise Architect utilise les gabarits de génération de code pour restituer leur comportement en conséquence.

## Actes

Action	Description
Appel Action	Un message qui appelle une opération.
Action Créer	Un message avec cycle de vie = Nouveau.
Action Détruire	Un message avec un cycle de vie = Supprimer.
Boucle Action	Un fragment combiné avec Type = Alt.
Action si	Un fragment combiné avec Type = boucle.
Affecter à	Un message d'appel avec un attribut cible valide défini à l'aide du champ « Attribuer à » est rendu dans le code comme l'attribut cible d'une Action d'appel.

## Notes

- Pour pouvoir générer du code à partir de modèles comportementaux, toutes les constructions comportementales doivent être contenues dans une classe
- Pour un diagramme d'interaction ( Séquence ), le moteur de génération de code comportemental s'attend à ce que le diagramme Séquence et tous ses messages et fragments d'interaction associés soient encapsulés dans un élément d'interaction.

## Génération de code – Statemachines

Une Statemachine illustre comment un objet (représenté par une classe) peut changer d'état, chaque changement d'état étant une transition initiée par un déclencheur découlant d'un événement, souvent sous des conditions ou des contraintes définies comme des gardes. Au fur et à mesure que vous modélisez la façon dont l'objet change d'état, vous pouvez générer et construire (compiler) du code à partir de celui-ci dans le langage logiciel approprié et exécuter le code, en visualisant l'exécution via le simulateur Modèle .

Il est également possible, dans Enterprise Architect , de combiner les Statemachines d'objets distincts mais liés pour voir comment ils interagissent (via Broadcast Événements ), et de créer et générer rapidement du code à partir de variantes du modèle. Par exemple, vous pouvez modéliser le comportement de :

- La roue arrière côté passager d'un véhicule en modes traction arrière et traction avant (un seul Statemachine )
- Le volant et les quatre roues motrices d'un véhicule en mode 4 roues motrices (cinq Statemachines )
- Les roues d'un véhicule tout-terrain et d'une voiture de sport (deux artefacts, instances d'une combinaison de Statemachines )

L'élément Statemachine Exécutable Artifact est d'une importance cruciale pour générer et tester le code de toutes ces options. Il agit comme conteneur et unité de génération de code pour vos modèles Statemachine .

Vous n'utilisez pas cette méthode pour générer du code pour les langages de définition de matériel, mais vous pouvez également générer du code HDL et du code logiciel à partir de Statemachines à l'aide des facilités génériques de génération de code dans Enterprise Architect (voir les procédures *Générer Code Source* ).



### Prérequis

- Sélectionnez « Paramètres > Modèle > Options > Ingénierie du code source » et, pour le langage de codage logiciel approprié (Java, C, C# ou ANSI C++), définissez l'option « Utiliser la nouvelle Statemachine Gabarit » sur « Vrai »
- Si vous travaillez en C++, sélectionnez « Paramètres > Modèle > Options > Ingénierie du code source > C++ » et définissez l'option « Version C++ » sur « ANSI »

Cette méthode de génération de code ne s'applique pas aux gabarits de génération de code Legacy Statemachine développés avant Enterprise Architect Release 11.0, ni à la génération de code Hardware Definition Language.

### Accéder

Faites glisser un artefact Statemachine Exécutable depuis la page « Simulation » de la boîte à outils Diagramme vers votre diagramme . La page « Simulation » de la boîte à outils Diagramme est accessible à l'aide de l'une des méthodes décrites dans ce tableau .

Ruban	Conception > Diagramme > Boîte à outils > Simulation
Raccourcis Clavier	Ctrl+Maj+3 > Simulation
Autre	Vous pouvez afficher ou masquer la boîte à outils Diagramme en cliquant sur les icônes  ou  à l'extrémité gauche de la barre de légende en haut de la Vue Diagramme .

### Préparez votre (vos) diagramme (s) Statemachine

Étape	Action
1	Pour chaque Statemachine vous souhaitez modéliser, créez un diagramme de classe.
2	Depuis la page « Classe » de la boîte à outils Diagramme , faites glisser l'icône « Classe » sur votre diagramme et donnez à l'élément un nom approprié.
3	<p>Cliquez-droit sur l'élément Class et sélectionnez l'option de menu contextuel 'Nouveau Diagramme enfant   Statemachine '.</p> <p>Donnez au diagramme de la Statemachine un nom approprié.</p>
4	Créez le modèle Statemachine pour refléter les transitions appropriées entre States .

## Configurer l'artefact Statemachine Exécutable

Étape	Action
1	Créez un nouveau diagramme de classe pour contenir la ou les Statemachine modélisées à partir desquelles vous avez l'intention de générer du code.
2	Depuis la page « Simulation » de la boîte à outils Diagramme , faites glisser l'icône « Statemachine Exécutable » sur le diagramme pour créer l'élément Artefact. Nommez l'élément et faites glisser ses bordures pour l'agrandir.
3	<p>Depuis la fenêtre Navigateur , faites glisser l'élément Class (premier) contenant un diagramme Statemachine sur l'élément Artifact du diagramme .</p> <p>La dialogue « Coller &lt;nom de l'élément&gt; » s'affiche. Dans le champ « Déposer sous », cliquez sur la flèche déroulante et sélectionnez la valeur « Propriété ».</p> <p>(Si le dialogue ne s'affiche pas, appuyez sur Ctrl pendant que vous faites glisser l'élément Classe depuis la fenêtre Navigateur .)</p>
4	Cliquez sur le bouton OK . L'élément Class est collé à l'intérieur de l'artefact en tant que partie.
5	<p>Répétez les étapes 3 et 4 pour toutes les autres classes avec Statemachines que vous souhaitez combiner et pour lesquelles vous souhaitez générer du code. Il peut s'agir de :</p> <ul style="list-style-type: none"> <li>• Répétez les « gouttes » de la même classe et de Statemachine , modélisation d'objets parallèles</li> <li>• Différentes classes et Statemachines , modélisation d'objets interactifs séparés</li> </ul>
6	<p>Cliquez-droit sur l'élément Artifact et sélectionnez l'option ' Propriétés &gt; Propriétés ', développez la catégorie 'Avancé' et, dans le champ 'Langue', cliquez sur la flèche déroulante et définissez la langue du code sur la même langue que celle définie pour les éléments Class.</p> <p>Vous pouvez maintenant faire glisser cet élément Statemachine Exécutable Artifact de la fenêtre Navigateur sur le diagramme autant de fois que vous le souhaitez et modifier les parties pour modéliser des variantes du système ou du processus, ou le même système ou processus avec différents langages de programmation.</p>

## Générer du code à partir d'un artefact

Étape	Action
1	<p>Cliquez sur l'élément Statemachine Exécutable Artifact et sélectionnez l'option de ruban 'Simulate &gt; States Exécutables &gt; Statemachine &gt; Générer '.</p> <p>La dialogue « Génération de code Statemachine exécutable » s'affiche.</p>
2	<p>Dans le champ « Répertoire de sortie du projet », saisissez ou recherchez le chemin du répertoire sous lequel créer les fichiers de sortie.</p> <p>Lors de la génération de code, tous les fichiers existants dans ce répertoire sont supprimés.</p>
3	<p>Sélectionnez le système cible. Si vous travaillez sous Windows sélectionnez l'option « Local ». Si vous travaillez sous Linux, choisissez l'option « À distance ». Le choix affecte les scripts générés pour support la Simulation .</p>
4	<p>Dans le champ « Emplacement du répertoire d'installation de &lt;compilateur&gt; », saisissez ou recherchez le chemin du répertoire d'installation du compilateur, qui sera automatiquement mappé au chemin local (affiché à gauche du champ). Pour chaque langage de programmation, les chemins peuvent ressembler à ces exemples :</p> <ul style="list-style-type: none"> <li>• Java JAVA_HOME C:\Program Files (x86)\Java\jdk1.7.0_17</li> <li>• C/C++ VC_HOME C:\Program Files (x86)\Microsoft Visual Studio 9.0</li> <li>• C# CS_HOME C:\Windows\Microsoft.NET\Framework\V3.5</li> </ul>
5	<p>Cliquez sur le bouton Générer . Les fichiers de code sont créés en fonction du langage de programmation.</p> <p>La fenêtre Sortie système s'affiche avec un onglet « Sortie Statemachine Exécutable », indiquant la progression et l'état de la génération.</p> <p>Lors de la génération du code, une fonction de validation automatique est exécutée pour vérifier les erreurs diagramme ou de modèle par rapport aux contraintes UML . Les éventuelles erreurs sont identifiées par des messages d'erreur dans l'onglet ' Statemachine Exécutable Output'.</p> <p>Double-cliquez sur un message d'erreur pour afficher la structure modélisation dans laquelle l'erreur se produit et corrigez l'erreur avant de régénérer le code.</p>
6	<p>Lorsque le code est généré sans erreur, cliquez sur l'élément Artifact et sélectionnez l'option de ruban « Simuler &gt; States Exécutables &gt; Statemachine &gt; Construire » pour compiler le code.</p> <p>La fenêtre Sortie système s'affiche avec un onglet « Créer », indiquant la progression et l'état de la compilation. Notez que la compilation inclut la configuration de l'opération de simulation.</p>

## Macros de génération de code

Vous pouvez également utiliser deux macros dans la génération de code pour Statemachines .

Nom de la macro	Description
ENVOYER_EVENEMENT	Envoyer un événement à un récepteur (la partie). Par exemple :
T	<code>%SEND_EVENT("événement1", "Partie1")%</code>

ÉVÉNEMENT DIFFUSÉ	Diffuser un événement à tous les récepteurs. Par exemple : %BROADCAST_EVENT("événement2")%
-------------------	---

## Exécuter/simuler du code à partir d'un artefact

Étape	Action
1	Sélectionnez l'option du ruban « Simuler > Simulation Dynamique > Simulateur > Appliquer l'espace de travail » pour afficher ensemble la fenêtre Simulation et la fenêtre Simulation Événements Placez les deux fenêtres dans une zone pratique de l'écran.
2	Sur le diagramme ou la fenêtre Navigateur , cliquez sur l'élément Artificiel et sélectionnez l'option de ruban 'Simulate > States Exécutables > Statemachine > Exécuter ' Le premier diagramme Statemachine de la série s'affiche avec la simulation du processus déjà démarrée. Dans la fenêtre Simulation , les étapes de traitement sont indiquées dans ce format : [03516677] Partie 1 [Classe 1]. Initial_367_TO_State4_142 Effet [03516683] Partie 1 [Classe 1]. StateMachine_State4 ENTRÉE [03516684] Partie 1 [Classe 1].StateMachine_State4 DO [03518375] Bloqué
3	Cliquez sur les boutons appropriés de la barre d'outils de la fenêtre de Simulation pour parcourir la simulation comme vous le souhaitez. Lorsque la simulation se termine au niveau de l'élément Quitter ou Terminer, cliquez sur le bouton Arrêter dans la barre d'outils de la fenêtre Simulation .
4	Lorsque la trace indique Bloqué, la simulation a atteint un point où un événement Déclencheur doit se produire avant que le traitement puisse continuer. Dans la fenêtre Simulation Événements , dans la colonne « Déclencheurs en Attente », double-cliquez sur le Déclencheur approprié. Lorsque le Déclencheur est déclenché, la simulation continue jusqu'au prochain point de pause, Déclencheur ou sortie.

## Notes

- Si vous apportez de petites modifications à un modèle Statemachine existant, vous pouvez combiner les opérations de génération, de construction et exécuter de code en sélectionnant l'option de ruban « Simuler > States Exécutables > Statemachine > Générer , construire et exécuter ».
- Vous pouvez également générer du code en JavaScript

## Statemachine héritée Gabarits

La génération de code fonctionne à l'aide d'un ensemble de gabarits de génération. À partir de la version 11.0 d'Enterprise Architect, un ensemble différent de gabarits est disponible par défaut pour la génération de code logiciel à partir d'un diagramme Statemachine en code Java, C, ANSI C++ ou C#. Vous pouvez toujours utiliser les gabarits d'origine, comme décrit ici, pour les modèles développés dans les versions antérieures d'Enterprise Architect, si vous ne souhaitez pas faciliter mettre à niveau pour les nouveaux gabarit.

### Basculer entre gabarits Legacy et Release 11

#### Accéder

Affichez la dialogue « Gérer les options Modèle », puis affichez la page « Spécifications de langue » pour la langue choisie, en utilisant l'une des méthodes décrites dans ce tableau. Si nécessaire, développez le groupement « Ingénierie Statemachine (pour le modèle actuel) » et définissez l'option « Utiliser la nouvelle Statemachine Gabarit » sur True (pour utiliser les gabarits ultérieurs) ou False (pour utiliser les gabarits hérités).

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > [nom de la langue]
-------	--

### Transformations héritées Gabarit

Une Statemachine dans une classe génère en interne un certain nombre de constructions dans des langages logiciels pour assurer l'exécution efficace des comportements des States (do, entry et exit) et également pour coder l'effet de transition approprié lorsque cela est nécessaire.

Objets Modèle	Objets de code
Énumérations	<ul style="list-style-type: none"> <li>StateType - se compose d'une énumération pour chacun des States contenus dans la Statemachine</li> <li>TransitionType – consiste en une énumération pour chaque transition à laquelle est associé un effet valide ; par exemple, ProcessOrder_Delivered_to_ProcessOrder_Closed</li> <li>CommandType – consiste en une énumération pour chacun des types de comportement qu'un State peut contenir (Do, Entry, Exit)</li> </ul>
Attributes	<ul style="list-style-type: none"> <li>currState:StateType - une variable pour contenir les informations sur State actuel</li> <li>nextState:StateType - une variable pour contenir les informations de State suivant, définies par les transitions de chaque State en conséquence</li> <li>currTransition:TransitionType - une variable pour contenir les informations de transition actuelles ; elle est définie si la transition est associée à un effet valide</li> <li>transcend:Boolean - un indicateur utilisé pour indiquer si une transition est impliquée dans la transcendance entre différentes Statemachines (ou états de sous-machines)</li> <li>xx_history:StateType - une variable d'historique pour chaque State Statemachine /sous-machine, pour contenir des informations sur le dernier</li> </ul>

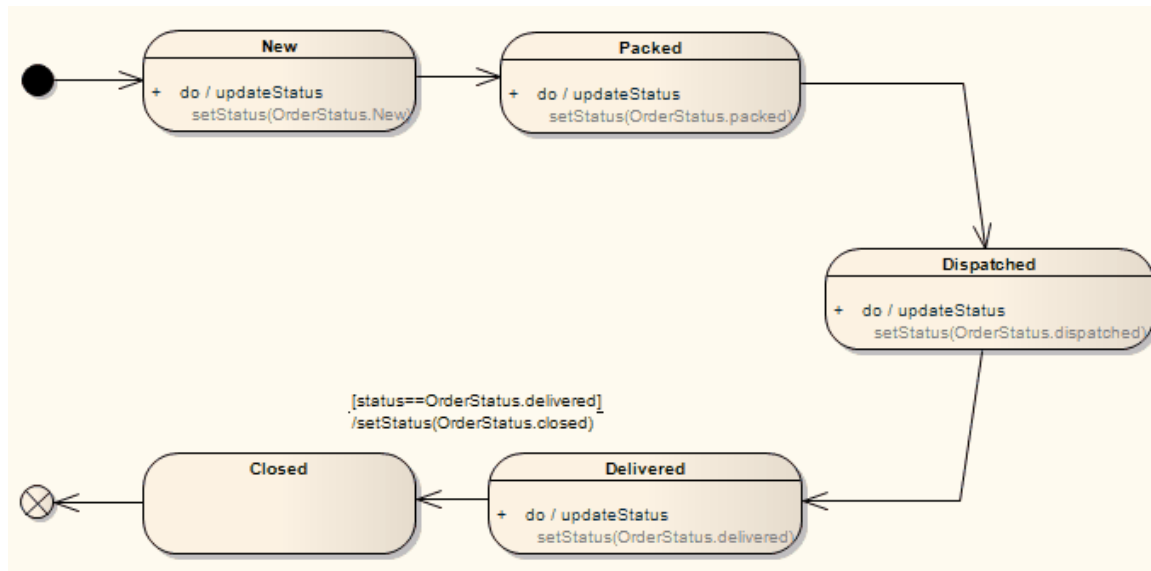
	State à partir duquel la transition a eu lieu
Opérations	<ul style="list-style-type: none"><li>• StatesProc - une procédure States , contenant une carte entre l'énumération d'un State et son fonctionnement ; elle déréférence les informations de State actuel pour invoquer la fonction de State respectif</li><li>• TransitionsProc - une procédure de transition, contenant une carte entre l'énumération de la transition et son effet ; elle invoque l'effet de la transition</li><li>• &lt;&lt;État&gt;&gt; - une opération pour chacun des States contenus dans la Statemachine ; cela restitue les comportements d'un State en fonction du CommandType d'entrée et exécute également ses transitions</li><li>• initializeStateMachine - une fonction qui initialise tous les attributs liés au framework</li><li>• runStateMachine - une fonction qui parcourt chaque State et exécute leurs comportements et transitions en conséquence</li></ul>

## Notes

- Pour pouvoir générer du code à partir de modèles comportementaux, toutes les constructions comportementales doivent être contenues dans une classe



## Code Java généré à partir d' StateMachine héritée Gabarit



```

enumération privée StateType : int
{
ProcessusCommande_Livrée,
ProcessusCommande_Emballée,
ProcessusCommande_Fermée,
ProcessusOrder_Dispatched,
ProcessOrder_New,
ST_NOSTATE
}
enumération privée TransitionType : int
{
ProcessOrder_Delivered_to_ProcessOrder_Closed,
TT_NOTRANSITION
}
enumération privée CommandType
{
Faire,
Entrée,
Sortie
}
État privéType currState;
État privéType nextState;
TransitionType privé currTransition;
transcendance booléenne privée ;
État privé Type ProcessOrder_history ;
processus privé voidOrder_Delivered(CommandType commande)

```

```
{
commutateur(commande)
{
affaire à faire :
{
// Faire des comportements..
setStatus(Livré);
// Les transitions State
si((statut==Livré))
{
nextState = StateType.ProcessOrder_Closed;
currTransition = TransitionType.ProcessOrder_Delivered_to_ProcessOrder_Closed;
}
casser;
}
défaut:
{
casser;
}
}
}
processus privé voidOrder_Packed(CommandType commande)
{
commutateur(commande)
{
affaire à faire :
{
// Faire des comportements..
setStatus(Emballé);
// Les transitions State
nextState = StateType.ProcessOrder_Dispatched;
casser;
}
défaut:
{
casser;
}
}
}
processus privé voidOrder_Closed(CommandType commande)
{
commutateur(commande)
```

```
{
affaire à faire :
{
// Faire des comportements..
// Les transitions State
casser;
}
défaut:
{
casser;
}
}
processus privé voidOrder_Dispatched(CommandType commande)
{
commutateur(commande)
{
affaire à faire :
{
// Faire des comportements..
setStatus(Expédié);
// Les transitions State
nextState = StateType.ProcessOrder_Delivered;
casser;
}
défaut:
{
casser;
}
}
}
processus privé voidOrder_New(CommandType commande)
{
commutateur(commande)
{
affaire à faire :
{
// Faire des comportements..
setStatus(nouveau);
// Les transitions State
nextState = StateType.ProcessOrder_Packed;
casser;
}
```

```
}
défaut:
{
casser;
}
}
}
privé void StatesProc(StateType currState, CommandType commande)
{
commutateur(état actuel)
{
cas ProcessOrder_Delivered :
{
processOrder_Delivered(commande);
casser;
}
cas ProcessOrder_Packed :
{
processOrder_Packed(commande);
casser;
}
dossier ProcessOrder_Closed :
{
processOrder_Closed(commande);
casser;
}
cas ProcessOrder_Dispatched :
{
processOrder_Dispatched(commande);
casser;
}
affaire ProcessOrder_New :
{
processOrder_New(commande);
casser;
}
défaut:
casser;
}
}
privé void TransitionsProc(TransitionType transition)
{
```

```
commutateur (transition)
{
dossier ProcessOrder_Delivered_to_ProcessOrder_Closed :
{
setStatus(fermé);
casser;
}
défaut:
casser;
}
}
privé void initializeStateMachine()
{
currState = StateType.ProcessOrder_New;
État suivant = Type d'État.ST_NOSTATE;
currTransition = TransitionType.TT_NOTRANSITION;
}
privé void runStateMachine()
{
tandis que (vrai)
{
si (currState == StateType.ST_NOSTATE)
{
casser;
}
currTransition = TransitionType.TT_NOTRANSITION;
ÉtatsProc(currState, CommandType.Do);
// puis vérifiez s'il existe une transition valide attribuée après le comportement do
si (nextState == StateType.ST_NOSTATE)
{
casser;
}
si (currTransition != TransitionType.TT_NOTRANSITION)
{
TransitionsProc(currTransition);
}
si (état actuel != état suivant)
{
ÉtatsProc(currState, CommandType.Exit);
StatesProc(nextState, CommandType.Entry);
currState = État suivant;
}
}
```

```
}  
}
```

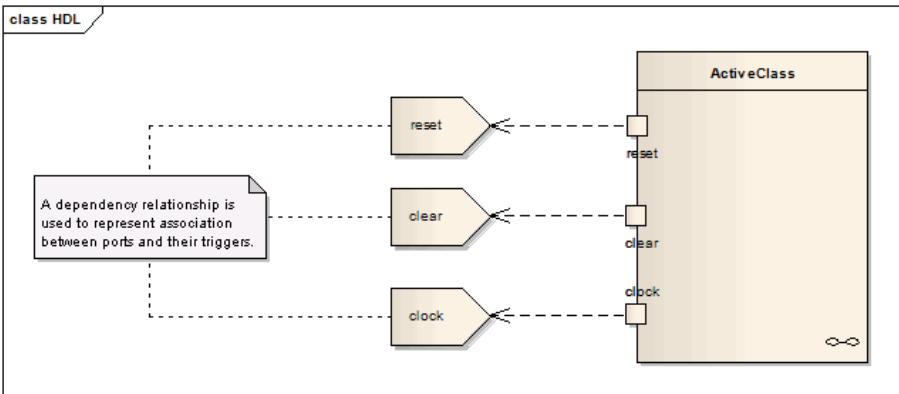
# Modélisation Statemachine pour les HDL

Pour générer efficacement du code HDL (Hardware Description Language) à partir de modèles Statemachine, appliquez les pratiques de conception décrites dans cette rubrique. Les langages de description du matériel incluent VHDL, Verilog et SystemC.

Dans un modèle de Statemachine HDL, vous pouvez vous attendre à :

- Désigner Déclencheurs de Conduite
- Établir une cartographie port-déclencheur
- Ajouter à la logique State Actif

## Opérations

Opération	Description
<p>Désigner Déclencheurs de Conduite</p>	<ul style="list-style-type: none"> <li>• Un Déclencheur « Change » est considéré comme un déclencheur asynchrone si :                     <ul style="list-style-type: none"> <li>- Il y a une transition de l' State actuel de la sous-machine (qui encapsule la logique réelle) qu'elle déclencheurs, et</li> <li>- L' State cible de cette transition a une auto-transition déclenchée par le même Déclencheur</li> </ul> </li> <li>• Déclencheurs Asynchrones doivent être modélisés selon ce motif :                     <ul style="list-style-type: none"> <li>- Le Déclencheur doit être de type Change (spécification : Vrai / Faux)</li> <li>- L' State actif ( State de sous-machine) doit avoir une transition déclenché par cela</li> <li>- L' State cible de la transition déclenchée doit avoir un auto transition avec le même Déclencheur</li> </ul> </li> <li>• Un Déclencheur de type 'Time', qui déclencheurs les transitions vers l'état actif (SubMachine State), est considéré comme l'Horloge ; la spécification de ce déclencheur doit être conforme au langage cible :                     <ul style="list-style-type: none"> <li>- VHDL - front montant / front descendant</li> <li>- Verilog - posedge / negedge</li> <li>- SystemC - positif / négatif</li> </ul> </li> </ul>
<p>Établir une cartographie des déclencheurs de port</p>	<p>Après modélisation avec succès les différents modes de fonctionnement du composant, et les Déclencheurs qui leur sont associés, vous devez associer les Déclencheurs aux Ports du composant.</p> <p>Une relation de dépendance du port au Déclencheur associé est utilisée pour signifier cette association.</p> 

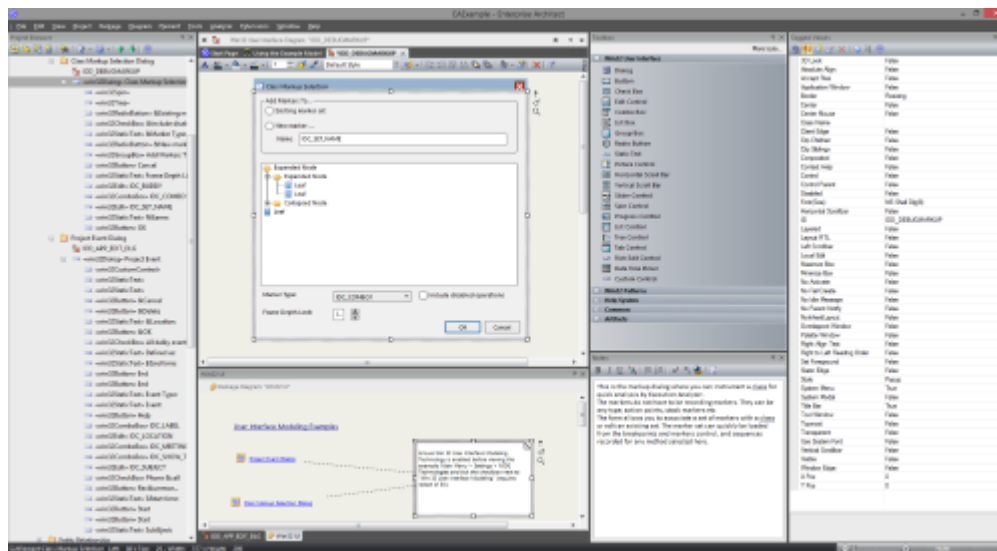
Logique State Actif	La désignation du Déclencheur pilote et l'établissement du mapping Port-Trigger mettent en place les préliminaires nécessaires à l'interprétation efficace des composants matériels. Nous modélisons maintenant la logique réelle Statemachine dans l' State Actif (SubMachine).
---------------------	---

## Notes

- Pour pouvoir générer du code à partir de modèles comportementaux, toutes les constructions comportementales doivent être contenues dans une classe
- Le moteur de génération de code actuel supporte qu'un seul Déclencheur d'horloge par composant



# Boîtes de dialogue Interface Utilisateur Win32



Grâce à la technologie MDG Win32 UI , vous pouvez concevoir des écrans d'interface utilisateur qui s'affichent sous forme de contrôles Win32®. L'interface utilisateur produite peut être utilisée dans n'importe quel script de définition de ressources. Les scripts de définition de ressources, ou fichiers RC, sont une technologie Microsoft qui, comme pour d'autres codes, peut être compilée et les ressources utilisées par les applications de bureau natives. Les écrans ou boîtes de dialogue d'interface utilisateur peuvent être créés à partir de zéro ou rétroconçus. Les modèles d'interface utilisateur peuvent également être rétroconçus à l'aide de la fonction de synchronisation de code (F7). modélisation d'interface se déroule sur diagrammes exactement de la même manière que vous travailleriez avec n'importe quelle technologie dans Enterprise Architect . Un aspect intéressant de la conception Interface Utilisateur dans Enterprise Architect est que les composants peuvent jouer un rôle actif dans la simulation de Statemachines et d'activités, permettant à une simulation d'interagir avec les utilisateurs, à la manière d'un vrai programme !

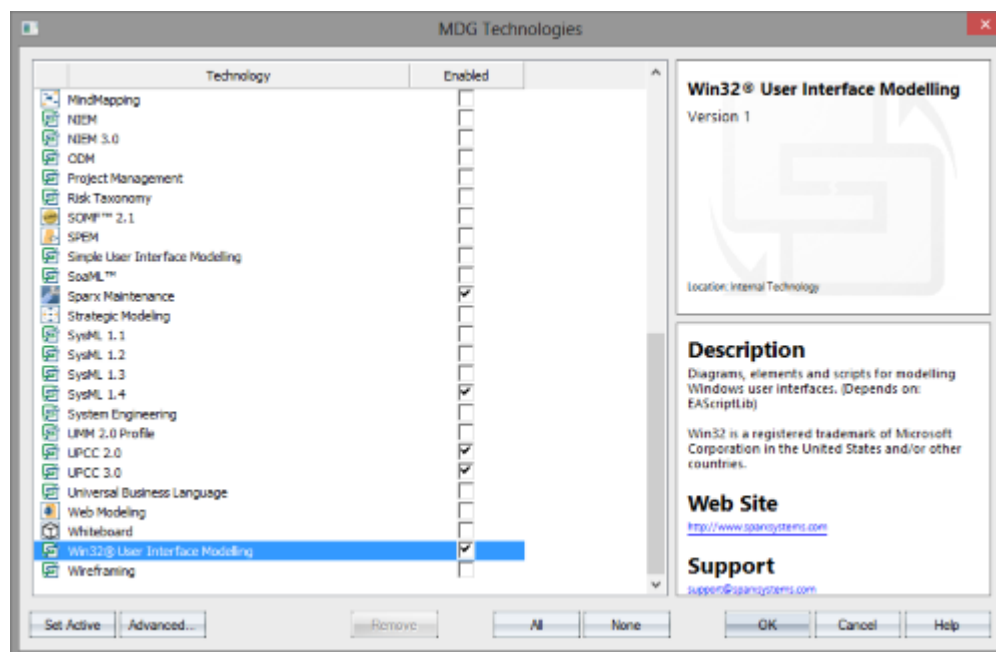
## Accéder

Ruban	Conception > Diagramme > Ajouter Diagramme > Type > Interface Utilisateur Win32
Menu Contexte	Cliquez-droit sur Paquetage   Ajouter Diagramme > Type   Interface Utilisateur Win32
Autre	Menu de la barre de légende de la fenêtre du Navigateur   Nouveau Diagramme   Interface Utilisateur Win32

## Support

La technologie MDG Win32® Interface Utilisateur est disponible dans les éditions Enterprise Architect Professional , Corporate , Unified et Ultimate

## Activation de la technologie Interface Utilisateur Win32



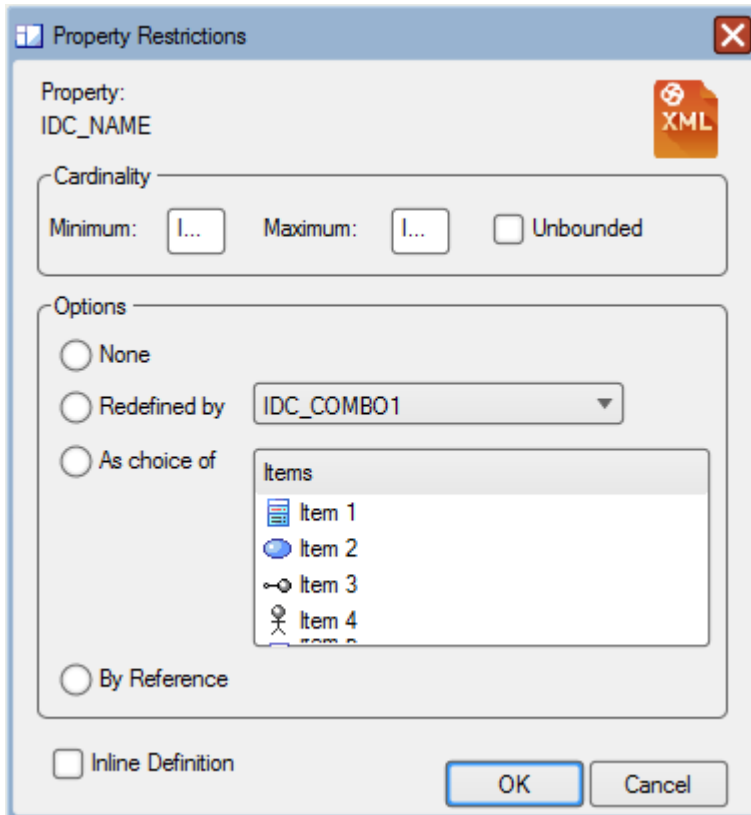
La technologie UI Win32® dans Enterprise Architect est activée ou désactivée à l'aide de la dialogue « MDG Technologies » (sélectionnez l'option de ruban « Spécialiser > Technologies > Gérer la technologie »).

## Technologie par défaut

Vous pouvez définir la technologie UI MDG Win32® comme technologie par défaut active pour accéder directement aux pages de la boîte à outils.

## Modélisation des dialogues UI

L'Interface Utilisateur Win32 MDG Technologie fournit les outils pour vous aider à concevoir une interface utilisateur qui émule étroitement le style visuel et les options disponibles pour les boîtes de dialogue Windows .



### Dialogue Win32

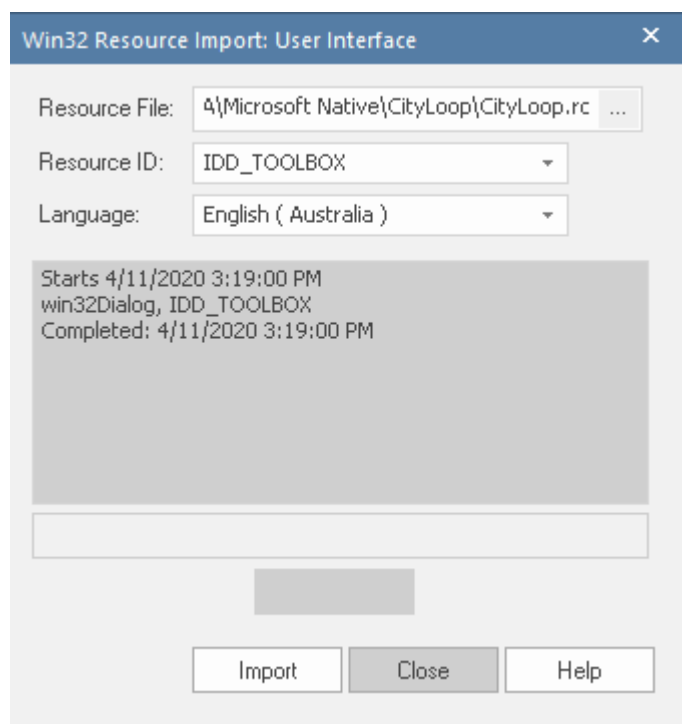
Ces composants d'interface utilisateur sont pris en charge, chacun correspondant à la ressource RC portant le nom équivalent.

Composant	Détails
dialogue win32	L'équivalent des ressources au format RC DIALOG et DIALOGEX.
Texte statique win32	L'équivalent des ressources au format RC LTEXT, RTEXT, CTEXT.
win32Modifier	L'équivalent de la ressource EDITTEXT au format RC.
Bouton win32	L'équivalent du format RC BUTTON, DEFPUSHBUTTON et d'autres ressources.
case à cocher win32	L'équivalent de la ressource CHECKBOX au format RC.
barre de défilement win32H	L'équivalent de la ressource SCROLLBAR au format RC avec le style SBS_HORZ
barre de défilement win32V	L'équivalent de la ressource SCROLLBAR au format RC avec le style SBS_VERT.

Boîte de groupe win32	L'équivalent de la ressource GROUPBOX au format RC.
boîte de dialogue Win32ComboBox	L'équivalent de la ressource COMBOBOX au format RC. Note : Lorsque vous faites glisser l'icône « Combo Box » (de type « Liste déroulante » ou « Liste déroulante ») sur un diagramme , la « poignée de suivi » du milieu de chaque côté de l'élément est blanche, ce qui indique que vous ne pouvez ajuster que la largeur de l'élément. Pour ajuster la hauteur de l'élément ainsi que sa largeur, cliquez sur la partie flèche déroulante de l'image ; la « poignée de suivi » du milieu sur le bord inférieur est maintenant blanche, ce qui indique que vous pouvez faire glisser la base vers le bas pour définir la hauteur virtuelle (la hauteur de l'élément lorsqu'il est développé pour afficher toutes les valeurs possibles dans la liste déroulante).
Liste de listes win32	L'équivalent de la ressource LISTBOX au format RC.
Bouton radio win32	L'équivalent de la ressource RADIOBUTTON au format RC.
Panneau d'onglets win32	L'équivalent de la ressource TABPANE au format RC.
win32Image	L'équivalent de la ressource STATIC au format RC avec le style SS_BITMAP. Le contrôle peut restituer une image lorsqu'il est appliqué à partir de votre modèle. Une image peut être appliquée en la sélectionnant d'abord et en appuyant sur Ctrl+Maj+W pour afficher le Gestionnaire d'images. Ensuite, vous devrez peut-être modifier la valeur de l' ID de ressource dans la Valeur Étiquetée appropriée.
Contrôle personnalisé win32	L'équivalent de la ressource CONTROL au format RC.

## Importer Dialogue unique à partir d'un fichier RC

Vous pouvez rapidement importer un seul dialogue par nom.



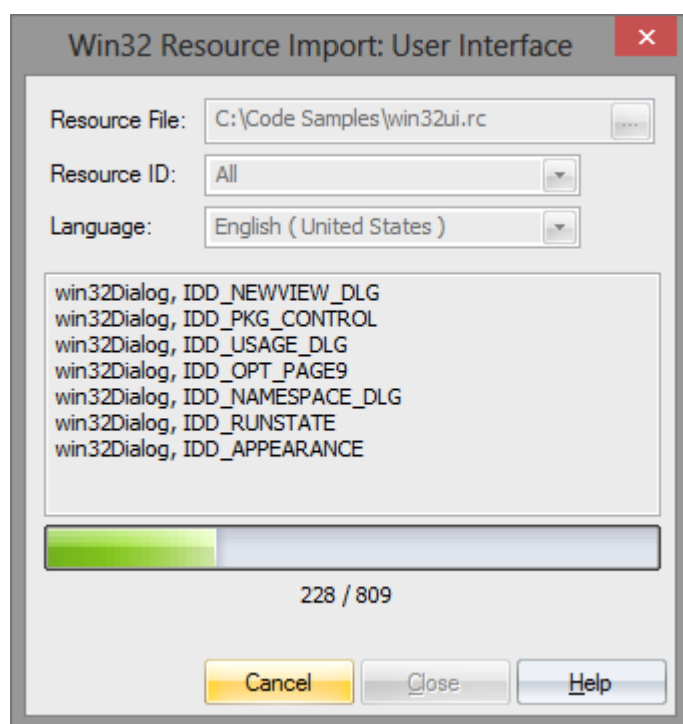
### Accéder

Dans la fenêtre Navigateur , cliquez sur la cible Paquetage .

Ruban	Développer > Code source > Fichiers > Importer un script de ressources
-------	--

## Importer toutes les boîtes de dialogue à partir du fichier RC

Toutes les boîtes de dialogue d'un seul fichier RC peuvent être importées dans votre modèle. Cette image a été capturée une minute après le début de l'importation, moment auquel plus de 200 grandes définitions dialogue avaient été importées.

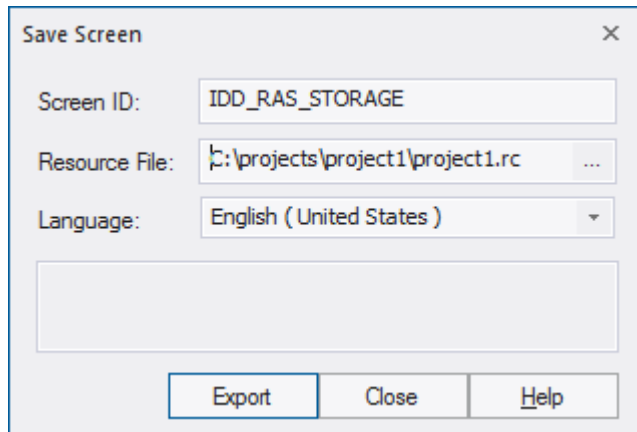


### Accéder

Ruban	Développer > Code source > Fichiers > Importer un script de ressources
-------	--

## Exporter Dialogue vers un fichier RC

Une fois la conception d'un écran modifiée ou une nouvelle créée, vous souhaitez peut-être la récupérer dans le fichier RC que vous avez utilisé pour créer votre application, afin de pouvoir voir à quoi elle ressemble avec des données réelles. Commencez par sélectionner l'élément Win32Dialog dans la fenêtre Navigateur , puis utilisez le ruban pour effectuer la synchronisation.



### Accéder

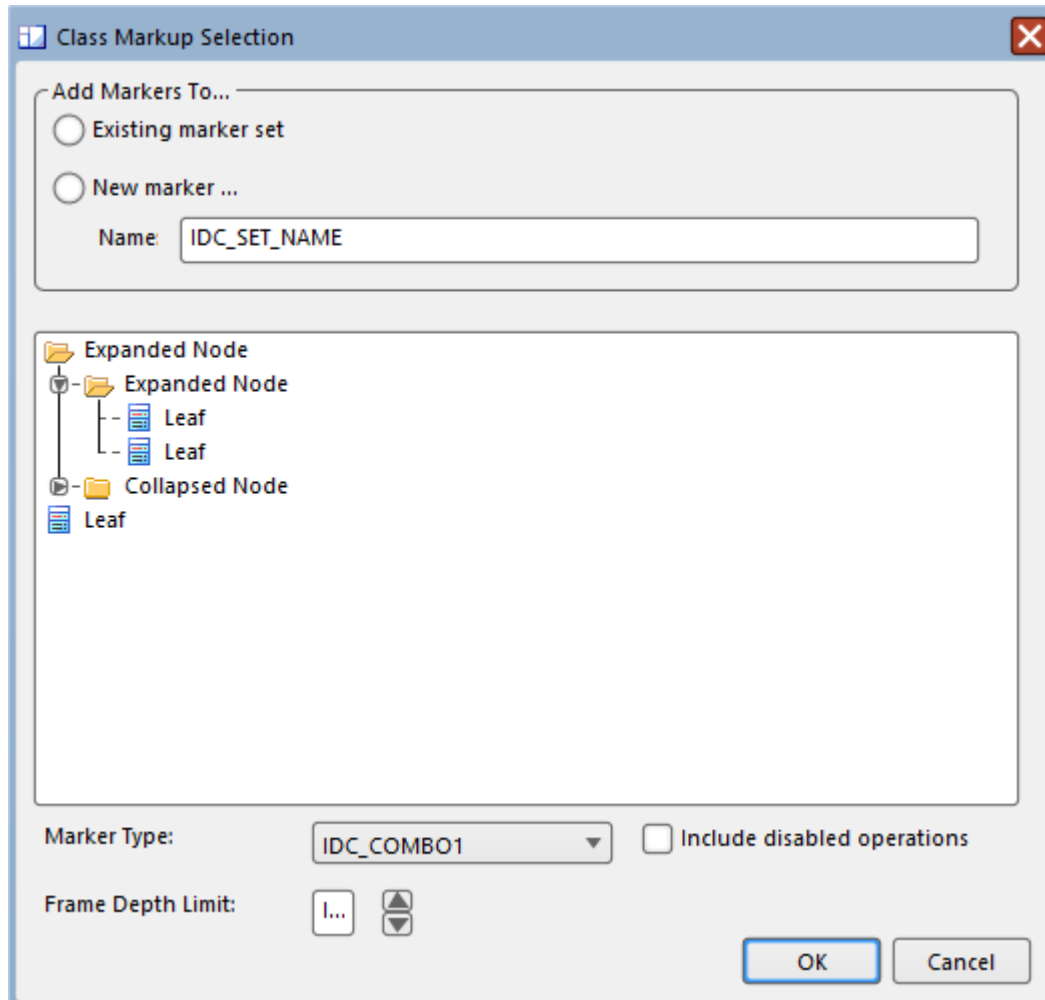
Cliquez sur l'élément win32Dialog.

Ruban	Développer > Code Source > Générer > Générer un seul élément
Raccourcis Clavier	F11

## Concevoir un nouveau Dialogue

Créer une nouvelle dialogue Win32 est simple et principalement visuelle. Vous aurez probablement besoin d'un espace de travail qui affiche :

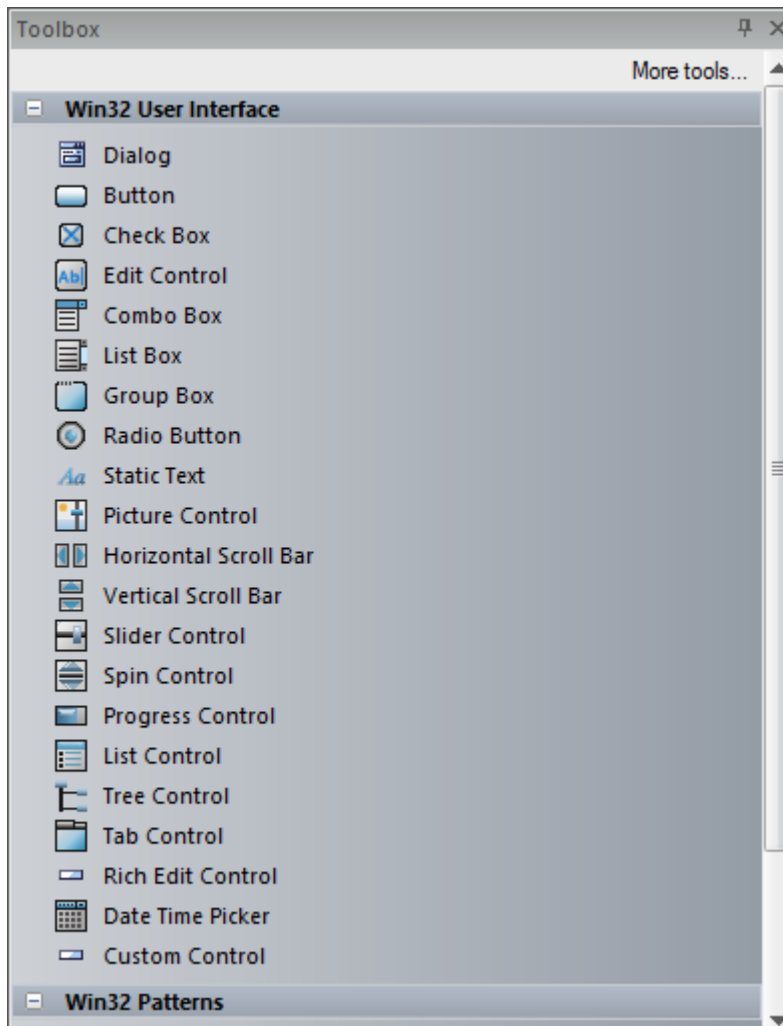
- Le nouveau diagramme (sélectionnez le chemin du ruban 'Conception > Diagramme > Ajouter Diagramme > Interface Utilisateur - Win32 > Interface Utilisateur - Win32')
- La boîte à outils Interface Utilisateur Win32 (sélectionnez l'option de ruban « Conception > Diagramme > Boîte à outils ») et
- L'onglet Valeur Étiquetés de la fenêtre Propriétés



### La boîte à outils UI

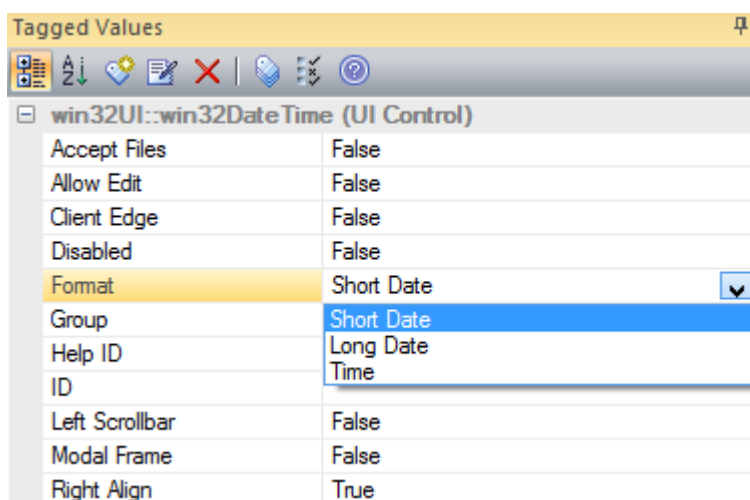
Tous les éléments RC courants peuvent être trouvés dans la boîte à outils UI





## L'onglet Étiquettes

Cet onglet est fourni dans la fenêtre Propriétés et dialogue « Propriétés » d'un objet , et c'est là que toutes les propriétés d'un contrôle peuvent être visualisées et modifiées.



## Utilisation du contrôle d'image

Les images de votre modèle (voir *Gestionnaire d'images*) peuvent être appliquées en sélectionnant le contrôle dans le dialogue et en appuyant sur Ctrl+Maj+W. Vous devrez peut-être saisir la valeur de l' ID de ressource dans la Valeur Étiquetée appropriée.

### Note


- Vous pouvez copier et coller Paquetages dialogue

## Motifs de Gang of Four (GoF)

Un Motif de conception est un gabarit permettant de résoudre des problèmes de conception récurrents. Il se compose d'une série d'éléments et de connecteurs qui peuvent être réutilisés dans un nouveau contexte. L'avantage d'utiliser Motifs est qu'ils ont été testés et affinés dans un certain nombre de contextes et constituent donc généralement des solutions robustes aux problèmes courants. Enterprise Architect fournit le Gang of Four Motifs en tant que MDG Technologie qui peut être chargée dans le référentiel actuel.

Les Motifs Gang of Four (Gof) sont un groupe de vingt-trois Motifs de conception publiés à l'origine dans un livre fondateur intitulé *Design Motifs : Elements of Reusable Object-Oriented Software* ; le terme « Gang of Four » fait référence aux quatre auteurs. Enterprise Architect affiche ces Motifs dans son moteur Motif , vous aidant à visualiser les éléments du Motif et à ajuster le Motif au contexte de votre problème de conception de logiciel.

### Motifs GoF dans Enterprise Architect

Fonctionnalités	Description
Facilités Motif GoF	<p>Les Motifs GoF sont fournis sous la forme de :</p> <ul style="list-style-type: none"><li>• Pages Motifs Comportementale GoF, Motifs Créationnels GoF et Motifs Structurels GoF dans la Boîte à Outils</li><li>• Entrées Motif Gang of Four dans le menu contextuel de la boîte à outils</li></ul> <p><b>Pages de la boîte à outils GoF Motif</b></p> <p>Vous pouvez accéder aux pages « GoF Motif » de la Boîte à Outils en cliquant sur  pour afficher la dialogue « Trouvez Item de Boîte à Outils » et en spécifiant « GoF Motifs » ; ces icônes sont disponibles :</p>



**GoF Behavioral Patterns**

- Chain of Responsibility
- Command
- Interpreter
- Iterator
- Mediator
- Memento
- Observer
- State
- Strategy
- Template Method
- Visitor

**GoF Creational Patterns**

- Abstract Factory
- Builder
- Factory Method
- Prototype
- Singleton

**GoF Structural Patterns**

- Adapter
- Bridge
- Composite
- Decorator
- Facade
- Flyweight
- Proxy

Lorsque vous faites glisser l'un des éléments Motif sur un nouveau diagramme , la dialogue « Ajouter Motif GoF < groupe motif >< type motif > » s'affiche ; si nécessaire, modifiez l'action et/ou la valeur par défaut des éléments composants, puis cliquez sur le bouton OK pour créer un diagramme basé sur le Motif .

## Paramètres de configuration



Vous pouvez définir les options de code par défaut telles que les éditeurs pour chacun des langages de programmation disponibles pour Enterprise Architect et les options spéciales pour la génération ou la rétro-ingénierie du code source. Ces options sont définies selon qu'elles s'appliquent à :

- Tous les utilisateurs du modèle actuel, définis dans la dialogue « Gérer les options Modèle », ou
- Tous les modèles auxquels vous accédez (les autres utilisateurs peuvent définir leurs propres paramètres qui s'appliquent aux mêmes modèles), définis dans la dialogue « Préférences »

Vous pouvez également :

- Pour chaque langage de programmation utilisé dans le modèle, pour tous les utilisateurs travaillant sur le modèle, définissez des classes de collection pour générer du code à partir de connecteurs d'association où le rôle cible a un paramètre de multiplicité supérieur à 1
- Définissez un chemin local pour vous-même, à l'aide de la dialogue « Chemin local » ; ces paramètres s'appliquent à tous les modèles Enterprise Architect auxquels vous accédez
- Définir des macros de langage dans le modèle, qui sont utiles dans la rétro-ingénierie et peuvent être exportées et importées vers le modèle

## Options d'ingénierie du code source

Les options « Ingénierie du code source » s'appliquent aux langages dans lesquels vous générez du code à partir d'Enterprise Architect . Elles sont divisées en options spécifiques au modèle et options spécifiques à l'utilisateur, comme expliqué ici.

### Options spécifiques au modèle

Ces options sont définies dans la dialogue « Gérer les options Modèle ».

#### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source
-------	---

### Types d'options

Type d'option	Détail
Options de génération de code source	Vous pouvez définir un certain nombre de paramètres pour générer du code dans le modèle, tels que la langue par défaut dans laquelle générer du code et le jeu de caractères Unicode pour la génération de code.
Options - Durée de vie Object	Vous pouvez configurer diverses options concernant la durée de vie Object .
Options de langage de code	Pour chacun des langages de code supporte Enterprise Architect , vous pouvez définir les options spécifiques au modèle et définir les classes de collection requises.

### Options spécifiques à l'utilisateur

Ces options sont définies dans la dialogue « Préférences ».

#### Accéder

Dans la dialogue « Préférences », cliquez sur « Ingénierie du code source » dans la liste de gauche.

Ruban	Démarrer > Apparence > Préférences > Préférences
Raccourcis Clavier	Ctrl+F9

## Types d'options

Type d'option	Détail
Options de génération de code source	Vous pouvez définir un certain nombre de paramètres pour générer du code dans n'importe quel modèle auquel vous accédez sous le même ID utilisateur.
Éditeurs de Code	Voici les options permettant d'accéder et de configurer l'éditeur de code source.
Attributs / Opérations	Utilisez ces options pour configurer les attributs et les opérations.
Options de langage de code	Pour chacun des langages de code supporte Enterprise Architect , vous pouvez définir les options spécifiques à l'utilisateur qui s'appliquent à tout modèle auquel vous accédez sous votre ID utilisateur.

## Options de génération de code


Lorsque vous générez du code pour votre modèle, vous pouvez définir certaines options. Celles-ci incluent :

- La langue par défaut
- S'il faut générer des méthodes pour les interfaces implémentées
- Les options Unicode pour la génération de code

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source
-------	---

### Configurer les options de génération de code

Option	Action
Toujours synchroniser avec le fichier existant (recommandé)	Sélectionnez le bouton radio pour synchroniser le code importé avec un fichier existant.
Remplacer (écraser) le fichier source existant	Sélectionnez le bouton radio pour écraser le fichier source existant avec le code importé.
Types de composants	Cliquez sur ce bouton pour ouvrir la dialogue « Importer les types de composants » afin de configurer l'importation des types de composants.
Langue par défaut pour la génération de code	Cliquez sur la flèche déroulante et sélectionnez la langue par défaut pour la génération de code.
Nom DDL Gabarits	Cliquez sur le bouton  pour définir les noms gabarit pour les gabarits Primary Key , de contrainte unique, Foreign Key et de nom d'index Foreign Key .
Nom par défaut pour l'attribut associé	Type un nom par défaut à générer à partir des attributs importés.
Générer des méthodes pour les interfaces implémentées	Cochez la case pour indiquer que les méthodes sont générées pour les interfaces implémentées.
Page de code pour l'édition de la source	Cliquez sur la flèche déroulante et sélectionnez le format d'intégration de caractères Unicode approprié à appliquer.

### Notes



- Il est utile de configurer ces paramètres, car ils servent de valeurs par défaut pour toutes les classes du modèle ; vous pouvez remplacer la plupart d'entre eux pour chaque classe à l'aide des paramètres personnalisés (à partir de la dialogue « Génération de code »).

## Types de composants d'importation

À l'aide de la dialogue « Importer les types de composants », vous pouvez configurer les éléments que vous souhaitez créer pour les fichiers de n'importe quelle extension trouvés lors de l'importation d'un répertoire de code source.

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source : Types de composants
-------	---

### Définir les types de composants d'importation

Option	Action
Extension	Type le nom de l'extension pour un type de composant.
Type	Cliquez sur la flèche déroulante et sélectionnez le type de composant.
Stéréotype	Type n'importe quel nom de stéréotype qui identifie davantage un composant de ce type.
Liste des composants	Répertorie les types de composants actuellement définis.
Sauvegarder	Cliquez sur ce bouton pour enregistrer la définition du composant et l'ajouter à la liste des composants.
Nouveau	Cliquez sur ce bouton pour effacer le dialogue afin de pouvoir définir un nouveau type de composant.
Supprimer	Cliquez sur ce bouton pour supprimer le type de composant sélectionné de la liste des composants.

### Notes

- Vous pouvez transporter ces types de composants d'importation entre les modèles, en utilisant les options de ruban « Paramètres > Modèle > Transférer > Exporter les données de référence » et « Importer les données de référence ».

## Options du code source

Vous pouvez définir un large éventail d'options pour générer du code dans les modèles avec lesquels vous travaillez. Celles-ci incluent :

- Comment formater le code généré
- Comment réagir à certains événements lors de la génération de code
- Faut-il générer un diagramme à partir du code ?

### Accéder

Dans la dialogue « Préférences », sélectionnez l'option « Ingénierie du code source »

Ruban	Démarrer > Apparence > Préférences > Préférences
Raccourcis Clavier	Ctrl+F9

### Configurer les options de génération de code

Champ	Action
Envelopper les longues lignes de commentaires à	Type le nombre de caractères à autoriser dans une ligne de commentaire avant de renvoyer le texte à la ligne suivante.
Diagramme Disposition automatique à l'importation	Cliquez sur la flèche déroulante et sélectionnez si et quand un diagramme est automatiquement généré lors de l'importation de code.
Type de Diagramme Disposition par défaut	Cliquez sur la flèche déroulante et sélectionnez le type disposition à appliquer aux diagrammes générés à partir du code.
Les fichiers de sortie utilisent à la fois CR et LF	Cochez la case pour inclure les retours chariot et les sauts de ligne ; définissez cette option en fonction du système d'exploitation actuellement utilisé, car le code peut ne pas s'afficher correctement.
Invite lors de la synchronisation (inversion)	Cochez la case pour afficher une prompt lorsque la synchronisation se produit.
Supprimer les sauts de ligne des commentaires lors de l'importation	Cochez la case pour supprimer les sauts de ligne des sections commentées lors de l'importation.
Générer automatiquement les noms de rôle lors de la création du code	Cochez la case pour générer des noms de rôle lors de la création de code.
Ne pas générer de membres lorsque la direction de l'association est « Non	Cochez la case pour empêcher la génération de membres si la direction de l'association n'est pas spécifiée.

spécifiée »	
Créer des dépendances pour les retours d'opération et les types de paramètres	Cochez la case pour générer des dépendances pour les retours d'opération et les types de paramètres.
Commentaires : Générer	Cochez la case pour générer des commentaires.
Commentaires : Inversé	Cochez la case pour générer des commentaires inversés.
Supprimer les préfixes lors de la génération des propriétés Get/Set	Type les préfixes, séparés par des points-virgules, utilisés dans vos conventions de nommage de variables, à supprimer dans les fonctions get/set correspondantes des variables.
Traiter comme des suffixes	Cochez la case pour utiliser les préfixes définis dans le champ « Supprimer les préfixes lors de la génération des propriétés Get/Set » comme suffixes.
Nom d'attribut en majuscule pour Propriétés	Cochez la case pour mettre en majuscule les noms d'attributs des propriétés.
Utiliser « Is » pour la propriété booléenne Get()	Cochez la case pour utiliser le mot-clé Is pour la propriété booléenne Get().

## Notes

- Il est utile de configurer ces paramètres, car ils servent de valeurs par défaut pour toutes les classes du modèle ; vous pouvez remplacer la plupart d'entre eux pour chaque classe à l'aide des paramètres personnalisés (à partir de la dialogue « Génération de code »).

## Options - Éditeurs de Code


Vous pouvez accéder aux options de l'éditeur de code source via la page « DDL » de la dialogue « Préférences ». Sur cette page, vous pouvez configurer les options de l'éditeur interne d' Enterprise Architect , ainsi que l'éditeur par défaut pour les scripts DDL. Vous pouvez configurer des éditeurs externes pour les langages de code sur chaque page d'options de langage.



### Accéder

Dans la dialogue « Préférences », sélectionnez l'option « Ingénierie du code source > Éditeurs de Code ».

Ruban	Démarrer > Apparence > Préférences > Préférences
Raccourcis Clavier	Ctrl+F9

### Options


Option	Action
Éditeur DDL	La valeur par défaut est vide, indiquant que l'éditeur de code Enterprise Architect est l'éditeur DDL utilisé. Vous pouvez sélectionner un autre éditeur par défaut si nécessaire ; cliquez sur le bouton  pour rechercher et sélectionner l'éditeur DDL requis. Le nom de l'éditeur s'affiche alors dans le champ « Éditeur DDL ».
Base de données par défaut	Cliquez sur la flèche déroulante et sélectionnez la base de données par défaut à utiliser.
Moteur de stockage MySQL	Cliquez sur la flèche déroulante et sélectionnez le moteur de stockage MySQL à utiliser.
Utiliser l'éditeur intégré si aucun éditeur externe n'est défini	Cochez la case pour utiliser l'éditeur intégré pour le code dans n'importe quelle langue si aucun éditeur externe n'est défini pour cette langue dans les options spécifiques à l'utilisateur.
Afficher les numéros de ligne	Cochez la case pour afficher les numéros de ligne dans l'éditeur.
Afficher l'arborescence de la structure	Cochez la case pour afficher un arbre avec les résultats de l'analyse du fichier ouvert (si le fichier est analysé avec succès).
Rétro-ingénierie automatique lors de l'enregistrement du fichier	Si vous cochez cette case, appuyez sur Ctrl+S pour enregistrer dans l'éditeur de code source pour effectuer automatiquement une rétro-ingénierie du code de la même manière que le bouton Enregistrer la source et resynchroniser la classe.
N'analysez pas les fichiers plus grands que	Cliquez sur la flèche déroulante et sélectionnez la limite supérieure de la taille du fichier pour l'analyse. La définition de cette option empêche la diminution des performances due à



	l'analyse de fichiers très volumineux.
Police , style et mise en évidence de la syntaxe	Cliquez sur le bouton  pour afficher la dialogue « Propriétés de la langue de l'éditeur », dans laquelle vous pouvez définir les propriétés de la langue de l'éditeur globales et spécifiques à la langue.
Configurer les associations de fichiers Enterprise Architect	Cliquez sur le bouton  pour afficher la dialogue « Définir des associations pour un programme » et sélectionnez les extensions de fichier pour les fichiers que vous souhaitez ouvrir via le gestionnaire de documents Enterprise Architect .

## Propriétés de la langue de l'éditeur

À l'aide de la dialogue « Propriétés du langage de l'éditeur », vous pouvez spécifier les propriétés de mise en surbrillance de la syntaxe pour n'importe lequel des langages de programmation supporte par Enterprise Architect lors de l'installation.


### Accéder

Dans la dialogue « Préférences », sélectionnez l'option « Ingénierie du code source | Éditeurs de Code » et cliquez sur le bouton  à côté de « Options de mise en évidence de la syntaxe ».

Ruban	Démarrer > Apparence > Préférences > Préférences, sélectionnez l'option « Source Code Engineering   Éditeurs de Code » > cliquez sur le bouton  à côté de « Options de mise en surbrillance de la syntaxe »
Autre	Dans la fenêtre Éditeur de Code , cliquez sur l'icône de la barre d'outils    Options de mise en surbrillance de la syntaxe

### Options

Panneau	Description
Panneau de langue	<p>Le panneau sur la gauche de le dialogue répertorie les langues pour lesquelles vous pouvez définir des propriétés.</p> <p>En haut de la liste se trouvent trois options non linguistiques :</p> <ul style="list-style-type: none"> <li>• (Thème sombre) - attribue un arrière-plan sombre aux champs de propriété et au panneau de code dans l'écran de l'éditeur de code (vous pouvez appliquer une couleur différente à des propriétés spécifiques)</li> <li>• (Thème clair) - attribue un arrière-plan pâle aux champs de propriété et au panneau de code dans l'écran de l'éditeur de code (vous pouvez appliquer une couleur différente à des propriétés spécifiques) Vous pouvez également définir les thèmes d'arrière-plan dans la dialogue « Apparence de l'application »</li> <li>• (Global) fournit des propriétés que vous pouvez définir pour tous les langages de programmation ; cependant, vous pouvez réinitialiser une propriété globale à une valeur différente pour un langage particulier, dans les propriétés spécifiques à ce langage La réinitialisation d'une propriété globale pour une langue n'affecte pas valeur de cette propriété pour les autres langues</li> </ul> <p>Cliquez sur la langue souhaitée dans la liste pour afficher les propriétés de cette langue :</p> <ul style="list-style-type: none"> <li>• Propriétés affichées en gras indiquent qu'il s'agit du niveau le plus élevé auquel cette propriété peut être définie (pour la plupart des options de langage autres que « Global », il s'agit en fait du seul point auquel la propriété est définie)</li> <li>• Propriétés affichées en police normale sont généralement les propriétés globales que vous pouvez réinitialiser uniquement pour la langue actuelle</li> </ul>

<p>Panneau Propriétés</p>	<p>Faites défiler les catégories de propriétés et les propriétés individuelles de la langue. Vous pouvez réduire et développer les catégories selon vos besoins, à l'aide de la zone d'extension située à côté du nom de la catégorie (☐).</p> <p>Lorsque vous cliquez sur le nom d'une propriété, une explication de cette propriété s'affiche dans le panneau en bas à droite de le dialogue .</p> <p>Pour définir une propriété, cliquez sur le champ valeur suivant le nom de la propriété ; selon le type de propriété, soit le champ est activé pour l'édition directe, soit une flèche déroulante ou un bouton  s'affiche (comme décrit pour l'onglet 'Tags' de la fenêtre Propriétés ) afin que vous puissiez sélectionner les valeurs pour définir la propriété.</p> <p>Sélectionnez ou saisissez les valeurs requises.</p> <p>Utilisez les icônes de la barre d'outils pour :</p> <ul style="list-style-type: none"> <li>• Enregistrez vos modifications dans les propriétés</li> <li>• Réinitialiser tous les champs de propriétés aux paramètres par défaut fournis avec Enterprise Architect</li> <li>• Réinitialiser le champ de style actuel au paramètre par défaut (non activé pour les champs sans style)</li> </ul>
<p>Affecter des touches aux macros</p>	<p>Dans la catégorie 'Macros' des propriétés, vous pouvez attribuer des combinaisons de touches (Ctrl+Alt+&lt;n&gt;) à des macros de codage que vous avez créées vous-même dans la ' Visionneuse de code source '.</p> <p>Lorsque vous cliquez sur le bouton Parcourir dans un champ « Macro » sélectionné, la dialogue « Ouvrir une macro » s'affiche ; cette dialogue répertorie les macros existantes et, si une combinaison de touches a été attribuée à une macro, quelle est cette combinaison de touches.</p> <p>Cliquez sur le nom de la macro et sur le bouton Ouvrir pour attribuer les touches sélectionnées à la macro.</p>

## Notes

- Vous ne pouvez actuellement pas définir de propriétés pour les langues supplémentaires que vous incluez via une MDG Technologie
- Vous pouvez redimensionner cette dialogue , si nécessaire



## Options - Durée de vie Object

Vous pouvez utiliser ces options pour configurer divers paramètres de durée de vie Object tels que :

- Définition des détails du constructeur lors de la génération de code
- Spécifier s'il faut créer un constructeur de copie
- Définition des détails du destructeur

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > Durée de vie Object
-------	---

### Options

Option	Action
Constructeur	<p>Si nécessaire, cochez les cases pour spécifier qu'un constructeur est généré et (pour C++) que le constructeur est en ligne.</p> <p>Cliquez sur la flèche déroulante et sélectionnez la visibilité appropriée du constructeur par défaut : Privé, Protégé ou Public.</p>
Constructeur de copie	<p>Si nécessaire, cochez les cases pour spécifier qu'un constructeur de copie est généré et (pour C++) que le constructeur de copie est en ligne.</p> <p>Cliquez sur la flèche déroulante et sélectionnez la visibilité appropriée du constructeur de copie par défaut : Privé, Protégé ou Public.</p>
Destructeur	<p>Si nécessaire, cochez les cases pour spécifier qu'un destructeur est généré et (pour C++) que le destructeur est en ligne et/ou virtuel.</p> <p>Cliquez sur la flèche déroulante et sélectionnez la visibilité appropriée du destructeur par défaut - Privé, Protégé ou Public.</p>

## Options - Attribut/Opérations

L'utilisation des attributs et des opérations peut être configurée de plusieurs manières. Vous pouvez définir des options pour :

- Supprimer les attributs du modèle non inclus dans le code lors de la synchronisation inverse
- Supprimer les méthodes de modèle non incluses dans le code lors de la synchronisation inverse
- Supprimer le code des fonctionnalités contenues dans le modèle lors de la synchronisation directe
- Supprimer les associations et agrégations de modèles qui correspondent à des attributs non inclus dans le code lors de la synchronisation inverse
- Définir si les corps des méthodes sont inclus et enregistrés dans le modèle lors de la rétro-ingénierie
- Créer fonctionnalités en succession rapide, en effaçant la fenêtre Propriétés lorsque vous cliquez sur « Enregistrer » afin de pouvoir saisir un autre nom fonctionnalité

Vous configurez ces options sur la page « Attribut/Opérations » de la dialogue « Préférences ».

### Accéder

Dans la dialogue « Préférences », sélectionnez l'option « Ingénierie du code source > Attribut/Opérations ».

Ruban	Démarrer > Apparence > Préférences > Préférences
Raccourcis Clavier	Ctrl+F9

### Options

Champ	Action
Lors de la synchronisation inverse, supprimez les attributs du modèle qui ne figurent pas dans le code	Cochez la case pour indiquer que lors de la synchronisation inverse, les attributs du modèle qui ne sont pas inclus dans le code sont automatiquement supprimés du modèle.
Lors de la synchronisation inverse, supprimez les associations de modèles qui ne figurent pas dans le code	Cochez la case pour indiquer que lors de la synchronisation inverse, les associations dans le modèle qui ne sont pas incluses dans le code sont automatiquement supprimées du modèle.
Lors de la synchronisation inverse, supprimez les méthodes de modèle qui ne figurent pas dans le code	Cochez la case pour indiquer que lors de la synchronisation inverse, les méthodes du modèle qui ne sont pas incluses dans le code sont automatiquement supprimées du modèle.
Inclure les corps de méthode dans le modèle lors de la rétro-ingénierie	Cochez la case pour indiquer que dans le code de rétro-ingénierie, les corps de méthode du code sont inclus dans votre modèle.
Après avoir enregistré, resélectionnez l'élément	Cochez la case pour indiquer qu'après l'enregistrement d'un attribut ou d'une opération, la définition des propriétés continue d'afficher les détails de la

modifié	fonctionnalité sélectionnée. Si cette option est désélectionnée, cela indique que les champs de la définition des propriétés seront effacés afin que vous puissiez saisir immédiatement un autre nom et des détails d'attribut ou d'opération.
Lors de la synchronisation directe, prompt à supprimer fonctionnalités de code non présentes dans le modèle	Cochez la case pour indiquer que, lors de la synchronisation directe, la dialogue « Synchroniser l'élément < nom paquetage >.< nom de l'élément> » s'affiche, afin que vous puissiez ignorer, réaffecter ou supprimer fonctionnalités du code qui ne sont pas dans le modèle.

## Conventions Modélisation



La synchronisation entre les modèles UML et le code de programmation est réalisée à l'aide d'un ensemble de conventions modélisation (mappings) entre les constructions UML et la syntaxe du code de programmation. Il est conseillé à l'ingénieur logiciel de se familiariser avec ces conventions afin de travailler avec le processus de génération de code pour les langages de programmation qu'il souhaite cibler. Il existe toute une gamme de constructions utilisées, notamment les éléments, fonctionnalités, les connecteurs, les extrémités de connecteur, les stéréotypes et Valeur Étiquetés. Le nouveau venu aura besoin d'un peu de temps pour se familiariser avec ces conventions, mais après peu de temps, il saura traduire sans effort entre le code de programmation et les constructions UML.

### Langues prises en charge

Langue
Scénario Action
Ada 2012 (éditions Unified et Ultimate )
C
C#
C++
Delphes
Java
PHP
Python
SystemC (éditions Unified et Ultimate )
Verilog (éditions Unified et Ultimate )
VHDL (éditions Unified et Ultimate )
Visual Basic
Visual Basic .NET

## Notes

Enterprise Architect intègre un certain nombre d'indicateurs de visibilité ou de valeurs de portée pour les langues prises en charge ; il s'agit notamment de :

- Toutes les langues - Public (+), Protégé (#) et Privé (-)
- Java - Paquetage (~)
- Delphi - Publié (^)
- C# - Interne (~), Interne protégé (^)
- ActionScript - Interne (~)
- VB.NET - Ami (~), ami protégé (^)
- PHP - Paquetage (~)
- Python - Paquetage (~)
- C - Paquetage (~)
- C++ - Paquetage (~)

## Conventions ActionScript

Enterprise Architect supporte l'ingénierie round d'ActionScript 2 et 3, lorsque ces conventions sont utilisées.

### Stéréotypes

Stéréotype	S'applique à
littéral	Opération Correspond à : une méthode littérale référencée par une variable.
obtenir une propriété	Opération Correspond à : une propriété « lecture ».
ensemble de propriétés	Opération Correspond à : une propriété « écriture ».

### Valeur Étiquetés

Étiquette	S'applique à
nom_attribut	Opération avec la propriété stéréotype get ou property set Correspond à : Le nom de la variable derrière cette propriété.
dynamique	Classe ou interface Correspond à : Le mot-clé « dynamique ».
final	ActionScript 3 : opération Correspond à : Le mot-clé « final ».
intrinsèque	ActionScript 2 : classe Correspond à : Le mot-clé « intrinsèque ».
espace de noms	ActionScript 3 : classe, interface, attribut, opération Correspond à : l'espace de noms de l'élément actuel.
outrepasser	ActionScript 3 : opération Correspond à : Le mot-clé « override ».
prototype	ActionScript 3 : Attribut Correspond à : Le mot-clé « prototype ».
repos	ActionScript 3 : Paramètre Correspond à : Le paramètre de repos (...)

## Conventions communes

- Les qualificateurs Paquetage (ActionScript 2) et Paquetages (ActionScript 3) sont générés lorsque le Paquetage actuel n'est pas une racine d'espace de noms
- Un type non spécifié est modélisé comme « var » ou un champ « Type » vide

## Conventions d'ActionScript 3

- La propriété Is Leaf d'une classe correspond au mot-clé sealed
- Si une étiquette d'espace de noms est spécifiée, elle remplace la portée spécifiée

## Conventions Ada 2012

Enterprise Architect supporte l'ingénierie round retour d'Ada 2012, où ces conventions sont utilisées.

### Stéréotypes

Stéréotype	S'applique à
Paquet ada	Classe Correspond à : une spécification Paquetage dans Ada 2012 sans enregistrement balisé.
Procédure ada	Classe Correspond à : une spécification de procédure dans Ada 2012.
déléguer	Opération Correspond à : Accès à un sous-programme.
énumération	Classe intérieure Correspond à : un type énuméré.
structure	Classe intérieure Correspond à : une définition d'enregistrement.
définition de type	Classe intérieure Correspond à : une définition de type, une définition de sous-type, une définition de type d'accès, un renommage.

### Valeur Étiquetés

Étiquette	S'applique à
Aspect	Classe interne avec typedef stéréotypé Opération Correspond à : Spécification d'aspect (Précondition et Postcondition de type de sous-programme 'invariant', sous-type 'prédicat').
Type d'unité instanciée	Classe interne avec typedef stéréotypé Correspond à : Le type de l'unité instanciée ( Paquetage / Procédure / Fonction).
Est-ce que Access	Paramètre Correspond à : Déterminer si le paramètre est une variable d'accès.
Est aliasé	Paramètre de fonction



	Correspond à : Paramètre de fonction aliasé.
Discriminant	Classe interne avec typedef stéréotypé Correspond à : Le discriminant du type.
Type de pièce	Classe interne avec typedef stéréotypé Correspond à : Le type de pièce (« renomme » ou « nouveau »).
Type	Classe interne avec typedef stéréotypé Correspond à : Si « Valeur » = « Sous-type », définir « sous-type » Si « Valeur » = « Accès », définissez « type d'accès ».

## Autres conventions

- Type approprié de fichiers sources : fichier de spécification Ada, .ads
- Ada 2012 importe Paquetages définis comme classe <<adaPackage>> ou classe, en fonction des paramètres des options Ada 2012
- Un Paquetage dans le fichier de spécifications Ada est importé en tant que Classe s'il contient un enregistrement balisé, dont le nom est régi par les options « Utiliser le nom de classe pour l'enregistrement balisé » et « Nom d'enregistrement balisé alternatif » ; tous les attributs définis dans cet enregistrement balisé sont absorbés en tant qu'attributs de la classe
- Une procédure/fonction dans un fichier de spécification Ada est considérée comme la fonction membre de la classe si son premier paramètre satisfait les conditions spécifiées dans les options « Style de paramètre de référence », « Ignorer le nom du paramètre de référence » et « Nom du paramètre de référence »
- L'option « Définir une référence pour un enregistrement balisé », si elle est activée, crée un type de référence pour la classe, dont le nom est déterminé par l'option « Nom Type de référence » ; par exemple :

```
Bonjour le monde.annonces
```

```
paquetage HelloWorld est
```

```
le type HelloWorld est un enregistrement tagué
```

```
Att1 : Naturel ;
```

```
Att3 : Integer ;
```

```
fin d'enregistrement;
```

```
-- Fonctions publiques
```

```
fonction MyPublicFunction (P: HelloWorld) renvoie String ;
```

```
procédure MyPublicFunction (P1 : entrée sortie HelloWorld ; AFlag : Boolean) ;
```

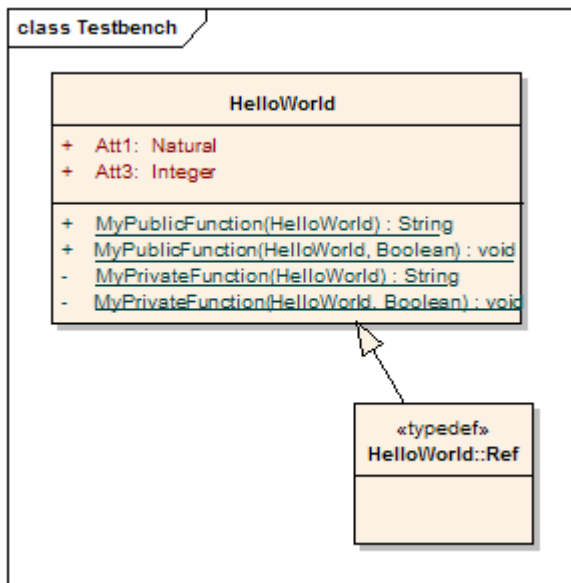
```
privé
```

```
-- Fonctions privées
```

```
fonction MyPrivateFunction (P: HelloWorld) renvoie String ;
```

```
procédure MyPrivateFunction (P1 : entrée sortie HelloWorld ; AFlag : Boolean) ;
```

```
fin HelloWorld;
```



## Notes

- support Ada 2012 est disponible dans les éditions Unified et Ultimate d' Enterprise Architect

## Conventions C

Enterprise Architect supporte l'ingénierie round retour de C, où ces conventions sont utilisées :

### Stéréotype

Stéréotype	S'applique à
énumération	Classe intérieure Correspond à : un type énuméré.
structure	Classe intérieure Correspond à : un type « struct ».
Attribut	Une structure de mot-clé dans la définition de variable.
définition de type	Classe intérieure Correspond à : une instruction « typedef », où le parent est le nom du type d'origine.
union	Classe intérieure Correspond à : Un type d'union.
Attribut	Une union de mots clés dans la définition de variable.

### Valeur Étiquetés

Étiquette	S'applique à
anonyme	Classe contenant également le typedef Valeur Étiquetée Correspond à : Le nom de cette classe étant défini uniquement par l'instruction typedef.
champ de bits	Attribut Correspond à : la taille, en bits, autorisée pour le stockage de cet attribut.
Emplacement du corps	Opération Correspond à : l'emplacement vers lequel le corps de la méthode est généré ; les valeurs attendues sont header, classDec ou classBody.
définition de type	Classe avec un stéréotype autre que « typedef » Correspond à : Cette classe étant définie dans une instruction « typedef ».
typeSynonymes	Classe Correspond à : le nom « typedef » et/ou les champs de ce type.

## Génération de code C pour Modèle UML

UML	Code C
Une classe	Une paire de fichiers C (.h + .c) Notes : le nom du fichier est le même que le nom de la classe
Opération (publique et protégée)	Déclaration de fonction dans le fichier .h et définition dans le fichier .c Notes :
Opération (privée)	Définition de fonction dans le fichier .c uniquement Notes :
Fonctionnement (statique)	Définition de fonction dans le fichier .c uniquement Notes : les fonctions statiques n'apparaîtront que dans le fichier .c, quelle que soit leur portée.
Attribut (public et protégé)	Définition de variable dans le fichier .h Notes :
Attribut (privé)	Définition de variable dans le fichier .c Notes :
Classe intérieure (sans stéréotype)	(N / A) Notes : Cette classe interne serait ignorée

## Capturez valeur #define à générer dans le code C

Par exemple, #define PI 3.14.

Étape	Processus
1	Ajoutez un attribut à la classe, avec le nom = PI et la valeur initiale = 3,14.
2	Dans le panneau des propriétés de la page « Attributes », mettez à jour les champs « Statique » et « Const ».
3	Sur l'onglet ' Valeur Étiquetés ' de la page ' Attributes ', ajoutez une étiquette appelée 'define' avec la valeur True.

## Notes

- Des conventions distinctes s'appliquent à la programmation orientée Object en C



# Programmation orientée Object en C

Dans Enterprise Architect , vous appliquez un certain nombre de conventions pour la programmation orientée objet en C. Pour configurer le système afin de support de C, vous devez définir l'option « Support orientée Object » sur True dans la page « Spécifications C » de la dialogue « Préférences ».

## Stéréotypes

Stéréotype	S'applique à
énumération	Classe Correspond à : un type énuméré.
structure	Classe Correspond à : un type « struct ».
Attribut	Une structure de mot-clé dans la définition de variable.
définition de type	Classe Correspond à : une instruction « typedef », où le parent est le nom du type d'origine.
union	Classe Correspond à : Un type d'union.
Attribut	Une union de mots clés dans la définition de variable.

## Valeur Étiquetés

Étiquette	S'applique à
anonyme	Classe avec le stéréotype « enumeration », « struct » ou « union » Correspond à : Le nom de cette classe étant défini uniquement par l'instruction typedef.
Emplacement du corps	Opération Correspond à : l'emplacement où le corps de la méthode est généré ; les valeurs attendues sont « header », « classDec » ou « classBody ».
définir	Attribut Correspond à : instruction « #define ».
définition de type	Classe avec le stéréotype « enumeration », « struct » ou « union » Correspond à : Cette classe étant définie dans une instruction « typedef ».

## Génération de code C orienté objet pour Modèle UML

L'idée de base de l'implémentation d'une classe UML dans le code C est de regrouper les variables de données (attributs UML) dans un type de structure ; cette structure est définie dans un fichier .h afin qu'elle puisse être partagée par d'autres classes et par le client qui y fait référence.

Une opération dans une classe UML est implémentée dans le code C en tant que fonction ; le nom de la fonction doit être un nom entièrement qualifié composé du nom de l'opération, ainsi que du nom de la classe pour indiquer que l'opération est destinée à cette classe.

Un délimiteur (spécifié dans l'option « Délimiteur Namespace » sur la page « Spécifications C ») est utilisé pour joindre le nom de la classe et le nom de la fonction (opération).

La fonction dans le code C doit également avoir un paramètre de référence à l'object Class - vous pouvez modifier les options « Référence comme paramètre d'opération », « Style de paramètre de référence » et « Nom du paramètre de référence » sur la page « Spécifications C » pour supporter ce paramètre de référence.

## Limitations de la programmation orientée objet en C

- Pas de mappage de portée pour un attribut : un attribut dans une classe UML est mappé à une variable de structure dans le code C et sa portée (privée, protégée ou publique) est ignorée
- Actuellement, une classe interne est ignorée : si une classe UML est la classe interne d'une autre classe UML, elle est ignorée lors de la génération du code C
- valeur initiale est ignorée : la valeur initiale d'un attribut dans une classe UML est ignorée dans le code C généré

## Conventions C#

Enterprise Architect supporte l'ingénierie round -retour de C# , où ces conventions sont utilisées.

### Stéréotypes

Stéréotype	S'applique à
énumération	Classe Correspond à : un type énuméré.
événement	Opération Correspond à : Un événement.
extension	Opération Correspond à : une méthode d'extension de classe, représentée dans le code par un paramètre « this » dans la signature.
indexeur	Opération Correspond à : une propriété agissant comme un index pour cette classe.
partiel	Opération Correspond à : Le mot clé « partial » sur une opération.
propriété	Opération Correspond à : une propriété contenant éventuellement du code de lecture et d'écriture.
enregistrer	Classe Correspond à : un type « enregistrement ».
structure_enregistrement	Classe Correspond à : Un type « record struct ».
structure	Classe Correspond à : un type « struct ».

### Valeur Étiquetés

Étiquette	S'applique à
argumentNom	Opération avec extension du stéréotype Correspond à : Le nom donné à ce paramètre.



nom_attribut	Opération avec une propriété ou un événement stéréotypé Correspond à : Le nom de la variable derrière cette propriété ou cet événement.
nom de la classe	Opération avec extension du stéréotype Correspond à : la classe à laquelle cette méthode est ajoutée.
constante	Attribut Correspond à : le mot-clé const.
définition	Opération avec stéréotype partiel Correspond à : s'il s'agit de la déclaration de la méthode ou de la définition.
déléguer	Opération Correspond à : Le mot-clé « délégué ».
enumType	Opération avec propriété de stéréotype Correspond à : le type de données sous lequel la propriété est représentée.
expressionCorps	Opération, Opération avec propriété stéréotype ou indexeur Correspond à : « Vrai » si le « Code de comportement » provient d'un membre de fonction du corps de l'expression.
extensionAttribut	Opération avec extension du stéréotype. Correspond à : L'attribut donné à ce paramètre.
externe	Opération Correspond à : Le mot-clé « extern ».
fixé	Attribut Correspond à : Le mot-clé « fixe ».
générique	Opération Correspond à : Les paramètres génériques pour cette opération.
contraintes génériques	Classe ou interface basée sur un modèle, opération avec étiquette « générique » Correspond à : Les contraintes sur les paramètres génériques de ce type ou de cette opération.
Outils	Opération Correspond à : Le nom de la méthode implémentée, y compris le nom de l'interface.
ImplémenteExplicit	Opération Correspond à : La présence du nom de l'interface source dans cette déclaration de méthode.
initialiseur	Opération Correspond à : une liste d'initialisation de constructeur.
nouveau	Classe, interface, fonctionnement

	Correspond à : Le mot-clé « nouveau ».
outrepasser	Opération Correspond à : Le mot-clé « override ».
paramètres	Paramètre Correspond à : une liste de paramètres utilisant le mot-clé « params ».
partiel	Classe, Interface Correspond à : Le mot-clé « partial ».
Initialiseur de propriété	Opération avec propriété de stéréotype Correspond à : un initialiseur de propriété.
lecture seule	Opération, classe <<struct>> Correspond à : Le mot-clé « readonly ».
Paramètres de position	Classe <<record>> Correspond à : Le paramètre de position dans la définition d'enregistrement.
réf	Opération, classe <<struct>> Correspond à : Le mot-clé « ref ».
scellé	Opération Correspond à : Le mot-clé « scellé ».
statique	Classe Correspond à : Le mot-clé « static ».
dangereux	Classe, Interface, Fonctionnement Correspond à : Le mot clé « unsafe ».
virtuel	Opération Correspond à : Le mot-clé « virtuel ».
écriture seule	Opération avec propriété de stéréotype Correspond à : Cette propriété définit uniquement le code « écriture ».

## Autres conventions

- Namespaces sont générés pour chaque Paquetage sous une racine d'espace de noms
- La propriété Const d'un attribut correspond au mot clé readonly, tandis que l' étiquette const correspond au mot clé const
- La valeur de inout pour la propriété Kind d'un paramètre correspond au mot-clé ref
- La valeur de out pour la propriété Kind d'un paramètre correspond au mot-clé out
- Les classes partielles peuvent être modélisées comme deux classes distinctes avec l' étiquette partielle

- La propriété Is Leaf d'une classe correspond au mot-clé sealed

## Conventions C++

Enterprise Architect supporte l'ingénierie round retour de C++, y compris les extensions Managed C++ et C++/CLI, où ces conventions sont utilisées.

### Stéréotypes

Stéréotype	S'applique à
énumération	Classe Correspond à : un type énuméré.
ami	Opération Correspond à : Le mot-clé « ami ».
obtenir une propriété	Opération Correspond à : une propriété « lecture ».
ensemble de propriétés	Opération Correspond à : une propriété « écriture ».
structure	Classe Correspond à : un type « struct ».
définition de type	Classe Correspond à : une instruction « typedef », où le parent est le nom du type d'origine.
alias	Classe Correspond à une déclaration « Alias », où le parent est le nom du type d'origine.
union	Classe Correspond à : Un type d'union.

### Valeur Étiquetés

Étiquette	S'applique à
afx_msg	Opération Correspond à : le mot-clé afx_msg.
anonyme	Classe contenant également le typedef Valeur Étiquetée Correspond à : Le nom de cette classe étant défini uniquement par l'instruction typedef.

nom_attribut	Opération avec la propriété stéréotype get ou property set Correspond à : Le nom de la variable derrière cette propriété.
champ de bits	Attribut Correspond à : la taille, en bits, autorisée pour le stockage de cet attribut.
Emplacement du corps	Opération Correspond à : l'emplacement vers lequel le corps de la méthode est généré ; les valeurs attendues sont header, classDec ou classBody.
rappel	Opération Correspond à : une référence à la macro CALLBACK.
constpr	Attribut et fonctionnement Correspond à : le mot-clé constexpr.
explicite	Opération Correspond à : Le mot-clé « explicite ».
initialiseur	Opération Correspond à : une liste d'initialisation de constructeur.
en ligne	Attribut et fonctionnement Correspond à : le mot-clé « inline » et la génération en ligne de la définition de la variable membre et du corps de la méthode.
mutable	Attribut Correspond à : Le mot-clé « mutable ».
portée	Classe avec énumération de stéréotypes Correspond à : soit le mot-clé « class » soit le mot-clé « struct ».
lance	Opération Correspond à : les exceptions levées par cette méthode.
définition de type	Classe avec un stéréotype autre que « typedef » Correspond à : Cette classe étant définie dans une instruction « typedef ».
typeSynonymes	Classe Correspond à : le nom « typedef » et/ou les champs de ce type.
volatil	Opération Correspond à : Le mot-clé « volatile ».

## Autres conventions

- Namespaces sont générés pour chaque Paquetage sous une racine d'espace de noms

- Les attributs par référence correspondent à un pointeur vers le type spécifié
- La propriété Transient d'un attribut correspond au mot clé volatile
- La propriété Abstract d'un attribut correspond au mot-clé virtual
- La propriété Const d'une opération correspond au mot-clé const, spécifiant un type de retour constant
- La propriété Is Query d'une opération correspond au mot-clé const, spécifiant que la méthode ne modifie aucun champ
- La propriété Pure d'une opération correspond à une méthode virtuelle pure utilisant la syntaxe "= 0"
- La propriété Fixed d'un paramètre correspond au mot-clé const

## Conventions C++ gérées

Ces conventions sont utilisées pour les extensions managées de C++ antérieures à C++/CLI. Pour configurer le système afin de générer du C++ managé, vous devez modifier la version C++ dans les options C++.

### Stéréotypes

Stéréotype	S'applique à
propriété	Opération Correspond à : Le mot-clé ' <code>__property</code> '.
obtenir une propriété	Opération Correspond à : le mot-clé ' <code>__property</code> ' et une propriété lue.
ensemble de propriétés	Opération Correspond à : le mot-clé ' <code>__property</code> ' et une propriété 'write'.
référence	Classe Correspond à : Le mot-clé « <code>__gc</code> ».
valeur	Classe Correspond à : Le mot-clé « <code>__value</code> ».

### Valeur Étiquetés

Étiquette	S'applique à
Type géré	Classe avec référence stéréotypée, valeur ou énumération ; Interface Correspond à : Le mot clé utilisé dans la déclaration de ce type ; les valeurs attendues sont « <code>class</code> » ou « <code>struct</code> ».

### Autres conventions

- Les étiquettes typedef et anonymes du C++ natif ne sont pas prises en charge
- La propriété Pure d'une opération correspond au mot clé `__abstract`

## Conventions C++/CLI

Ces conventions sont utilisées pour modélisation des extensions C++/CLI vers C++. Pour configurer le système afin de générer du C++/CLI managé, vous devez modifier la version C++ dans les options C++.

### Stéréotypes

Stéréotype	S'applique à
événement	Opération Description : définit un événement pour fournir l'accès au gestionnaire d'événements pour cette classe.
propriété	Opération, Attribut Description : Il s'agit d'une propriété contenant éventuellement du code de lecture et d'écriture.
référence	Classe Description : Correspond au mot-clé « ref class » ou « ref struct ».
valeur	Classe Description : Correspond au mot-clé « valeur class » ou « valeur struct ».

### Valeur Étiquetés

Étiquette	S'applique à
nom_attribut	Opération avec une propriété ou un événement stéréotypé Description : Le nom de la variable derrière cette propriété ou cet événement.
générique	Opération Description : Définit les paramètres génériques pour cette opération.
contraintes génériques	Classe ou interface modélisée, opération avec étiquette générique Description : Définit les contraintes sur les paramètres génériques pour cette opération.
initonly	Attribut Description : Correspond au mot-clé « initonly ».
littéral	Attribut Description : Correspond au mot-clé littéral.
Type géré	Classe avec référence stéréotypée, valeur ou énumération ; Interface Description : Correspond soit au mot-clé « class » soit au mot-clé « struct ».



## Autres conventions

- Les étiquettes typedef et anonymes ne sont pas utilisées
- Les stéréotypes de propriété get/property set ne sont pas utilisés
- La propriété Pure d'une opération correspond au mot-clé abstract

## Conventions Delphi

Enterprise Architect supporte l'ingénierie round de Delphi, où ces conventions sont utilisées :

### Stéréotypes

Stéréotype	S'applique à
constructeur	Opération Correspond à : Un constructeur.
destructeur	Opération Correspond à : Un destructeur.
interface d'affichage	Classe, Interface Correspond à : une interface de répartition.
énumération	Classe Correspond à : un type énuméré.
métaclass	Classe Correspond à : un type de métaclass.
object	Classe Correspond à : Un type object .
opérateur	Opération Correspond à : Un opérateur.
obtenir une propriété	Opération Correspond à : une propriété « lecture ».
ensemble de propriétés	Opération Correspond à : une propriété « écriture ».
structure	Classe Correspond à : un type d'enregistrement.

### Valeur Étiquetés

Étiquette	S'applique à
nom_attribut	Opération avec la propriété stéréotype get ou property set Correspond à : Le nom de la variable derrière cette propriété.

surcharge	Opération Correspond à : Le mot-clé « surcharge ».
outrepasser	Opération Correspond à : Le mot-clé « override ».
emballé	Classe Correspond à : Le mot-clé « packed ».
propriété	Classe Correspond à : une propriété ; voir <i>Delphi Propriétés</i> pour plus d'informations.
réintroduire	Opération Correspond à : Le mot-clé « réintroduire ».

## Autres conventions

- La propriété Static d'un attribut ou d'une opération correspond au mot-clé 'class'
- La propriété Fixed d'un paramètre correspond au mot-clé 'const'
- La valeur de inout pour la propriété Kind d'un paramètre correspond au mot-clé 'Var'
- La valeur de out pour la propriété Kind d'un paramètre correspond au mot-clé 'Out'

## Conventions Java

Enterprise Architect supporte l'ingénierie round -retour de Java - y compris les extensions AspectJ - lorsque ces conventions sont utilisées.

### Stéréotypes

Stéréotype	S'applique à
annotation	Interface Correspond à : un type d'annotation.
Constructeur Compact	Opération Correspond à : Un constructeur canonique compact pour l'enregistrement.
enregistrer	Classe Correspond à : un type d'enregistrement.
défaut	Opération Correspond à : Le mot clé « default ».
énumération	Attributs dans une énumération stéréotypée de classe Correspond à : une option énumérée, distinguée des autres attributs qui n'ont pas de stéréotype.
énumération	Classe Correspond à : un type énuméré.
opérateur	Opération Correspond à : Un opérateur.
obtenir une propriété	Opération Correspond à : une propriété « lecture ».
ensemble de propriétés	Opération Correspond à : une propriété « écriture ».
statique	Classe ou interface Correspond à : Le mot-clé « static ».

### Valeur Étiquetés

Étiquette	S'applique à

annotations	Rien Correspond à : Les annotations sur la fonctionnalité de code actuelle.
arguments	Attribut avec énumération de stéréotypes Correspond à : Les arguments qui s'appliquent à cette valeur énumérée.
nom_attribut	Opération avec la propriété stéréotype get ou property set Correspond à : Le nom de la variable derrière cette propriété.
dynamique	Classe ou interface Correspond à : Le mot-clé « dynamique ».
générique	Opération Correspond à : Les paramètres génériques de cette opération.
non scellé	Classe, Interface Correspond à : Le mot-clé « non scellé ».
permis	Classe, Interface Correspond à : L'expression « permet ».
Liste de paramètres	Paramètre Correspond à : une liste de paramètres avec la syntaxe ....
scellé	Classe, Interface Correspond à : Le mot-clé « scellé ».
En-tête d'enregistrement	<<record>>Class Correspond à : l'en-tête d'enregistrement de la définition d'enregistrement.
lance	Opération Correspond à : les exceptions levées par cette méthode.
transitoire	Attribut Correspond à : Le mot-clé « transient ».

## Autres conventions

- Les instructions Paquetage sont générées lorsque le Paquetage actuel n'est pas une racine d'espace de noms
- La propriété Const d'un attribut ou d'une opération correspond au mot-clé final
- La propriété Transient d'un attribut correspond au mot clé volatile
- La propriété Fixed d'un paramètre correspond au mot clé final

## Conventions d'AspectJ

Ce sont les conventions utilisées pour prendre en charge les extensions AspectJ pour Java.

### Stéréotypes

Stéréotype	S'applique à
conseil	Opération Correspond à : Un conseil dans un aspect AspectJ.
aspect	Classe Correspond à : Un aspectJ aspect.
coupe en pointe	Opération Correspond à : un « pointcut » dans un aspect AspectJ.

### Valeur Étiquetés

Étiquette	S'applique à
nom de la classe	Attribut ou opération dans un aspect stéréotypé de classe Correspond à : les classes auxquelles appartient ce membre intertype AspectJ.

### Autres conventions

- Les spécifications d'un pointcut sont incluses dans le champ « Comportement » de la méthode

## Conventions PHP

Enterprise Architect supporte l'ingénierie round de PHP 4 et 5, où ces conventions sont utilisées.

### Stéréotypes

Stéréotype	S'applique à
trait	Classe Correspond à : Un « trait ».
obtenir une propriété	Opération Correspond à : une propriété « lecture ».
ensemble de propriétés	Opération Correspond à : une propriété « écriture ».

### Valeur Étiquetés

Étiquette	S'applique à
nom_attribut	Opération avec la propriété stéréotype get ou property set Correspond à : Le nom de la variable derrière cette propriété.
final	Opérations en PHP 5 Correspond à : Le mot-clé « final ».

### Conventions communes

- Un type non spécifié est modélisé comme var
- Méthodes renvoyant une référence sont générées en définissant le Type de retour sur var\*
- Les paramètres de référence sont générés à partir de paramètres avec le paramètre Kind défini sur inout ou out

### Conventions PHP 5

- Le modificateur de classe final correspond à la propriété Is Leaf
- Le modificateur de classe abstrait correspond à la propriété Abstract
- L'indication du type de paramètre est prise en charge en définissant le Type d'un paramètre
- La valeur de inout ou out pour la propriété Kind d'un paramètre correspond à un paramètre de référence





# Conventions Python

Enterprise Architect supporte l'ingénierie round -retour de Python, où ces conventions sont utilisées.

## Stéréotypes

Stéréotype	Correspond à
TypeAlias	Classe Correspond à : Alias Type explicite

## Valeur Étiquetés

Étiquette	S'applique à
asynchrone	Opération Correspond à : Le mot clé « async » dans la définition de fonction.
Décorateurs	Classe, Opération Correspond à : Les décorateurs appliqués à cet élément dans la source.

## Autres conventions

- Les membres Modèle avec une portée privée correspondent aux membres du code avec deux traits de soulignement au début
- Attributs ne sont générés que lorsque la valeur initiale n'est pas vide
- Tous les types sont rétro-conçus comme var

## Conventions SystemC

Enterprise Architect supporte l'ingénierie aller-retour de SystemC, où ces conventions sont utilisées.

### Stéréotypes

Stéréotype	S'applique à
déléguer	Méthode Correspond à : Un délégué.
énumération	Classe intérieure Correspond à : un type d'énumération.
ami	Méthode Correspond à : Une méthode amie.
propriété	Méthode Correspond à : une définition de propriété.
sc_ctor	Méthode Correspond à : un constructeur SystemC.
module_sc	Classe Correspond à : un module SystemC.
sc_port	Attribut Correspond à : Un port.
signal_sc	Attribut Correspond à : Un signal.
structure	Classe intérieure Correspond à : une structure ou une union.

### Valeur Étiquetés

Étiquette	S'applique à
gentil	Attribut (Port) Correspond à : Type de port (cadencé, fifo, maître, esclave, résolu, vecteur).
mode	Attribut (Port) Correspond à : Mode port (entrée, sortie, entrée-sortie).

remplace	Méthode Correspond à : La liste d'héritage d'une déclaration de méthode.
lancer	Méthode Correspond à : La spécification d'exception d'une méthode.

## Autres conventions





- SystemC hérite également de la plupart des stéréotypes et Valeur Étiquetés du C++

## Pages de la boîte à outils SystemC

Pour modéliser une conception SystemC, faites glisser ces icônes sur un diagramme à partir de la page « Constructions SystemC » de la boîte à outils Diagramme .

Page	Icône
SystèmeC	Module Action : définit un module SystemC. Un élément de classe stéréotypé sc_module.
Fonctionnalités de SystemC	Port Action : définit un port SystemC. Un attribut stéréotypé sc_port.

## Accéder

Ruban	Conception > Diagramme > Toolbox :  > Spécifiez 'SystemC Constructs' dans les boîtes de dialogue ' Trouvez Item de Boîte à Outils '
Raccourcis Clavier	Ctrl+Shift+3 :  > Spécifiez 'SystemC Constructs' dans la dialogue ' Trouvez Item de Boîte à Outils '
Autre	Vous pouvez afficher ou masquer la boîte à outils Diagramme en cliquant sur les icônes  ou  à l'extrémité gauche de la barre de légende en haut de la Vue Diagramme .

## Conventions VB.NET

Enterprise Architect supporte l'ingénierie aller-retour de Visual Basic.NET, où ces conventions sont utilisées. Les versions antérieures de Visual Basic sont prises en charge en tant que langage différent.

### Stéréotypes

Stéréotype	S'applique à
événement	Opération Correspond à : une déclaration d'événement.
importer	Opération Correspond à : une opération à importer depuis une autre bibliothèque.
module	Classe Correspond à : Un module.
opérateur	Opération Correspond à : une définition de surcharge d'opérateur.
partiel	Opération Correspond à : Le mot clé « partial » sur une opération.
propriété	Opération Correspond à : une propriété contenant éventuellement du code de lecture et d'écriture.

### Valeur Étiquetés

Étiquette	S'applique à
Alias	Opération avec importation de stéréotypes Correspond à : l'alias de cette opération importée.
nom_attribut	Opération avec propriété de stéréotype Correspond à : Le nom de la variable derrière cette propriété.
Jeu de caractères	Opération avec importation de stéréotypes Correspond à : la clause de jeu de caractères pour cette importation - l'une des valeurs « Ansi », « Unicode » ou « Auto ».
déléguer	Opération Correspond à : Le mot-clé « délégué ».

enumTag	Opération avec propriété de stéréotype Correspond à : le type de données sous lequel cette propriété est représentée.
Poignées	Opération Correspond à : la clause « handles » sur cette opération.
Outils	Opération Correspond à : la clause « implements » sur cette opération.
Lib	Opération avec importation de stéréotypes Correspond à : la bibliothèque d'où provient cette importation.
Doit remplacer	Opération Correspond à : le mot-clé « MustOverride ».
Rétrécissement	Opération avec l'opérateur stéréotype Correspond à : Le mot-clé « Rétrécissement ».
Non remplaçable	Opération Correspond à : le mot-clé « NotOverrideable ».
Surcharges	Opération Correspond à : Le mot-clé « surcharges ».
Remplacements	Opération Correspond à : Le mot-clé « overrides ».
tableau de paramètres	Paramètre Correspond à : une liste de paramètres utilisant le mot-clé « ParamArray ».
partiel	Classe, Interface Correspond à : Le mot-clé « partial ».
lecture seule	Opération avec propriété de stéréotype Correspond à : Cette propriété définit uniquement le code « lecture ».
ombres	Classe, interface, fonctionnement Correspond à : Le mot-clé « Ombres ».
Commun	Attribut Correspond à : Le mot-clé « Partagé ».
Élargissement	Opération avec l'opérateur stéréotype Correspond à : Le mot-clé « Élargissement ».
écriture seule	Opération avec propriété de stéréotype Correspond à : Cette propriété définit uniquement le code « écriture ».

## Autres conventions

- Namespaces sont générés pour chaque Paquetage sous une racine d'espace de noms
- La propriété Is Leaf d'une classe correspond au mot-clé NotInheritable
- La propriété Abstract d'une classe correspond au mot-clé MustInherit
- La propriété Static d'un attribut ou d'une opération correspond au mot-clé Shared
- La propriété Abstract d'une opération correspond au mot clé MustOverride
- La valeur de in pour la propriété Kind d'un paramètre correspond au mot-clé ByVal
- La valeur de inout ou out pour la propriété Kind d'un paramètre correspond au mot-clé ByRef

# Conventions Verilog

Enterprise Architect supporte l'ingénierie aller-retour de Verilog, où ces conventions sont utilisées.

## Stéréotypes

Stéréotype	S'applique à
asynchrone	Méthode Correspond à : Un processus simultané.
énumération	Classe intérieure Correspond à : un type d'énumération.
initialiseur	Méthode Correspond à : un processus d'initialisation.
module	Classe Correspond à : Un module.
partie	Attribut Correspond à : une instanciation de composant.
port	Attribut Correspond à : Un port.
synchrone	Méthode Correspond à : Un processus séquentiel.

## Valeur Étiquetés

Étiquette	S'applique à
gentil	Attribut (signal) Correspond à : Le type de signal (tel que registre, bus).
mode	Attribut (Port) Correspond à : Le mode Port (in, out, inout).
Plan du port	Attribut (partie) Correspond à : la carte générique/port du composant instancié.
sensibilité	Méthode Correspond à : La liste de sensibilité d'un processus séquentiel.

taper	Attribut Correspond à : la plage ou le type de valeur d'un attribut.
-------	---

## Pages de la boîte à outils Verilog

Accès : « Conception > Diagramme > Boîte à outils : icône « Hamburger » > HDL | Constructions Verilog »

Faites glisser ces icônes sur un diagramme pour modéliser une conception Verilog.

Item	Action
Module	Définit un module Verilog. Un élément de classe stéréotypé de module.
Énumération	Définit un Type énuméré. Un élément d'énumération.
Port	Définit un port Verilog. Un attribut stéréotypé de port.
Partie	Définit une instanciation de composant Verilog. Un attribut partiellement stéréotypé.
Attribut	Définit un attribut.
Procédure	Définit un processus Verilog : <ul style="list-style-type: none"> <li>• Concurrent - Une méthode stéréotypée asynchrone</li> <li>• Séquentiel - Une méthode synchrone-stéréotypée</li> <li>• Initializer - Une méthode stéréotypée d'initialisation</li> </ul>



## Conventions VHDL

Enterprise Architect supporte l'ingénierie aller-retour de VHDL, lorsque ces conventions sont utilisées.

### Stéréotypes

Stéréotype	S'applique à
architecture	Classe Correspond à : Une architecture .
asynchrone	Méthode Correspond à : Un processus asynchrone.
configuration	Méthode Correspond à : Une configuration.
énumération	Classe intérieure Correspond à : un type énuméré.
entité	Interface Correspond à : Une entité.
partie	Attribut Correspond à : une instanciation de composant.
port	Attribut Correspond à : Un port.
signal	Attribut Correspond à : Une déclaration de signal.
structure	Classe intérieure Correspond à : une définition d'enregistrement.
synchrone	Méthode Correspond à : Un processus synchrone.
définition de type	Classe intérieure Correspond à : une définition de type ou de sous-type.

### Valeur Étiquetés





Étiquette	S'applique à
-----------	--------------

estGénérique	Attribut (port) Correspond à : La déclaration « port » dans une interface générique.
estSubType	Classe interne (typedef) Correspond à : une définition de sous-type.
gentil	Attribut (signal) Correspond à : Le type de signal (tel que « registre », « bus »).
mode	Attribut (Port) Correspond à : Le mode Port ('in', 'out', 'inout', 'buffer', 'linkage').
plan du port	Attribut (partie) Correspond à : la carte générique/port du composant instancié.
sensibilité	Méthode (synchrone) Correspond à : La liste de « sensibilité » d'un processus synchrone.
taper	Classe interne (typedef) Correspond à : L'indication « type » d'une déclaration « type ».
typeNameSpace	Attribut (partie) Correspond à : l'espace de noms « type » du composant instancié.

## Pages de la boîte à outils VHDL

### Accéder

Pour modéliser une conception VHDL, faites glisser les icônes depuis les pages de la boîte à outils VHDL et déposez-les sur votre diagramme .

Ruban	Conception > Diagramme > Toolbox :  > Spécifiez 'VHDL Constructs' dans la dialogue ' Trouvez Item de Boîte à Outils '
Raccourcis Clavier	Ctrl+Shift+3 :  > Spécifiez 'VHDL Constructs' dans la dialogue ' Trouvez Item de Boîte à Outils '
Autre	Vous pouvez afficher ou masquer la boîte à outils Diagramme en cliquant sur les icônes  ou  à l'extrémité gauche de la barre de légende en haut de la Vue Diagramme .

## Page de la boîte à outils VHDL

Item	Action
Architecture	Définit une architecture à associer à une entité VHDL. Un élément de classe stéréotypé en termes d'architecture.
Entité	Définit une entité VHDL pour contenir les définitions de port. Un élément d'interface stéréotypé d'entité.
Énumération	Définit un Type énuméré. Un élément d'énumération.
Structure	Définit un enregistrement VHDL. Un élément de classe stéréotypé de type struct.
Définition de type	Définit un type ou sous-type VHDL. Un élément de classe stéréotypé typedef.

## Page Boîte à Outils Fonctionnalités VHDL

Item	Action
Partie	Définit une instanciation de composant VHDL. Un attribut partiellement stéréotypé.
Port	Définit un port VHDL. Un attribut stéréotypé du port.
Signal	Définit un signal VHDL. Un attribut stéréotypé du signal.
Procédure	Définit un processus VHDL : <ul style="list-style-type: none"> <li>• Concurrent - Une méthode stéréotypée asynchrone</li> <li>• Séquentiel - Une méthode synchrone-stéréotypée</li> <li>• Configuration - Une méthode stéréotypée de configuration</li> </ul>

## Conventions de Visual Basic

Enterprise Architect supporte l'ingénierie round de Visual Basic 5 et 6, où ces conventions sont utilisées. Visual Basic .NET est pris en charge comme langage différent.

### Stéréotypes

Stéréotype	S'applique à
mondial	Attribut Correspond à : Le mot-clé « Global ».
importer	Opération Correspond à : une opération à importer depuis une autre bibliothèque.
obtenir une propriété	Opération Correspond à : Une propriété 'get'.
ensemble de propriétés	Opération Correspond à : un « ensemble » de propriétés.
propriété à louer	Opération Correspond à : Une propriété « let ».
avec des événements	Attribut Correspond à : le mot-clé « WithEvents ».

### Valeur Étiquetés

Étiquette	S'applique à
Alias	Opération avec importation de stéréotypes Correspond à : l'alias de cette opération importée.
nom_attribut	Opération avec propriété de stéréotype get, propriété set ou propriété let Correspond à : Le nom de la variable derrière cette propriété.
Lib	Opération avec importation de stéréotypes Correspond à : la bibliothèque d'où provient cette importation.
Nouveau	Attribut Correspond à : Le mot-clé « nouveau ».

## Autres conventions

- La valeur de in pour la propriété Kind d'un paramètre correspond au mot-clé ByVal
- La valeur de inout ou out pour la propriété Kind d'un paramètre correspond au mot-clé ByRef

## Options de langue

Vous pouvez configurer différentes options pour déterminer la manière dont Enterprise Architect gère un langage particulier lors de la génération et de la rétro-ingénierie du code. Ces options sont spécifiques à :

- Votre ID utilisateur, pour tous les modèles ou
- Le modèle dans lequel ils sont définis, pour tous les utilisateurs

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > <nom de la langue> Paramètres > Modèle > Options > Ingénierie du code source > <nom de la langue>
Raccourcis Clavier	Ctrl+F9 ( dialogue « Préférences »)

### Langues prises en charge

Langue
Scénario Action
Ada 2012 (dans les éditions Unified et Ultimate d' Enterprise Architect )
ArcGIS
Norme ANSI C
C#
C++
Delphes
Java
PHP
Python
SystèmeC
Verilog (éditions Unified et Ultimate )
VHDL (éditions Unified et Ultimate )

---

Visual Basic
Visual Basic .NET

## Options ActionScript - Utilisateur


Si vous avez l'intention de générer du code ActionScript à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications ActionScript » de la dialogue « Préférences » pour :

- Spécifier le répertoire source par défaut
- Spécifier l'éditeur pour le code ActionScript

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > ActionScript
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code ActionScript. Cochez cette case pour désactiver support du code ActionScript.
Options pour l'utilisateur actuel	Dans les champs « Répertoire source par défaut » et « Éditeur », cliquez sur le bouton  et recherchez le répertoire source et l'éditeur de fichiers externes que vous utiliserez.

### Notes

- Ces options s'appliquent à tous les modèles auxquels vous accédez



## Options ActionScript - Modèle

Si vous avez l'intention de générer du code ActionScript à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications ActionScript » de la dialogue « Gérer les options Modèle » pour :

- Spécifiez la version ActionScript par défaut à générer (AS2.0 ou AS3.0)
- Spécifier les extensions de fichier par défaut
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > ActionScript
-------	--

### Options

Option	Action
Options pour le modèle actuel	Type la version ActionScript par défaut et l'extension de fichier par défaut à appliquer lors de la génération du code source ActionScript.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options Ada 2012 - Utilisateur

Si vous souhaitez générer du code Ada 2012 à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Ada » de la dialogue « Préférences » pour :

- Informez le processus d'ingénierie inverse si le nom de l'enregistrement balisé est le même que le nom Paquetage
- Indiquez au moteur le nom de l'enregistrement balisé alternatif à localiser
- Spécifiez si le moteur doit créer un type de référence pour l'enregistrement balisé (si aucun n'est défini)
- Indiquez le nom du type de référence à créer (la valeur par défaut est Ref)
- Spécifier le paramètre de référence d'un type Référence / Accès
- Dites au moteur d'ignorer le nom du paramètre de référence
- Indiquer le nom du paramètre de référence à localiser

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > Ada
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code Ada 2012. Cochez cette case pour désactiver support du code Ada 2012.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

### Notes

- support Ada 2012 est disponible dans les éditions Unified et Ultimate d' Enterprise Architect

## Options Ada 2012 - Modèle

Si vous avez l'intention de générer du code Ada 2012 à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Ada » de la dialogue « Gérer les options Modèle » pour :

- Spécifiez l'extension de fichier par défaut et
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > Ada
-------	---

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source Ada.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles
- support Ada 2012 est disponible dans les éditions Unified et Ultimate d' Enterprise Architect

## Options ArcGIS - Utilisateur

Si vous avez l'intention de générer du code ArcGIS à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « ArcGIS » de la dialogue « Préférences » pour :

- Spécifier le répertoire source par défaut
- Spécifier l'éditeur pour le code ArcGIS

ArcGIS doit être activé dans la dialogue « MDG Technologies » (Spécialiser > Technologies > Gérer la technologie) pour que la page « ArcGIS » soit disponible.

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > ArcGIS
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code ArcGIS. Cochez cette case pour désactiver support du code ArcGIS.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

## Options ArcGIS - Modèle

Si vous avez l'intention de générer du code ArcGIS à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « ArcGIS » de la dialogue « Gérer les options Modèle » pour :

- Spécifier les extensions de fichier par défaut
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > ArcGIS
-------	--

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source ArcGIS.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles



## Options C - Utilisateur

Si vous avez l'intention de générer du code C à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications C » de la dialogue « Préférences ».

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > C
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code C. Sélectionnez cette option pour désactiver support du code C.
Options pour l'utilisateur actuel	<p>Dans les champs valeur , spécifiez les options qui s'appliquent sous votre propre ID d'utilisateur dans tous les modèles auxquels vous accédez :</p> <ul style="list-style-type: none"> <li>• Le type d'attribut par défaut à créer (fixé comme int )</li> <li>• Si une constante #define est importée en tant qu'attribut dans le code C importé (si « Programmation orientée Object » est défini sur True sur la page « Spécifications C » de la dialogue « Gérer les options Modèle »)</li> <li>• S'il faut générer des commentaires pour les méthodes C de la déclaration et effectuer une rétro-ingénierie des commentaires à partir de la déclaration</li> <li>• S'il faut générer des commentaires pour les méthodes C de l'implémentation et effectuer une rétro-ingénierie des commentaires à partir de l'implémentation</li> <li>• S'il faut mettre à jour les commentaires lors de la régénération du code à partir du modèle</li> <li>• S'il faut mettre à jour le fichier d'implémentation lors de la régénération du code à partir du modèle</li> <li>• L'emplacement du répertoire de code source par défaut (cliquez sur le bouton  )</li> <li>• Les extensions de fichiers par défaut à lire lors de l'importation d'un répertoire de code C</li> <li>• L' Éditeur de Code à utiliser (cliquez sur le bouton  )</li> <li>• Le chemin de recherche du fichier d'implémentation par rapport au chemin du fichier d'en-tête</li> </ul>

## Options C - Modèle

Si vous avez l'intention de générer du code C à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications C » de la dialogue « Gérer les options Modèle » pour :

- Spécifier les extensions de fichier par défaut (en-tête et source)
- Définir support de la programmation orientée Object
- Définir les options d'ingénierie Statemachine
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > C
-------	---

### Options

Option	Action
Options pour le modèle actuel	<p>Dans les champs valeur , spécifiez ces options :</p> <ul style="list-style-type: none"> <li>• Les extensions de fichier source et d'en-tête par défaut pour les fichiers de code</li> <li>• Support de la programmation orientée Object ; si cette valeur est vraie, définissez : <ul style="list-style-type: none"> <li>- Le caractère délimiteur Namespace</li> <li>- Si le premier paramètre d'une opération est une référence de classe</li> <li>- Le style de référence des paramètres dans le code C généré</li> <li>- Le nom du paramètre de référence dans le code généré</li> <li>- Le nom du constructeur par défaut dans le code généré</li> <li>- Le nom du destructeur par défaut dans le code généré</li> </ul> </li> </ul>
Ingénierie Statemachine	<p>Dans les champs valeur , utilisez les flèches déroulantes pour définir les options sur True ou False ; ces options s'appliquent uniquement à la génération de code à partir de modèles Statemachine dans le modèle actuel :</p> <ul style="list-style-type: none"> <li>• « Utiliser la nouvelle Statemachine Gabarit » : définissez sur True pour utiliser les gabarits de génération de code d' Enterprise Architect Release 11 et versions ultérieures, définissez sur False pour appliquer les gabarits hérités d'EASL</li> <li>• Générer un code de trace - définissez sur True pour générer un code de trace, False pour l'omettre</li> </ul>
Cours de collection	<p>Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.</p>

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles





## Options C# - Utilisateur

Si vous avez l'intention de générer du code C# à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications C# de la dialogue « Préférences »

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > C#
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code C# . Cochez cette case pour désactiver support du code C# .
Options pour l'utilisateur actuel	<p>Dans les champs valeur , spécifiez les options qui s'appliquent sous votre propre ID d'utilisateur dans tous les modèles auxquels vous accédez :</p> <ul style="list-style-type: none"> <li>• Le type d'attribut par défaut à créer</li> <li>• Si Namespaces doivent être générés lors de la génération de classes C#</li> <li>• S'il faut supprimer les nouvelles lignes (retours chariot durs) de l' étiquette de résumé lors de l'importation de commentaires de style XML.NET</li> <li>• S'il faut générer une méthode Finalizer lors de la génération de code pour une classe C#</li> <li>• S'il faut générer une méthode Dispose lors de la génération de code pour une classe C#</li> <li>• L'emplacement du répertoire de code source par défaut (cliquez sur le bouton  )</li> <li>• L' Éditeur de Code à utiliser (cliquez sur le bouton  )</li> </ul>

## Options C# - Modèle

Si vous avez l'intention de générer du code C# à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications C# » de la dialogue « Gérer les options Modèle » pour :

- Spécifiez l'extension de fichier par défaut
- Indiquer des classes de collection supplémentaires - pour définir des classes de collection personnalisées, qui peuvent être de simples substitutions (telles que `CArray<#TYPE#>`) ou un mélange d'autres chaînes et substitutions (telles que `Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>`) ; ces classes de collection sont définies par défaut :  
- `Liste<#TYPE#>`; `Pile<#TYPE#>`; `File d'attente<#TYPE#>`;
- Définir les options d'ingénierie Statemachine
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > C#
-------	--

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source C# et une liste de toutes les classes de collection supplémentaires que vous souhaitez définir.
Ingénierie Statemachine	Dans les champs valeur , utilisez les flèches déroulantes pour définir les options sur True ou False ; ces options s'appliquent uniquement à la génération de code à partir de modèles Statemachine dans le modèle actuel : <ul style="list-style-type: none"> <li>• « Utiliser la nouvelle Statemachine Gabarit » : définissez sur True pour utiliser les gabarits de génération de code d' Enterprise Architect Release 11 et versions ultérieures, définissez sur False pour appliquer les gabarits hérités d'EASL</li> <li>• « Générer un code de trace » : définissez sur True pour générer un code de trace, False pour l'omettre</li> </ul>
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles



## Options C++ - Utilisateur

Si vous avez l'intention de générer du code C++ à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications C++ » de la dialogue « Préférences ».

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > C++
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code C++. Sélectionnez cette option pour désactiver support du code C++.
Options pour l'utilisateur actuel	<p>Dans les champs valeur , spécifiez les options qui s'appliquent sous votre propre ID d'utilisateur dans tous les modèles auxquels vous accédez :</p> <ul style="list-style-type: none"> <li>• Le type d'attribut par défaut à créer</li> <li>• Si Namespaces doivent être générés lors de la génération de classes C++</li> <li>• Quel style appliquer lors de la génération et du traitement des commentaires pour C++</li> <li>• S'il faut générer des commentaires pour les méthodes C++ de la déclaration ou procéder à une rétro-ingénierie des commentaires à partir de la déclaration</li> <li>• Faut-il générer des commentaires pour les méthodes C++ de l'implémentation ou procéder à une rétro-ingénierie des commentaires à partir de l'implémentation ?</li> <li>• S'il faut mettre à jour les commentaires lors de la régénération du code à partir du modèle</li> <li>• S'il faut mettre à jour le fichier d'implémentation lors de la régénération du code à partir du modèle</li> <li>• L'emplacement du répertoire de code source par défaut (cliquez sur le bouton  )</li> <li>• Les extensions de fichier par défaut à lire lors de l'importation d'un répertoire de code C++</li> <li>• L' Éditeur de Code à utiliser (cliquez sur le bouton  )</li> <li>• Le chemin de recherche du fichier d'implémentation par rapport au chemin du fichier d'en-tête</li> </ul>

## Options C++ - Modèle

Si vous avez l'intention de générer du code C++ à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications C++ » de la dialogue « Gérer les options Modèle » pour :

- Indiquez la version de C++ à générer ; cela contrôle l'ensemble des gabarits utilisés et la manière dont les propriétés sont créées
- Spécifier le type de référence par défaut utilisé lorsqu'un type est spécifié par référence
- Spécifier les extensions de fichier par défaut
- Spécifier les préfixes Get/Set par défaut
- Spécifier les définitions de classe de collection pour les connecteurs d'association
- Définir des classes de collection supplémentaires - pour définir des classes de collection personnalisées, qui peuvent être de simples substitutions (telles que `CArray<#TYPE#>`) ou un mélange d'autres chaînes et substitutions (telles que `Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>`) ; ces classes de collection sont définies par défaut :  
- `CArray<#TYPE#>;CMap<CString,LPCTSTR,#TYPE#*,#TYPE#*>;`
- Définir les options d'ingénierie Statemachine

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > C++
-------	---

### Options

Option	Action
Options pour le modèle actuel	<p>Dans les champs valeur , spécifiez les options qui affectent tous les utilisateurs du modèle actuel :</p> <ul style="list-style-type: none"> <li>• La version de C++ que vous utilisez (qui détermine les gabarits à utiliser lors de la génération de code)</li> <li>• Le type de référence par défaut à utiliser lors de la création de propriétés pour les attributs C++ par référence</li> <li>• Les extensions de fichier source et d'en-tête par défaut pour les fichiers de code</li> <li>• Le préfixe « Get » par défaut</li> <li>• Le préfixe « Set » par défaut</li> <li>• Les classes de collection supplémentaires</li> </ul>
Options d'ingénierie Statemachine	<p>Dans les champs valeur , utilisez les flèches déroulantes pour définir les options sur True ou False ; ces options s'appliquent uniquement à la génération de code à partir de modèles Statemachine dans le modèle actuel :</p> <ul style="list-style-type: none"> <li>• « Utiliser la nouvelle Statemachine Gabarit » : définissez sur True pour utiliser les gabarits de génération de code d' Enterprise Architect Release 11 et versions ultérieures, définissez sur False pour appliquer les gabarits hérités d'EASL</li> <li>• « Générer un code de trace » : définissez sur True pour générer un code de trace, False pour l'omettre</li> </ul>

Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.
---------------------	--

## Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options Delphi - Utilisateur

Si vous avez l'intention de générer du code Delphi à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications Delphi » de la dialogue « Préférences » pour :

- Définir le type d'attribut par défaut
- Indiquer un répertoire source par défaut
- Définir l'éditeur de code par défaut à utiliser pour modifier le code source Delphi

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > Delphi
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code Delphi. Sélectionnez cette option pour désactiver support du code Delphi.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

## Options Delphi - Modèle

Si vous avez l'intention de générer du code Delphi à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications Delphi » de la dialogue « Gérer les options Modèle » pour :

- Spécifier les extensions de fichier par défaut (en-tête et source)
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > Delphi
-------	--

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source Delphi.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

# Propriétés Delphi

Enterprise Architect offre support complet pour les propriétés Delphi. Celles-ci sont implémentées sous forme de Valeur Étiquetés , avec un éditeur de propriétés spécialisé pour aider à créer et modifier les propriétés de classe. En utilisant l'option de menu contextuel de l'élément « Visibilité Fonctionnalité », vous pouvez afficher le compartiment « étiquettes » qui contient les propriétés. Les classes Delphi importées avec des propriétés ont cette fonctionnalité automatiquement rendue visible pour votre commodité.

## Activer manuellement l'éditeur de propriétés

- Dans la classe sélectionnée, définissez le langage de génération de code sur « Delphi »
- Cliquez-droit sur la Classe et sélectionnez 'Delphi Propriétés ' pour ouvrir l'éditeur

En utilisant l'éditeur Delphi Propriétés , vous pouvez créer des propriétés rapidement et simplement ; à partir de là, vous pouvez :

- Modifier le nom et la portée (seuls les publics et les publiés sont actuellement pris en charge)
- Modifier le type de propriété (la liste déroulante inclut toutes les classes définies dans le projet)
- Définissez les informations de lecture et d'écriture (les listes déroulantes contiennent tous les attributs et opérations de la classe actuelle ; vous pouvez également saisir du texte libre)
- Réglez « Stocké » sur Vrai ou Faux
- Définir les informations sur les outils
- Définir la valeur par défaut, si elle existe

## Notes

- Lorsque vous utilisez la dialogue « Créer une propriété » à partir de l'écran « Attribut », le système génère une paire de fonctions Get et Set avec la définition de propriété requise en tant que Valeur Étiquetés ; vous pouvez modifier manuellement ces Valeur Étiquetés si nécessaire
- Les propriétés publiques sont affichées avec un préfixe « + » et publiées avec un « ^ »
- Lors de la création d'une propriété dans la dialogue « Créer une implémentation de propriété » (accessible via la dialogue « Attributs »), vous pouvez définir la portée sur « Publié » si le type de propriété est Delphi
- Seuls « Public » et « Publié » sont pris en charge
- Si vous modifiez le nom d'une propriété et effectuez une ingénierie directe, une nouvelle propriété est ajoutée, mais vous devez supprimer manuellement l'ancienne du fichier source





## Options Java - Utilisateur

Si vous avez l'intention de générer du code Java à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications Java » de la dialogue « Préférences ».

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > Java
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code Java. Cochez cette case pour désactiver support du code Java.
Options pour l'utilisateur actuel	Dans les champs valeur , spécifiez les options qui s'appliquent sous votre propre ID d'utilisateur dans tous les modèles auxquels vous accédez ; le : <ul style="list-style-type: none"><li>Type d'attribut par défaut à créer (sélectionnez dans la liste déroulante)</li><li>Emplacement du répertoire de code source par défaut (cliquez sur le bouton  )</li><li>Éditeur de Code à utiliser (cliquez sur le bouton  )</li></ul>

## Options Java - Modèle

Si vous avez l'intention de générer du code Java à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications Java » de la dialogue « Gérer les options Modèle » pour :

- Spécifiez l'extension de fichier par défaut
- Spécifier un préfixe « Get » par défaut
- Spécifier un préfixe « Set » par défaut
- Définir les options d'ingénierie Statemachine
- Spécifier les définitions de classe de collection pour les connecteurs d'association
- Définir des classes de collection supplémentaires - pour définir des classes de collection personnalisées, qui peuvent être de simples substitutions (telles que `CArray<#TYPE#>`) ou un mélange d'autres chaînes et substitutions (telles que `Cmap<CString,LPCTSTR,#TYPE#*,#TYPE#*>`) ; ces classes de collection sont définies par défaut :
  - `HashSet<#TYPE#>`;
  - `Map< String ,#TYPE#>`;

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > Java
-------	--

### Options

Option	Action
Options pour le modèle actuel	<p>Dans les champs valeur , spécifiez les options qui affectent tous les utilisateurs du modèle actuel ; les :</p> <ul style="list-style-type: none"> <li>• Extension de fichier par défaut pour les fichiers de code</li> <li>• Les préfixes Get et Set par défaut</li> <li>• Les classes de collection par défaut et supplémentaires</li> </ul>
Ingénierie Statemachine	<p>Dans les champs valeur , utilisez les flèches déroulantes pour définir les options sur True ou False ; ces options s'appliquent uniquement à la génération de code à partir de modèles Statemachine dans le modèle actuel :</p> <ul style="list-style-type: none"> <li>• « Utiliser la nouvelle Statemachine Gabarit » : définissez sur True pour utiliser les gabarits de génération de code d' Enterprise Architect Release 11 et versions ultérieures, définissez sur False pour appliquer les gabarits hérités d'EASL</li> <li>• « Générer un code de trace » : définissez sur True pour générer un code de trace, False pour l'omettre</li> </ul>
Cours de collection	<p>Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.</p>

## Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options MySQL - Utilisateur

Si vous avez l'intention de générer du code MySQL à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « MySQL » de la dialogue « Préférences » pour :

- Spécifier un type d'attribut par défaut
- Spécifier un répertoire source par défaut
- Spécifier les extensions de nom de fichier pour les fichiers à importer
- Spécifier un éditeur pour modifier le code
- Spécifier un propriétaire par défaut

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > MySQL
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code MySQL. Sélectionnez cette option pour désactiver support du code MySQL.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

## Options MySQL - Modèle

Si vous avez l'intention de générer du code MySQL à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « MySQL » de la dialogue « Gérer les options Modèle » pour :

- Spécifier l'extension de fichier par défaut
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > MySQL
-------	---

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source MySQL.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options PHP - Utilisateur

Si vous avez l'intention de générer du code PHP à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications PHP » de la dialogue « Préférences » pour :

- Définissez une liste d'extensions séparées par des points-virgules à examiner lors de l'importation d'un code de répertoire pour PHP
- Définir un répertoire par défaut pour ouvrir et enregistrer le code source PHP
- Spécifiez l'éditeur par défaut à utiliser lors de l'édition du code PHP

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > PHP
Raccourcis Clavier	Ctrl+F9   Ingénierie du code source   PHP

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code PHP. Sélectionnez cette option pour désactiver support du code PHP.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

## Options PHP - Modèle

Si vous avez l'intention de générer du code PHP à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications PHP » de la dialogue « Gérer les options Modèle » pour :

- Spécifiez la version PHP par défaut à générer
- Définir l'extension de fichier par défaut
- Spécifier un préfixe « Get » par défaut
- Spécifier un préfixe « Set » par défaut

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > PHP
-------	---

### Options

Option	Action
Options pour le modèle actuel	Type la version PHP par défaut, l'extension de fichier par défaut à appliquer lors de la génération du code source PHP et les préfixes par défaut « Get » et « Set ».

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options Python - Utilisateur

Si vous avez l'intention de générer du code Python à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications Python » de la dialogue « Préférences » pour :

- Spécifiez le répertoire source par défaut à utiliser
- Spécifiez l'éditeur par défaut utilisé pour écrire et modifier le code Python

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > Python
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code Python. Sélectionnez cette option pour désactiver support du code Python.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.



## Options Python - Modèle

Si vous avez l'intention de générer du code Python à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications Python » de la dialogue « Gérer les options Modèle » pour :

- Spécifiez l'extension de fichier par défaut

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > Python
-------	--

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source Python.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options SystemC - Utilisateur

Si vous avez l'intention de générer du code SystemC à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « SystemC » de la dialogue « Préférences » pour :

- Spécifier un répertoire source par défaut
- Spécifier un éditeur pour modifier le code

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > SystemC
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code SystemC. Sélectionnez cette option pour désactiver support du code SystemC.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

## Options SystemC - Modèle

Si vous avez l'intention de générer du code SystemC à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « SystemC » de la dialogue « Gérer les options Modèle » pour :

- Spécifier l'extension de fichier par défaut
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > SystemC
-------	---

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source SystemC.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options Teradata - Utilisateur

Si vous avez l'intention de générer du code Teradata à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Teradata » de la dialogue « Préférences » pour :

- Spécifier un type d'attribut par défaut
- Spécifier un répertoire source par défaut
- Spécifier un éditeur pour modifier le code

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > Teradata
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code Teradata. Sélectionnez cette option pour désactiver support du code Teradata.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

## Options Teradata - Modèle

Si vous avez l'intention de générer du code Teradata à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Teradata » de la dialogue « Gérer les options Modèle » pour :

- Spécifiez l'extension de fichier par défaut
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > Teradata
-------	--

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source Teradata.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options VB.NET - Utilisateur

Si vous avez l'intention de générer du code VB.NET à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications VB.NET » de la dialogue « Préférences » pour :

- Spécifier le type d'attribut par défaut
- Indiquer s'il faut générer des espaces de noms
- Spécifier un répertoire source par défaut
- Spécifier un éditeur pour modifier le code

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > VB.Net
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code VB.NET. Sélectionnez cette option pour désactiver support du code VB.NET.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

## Options VB.NET - Modèle

Si vous avez l'intention de générer du code VB.NET à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications VB.Net » de la dialogue « Gérer les options Modèle » pour :

- Spécifiez l'extension de fichier par défaut
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > VB.Net
-------	--

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source VB.Net.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options Verilog - Utilisateur

Si vous avez l'intention de générer du code Verilog à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Verilog » de la dialogue « Préférences » pour :

- Spécifier un répertoire source par défaut
- Spécifier un éditeur pour modifier le code

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > Verilog
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code Verilog. Sélectionnez cette option pour désactiver support du code Verilog.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.



## Options Verilog - Modèle

Si vous avez l'intention de générer du code Verilog à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Verilog » de la dialogue « Gérer les options Modèle » pour :

- Spécifier l'extension de fichier par défaut
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > Verilog
-------	---

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source Verilog.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options VHDL - Utilisateur

Si vous avez l'intention de générer du code VHDL à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « VHDL » de la dialogue « Préférences » pour :

- Spécifier un répertoire source par défaut
- Spécifier un éditeur pour modifier le code

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > VHDL
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code VHDL. Sélectionnez cette option pour désactiver support du code VHDL.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisateur actuel ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

## Options VHDL - Modèle

Si vous avez l'intention de générer du code VHDL à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « VHDL » de la dialogue « Gérer les options Modèle » pour :

- Spécifier l'extension de fichier par défaut
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > VHDL
-------	--

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source VHDL.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options Visual Basic - Utilisateur

Si vous avez l'intention de générer du code Visual Basic à partir de votre modèle, vous pouvez configurer les options de génération de code à l'aide de la page « Spécifications VB » de la dialogue « Préférences » pour :

- Spécifier le type d'attribut par défaut
- Définir le répertoire source par défaut
- Définir les extensions de fichier pour rechercher les fichiers de code à importer
- Définir l'éditeur par défaut à utiliser pour éditer le code source

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > Visual Basic
Raccourcis Clavier	Ctrl+F9

### Options

Option	Action
Désactiver la langue	Laissez cette case à cocher décochée pour support la génération de code Visual Basic. Sélectionnez cette option pour désactiver support du code Visual Basic.
Options pour l'utilisateur actuel	Spécifie les options utilisées pour l'utilisation actuelle ; ces options s'appliquent à tous les modèles auxquels l'utilisateur accède.

## Options Visual Basic - Modèle

Si vous avez l'intention de générer du code Visual Basic à partir de votre modèle, vous pouvez configurer les options de génération de code spécifiques au modèle à l'aide de la page « Spécifications VB » de la dialogue « Gérer les options Modèle » pour :

- Spécifiez la version par défaut de Visual Basic à générer
- Indiquer l'extension de fichier par défaut lors de la lecture/écriture
- Indiquez le mode de transaction Microsoft Transaction Server (MTS) pour les objets MTS
- Spécifiez si une classe utilise une utilisation multiple (vrai ou faux)
- Spécifier si une classe utilise la propriété Persistable
- Indiquer les comportements de liaison de données et de source de données
- Définir l'espace de noms global
- Définir l'attribut Exposed
- Indiquez si l'attribut Créable est Vrai ou Faux
- Spécifier les définitions de classe de collection pour les connecteurs d'association

### Accéder

Ruban	Paramètres > Modèle > Options > Ingénierie du code source > Visual Basic
-------	--

### Options

Option	Action
Options pour le modèle actuel	Type l'extension de fichier par défaut à appliquer lors de la génération du code source Visual Basic, puis cliquez sur la flèche déroulante dans chacun des autres champs et sélectionnez la valeur appropriée.
Cours de collection	Cliquez sur ce bouton pour ouvrir la dialogue « Classes de collection pour les rôles d'association », à travers laquelle vous spécifiez les définitions de classes de collection pour les connecteurs d'association.

### Notes

- Ces options affectent tous les utilisateurs du modèle actuel ; cependant, elles ne s'appliquent pas aux autres modèles

## Options linguistiques MDG Technologie

Si vous avez chargé une MDG Technologie qui spécifie un module de code dans votre dossier *Sparx Systems > EA > MDG Technologies*, la langue est incluse dans la liste « Ingénierie du code source » de la dialogue « Préférences ». La langue n'est répertoriée dans la dialogue « Préférences » que si un fichier MDG Technologie l'utilise réellement dans votre modèle.

### Accéder

Ruban	Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > MDG
Raccourcis Clavier	Ctrl+F9

### Options

Champ	Action
Extension par défaut	Extension par défaut pour les fichiers sources générés ; affichée si l'option est dans la technologie. Ceci est enregistré par projet.
Importer des extensions de fichiers	Dossier par défaut à partir duquel importer les fichiers sources ; affichée si la technologie supporte les espaces de noms. Ceci est enregistré une fois pour tous les projets.
Générer Namespaces	Indique si les espaces de noms sont générés ou non.
Répertoire source par défaut	Le répertoire par défaut pour enregistrer les fichiers sources générés. Ceci est toujours affiché.
Éditeur	Indique l'éditeur utilisé pour éditer les fichiers sources.
Type att	Indique le type d'attribut par défaut.

### Notes

- Ces options sont définies dans la technologie à l'intérieur de l'étiquette `<CodeOptions>` d'un module de code, comme indiqué :  
`<CodeOption name="DefaultExtension">.rb</CodeOption>`

## Options de réinitialisation

Enterprise Architect stocke certaines des options d'une classe lors de sa création initiale. Certaines sont globales ; par exemple, \$LinkClass est stocké lors de la création initiale de la classe. Ainsi, dans les classes existantes, la modification globale dans la dialogue « Préférences » ne sera pas automatiquement prise en compte. Vous devez modifier les options de la classe existante.

### Modifier les options pour une seule classe

Étape	Action
1	<p>Cliquez sur la classe à modifier et sélectionnez l'option de ruban « Développer &gt; Code source &gt; Générer &gt; Générer un élément unique ».</p> <p>La dialogue ' Générer Code' s'affiche.</p>
2	<p>Cliquez sur le bouton Avancé.</p> <p>La dialogue « Options Object » s'affiche.</p>
3	<p>Cliquez sur l'option « Attributes / Opérations ».</p>
4	<p>Modifiez les options et cliquez sur le bouton Fermer pour appliquer les modifications.</p>

### Modifier les options pour toutes les classes d'un Paquetage

Étape	Action
1	<p>Cliquez sur Paquetage dans la fenêtre Navigateur et sélectionnez l'option de ruban « Développer &gt; Préférences &gt; Options &gt; Réinitialiser la langue source ».</p> <p>La dialogue « Gérer la génération de code » s'affiche.</p>
2	<p>Dans le champ « Où se trouve la langue : », cliquez sur la flèche déroulante et sélectionnez la langue que vous souhaitez modifier.</p>
3	<p>Dans le champ « Convertir en : », cliquez sur la flèche déroulante et sélectionnez la langue vers laquelle vous souhaitez changer.</p>
4	<p>Cochez la case en regard de chaque option à appliquer aux éléments de classe modifiés dans le Paquetage :</p> <ul style="list-style-type: none"> <li>• Effacer les noms de fichiers pour générer du code</li> <li>• Réinitialiser les options par défaut sur chaque classe</li> <li>• Traiter Paquetages enfants sous le Paquetage sélectionné</li> </ul>
5	<p>Cliquez sur le bouton OK pour appliquer les modifications.</p>

## Classes de collection d'ensembles

À l'aide d' Enterprise Architect , vous pouvez définir des classes de collection pour générer du code à partir de connecteurs d'association où le rôle cible a un paramètre de multiplicité supérieur à 1.

### Tâches

Tâche	Détail
Définition des classes de collection	<p>Dans la section « Ingénierie du code source » de la dialogue « Gérer les options Modèle » (sélectionnez l'option de ruban « Paramètres &gt; Modèle &gt; Options &gt; Ingénierie du code source »), sur chaque page de langue, cliquez sur le bouton Classes de collection.</p> <p>La dialogue « Classes de collection pour les rôles d'association » s'affiche. Dans cette dialogue , vous pouvez définir :</p> <ul style="list-style-type: none"> <li>• La classe de collection par défaut pour les rôles 1..*</li> <li>• La classe de collection ordonnée à utiliser pour les rôles 1..*</li> <li>• La classe de collection qualifiée à utiliser pour les rôles 1..*</li> </ul>
Définition des classes de collection pour une classe spécifique	<p>Les classes de collection spécifiques à la classe peuvent être définies en cliquant sur le bouton Classes de collection dans la dialogue « Propriétés » de la classe de l'élément.</p>
Priorité de génération de code	<p>Lorsque Enterprise Architect génère du code pour un connecteur qui a un rôle de multiplicité &gt; 1 :</p> <ol style="list-style-type: none"> <li>1. Si le qualificateur est défini, utilisez la collection qualifiée : <ul style="list-style-type: none"> <li>- pour la classe si définie</li> <li>- sinon utiliser la collection qualifiée en langage de code</li> </ul> </li> <li>2. Si l'option « Ordre » est définie, utilisez la collection ordonnée : <ul style="list-style-type: none"> <li>- pour la classe si définie</li> <li>- sinon utiliser la collection ordonnée du langage du code</li> </ul> </li> <li>3. Sinon, utilisez la collection par défaut : <ul style="list-style-type: none"> <li>- pour la classe si définie</li> <li>- sinon utilisez la collection par défaut du langage de code</li> </ul> </li> </ol>
Utilisation Marqueurs	<p>Vous pouvez inclure le marqueur #TYPE# dans le nom de la collection ; Enterprise Architect le remplace par le nom de la classe collectée au moment de la génération de la source (par exemple, Vector&lt;#TYPE#&gt; deviendrait Vector&lt;foo&gt;).</p> <p>À l'inverse, lors de la rétro-ingénierie, un connecteur d'association est également créé si une entrée correspondante (par exemple, foo si foo est trouvé dans le modèle) est définie comme une classe de collection.</p>
Classes de collection supplémentaires	<p>Des classes de collection supplémentaires peuvent être définies dans les pages d'options de langage spécifiques au modèle pour C# , C++ et Java.</p>
Type de membre	<p>Dans l'onglet « Rôle(s) » de la dialogue « Propriétés » de l'Association (accessible depuis le menu contextuel cliquez-droit de n'importe quelle Association), il existe un champ « Type de membre » pour chacun des rôles source et cible.</p> <p>Si vous définissez cette option, la valeur que vous entrez remplace toutes les</p>



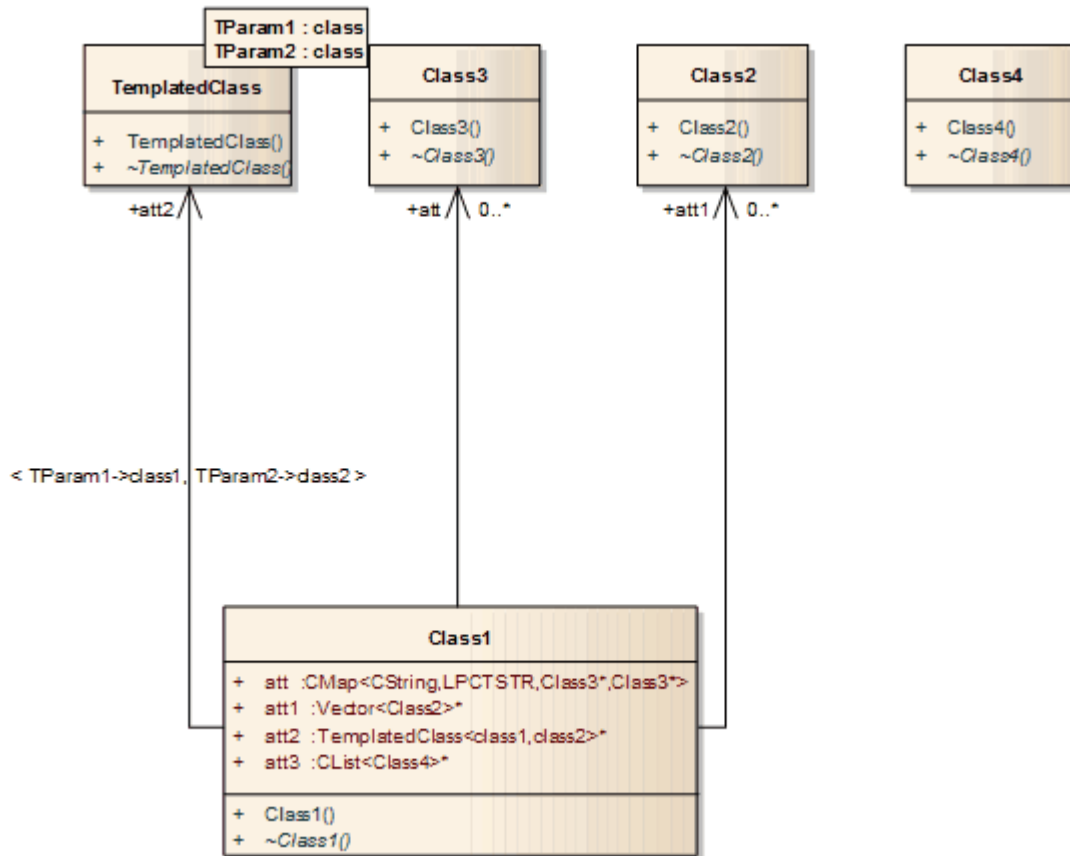
	options répertoriées.
--	-----------------------

## Exemple d'utilisation des classes de collection

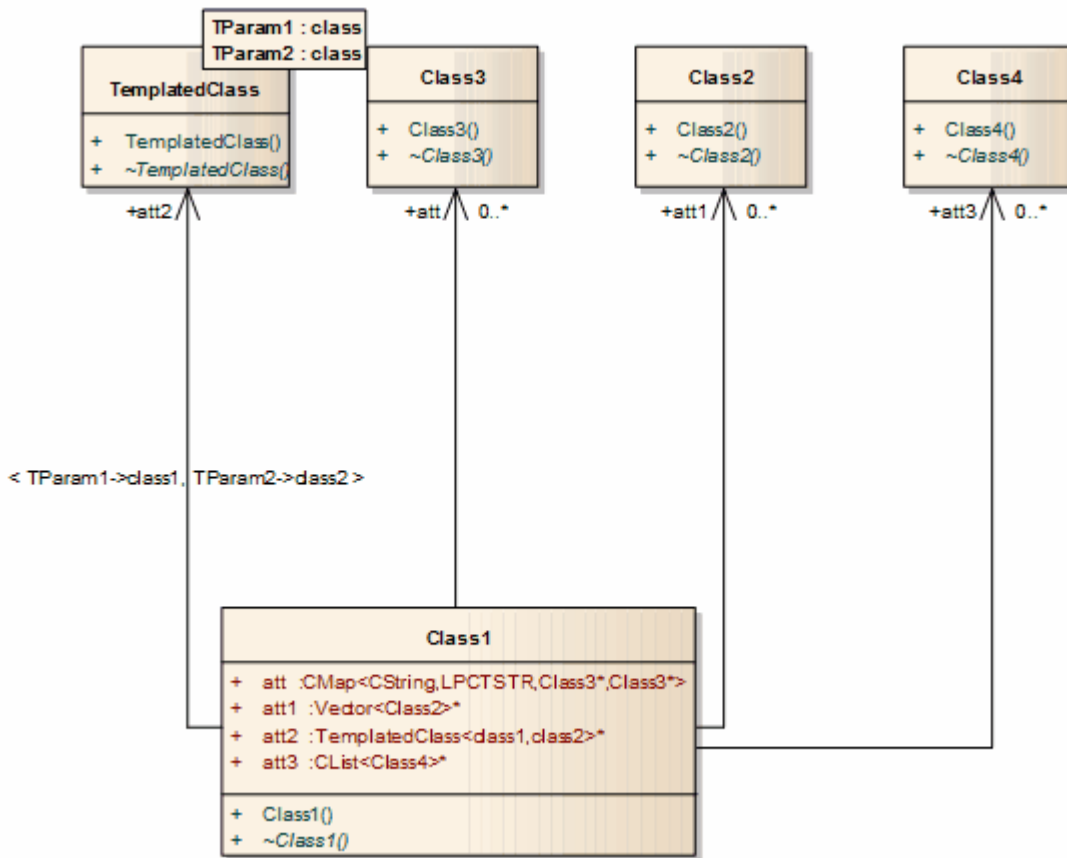
Considérez ce code source :

```
classe Classe1
{
public:
Classe1();
virtuel ~Classe1();
CMap<CString,LPCTSTR,Class3*,Class3*> att;
Vecteur<Classe2> *att1;
TemplatedClass<classe1,classe2> *att2;
CList<Class4> *att3;
};
classe Classe2
{
public:
Classe2();
virtuel ~Classe2();
};
classe Classe3
{
public:
Classe3();
virtuel ~Classe3();
};
classe Classe4
{
public:
Classe4();
virtuel ~Classe4();
};
modèle<classe TParam1, classe TParam2>
classe TemplatedClass
{
public:
Classe modèle() {
}
virtuel ~TemplatedClass() {
}
};
```

Si ce code est importé dans le système avec les options d'importation par défaut, ce diagramme est généré :



Si, toutefois, vous entrez la valeur « CList<#Type#> » dans le champ « Classes de collection supplémentaires » dans la page d'options de langage spécifiques au modèle ( C# , Java, C++), un connecteur d'association est également créé pour la classe 4 :



## Chemins locaux

Lorsqu'une équipe de développeurs travaille sur le même modèle Enterprise Architect, chaque développeur peut stocker sa version du code source dans son système de fichiers local, mais pas toujours au même emplacement que ses collègues développeurs. Pour gérer ce scénario dans Enterprise Architect, vous pouvez définir des chemins locaux pour chaque utilisateur, dans la dialogue « Chemins locaux ».

Vous pouvez utiliser des chemins locaux pour générer du code et de la rétro-ingénierie, ainsi que dans Contrôle de Version, pour développer des schémas XML et générer des documents et des rapports Web.

Les chemins locaux peuvent prendre un peu de temps à configurer, mais si vous souhaitez travailler en collaboration sur la source et le modèle simultanément, l'effort en vaut la peine.

Par exemple, si :

- Le développeur A stocke ses fichiers .java dans un répertoire C:\Java\Source, tandis que le développeur B stocke les siens dans D:\Source, et
- Les deux développeurs souhaitent générer et rétroconcevoir le même modèle Enterprise Architect situé sur un lecteur réseau partagé (ou répliqué)

Le développeur A peut définir un chemin local de :

```
JAVA_SOURCE = "C:\Java\Source"
```

Toutes les classes générées et stockées dans le projet Enterprise Architect sont stockées comme :

```
%JAVA_SOURCE%\<xxx.java>
```

Le développeur B définit un chemin local comme :

```
JAVA_SOURCE = "D:\Source"
```

Désormais, Enterprise Architect stocke tous les fichiers Java dans ces répertoires comme suit :

```
%JAVA_SOURCE%\<nom de fichier>
```

Sur la machine de chaque développeur, le nom de fichier est étendu à la version locale correcte.

### Accéder

Ruban	Développer > Code source > Options > Configurer les chemins locaux
-------	--



## Dialogue sur les chemins locaux

À l'aide de la dialogue « Chemins locaux », vous pouvez configurer des chemins locaux pour un seul utilisateur sur une machine particulière. Pour une description de l'utilisation des chemins locaux, consultez la rubrique *Chemins locaux*.

### Accéder

Ruban	Développer > Code source > Options > Configurer les chemins locaux
-------	--

### Options

Option	Action
Chemin	Type ou recherchez dans votre navigateur le chemin du répertoire local dans le système de fichiers (par exemple, d:\java\source).
ID	Type l' ID partagé qui remplace le chemin local (par exemple, JAVA_SRC).
Type	Cliquez sur la flèche déroulante et sélectionnez le type de chemin à appliquer (par exemple, Java).
Chemins relatifs	Répertorie les chemins actuellement définis pour le modèle, la valeur par défaut étant le plus récent en haut. Si vous souhaitez modifier la séquence des chemins dans la liste, cliquez sur un chemin et utilisez les boutons   pour déplacer le chemin vers le haut ou vers le bas d'une position dans la liste.
Appliquer le chemin	Cliquez sur un chemin dans la liste « Chemins relatifs » et cliquez sur ce bouton pour mettre à jour tous les noms de chemin complets existants dans le modèle avec le nom de chemin relatif partagé. Par exemple : d:\java\source\main.java pourrait devenir %JAVA_SRC%\main.java
Développer le chemin	Cliquez sur un chemin dans la liste « Chemins relatifs » et cliquez sur ce bouton pour supprimer le chemin relatif et le remplacer par le nom du chemin complet (l'effet inverse du bouton Appliquer le chemin).
Nouveau	Cliquez sur ce bouton pour effacer les champs de données afin de pouvoir définir un autre chemin local.
Sauvegarder	Lorsque vous avez défini un chemin local, cliquez sur ce bouton pour l'enregistrer et l'ajouter à la liste « Chemins relatifs ».
Supprimer	Cliquez sur un chemin dans la liste « Chemins relatifs » et cliquez sur ce bouton pour supprimer complètement le chemin de la liste.
Fermer	Cliquez sur ce bouton pour fermer le dialogue et enregistrer les modifications

	apportées à la liste.
--	-----------------------

## Notes

- Vous pouvez également configurer un lien hypertexte (pour une commande Enterprise Architect ) sur un diagramme pour accéder à la dialogue « Chemins locaux » pour changer, mettre à jour ou développer votre chemin local actuel.
- Si l'acte d'étendre ou d'appliquer un chemin pour un fichier lié crée un enregistrement en double, le processus ignorera cet enregistrement et affichera un message à la fin du processus

## Macros de langage

Lors de la rétro-ingénierie d'un langage tel que C++, vous pouvez trouver des directives de préprocesseur dispersées dans le code. Cela peut faciliter la gestion du code, mais peut entraver l'analyse du langage C++ sous-jacent.

Pour remédier à cela, vous pouvez inclure un nombre quelconque de définitions de macros, qui sont ignorées pendant la phase d'analyse de la rétro-ingénierie. Il est toujours préférable, si vous disposez de facilité, de prétraiter le code en utilisant d'abord le compilateur approprié ; de cette façon, les définitions et définitions de macros complexes sont étendues et peuvent être facilement analysées. Si vous ne disposez pas de facilité, cette option constitue un substitut pratique.

### Accéder

Ruban	Paramètres > Données de référence > Paramètres > Macros du préprocesseur ou Développer > Code source > Options > Configurer > Définir les macros du préprocesseur
-------	--

### Définir une macro

Étape	Action
1	Sélectionnez l'option de menu « Macros du préprocesseur ». La dialogue « Macros de langage » s'affiche.
2	Cliquez sur le bouton Ajouter nouveau.
3	Entrez les détails de votre macro.
4	Cliquez sur le bouton OK .

### Macros intégrées dans les déclarations

Les macros sont parfois utilisées dans la déclaration de classes et d'opérations, comme dans ces exemples :

```
classe __declspec Foo
{
int __declspec Barre( int p);
};
```

Si declspec est défini comme une macro C++, comme indiqué, la classe et l'opération importées contiennent une Valeur Étiquetée appelée DeclMacro1 avec valeur \_\_declspec (les macros suivantes seraient définies comme DeclMacro2, DeclMacro3 et ainsi de suite).

Lors de l'ingénierie prospective, ces Valeur Étiquetés sont utilisées pour régénérer les macros dans le code.



## Définir des macros complexes

Il est parfois utile de définir des règles pour des macros complexes pouvant s'étendre sur plusieurs lignes ; Enterprise Architect ignore toute la section de code définie par la règle.

De telles macros peuvent être définies dans Enterprise Architect comme dans ces deux exemples ; les deux types peuvent être combinés dans une seule définition.

### Bloc macros

```
DEBUT_PARTIE_INTERFACE ^ FIN_PARTIE_INTERFACE
```

Le symbole ^ représente le corps de la macro - cela permet de passer d'une macro à une autre ; les espaces entourant le symbole ^ sont obligatoires.

### Macros de fonctions

```
RTTI_EMULATION()
```

Enterprise Architect ignore le jeton, y compris tout ce qui se trouve à l'intérieur des parenthèses.

Les macros de fonction peuvent également inclure le corps de la fonction :

```
RTTI_EMULATION() {}
```

Dans ce cas, Enterprise Architect ignore le jeton, y compris tout ce qui se trouve entre parenthèses et entre accolades. Note que si la macro de fonction inclut le corps de la fonction, elle ne peut pas être combinée avec une macro Bloc .

## Notes

- Vous pouvez transporter ces définitions de macro de langage (ou macro de préprocesseur) entre les modèles, en utilisant les options « Paramètres > Modèle > Transférer > Exporter les données de référence » et « Importer les données de référence » ; les macros sont exportées sous forme de liste de macros.

# Développement de langages de programmation

Vous pouvez utiliser une gamme de langages de programmation établis dans Enterprise Architect , mais si ceux-ci ne sont pas adaptés à vos besoins, vous pouvez développer le vôtre. Vous l'appliquerez ensuite à vos modèles via une MDG Technologie que vous pourriez développer uniquement à cette fin, ou à des fins plus larges. Après avoir développé le langage, vous pouvez également écrire gabarits de transformation MDA pour convertir un Modèle indépendant de la plate-forme ou un modèle dans un autre langage en un modèle pour votre nouveau langage, ou vice-versa.

## Accéder

Ruban	Développer > Code source > Options > Modifier le code Gabarits
Raccourcis Clavier	Ctrl+Maj+P

## Développer un langage de programmation

Étape	Description
1	<p>Dans l'éditeur Code Gabarit , cliquez sur le bouton Nouveau langage et, dans la dialogue « Types de données des langages de programmation », cliquez sur le bouton Ajouter un produit.</p> <p>Saisissez le nom de votre nouveau langage de programmation et définissez les types de données correspondants. Vous ne pouvez pas accéder au nouveau langage dans l'éditeur Code Gabarit tant qu'au moins un type de données n'a pas été ajouté au langage.</p>
2	<p>Après avoir défini tous les types de données dont vous avez besoin, cliquez sur le bouton Fermer, sélectionnez la langue dans le champ « Langue » de l'éditeur de code Gabarit et commencez à éditer ou à créer les gabarits de code pour la nouvelle langue.</p> <p>Les gabarits du code définissent comment le système doit fonctionner :</p> <ul style="list-style-type: none"> <li>• Ingénierie de code avancée de vos modèles dans le nouveau langage</li> <li>• Génération de code Comportementale (si cela est approprié)</li> </ul>
3	<p>Si vous le préférez, vous pouvez également définir des options de code source pour votre nouveau langage. Il s'agit de paramètres supplémentaires pour le langage qui ne sont pas fournis par les types de données ou gabarits de code et qui aident à définir la manière dont le système gère ce langage lors de la génération et de la rétro-ingénierie du code.</p> <p>Les options de code sont mises à disposition de vos modèles uniquement via une MDG Technologie .</p>
4	<p>Définir une grammaire pour votre langue est une étape facultative qui offre deux avantages principaux :</p> <ul style="list-style-type: none"> <li>• Rétro-ingénierie du code existant dans votre modèle</li> <li>• Synchronisation lors de la génération de code afin que les modifications apportées au fichier depuis sa dernière génération ne soient pas perdues.</li> </ul> <p>Pour accéder à l'éditeur de grammaire, sélectionnez l'option de ruban « Développer &gt; Code source &gt; Éditeur de grammaire ».</p>
5	<p>Si vous souhaitez effectuer des transformations MDA vers (ou depuis) votre nouveau langage de programmation, vous pouvez également modifier et créer gabarits de transformation pour celui-ci. Le</p>

	processus de création gabarits de transformation est très similaire à celui de création gabarits de code.
6	Après avoir créé les types de données, gabarits de code, les options de code, la grammaire et gabarits de transformation pour votre nouveau langage, vous pouvez les incorporer et les distribuer dans un MDG Technologie .

## Cadre de code Gabarit

Lorsque vous utilisez Enterprise Architect pour générer du code à partir d'un modèle ou pour transformer le modèle, le système fait référence au Code Gabarit Framework (CTF) pour les paramètres qui définissent comment il doit :

- Ingénierie avancée d'un modèle UML
- Générer du code Comportementale
- Effectuer une transformation Model Driven Architecture (MDA)
- Générer du DDL en modélisation de base de données

Une gamme de gabarits standards est disponible pour la génération directe de code et pour la transformation ; si vous ne souhaitez pas utiliser les configurations CTF standards, vous pouvez les personnaliser pour répondre à vos besoins.

### CTF Gabarits

Type Gabarit	Détail
Code Gabarits	<p>Lorsque vous effectuez l'ingénierie directe d'un modèle de classe, les gabarits de code définissent la manière dont le code squelettique doit être généré pour un langage de programmation donné. Les gabarits d'un langage sont automatiquement associés au langage.</p> <p>Les gabarits sont écrits sous forme de texte brut avec une syntaxe qui partage certains aspects des langages de balisage et des langages de script.</p>
Transformation du Modèle Gabarits	<p>Transformation du Modèle Gabarits fournissent une méthode entièrement configurable pour définir la manière dont les transformations Model Driven Architecture (MDA) convertissent les éléments et les fragments de modèle d'un domaine à un autre.</p> <p>Ce processus est à deux niveaux. Il crée un langage intermédiaire (qui peut être visualisé pour le débogage) qui est ensuite traité pour créer les objets.</p>
Gabarits de génération de code Comportementale	<p>Enterprise Architect supporte la génération de code définissable par l'utilisateur des modèles Comportementale UML .</p> <p>Ceci applique le framework Code Gabarit standard mais inclut des macros de génération de code spécifiques Bibliothèque Simulation Enterprise Architect (EASL).</p>
DDL Gabarits	<p>Gabarits DDL sont très similaires aux gabarits de génération de code, mais ils ont été étendus pour support la génération DDL avec leur propre ensemble de gabarits de base, de macros, de macros de fonctions et d'options gabarit .</p>

## Code Gabarit Personnalisation

Enterprise Architect vous aide à générer du code source à partir de modèles UML pour une large gamme de langages de programmation. gabarits standards (mappings) sont fournis prêts à l'emploi, mais vous pouvez personnaliser la manière dont le code est généré en utilisant le framework Code Gabarit (CTF) pratique et flexible. Ce framework sophistiqué vous permet de personnaliser chaque détail de la manière dont le code est généré, y compris la facilité de création de nouveaux gabarits pour les langages non pris en charge dans le produit de base. Par exemple, JavaScript ne fait pas partie des langages pris en charge, mais une série de gabarits peut être écrite rapidement pour générer JavaScript à partir de modèles UML . Dans ces cas, gabarits existants servent de point de départ et de référence utiles pour les nouveaux langages.

Le framework gabarit de code fournit également le mécanisme de génération de modèles comportementaux et est utilisé pour les gabarits de transformation.

### Fonctionnalités

Fonctionnalité	Détail
Gabarits par défaut	Les codes par défaut Gabarits sont intégrés à Enterprise Architect pour les langages pris en charge par l'ingénierie directe.
Éditeur de code Gabarit	Un éditeur de code Gabarit est fourni pour créer et maintenir des code Gabarits définis par l'utilisateur.
Code de personnalisation Gabarits	Descriptions de la syntaxe gabarit et des macros et fonctions que vous pouvez utiliser pour contrôler les effets des gabarits .
Synchroniser le code	Un sous-ensemble du Code Gabarits par défaut pour synchroniser le code.

## Coder et transformer Gabarits

gabarits de code et de transformation ( Transformation du Modèle ) définissent la manière dont le système doit générer ou transformer du code dans l'un ou l'autre des langages de programmation supportés par Enterprise Architect . Chaque langage dispose d'une large gamme de gabarits de base, chacun définissant la manière dont une structure de code particulière est générée. Vous pouvez utiliser ces gabarits de base tels quels, ou vous pouvez personnaliser et ajouter des gabarits pour mieux supporter votre utilisation des langages standard, ou d'autres langages que vous pourriez définir pour le système. Vous pouvez réviser, mettre à jour et créer des gabarits via l'éditeur Gabarit de code ou l'éditeur Gabarit de transformation.

L'ordre dans lequel les gabarits de base sont listés dans les deux éditeurs est lié à l'ordre hiérarchique des objets et de leurs parties à traiter. Les appels sont effectués de certains gabarits de base vers d'autres, et vous pouvez ajouter d'autres appels à la fois aux gabarits de base et à vos propres gabarits personnalisés. Par défaut, le gabarit du fichier est le point de départ d'un processus de génération de code via les gabarits ; un fichier est constitué de classes qui peuvent contenir des attributs et des opérations.

### Accéder

Ruban	Développer > Code source > Options > Modifier le code Gabarits Conception > Paquetage > Transformation > Transformation Gabarits
Raccourcis Clavier	Ctrl+Maj+P (Génération de code Gabarits ) Ctrl+Alt+H ( Gabarits de transformation MDA)

### Application des Gabarits

Action	Détail
Appel Gabarits	<p>Dans n'importe quel gabarit , vous pouvez appeler d'autres gabarits en utilisant %TemplateName%. Les signes de pourcentage (%) englobants indiquent une macro.</p> <p>Vous utiliseriez ceci pour un seul appel au gabarit ClassBody, %ClassBody%, comme indiqué :</p> <pre>% liste = "Nom du modèle" @separator= " \n" @indent= " " %</pre> <p>La macro %list effectue un passage itératif sur tous les objets dans la portée du gabarit actuel et appelle le TemplateName pour chacun d'eux :</p> <pre>% liste = "ClassBody" @separator= " \n" @indent= " " %</pre> <p>Après la génération ou la transformation, chaque macro est substituée pour produire la sortie générée ; pour un langage tel que C++, le résultat du traitement de ce gabarit pourrait être :</p> <pre>/**  * Ceci est un exemple note de cours générée à l'aide gabarits de code  * @auteur Sparx Systems  */ classe ClasseA : publique ClasseB {</pre>

	<pre>... }</pre>
Exécution du Code Gabarits	<p>Chaque gabarit peut agir uniquement sur un type d'élément particulier ; par exemple, le gabarit ClassNotes agit uniquement sur les éléments de classe et d'interface UML .</p> <p>L'élément à partir duquel le code est en cours de génération est dit dans la portée ; si l'élément dans la portée est stéréotypé, le système recherche un gabarit qui a été défini pour ce stéréotype. Si un gabarit spécialisé est trouvé, il est exécuté ; sinon, l'implémentation par défaut du gabarit de base est utilisée.</p> <p>Gabarits sont traités séquentiellement, ligne par ligne, en remplaçant chaque macro par sa valeur de texte sous-jacente du modèle.</p>
Transférer Gabarits entre les projets	<p>Si vous modifiez un gabarit de génération ou de transformation de code de base, ou créez un gabarit personnalisé, vous pouvez les copier d'un projet à un autre en tant que données de référence.</p>

## Base Gabarits

Le framework Code Gabarit se compose d'un certain nombre de gabarits de base. Chaque gabarit de base transforme des aspects particuliers de l'UML en parties correspondantes de langages orientés objet.

Les gabarits de base forment une hiérarchie, qui varie légèrement selon les différents langages de programmation. Dans une hiérarchie gabarit typique propre à un langage tel que C# ou Java (qui n'ont pas de fichiers d'en-tête), les gabarits peuvent être modélisés comme des classes, mais sont généralement simplement du texte brut. Cette hiérarchie serait légèrement plus compliquée pour des langages tels que C++ et Delphi, qui ont des gabarits d'implémentation distincts.

Chacun des gabarits de base doit être spécialisé pour être utile dans l'ingénierie de code ; en particulier, chaque gabarit est spécialisé pour les langages pris en charge (ou « produits »). Par exemple, il existe un gabarit `ClassBody` défini pour C++, un autre pour C#, un autre pour Java, et ainsi de suite ; en spécialisant les gabarits, vous pouvez adapter le code généré pour l'entité UML correspondante.

Une fois les gabarits de base spécialisés pour une langue donnée, ils peuvent être encore plus spécialisés en fonction de :

- Un stéréotype de classe, ou
- Un stéréotype de fonctionnalité (où la fonctionnalité peut être une opération ou un attribut)

Ce type de spécialisation permet, par exemple, à une opération C# stéréotypée comme « propriété » d'avoir un gabarit de corps d'opération différent d'une opération ordinaire ; le gabarit de corps d'opération peut alors être davantage spécialisé, en fonction du stéréotype de classe.

### gabarits de base utilisés dans le CTF

Gabarit	Description
Attribut	Un gabarit de niveau supérieur pour générer des variables membres à partir d'attributs UML .
Déclaration d'attribut	Utilisé par l'attribut gabarit pour générer une déclaration de variable membre.
Notes sur les attributs	Utilisé par l'attribut gabarit pour générer notes de variables membres.
Classe	Un gabarit de niveau supérieur pour générer des classes à partir de classes UML .
Base de classe	Utilisé par le gabarit de classe pour générer un nom de classe de base dans la liste d'héritage d'une classe dérivée, où la classe de base n'existe pas dans le modèle.
Corps de la classe	Utilisé par la classe gabarit pour générer le corps d'une classe.
Déclaration de classe	Utilisé par le gabarit de classe pour générer la déclaration d'une classe.
Interface de classe	Utilisé par le gabarit de classe pour générer un nom d'interface dans la liste d'héritage d'une classe dérivée, où l'interface n'existe pas dans le modèle.
Notes de cours	Utilisé par le gabarit de classe pour générer les notes de classe.
Déposer	Un gabarit de niveau supérieur pour générer le fichier source. Pour les langages tels que C++, cela correspond au fichier d'en-tête.
Section d'importation	Utilisé dans le gabarit du fichier pour générer des dépendances externes.



Attribut lié	Un gabarit de niveau supérieur pour générer des attributs dérivés d'associations UML .
Notes sur les attributs liés	Utilisé par le gabarit d'attribut lié pour générer les notes d'attribut.
Déclaration d'attribut lié	Utilisé par le gabarit d'attribut lié pour générer la déclaration d'attribut.
Base de classe liée	Utilisé par le gabarit de classe pour générer un nom de classe de base dans la liste d'héritage d'une classe dérivée, pour un élément de classe dans le modèle qui est un parent de la classe actuelle.
Interface de classe liée	Utilisé par le gabarit de classe pour générer un nom d'interface dans la liste d'héritage d'une classe dérivée, pour un élément d'interface dans le modèle qui est un parent de la classe actuelle.
Namespace	Un gabarit de niveau supérieur pour générer des espaces de noms à partir de Paquetages UML (bien que tous les langages n'aient pas d'espaces de noms, ce gabarit peut être utilisé pour générer une construction équivalente, comme Paquetages en Java).
Corps Namespace	Utilisé par le gabarit Namespace pour générer le corps d'un espace de noms.
Déclaration Namespace	Utilisé par le gabarit Namespace pour générer la déclaration d'espace de noms.
Opération	Un gabarit de niveau supérieur pour générer des opérations à partir des opérations d'une classe UML .
Opération Corps	Utilisé par le gabarit d'opération pour générer le corps d'une opération UML .
Déclaration d'opération	Utilisé par le gabarit d'opération pour générer la déclaration d'opération.
Notes d'opération	Utilisé par le gabarit d'opération pour générer la documentation d'une opération.
Paramètre	Utilisé par le gabarit de déclaration d'opération pour générer des paramètres.

## Gabarits pour générer du code pour les langages avec des sections d'interface et d'implémentation séparées

Gabarit	Description
Implémentation de classe	Un gabarit de niveau supérieur pour générer l'implémentation d'une classe.
Corps de classe Impl	Utilisé par le gabarit de classe Impl pour générer l'implémentation des membres de la classe.
Fichier Imp	Un gabarit de niveau supérieur pour générer le fichier d'implémentation.
Notes de fichier Impl	Utilisé par le gabarit File Impl pour générer notes dans le fichier source.

Section d'importation Impl	Utilisé par le gabarit File Impl pour générer des dépendances externes.
Opération Impl	Un gabarit de niveau supérieur pour générer des opérations à partir des opérations d'une classe UML .
Opération Body Impl	Utilisé par le gabarit d'opération pour générer le corps d'une opération UML .
Déclaration d'opération Impl	Utilisé par le gabarit d'opération pour générer la déclaration d'opération.
Notes d'opération Impl	Utilisé par le gabarit d'opération pour générer la documentation d'une opération.

# Génération et transformation de code d'exportation

## Gabarits

Il est possible d'exporter gabarits de génération et de transformation de code de votre modèle vers un fichier .xml. Vous pouvez ensuite importer ce fichier - et donc les gabarits - dans d'autres modèles, en tant que données de référence. Vous pouvez exporter gabarits personnalisés, qui incluent ceux que vous ou d'autres utilisateurs avez créés et mis à jour, et gabarits de base (standard) qui ont été personnalisés. Vous n'avez pas besoin d'exporter gabarits de base qui n'ont pas été modifiés, car ils sont disponibles dans chaque installation d' Enterprise Architect .

### Accéder

Ruban	Paramètres > Modèle > Transfert > Exporter les données de référence
-------	---

### Exporter un gabarit de Génération de Code ou gabarit de Transformation

Étape	Action
1	<p>Dans la dialogue « Exporter les données de référence », dans la liste « Nom », sélectionnez les gabarits à exporter.</p> <p>La liste comprend tous gabarits de génération ou de transformation de code standard qui ont été modifiés, ainsi que tous gabarits personnalisés que vous avez créés ou modifiés.</p> <p>Vous pouvez sélectionner un ou plusieurs gabarits à exporter vers un seul fichier XML, en appuyant sur Ctrl ou Maj lorsque vous cliquez sur les noms gabarit .</p>
2	Cliquez sur le bouton Exporter.
3	Lorsque vous y êtes invité, entrez un nom de fichier valide avec une extension .xml.
4	<p>Cliquez sur le bouton Enregistrer et sur le bouton OK .</p> <p>Cela exporte le(s) gabarit (s) vers le fichier ; vous pouvez utiliser n'importe quel visualiseur de texte ou XML pour examiner le fichier.</p>

# Génération et transformation de code d'importation

## Gabarits

Si vous avez exporté gabarits de génération et/ou de transformation de code à partir d'un modèle Enterprise Architect , vous pouvez les importer dans d'autres modèles Enterprise Architect en tant que données de référence.

### Accéder

Ruban	Paramètres > Modèle > Transfert > Importer les données de référence
-------	---

### Génération et/ou transformation de code d'importation Gabarits

Étape	Action
1	Dans la dialogue « Importer des données de référence », cliquez sur le bouton Sélectionner un fichier et accédez au fichier .xml contenant les gabarits de génération ou de transformation de code requis.
2	Sélectionnez le nom d'un ou plusieurs jeux de données gabarit et cliquez sur le bouton Importer.

## Synchroniser le code

Enterprise Architect utilise gabarits de code lors de la synchronisation directe de ces langages de programmation :

- ActionScript
- C
- C++
- C#
- Delphi
- Java
- PHP
- Python
- VB
- VB.Net

Trois types de changements peuvent survenir dans la source lorsqu'elle est synchronisée avec le modèle UML :

- Les sections existantes sont synchronisées : par exemple, le type de retour dans une déclaration d'opération est mis à jour
- De nouvelles sections sont ajoutées aux fonctionnalités existantes : par exemple, Notes sont ajoutées à une déclaration de classe là où il n'y en avait pas auparavant
- De nouvelles fonctionnalités et éléments sont ajoutés : par exemple, une nouvelle opération est ajoutée à une classe

Chacun de ces changements a un effet différent sur le CTF et doit être géré différemment par Enterprise Architect , comme décrit dans ces rubriques :

- *Synchroniser les sections existantes*
- *Ajouter de nouvelles sections aux Fonctionnalités existantes*
- *Ajouter de nouvelles Fonctionnalités et éléments*

### Sections de code pouvant être synchronisées

Seul un sous-ensemble des gabarits de base CTF est utilisé lors de la synchronisation. Ce sous-ensemble correspond aux sections distinctes qu'Enterprise Architect reconnaît dans le code source.

Code Gabarit	Section du code
Notes de cours	Commentaires précédant la déclaration de classe.
Déclaration de classe	Jusqu'aux parents de la classe inclus.
Notes sur les attributs	Commentaires précédant une déclaration d'attribut.
Déclaration d'attribut	Jusqu'au caractère final inclus.
Notes d'opération	Commentaires précédant une déclaration d'opération.
Notes d'opération Impl	Quant Notes d'opération.
Déclaration d'opération	Jusqu'au caractère final inclus.
Déclaration d'opération	

Impl	Jusqu'au caractère final inclus.
Opération Corps	Tout ce qui se trouve entre et y compris les accolades.
Opération Body Impl	Quant à l'Opération Corps.

## Synchroniser les sections existantes

Lorsqu'une section existante dans le code source diffère du résultat généré par le gabarit correspondant, cette section est remplacée.

Considérez, par exemple, cette déclaration de classe C++ :

```
(asm) classe A : public B
```

Supposons maintenant que vous ajoutiez une relation d'héritage de la classe A à la classe C ; la déclaration de classe entière serait remplacée par quelque chose ressemblant à ceci :

```
(asm) classe A : public B, public C
```

## Ajouter de nouvelles sections

Ces sections peuvent être ajoutées aux fonctionnalités existantes dans le code source, en tant que nouvelles sections :

- Notes de cours
- Notes sur les attributs
- Notes d'opération
- Notes d'opération Impl
- Opération Corps
- Opération Body Impl

Supposons que, dans cet exemple, la classe A n'avait aucune note lorsque vous avez généré le code à l'origine :

```
(asm) classe A : public B, public C
```

Si vous spécifiez maintenant une note dans le modèle pour la classe A, Enterprise Architect tente d'ajouter la nouvelle note à partir du modèle lors de la synchronisation, en exécutant le gabarit Class Notes .

Pour faire de la place à l'insertion de la nouvelle section, vous pouvez spécifier la quantité d'espace blanc à ajouter à la section via des macros de synchronisation.



## Ajouter de nouvelles Fonctionnalités et éléments

Ces fonctionnalités et éléments peuvent être ajoutés au code source lors de la synchronisation :

- Attributes
- Classes internes
- Opérations

Ils sont ajoutés en exécutant les gabarits pertinents pour chaque nouvel élément ou fonctionnalité du modèle.

Enterprise Architect tente de préserver l'indentation appropriée des nouvelles fonctionnalités dans le code, en trouvant les indentations spécifiées dans les macros de liste de la classe ; pour les langages qui utilisent des espaces de noms, la macro « synchNamespaceBodyIndent » est disponible.

Les classes définies dans un espace de noms (non global) sont indentées en fonction de la valeur définie pour cette macro, lors de la synchronisation.

La valeur est ignorée :

- Pour les classes définies dans un Paquetage configuré comme espace de noms racine, ou
- Si l'option « Générer Namespaces » est définie sur False dans la page de langue appropriée ( C# , C++ ou VB.Net) dans la dialogue « Préférences » (' Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source > <langue>')

# L'éditeur Code Gabarit

L'éditeur de code Gabarit fournit les facilités de l' Éditeur de Code Common, y compris Intelli-sense pour les différentes macros. Pour plus d'informations sur Intelli-sense et l' Éditeur de Code Common, consultez la rubrique *Édition du code source* .

## Accéder

Ruban	Développer > Code source > Options > Modifier le code Gabarits
Raccourcis Clavier	Ctrl+Maj+P

## Options

Option	Action
Langue	Sélectionnez le langage de programmation.
Nouvelle langue	Affichez la dialogue « Types de données des langages de programmation », qui vous permet d'inclure des langages de programmation autres que ceux pris en charge par Enterprise Architect , pour lesquels créer ou modifier gabarits de code.
Gabarit	Affiche le contenu du gabarit actif et ouvre l'éditeur de modification gabarits .
Gabarits	Lister les gabarits du code de base ; le gabarit actif est mis en surbrillance. Le champ « Modifié » indique si vous avez modifié le gabarit par défaut pour la langue actuelle.
Les stéréotypes supplantent	Lister les gabarits stéréotypés, pour le gabarit de base actif. Le champ « Modifié » indique si vous avez modifié un gabarit stéréotypé par défaut.
Ajouter un nouveau Gabarit personnalisé	Invoyer une dialogue pour créer un gabarit stéréotypé personnalisé.
Ajouter un nouveau remplacement stéréotypé	Appeler une dialogue pour ajouter un gabarit stéréotypé, pour le gabarit de base actuellement sélectionné.
Obtenir Gabarit par défaut	Mettre à jour l'affichage de l'éditeur avec la version par défaut du gabarit actif.
Sauvegarder	Remplacer les gabarits actifs par le contenu de l'éditeur.
Supprimer	Si vous avez remplacé le gabarit actif, le remplacement est supprimé et remplacé par le gabarit de code par défaut correspondant.

## Notes

- Gabarits de code modifiés et définis par l'utilisateur peuvent être importés et exportés en tant que données de référence (voir la rubrique *Partage des données de référence*) ; les gabarits définis pour chaque langue sont indiqués dans la dialogue « Exporter les données de référence » par le nom de la langue avec le suffixe `_Code_Templates` - si aucun gabarits n'existe pour une langue, il n'y a pas d'entrée pour la langue dans le dialogue

## Créer un nouveau Gabarit personnalisé

La dialogue Créer un nouveau Gabarit personnalisé offre la possibilité de créer un gabarit personnalisé pour le langage de programmation ou le système de gestion de base de données (SGBD) actuel, en fonction des informations modifiées avec l'éditeur de code Gabarit .

Lorsque cette dialogue est chargée, il vous sera demandé de saisir une valeur pour Type Gabarit et le nom Gabarit . Pour enregistrer un nouveau gabarit Type et le nom sont tous deux requis.

### Options

Option	Action
Type Gabarit	Choisissez le type de Gabarit pour le nouveau Custom Gabarit .
Nom de Gabarit	Entrez un nom pour le nouveau Gabarit personnalisé.
OK	Enregistrez les détails du nouveau Custom Gabarit .
Annuler	Fermez la dialogue Créer un nouveau Gabarit personnalisé et perdez toutes les modifications non enregistrées.

### Note :

Tous gabarits de type " <none> » sont traités comme des fonctions. Par conséquent, Enterprise Architect supprimera automatiquement tous les caractères d'espace saisis dans le nom.

# Syntaxe du code Gabarit

Les Code Gabarits sont écrits à l'aide de l'éditeur Code Gabarit d' Enterprise Architect . L'éditeur Code Gabarit supporte la coloration syntaxique du langage Code Gabarit Framework.

## Éléments de syntaxe

Éléments	Détail
Constructions de base	<p>Gabarits peuvent contenir :</p> <ul style="list-style-type: none"><li>• Texte littéral</li><li>• Variables</li><li>• Macros</li><li>• Appels vers d'autres gabarits</li></ul>
Commentaires	<p>Si vous souhaitez ajouter des commentaires aux gabarits , utilisez la commande : <code>\$COMMENT="texte"</code> où « texte » est le texte du commentaire ; celui-ci doit être placé entre guillemets. La commande est sensible à la casse et doit être saisie en majuscules.</p>

## Texte littéral

Tout texte dans un gabarit donné qui ne fait pas partie d'une macro ou d'une définition/référence de variable est considéré comme du texte littéral. À l'exception des lignes vides, qui sont ignorées, le texte littéral est directement substitué à partir du gabarit dans le code généré.

Considérez cet extrait du gabarit de déclaration de classe Java :

```
$bases = "Base"  
classe % className % $bases
```

Sur la dernière ligne, le mot « class », y compris l'espace qui suit, serait traité comme du texte littéral et donc pour une classe nommée « foo », renverrait la sortie :

```
classe fooBase
```

Une ligne vide suivant la variable \$bases n'aurait aucun effet sur la sortie.

### Insertion de caractères système :

Les caractères %, \$, " et \ ont une signification particulière dans la syntaxe gabarit et ne peuvent pas toujours être utilisés comme texte littéral. Si ces caractères doivent être générés à partir des gabarits, ils peuvent être reproduits en toute sécurité à l'aide de ces macros de substitution directe :

Macro	Action
%dl%	Produire un caractère \$ littéral.
%pc%	Produire un caractère % littéral.
%qt%	Produire un caractère " littéral.
%sl%	Produire un caractère \ littéral

### Notes

Les opérateurs de conjonction String (« + », « += ») ne sont pas obligatoires mais peuvent être utilisés

# Variables

Les variables Gabarit offrent un moyen pratique de stocker et de récupérer des données dans un gabarit . Cette section explique comment les variables sont définies et référencées.

## Définitions des variables

Les définitions de variables prennent la forme de base :

```
$<nom> = <valeur>
```

où <nom> peut être n'importe quelle séquence alphanumérique et <valeur> est dérivé d'une macro ou d'une autre variable.

Un exemple simple de définition serait :

```
$foo = %className%
```

Les variables peuvent être définies à l'aide de valeurs provenant de :

- Substitution, fonction ou macros de liste
- Littéraux String , placés entre guillemets doubles
- Références variables

## Règles de définition

Ces règles s'appliquent aux définitions de variables :

- Les variables ont une portée globale dans le gabarit dans lequel elles sont définies et ne sont pas accessibles aux autres gabarits
- Chaque variable doit être définie au début d'une ligne, sans aucun espace intermédiaire
- Les variables sont indiquées en préfixant le nom avec \$, comme dans \$foo
- Les variables n'ont pas besoin d'être déclarées avant d'être définies
- Les variables doivent être définies à l'aide de l'opérateur d'affectation (=) ou de l'opérateur d'addition-affectation (+=)
- Plusieurs termes peuvent être combinés dans une seule définition à l'aide de l'opérateur d'addition (+)

## Exemples

Utilisation d'une macro de substitution :

```
$foo = %opTag:"bar"%
```

En utilisant une string littérale :

```
$foo = "barre"
```

En utilisant une autre variable :

```
$foo = $bar
```

Utilisation d'une macro de liste :

```
$sops = %list="Opération" @separator="\n\n" @indent="\t"%
```

En utilisant l'opérateur d'addition-affectation (+=) :

```
$body += %list="Opération" @separator="\n\n" @indent="\t"%
```

Cette définition est équivalente à :

```
$body = $body + %list="Opération" @separator="\n\n" @indent="\t"%
```

Utilisation de plusieurs termes :

```
$templateArgs = %list="Paramètre de classe" @séparateur=", "%
```

```
$template = "modèle< " + $templateArgs + "> "
```

## Références variables

Les valeurs des variables peuvent être récupérées en utilisant une référence de la forme :

```
$<nom>
```

où <nom> peut être une variable précédemment définie.

Des références variables peuvent être utilisées :

- Dans le cadre d'une macro, comme l'argument d'une macro de fonction
- En tant que terme dans une définition de variable
- En tant que substitution directe de la valeur de la variable dans la sortie

Il est légal de référencer une variable avant qu'elle ne soit définie. Dans ce cas, la variable est supposée contenir une string vide valeur : " "

## Références variables - Exemple 1

Utilisation de variables dans le cadre d'une macro. Ceci est un extrait du gabarit par défaut de C++ ClassNotes.

```
$wrapLen = %genOptWrapComment%
```

```
$style = %genOptCPPCommentStyle% (Définir des variables pour stocker les options de style et de longueur d'habillage)
```

```
%if $style == "XML.NET"% (Référence à $style dans le cadre d'une condition)
```

```
%XML_COMMENT($wrapLen)%
```

```
%autre%
```

```
%CSTYLE_COMMENT($wrapLen)% (Référence à $wrapLen comme argument de la macro de fonction)
```

```
%finSi%
```

## Références de variables - Exemple 2

Utilisation de références de variables dans le cadre d'une définition de variable.

```
$foo = "foo" (Définir nos variables)
```

```
$bar = "barre"
```

```
$foobar = $foo + $bar ($foobar contient maintenant la valeur foobar)
```

## Références de variables - Exemple 3

Substitution de valeurs de variables dans la sortie.



\$bases=%classInherits% (Stocker le résultat du gabarit ClassInherits dans \$bases)

Classe %className%\$bases (affiche maintenant la valeur de \$bases après le nom de la classe)

# Macros

Les macros permettent d'accéder aux champs d'éléments du modèle UML et sont également utilisées pour structurer la sortie générée. Toutes les macros sont entourées de signes de pourcentage (%), comme indiqué :

%<nom de la macro>%

En général, les macros (y compris les délimiteurs %) remplacent le texte littéral dans la sortie. Par exemple, considérez cet élément du gabarit de déclaration de classe :

... classe %className% ...

La macro de substitution de champ, %className%, entraînerait la substitution du nom de classe actuel dans la sortie. Ainsi, si la classe générée était nommée Foo, la sortie serait :

... classe Foo ...

Le CTF contient un certain nombre de types de macros :

- [Template Substitution Macros](#)
- [Field Substitution Macros](#)
- [Substitution Examples](#)
- [Attribute Field Substitution Macros](#)
- [Class Field Substitution Macros](#)
- [Code Generation Option Field Substitution Macros](#)
- [Connector Field Substitution Macros](#)
- [Constraint Field Substitution Macros](#)
- [Effort Field Substitution Macros](#)
- [File Field Substitution Macros](#)
- [File Import Field Substitution Macros](#)
- [Link Field Substitution Macros](#)
- [Linked File Field Substitution Macros](#)
- [Metric Field Substitution Macros](#)
- [Operation Field Substitution Macros](#)
- [Package Field Substitution Macros](#)
- [Parameter Field Substitution Macros](#)
- [Problem Field Substitution Macros](#)
- [Requirement Field Substitution Macros](#)
- [Resource Field Substitution Macros](#)
- [Risk Field Substitution Macros](#)
- [Scenario Field Substitution Macros](#)
- [Tagged Value Substitution Macros](#)
- [Template Parameter Substitution Macros](#)
- [Test Field Substitution Macros](#)
- [Function Macros](#)
- [Control Macros](#)
- [List Macro](#)
- [Branching Macros](#)
- [Synchronization Macros](#)

- [The Processing Instruction \(PI\) Macro](#)
- [EASL Code Generation Macros](#)

# Macros de substitution Gabarit

Les macros de substitution Gabarit correspondent aux gabarits de base et aboutissent à l'exécution du gabarit nommé. Par convention, les macros gabarit sont nommées selon la casse Pascal.

Structure : %<Nom du modèle>%

où <TemplateName> peut être l'un des gabarits répertoriés dans cette rubrique.

Lorsqu'un gabarit est référencé à partir d'un autre gabarit, il est généré par rapport aux éléments actuellement concernés. Le gabarit spécifique est sélectionné en fonction des stéréotypes des éléments concernés.

Comme indiqué précédemment, il existe une hiérarchie implicite entre les différents gabarits. Il convient de prendre certaines précautions afin de préserver une hiérarchie raisonnable des références gabarit. Par exemple, il n'est pas judicieux d'utiliser la macro %ClassInherits% dans l'un des gabarits Attribute ou Operation. Inversement, les gabarits Operation et Attribute sont conçus pour être utilisés dans le gabarit ClassBody.

## Macros de substitution Gabarit dans le CTF

- Attribut
- Déclaration d'attribut
- Déclaration d'attributImpl
- Notes sur les attributs
- Classe
- Base de classe
- Corps de classe
- ClasseBodyImpl
- Déclaration de classe
- Déclaration de classeImpl
- ClasseImpl
- La classe hérite
- Interface de classe
- Notes de cours
- Paramètre de classe
- Déposer
- FichierImpl
- Section d'importation
- ImportationSectionImpl
- Classe intérieure
- Implémentation de la classe intérieure
- Attribut lié
- Déclaration d'attribut lié
- Notes sur les attributs liés
- Base de classe liée
- Interface de classe liée
- Namespace
- Espace de nomsCorps

- Déclaration d'espace de noms
- Espace de nomsImpl
- Opération
- OpérationCorps
- OpérationBodyImpl
- Déclaration d'opération
- OpérationDéclarationImpl
- OpérationImpl
- Notes d'opération
- Paramètre

## Macros de substitution de champs

Les macros de substitution de champs permettent d'accéder aux données de votre modèle. Elles permettent notamment d'accéder aux champs de données de :

- Paquetages
- Cours
- Attributs
- Opérations et
- Paramètres

Les macros de substitution de champ sont nommées selon la casse Camel. Par convention, la macro est préfixée par une forme abrégée de l'élément de modèle correspondant. Par exemple, les macros liées aux attributs commencent par att, comme dans la macro %attName%, pour accéder au nom de l'attribut concerné.

Les macros qui représentent des cases à cocher renvoient une valeur de T si la case est sélectionnée. Sinon, la valeur est vide.

Ce tableau répertorie un petit nombre de macros de substitution de champs de projet. Les macros spécifiques au type sont répertoriées dans les sous-rubriques de cette section *Macros de substitution de champs* .

### Macros de projet

Nom de la macro	Description
Date et heure	L'heure actuelle au format : JJ-MMM-AAAA HH:MM:SS AM/PM.
eaGUID	Un GUID unique pour cette génération.
Version ea	Version du programme (située dans la dialogue « À propos Enterprise Architect » en sélectionnant « Démarrer > Aide > Aide > À propos d'EA »).

# Exemples de substitution

Les macros de substitution de champ peuvent être utilisées de deux manières :

- Substitution directe ou
- Substitution conditionnelle

## Substitution directe

Ce formulaire remplace directement la valeur correspondante de l'élément concerné dans la sortie.

Structure : %<macroName>%

Où <macroName> peut être l'une des macros répertoriées dans les tableaux Macros de substitution de champ.

## Exemples

- %Nom de classe%
- %opName%
- %attName%

## Substitution conditionnelle

Cette forme de macro permet de réaliser des substitutions alternatives en fonction de la valeur de la macro.

Structure : %<macroName> (== "<text> " ) ? <subTrue> (: <subFalse> ) %

Où:

- () indique que les valeurs entre parenthèses sont facultatives
- <text> est une string représentant une valeur possible pour la macro
- <subTrue> et <subFalse> peuvent être une combinaison de chaînes entre guillemets et du mot-clé valeur ; lorsque la valeur est utilisée, elle est remplacée par la valeur de la macro dans la sortie

## Exemples

- %classAbstract=="T" ? "pure" : " "%
- %opStereotype=="opérateur" ? "opérateur" : " "%
- %paramDefault != " " ? " = " valeur : " "%

Ces trois exemples ne génèrent aucun résultat si la condition échoue. Dans ce cas, la condition False peut être omise, ce qui donne cette utilisation :

- %classAbstract=="T" ? "pur"%
- %opStereotype=="opérateur" ? "opérateur"%
- %paramDefault != " " ? " = "valeur%

Le troisième exemple des deux blocs montre une comparaison vérifiant une valeur non vide ou une existence. Ce test peut également être omis.

- %paramDefault ? " = " valeur : " "%

- `%paramDefault ? " = " valeur%`

Tous ces exemples contenant `paramDefault` sont équivalents. Si le paramètre concerné avait une valeur par défaut de 10, la sortie de chacun d'eux serait normalement :

= 10

## Notes

- Dans une macro de substitution conditionnelle, tout espace blanc suivant `<macroName>` est ignoré ; si un espace blanc est requis dans la sortie, il doit être inclus dans les chaînes de substitution entre guillemets



## Macros de substitution de champs d'attributs

Ce tableau répertorie chacune des macros de substitution de champ d'attribut.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros d'attributs

Nom de la macro	Description
attAlias	dialogue « Attributs » : Alias.
attAutoriser les doublons	dialogue « Détails Attributs » : case à cocher « Autoriser les doublons ».
attClassifierGUID	Le GUID unique pour le classificateur de l'attribut actuel.
attCollection	dialogue « Détails Attributs » : case à cocher « L'attribut est une collection ».
attConst	Boîte dialogue « Attributs » : case à cocher « Const ».
attContainerType	dialogue « Détails Attributs » : Type de conteneur.
attConfinement	dialogue « Attributs » : Confinement.
attDérivé	dialogue « Attributs » : case à cocher « Dérivé ».
attGUID	Le GUID unique pour l'attribut actuel.
attInitial	dialogue « Attributs » : Initial.
attIsEnumLiteral	dialogue « Attributs » : case à cocher « Est littéral ».
ID d'identification d'attrance	Boîte de dialogue « Détails Attributs » : case à cocher « isID ».
attLongueur	dialogue « Colonne » : Durée.
attLimite inférieure	dialogue « Détails Attributs » : limite inférieure.
Nom d'utilisateur	dialogue « Attributs » : Nom.
attNotes	dialogue « Attributs » : Notes .
attOrderedMultiplicité	dialogue « Détails Attributs » : case à cocher « Multiplicité ordonnée ».
attProperty	dialogue « Attributs » : case à cocher « Propriété ».
attQualType	Le type d'attribut qualifié par le chemin de l'espace de noms (en cas de génération d'espaces de noms) et le chemin du classificateur (délimité par des points). Si le

	classificateur d'attribut n'a pas été défini, est équivalent à la macro attType.
attScope	dialogue « Attributes » : Portée.
attStatique	Boîte de dialogue « Attributes » : case à cocher « Statique ».
attStéréotype	dialogue « Attributes » : Stéréotype.
attType	dialogue « Attributes » : Type .
attLimite supérieure	dialogue « Détails Attributes » : limite supérieure.
attVolatile	Boîte de dialogue « Détails Attributes » : case à cocher « Transient ».

## Macros de substitution de champs de classe

Ce tableau fournit une liste de méthodes permettant d'accéder à chaque propriété de classe disponible dans les gabarits de génération et de transformation de code.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros de classe

Nom de la macro	Description
élémentType	Le type d'élément : Interface ou Classe.
Résumé de la classe	dialogue Classe ' Propriétés ' : case à cocher 'Résumé' (onglet 'Détails').
classeAlias	dialogue de la classe ' Propriétés ' : champ 'Alias'.
Arguments de classe	dialogue de classe « Détail » : C++ Gabarits : Arguments.
classeAuteur	dialogue de la classe ' Propriétés ' : champ 'Auteur'.
nom de base de la classe	dialogue « Hiérarchie Type » : Nom de la classe (à utiliser lorsqu'il n'existe aucun connecteur entre les classes enfant et de base).
classeBaseScope	La portée de l'héritage en tant qu'ingénierie inverse. (À utiliser lorsqu'il n'existe aucun connecteur entre les classes enfant et de base.)
classeBaseVirtual	La propriété virtuelle de l'héritage telle qu'elle a été conçue à rebours. (À utiliser lorsqu'il n'existe aucun connecteur entre les classes enfant et de base.)
classeComplexité	dialogue de la classe ' Propriétés ' : champ 'Complexité'.
classeCréé	La date et l'heure de création de la classe.
classeGUID	Le GUID unique pour la classe actuelle.
classeHasConstructor	Examine la liste des méthodes de l' object actuel et, selon les conventions du langage actuel, renvoie T si l'une d'elles est un constructeur par défaut. Généralement utilisé avec la macro genOptGenConstructor.
classeHasCopyConstructor	Examine la liste des méthodes de l' object actuel et, selon les conventions du langage actuel, renvoie T si l'une d'elles est un constructeur de copie. Généralement utilisé avec la macro genOptGenCopyConstructor.
classeHasDestructor	Examine la liste des méthodes de l' object actuel et, selon les conventions du langage actuel, renvoie T si l'une d'elles est un destructeur. Généralement utilisé avec la macro genOptGenDestructor.
classeHasParent	Vrai, si la classe dans la portée a une ou plusieurs classes de base.

classeHasStereotype	<p>Vrai, si la classe dans la portée a un stéréotype qui correspond à un nom de stéréotype (que vous pouvez éventuellement spécifier comme entièrement qualifié). Il vérifie donc tous les stéréotypes qu'une classe possède et renvoie « T » si l'un d'entre eux correspond au stéréotype spécifié ou à une spécialisation de celui-ci. Par exemple :</p> <ul style="list-style-type: none"> <li>• <code>%classHasStereotype:"block"%</code> renverra 'T' pour toute classe stéréotypée par bloc de n'importe quelle version SysML, y compris <code>associationBlock</code></li> <li>• <code>%classHasStereotype:"SysML1.4::block"%</code> correspondra spécifiquement aux versions SysML 1.4</li> </ul> <p>Comparez ceci avec la classe <code>Stereotype</code>, plus tard.</p>
Importations de classe	dialogue « Code Gen » : Importations.
classeIsActive	dialogue de classe « Avancé » : case à cocher « Est Actif ».
classeIsAssociationClass	Vrai, si l'Association est un connecteur <code>AssociationClass</code> .
classeIsInstancié	Vrai, si la classe est une classe gabarit instanciée.
classeIsLeaf	dialogue de classe « Avancé » : case à cocher « Est une feuille ».
classeIsRoot	dialogue de classe « Avancé » : case à cocher « Est root ».
classeIsSpecification	dialogue de classe « Avancé » : case à cocher « Est-ce Spécification ».
classeMots-clés	Boîte dialogue Classe ' Propriétés ' : champ 'Mots clés'.
classeLangue	dialogue de la classe ' Propriétés ' : champ 'Langue'.
classeMacros	Une liste séparée par des espaces de macros définies pour la classe.
classeModifié	La date et l'heure de la dernière modification de la classe.
classeMultiplicité	dialogue de classe « Avancé » : Multiplicité.
nom de la classe	Boîte dialogue Classe ' Propriétés ' : champ 'Nom'.
Notes de cours	dialogue Classe ' Propriétés ' : champ ' Note '.
classeParamDefault	dialogue « Détail » de la classe.
nom du paramètre de classe	dialogue « Détail » de la classe.
classeParamType	dialogue « Détail » de la classe.
classePersistence	dialogue Classe ' Propriétés ' : Champ 'Persistence' (onglet 'Détails')
classePhase	Boîte dialogue Classe ' Propriétés ' : champ 'Phase'.
nomQualClasse	Le nom de la classe préfixé par ses classes externes. Les noms de classe sont séparés par deux points (::).

portée de la classe	dialogue Classe ' Propriétés ' : champ 'Scope'.
classeStéréotype	dialogue « Propriétés » de la classe : champ « Stéréotype ». Récupère le nom du premier stéréotype appliqué à la classe. Lorsqu'il est utilisé dans une comparaison, il vérifie si ce premier stéréotype correspond exactement à une string . Par exemple : %classStereotype=="enumeration" ? "enum" : "class"% Comparez ceci avec classHasStereotype, plus tôt.
Statut de la classe	Boîte dialogue Classe ' Propriétés ' : champ 'Statut'.
classeVersion	Boîte dialogue Classe ' Propriétés ' : champ 'Version'.

## Option de génération de code Macros de substitution de champ

Les macros de substitution de champ d'option de génération de code fonctionnent sur les options de génération de code source définies dans les pages « Ingénierie du code source » de :

- dialogue « Préférences » (' Démarrer > Apparence > Préférences > Préférences > Ingénierie du code source') pour les options spécifiques à l'utilisateur, ou
- dialogue « Gérer les options Modèle » (« Paramètres > Modèle > Options ») pour les options spécifiques au modèle

Pour plus d'informations sur la division des options, consultez la rubrique *Options d'ingénierie du code source* .

Les macros de substitution de champ sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide. Ce tableau répertorie chacune des macros de substitution de champ d'option de génération de code.

### Macros d'options de génération de code

Nom de la macro	Description
genOptActionScriptVersion	Page des spécifications ActionScript : version par défaut.
genOptCDefaultAttributeType	Page Spécifications C : Type d'attribut par défaut.
genOptCGenMethodNotesInBody	Page Spécifications C : Notes de méthode en cours d'implémentation.
genOptCGenMethodNotesInHeader	Page Spécifications C : Notes de méthode dans l'en-tête.
Notes de synchronisation genOptC	Page Spécifications C : Synchroniser Notes dans la génération.
fichier genOptCSynchCFile	Page Spécifications C : Synchroniser le fichier d'implémentation dans la génération.
genOptCDefaultSourceDirectory	Page des spécifications C : Répertoire source par défaut.
genOptCNamespaceDelimiter	Page de spécifications C : délimiteur Namespace .
genOptCOperationRefParam	Page Spécifications C : Référence comme paramètre de fonctionnement.
genOptCOperationRefParamStyle	Page Spécifications C : Style de paramètre de référence.
genOptCOperationRefParamName	Page Spécifications C : Nom du paramètre de référence.

genOptCConstructorName	Page Spécifications C : nom du constructeur par défaut.
genOptCDestructorName	Page des spécifications C : nom du destructeur par défaut.
genOptCPPCommentStyle	Page des spécifications C++ : Style de commentaire.
genOptCPPDefaultAttributeType	Page des spécifications C++ : Type d'attribut par défaut.
genOptCPPDefaultReferenceType	Page des spécifications C++ : Type de référence par défaut.
genOptCPPDefaultSourceDirectory	Page des spécifications C++ : Répertoire source par défaut.
genOptCPPGenMethodNotesInHeader	Page des spécifications C++ : case à cocher « Notes de méthode dans l'en-tête ».
genOptCPPGenMethodNotesInBody	Page des spécifications C++ : case à cocher Notes de méthode dans le corps.
genOptCPPGetPrefix	Page des spécifications C++ : obtenir le préfixe.
Extension de titre genOptCPPH	Page des spécifications C++ : extension d'en-tête.
genOptCPPSetPrefix	Page des spécifications C++ : définir le préfixe.
Extension de source genOptCPP	Page des spécifications C++ : extension de source.
Notes de synchronisation genOptCPPS	Page des spécifications C++ : Synchroniser Notes .
genOptCPPSynchCPPFile	Page des spécifications C++ : Synchroniser le fichier CPP.
genOptCSDefaultAttributeType	Page des spécifications C# : Type d'attribut par défaut.
Extension de source genOptCS	Page des spécifications C# : extension de fichier par défaut.
genOptCSGenDispose	Page Spécifications C# : Générer Dispose.
genOptCSGenFinalizer	Page Spécifications C# : Générer Finalizer.
genOptCSGenNamespace	Page Spécifications C# : Générer Namespace .
genOptCSDefaultSourceDirectory	Page des spécifications C# : Répertoire source par défaut.
genOptDefaultAssocAttName	Page d'ingénierie du code source : nom par défaut pour l'attribut associé.

me	
genOptDefaultConstructorScope	Page Durées de vie Object : visibilité du constructeur par défaut.
genOptDefaultCopyConstructorScope	Page Durées de vie Object : visibilité du constructeur de copie par défaut.
Base de données genOptDefault	Page Éditeurs de Code : Base de données par défaut.
genOptDefaultDestructorScope	Page Durées de vie Object : Visibilité du constructeur du destructeur par défaut.
genOptGenCapitalizedProperties	Page « Ingénierie du code source » : case à cocher « Mettre en majuscules les noms d'attributs pour Propriétés ».
genOptGenComments	Page 'Ingénierie du Code Source' : case à cocher 'Commentaires - Générer '.
genOptGenConstructeur	Page Durées de vie Object : case à cocher « Générer un constructeur ».
genOptGenConstructorInline	Page Durées de vie Object : case à cocher « Constructeur en ligne ».
genOptGenCopyConstructor	Page Durées de vie Object : case à cocher « Générer un constructeur de copie ».
genOptGenCopyConstructorInline	Page Durées de vie Object : case à cocher « Copier le constructeur en ligne ».
genOptGenDestructeur	Page Durées de vie Object : case à cocher « Générer un destructeur ».
genOptGenDestructeurInline	Page Durées de vie Object : case à cocher « Destructeur en ligne ».
genOptGenDestructeurVirtual	Page Durées de vie Object : case à cocher « Destructeur virtuel ».
genOptGenImplementedInterfaceOps	Page « Génération de code » : case à cocher « Générer des méthodes pour les interfaces implémentées ».
genOptGenPrefixBoolProperties	Page « Ingénierie du code source » : case à cocher « Utiliser « Est » pour la propriété booléenne Get() ».
genOptGenRoleNames	Page « Ingénierie du code source » : case à cocher « Générer automatiquement les noms de rôle lors de la création du code ».
genOptGenUnspecAssocDir	Page « Ingénierie du code source » : case à cocher « Ne pas générer de membres lorsque la direction de l'association n'est pas spécifiée ».
genOptJavaDefaultAttributeType	Page des spécifications Java : type d'attribut par défaut.



genOptJavaGetPrefix	Page des spécifications Java : obtenir le préfixe.
genOptJavaDefaultSource Directory	Page des spécifications Java : répertoire source par défaut.
genOptJavaSetPrefix	Page des spécifications Java : définir le préfixe.
Extension source genOptJava	Page des spécifications Java : extension du code source.
genOptPHPDefaultSource Directory	Page des spécifications PHP : Répertoire source par défaut.
genOptPHPGetPrefix	Page de spécifications PHP : obtenir le préfixe.
genOptPHPSetPrefix	Page de spécifications PHP : définir le préfixe.
Extension de source genOptPHP	Page de spécifications PHP : extension de fichier par défaut.
Version genOptPHP	Page des spécifications PHP : Version PHP.
Préfixe de propriété genOpt	Page « Ingénierie du code source » : supprimez les préfixes lors de la génération des propriétés Get/Set.
genOptVBMultiUse	Page Spécifications VB : case à cocher « Multi-usage ».
genOptVBPersistable	Page Spécifications VB : case à cocher « Persistant ».
Comportement de liaison de données genOptVB	Page des spécifications VB : case à cocher « Comportement de liaison de données ».
Comportement de genOptVBDataSource	Page Spécifications VB : case à cocher « Comportement de la source de données ».
genOptVBGlobal	Page des spécifications VB : case à cocher « Espace de noms global ».
genOptVBCréable	Page des spécifications VB : case à cocher « Créable ».
genOptVBExposed	Page Spécifications VB : case à cocher « Exposé ».
genOptVBMTS	Page des spécifications VB : mode de transaction MTS.
genOptVBNetGenNamespace	Page Spécifications VB.Net : Générer Namespace .
Version genOptVB	Page des spécifications VB : version par défaut.
Commentaire genOptWrap	Page « Ingénierie du code source » : longueur d'encapsulation pour les lignes de commentaires.



## Macros de substitution de champs de connecteur

Ce tableau répertorie chacune des macros de substitution de champ de connecteur.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros de connecteur

Nom de la macro	Description
connecteurAlias	dialogue ' Propriétés ' du connecteur : champ 'Alias'.
connecteurAssociationClassesElemGUID	Le GUID de l'élément de classe d'association du connecteur.
connecteurAssociationClassesElemName	Le nom de l'élément de classe d'association du connecteur.
connecteurDestAccess	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Accès.
connecteurDestAggregation	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Agrégation.
connecteurDestAlias	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Alias.
connecteurDestAllowDuplicates	dialogue « Propriétés » du connecteur, onglet « Rôle cible » : case à cocher « Autoriser les doublons ».
connecteurDestChangeable	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Modifiable.
connecteurDestConstraint	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Contrainte(s).
connecteurDestContainment	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Confinement.
connecteurDestDerived	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : case à cocher 'Dérivé'.
connecteurDestDerivedUnion	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : case à cocher 'DerivedUnion'.
connecteurDestElem*	Un ensemble de macros qui accèdent à une propriété de l'élément à l'extrémité cible d'un connecteur. Le * (astérisque) est un caractère générique qui correspond à n'importe quelle macro de substitution de classe dans la liste des macros de classe. Par exemple : <ul style="list-style-type: none"> <li>connecteurDestElemAlias (classAlias)</li> <li>connecteurDestElemAuthor (classAuthor)</li> </ul>
connecteurDestElemType	Le type d'élément de l'élément de destination du connecteur. (Séparé des macros connectorDestElem* car il n'y a pas de macro de substitution classType.)

connecteurDestFeature*	<p>Un ensemble de macros qui accèdent à une propriété de la fonctionnalité à l'extrémité cible d'un connecteur. Le * (astérisque) est un caractère générique qui correspond à toute macro de substitution d'attribut ou d'opération dans la liste des macros d'attribut ou d'opération, en fonction du connectorDestFeatureType.</p> <p>Par exemple:</p> <ul style="list-style-type: none"> <li>• connectorDestFeatureReturnClassifierGUID - GUID du classificateur de retour d'une opération</li> <li>• connectorDestFeatureContainment - confinement d'un attribut</li> </ul>
connecteurDestFeatureType	<p>Le type de fonctionnalité de destination du connecteur.</p> <ul style="list-style-type: none"> <li>• connectorDestFeatureType="Attribut" ou "Opération"</li> </ul>
connecteurDestMemberType	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Type de membre .
connecteurDestMultiplicité	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Multiplicité.
connecteurDestNavigabilité	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Navigabilité.
connecteurDestNotes	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Notes sur le rôle .
connecteurDestOrdered	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : case à cocher 'Ordonné'.
connecteurDestOwned	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : case à cocher 'Possédé'.
connecteurDestQualifier	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Qualificateur(s).
connecteurDestRole	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Rôle.
connecteurDestScope	dialogue ' Propriétés ' du connecteur, onglet 'Rôle cible' : Périmètre cible.
connecteurDestStereotype	Connecteur dialogue ' Propriétés ', onglet ' Rôle cible ' : Stéréotype.
connecteurDirection	Propriétés du Connecteur : Direction.
connecteurEffet	dialogue « Contraintes de transition » : champ « Effet ».
connecteurGuard	Boîtes de dialogue « Flux Object » et « Contraintes de transition » : champ « Garde ».
connecteurGUID	Le GUID unique pour le connecteur actuel.
connecteurIsAssociationClass	Vrai, si le connecteur est un connecteur AssociationClass.
nom du connecteur	Propriétés du Connecteur : Nom.
connecteurNotes	Propriétés du Connecteur : Notes .
connecteurSourceAccess	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Accès.

connecteurSourceAggregation	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Agrégation.
connecteurSourceAlias	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Alias.
connecteurSourceAllowDuplicates	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : case à cocher Autoriser les doublons.
connecteurSourceChangeable	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Modifiable.
connecteurSourceConstraint	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Contrainte(s).
connecteurSourceContainment	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Confinement.
connecteurSourceDérivé	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : case à cocher 'Dérivé'.
connecteurSourceDerivedUnion	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : case à cocher ' DerivedUnion '.
connecteurSourceElem*	<p>Un ensemble de macros qui accèdent à une propriété de l'élément à l'extrémité source d'un connecteur. Le * (astérisque) est un caractère générique qui correspond à n'importe quelle macro de substitution de classe dans la liste des macros de classe. Par exemple :</p> <ul style="list-style-type: none"> <li>• connecteurSourceElemAlias (classAlias)</li> <li>• connecteurSourceElemAuthor (classAuthor)</li> </ul>
connecteurSourceElemType	Le type d'élément de l'élément source du connecteur. (Séparé des macros connectorSourceElem* car il n'y a pas de macro de substitution classType.)
connecteurSourceFonctionnalité*	<p>Un ensemble de macros qui accèdent à une propriété de la fonctionnalité à l'extrémité source d'un connecteur. Le * (astérisque) est un caractère générique qui correspond à toute macro de substitution d'attribut ou d'opération dans la liste des macros d'attribut ou des macros d'opération, en fonction du connectorSourceFeatureType. Par exemple :</p> <ul style="list-style-type: none"> <li>• connectorSourceFeatureCode - Code de l'opération</li> <li>• connectorSourceFeatureInitial - Initiale de l'attribut</li> </ul>
connecteurSourceFeatureType	<p>Le type de fonctionnalité source du connecteur.</p> <ul style="list-style-type: none"> <li>• connectorSourceFeatureType="Attribut" ou "Opération"</li> </ul>
connecteurSourceMemberType	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Type de membre .
connecteurSourceMultiplicité	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Multiplicité.
connecteurSourceNavigabilité	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Navigabilité.
connecteurSourceNotes	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Notes sur le rôle .

connecteurSourceCommandé	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : case à cocher 'Ordonné'.
connecteurSourcePropriété	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : case à cocher 'Détenu'.
connecteurSourceQualifier	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Qualificateur(s).
connecteurSourceRole	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Rôle.
connecteurSourceScope	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Périmètre cible.
connecteurSourceStéréotype	dialogue ' Propriétés ' du connecteur, onglet 'Rôle source' : Stéréotype.
connecteurStéréotype	dialogue ' Propriétés ' du connecteur : champ 'Stéréotype'.
connecteurTrigger	dialogue 'Contraintes de Transition' : champ ' Déclencheur '.
Type de connecteur	Le type de connecteur ; par exemple, Association ou Généralisation.
connecteurPoids	dialogue « Contraintes de flux Object » : champ « Poids ».

## Macros de substitution de champs de contraintes

Ce tableau répertorie chacune des macros de substitution de champ « Contrainte ».

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros de contraintes

Nom de la macro	Description
nom de contrainte	dialogue « Classe », onglet « Contraintes » : Nom.
Notes sur les contraintes	dialogue « Classe », onglet « Contraintes » : Notes .
Statut de contrainte	dialogue « Classe », onglet « Contraintes » : Statut.
type de contrainte	dialogue 'Classe', onglet 'Contraintes' : Type .
contraintePoids	dialogue 'Classe', onglet 'Contraintes' : touches de classement (main en haut/en bas).

## Macros de substitution de champs d'effort

Ce tableau répertorie chacune des macros de substitution de champ « Effort ».

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros d'effort

Nom de la macro	Description
effortNom	Fenêtre d'effort : Effort.
effortNotes	Fenêtre d'effort : Notes (sans étiquette).
effortTemps	Fenêtre d'effort : Temps.
effortType	Fenêtre d'effort : Type .



## Macros de substitution de champs de fichiers

Ce tableau répertorie chacune des macros de substitution de champ de fichier.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros de fichiers

Nom de la macro	Description
extension de fichier	L'extension du type de fichier du fichier en cours de génération.
nom de fichier	Le nom du fichier en cours de génération.
nom_fichierImpl	Le nom du fichier d'implémentation pour cette génération, le cas échéant.
en-têtes de fichier	dialogue « Code Gen » : en-têtes.
Importations de fichiers	dialogue « Code Gen » : Importations. Pour les langues prises en charge, cela inclut également les dépendances dérivées de ces types de relations : <ul style="list-style-type: none"><li>• Agrégation</li><li>• Association</li><li>• Classificateur d'attributs</li><li>• Type de retour de la méthode</li><li>• Classificateur de paramètres de méthode</li><li>• Généralisation</li><li>• Réalisation (à l'interface)</li><li>• Liaison Gabarit (C++)</li><li>• Dépendance</li></ul>
chemin du fichier	Le chemin complet du fichier en cours de génération.
chemin du fichierImpl	Le chemin complet du fichier d'implémentation pour cette génération, le cas échéant.

## Macros de substitution de champs d'importation de fichiers

Ce tableau répertorie chacune des macros de substitution de champ d'importation de fichier.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de T si la case est sélectionnée. Sinon, la valeur est vide.

### Macros d'importation de fichiers

Nom de la macro	Description
Nom de la classe d'importation	Le nom de la classe en cours d'importation.
importer le nom du fichier	Le nom de fichier de la classe en cours d'importation.
importerFilePath	Le chemin complet de la classe en cours d'importation.
importerDeL'agrégation	T si la classe possède un connecteur d'agrégation vers une classe dans ce fichier, F sinon.
importerDeL'Association	T si la classe a un connecteur d'association vers une classe dans ce fichier, F sinon.
importerDeAtt	T si un attribut d'une Classe dans le fichier courant est du type de cette Classe, F sinon.
importerDeDépendance	T si la classe a un connecteur de dépendance vers une classe dans ce fichier, F sinon.
importerDeGénéralités	T si la classe a un connecteur de généralisation vers une classe dans ce fichier, F sinon.
importerDeMeth	T si le type de retour d'une méthode d'une classe dans le fichier courant est le type de cette classe, F sinon.
importerDeParam	T si un paramètre de méthode d'une classe dans le fichier courant est du type de cette classe ; sinon F.
importerDePropertyType	T si la classe possède une propriété (Partie/Port) typant une autre classe, F sinon.
importerDeRéalisation	T si la classe possède un connecteur de réalisation vers une classe dans ce fichier, F sinon.
importerFromTemplateBinding	T si la classe a un connecteur TemplateBinding vers une classe dans ce fichier, F sinon.
importerDansFichier	T si la classe est dans le fichier courant, F sinon.
importerPackagePath	Le chemin Paquetage avec un séparateur « . » de la classe en cours d'importation.

---

ImporterRelativeFilePath	Le chemin d'accès relatif au fichier de la classe en cours d'importation à partir du chemin d'accès au fichier en cours de génération.
--------------------------	--

## Macros de substitution de champs de liens

Si vous souhaitez donner accès aux données concernant les connecteurs du modèle, en particulier les associations et les généralisations, vous pouvez utiliser les macros « Substitution de champ de lien ». Les noms des macros sont en majuscules Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée ; sinon, la valeur est vide.

### Liens macros

Nom de la macro	Description/Résultat
lienAttAccess	dialogue ' Propriétés ' de l'association, Rôle cible : champ 'Accès'.
lienAttAggregation	Association dialogue ' Propriétés ', Rôle Source ou Cible : Agrégation.
lienAttCollectionClass	La collection appropriée à l'attribut lié dans la portée.
lienAttContainment	Association ' Propriétés ' dialogue , Rôle cible : Confinement.
lienAttName	dialogue 'Association Propriétés ' : Cible.
lienAttNotes	dialogue ' Propriétés ' de l'association, Rôle cible : Notes sur le rôle .
lienAttOwnedByAssociation	Vrai, si la case à cocher « Propriété » sur la page « Rôle(s) » de la dialogue « Propriétés » de l'Association n'est pas sélectionnée.
lienAttOwnedByClass	Vrai, si la case à cocher « Propriété » sur la page « Rôle(s) » de la dialogue « Propriétés » de l'Association est sélectionnée.
lienAttQualName	La cible de l'association qualifiée par le chemin de l'espace de noms (si vous générez des espaces de noms) et le chemin du classificateur (délimité par des points).
lienAttRole	dialogue ' Propriétés ' de l'association, Rôle cible : Rôle.
lienAttRoleAlias	dialogue 'Association Propriétés Rôle Cible' : Alias
lienAttStereotype	Association dialogue ' Propriétés ', Rôle cible : Stéréotype.
lienAttTargetScope	dialogue ' Propriétés ' de l'association, Rôle cible : Périmètre cible.
lienCarte	Lien dialogue ' Propriétés ', Rôle cible : Multiplicité.
lienGUID	Le GUID unique pour le connecteur actuel.
lienIsAssociationClass	Vrai, si l'Association est un connecteur AssociationClass.
lienEstLié	Renvoie T si des TemplateBindings sont spécifiés sur le connecteur.
lienParamSubs	Renvoie une liste séparée par des virgules des arguments spécifiés.

lienParentName	Généralisation dialogue ' Propriétés ' : champ 'Cible'.
lienParentQualName	La cible de généralisation qualifiée par le chemin de l'espace de noms (si vous générez des espaces de noms) et le chemin du classificateur (délimité par des points).
lienStéréotype	Le stéréotype du connecteur actuel.
lienHéritage Virtuel	Généralisation dialogue ' Propriétés ' : champ 'Héritage Virtuel'.

## Macros de substitution de champs de fichiers liés

Ce tableau répertorie chacune des macros de substitution de champ « Fichier lié ».

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros de fichiers liés

Nom de la macro	Description
fichier liéDernière écriture	dialogue Classe ' Propriétés ' : onglet 'Fichiers', champ 'Dernière écriture'.
Notes de fichiers liés	dialogue Classe ' Propriétés ' : onglet ' Fichiers ', champ ' Notes '.
chemin du fichier lié	dialogue Classe ' Propriétés ' : onglet 'Fichiers', champ 'Chemin du fichier'.
Taille du fichier lié	dialogue Classe ' Propriétés ' : onglet 'Fichiers', champ 'Taille'.
Type de fichier lié	dialogue Classe ' Propriétés ' : onglet ' Fichiers ', champ ' Type '.

## Macros de substitution de champs métriques

Ce tableau répertorie chacune des macros de substitution de champ métrique.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros métriques

Nom de la macro	Description
nom métrique	Écran Métriques : champ « Métrique ».
Notes métriques	Écran Métriques : champ ( Notes ).
type métrique	Écran métriques : champ « Type ».
Poids métrique	Écran de métriques : champ « Poids ».

## Macros de substitution de champs d'opération

Les macros 'Substitution de champs d'opération' permettent d'accéder aux données concernant les opérations dans le modèle. Les noms des macros sont en majuscules Camel. Les macros qui représentent des cases à cocher renvoient une valeur de 'T' si la case est sélectionnée ; sinon la valeur est vide.

### Macros de substitution de champs d'opération

Nom de la macro	Description/Résultat
opRésumé	dialogue « Opération » : case à cocher « Virtuel ».
opAlias	dialogue « Opération » : Alias.
comportementop	dialogue « Opération Comportement » : Comportement.
Code d'opération	dialogue « Opération Comportement » : Code de comportement.
OpConcurrence	dialogue « Opération » : Concurrence.
opConst	dialogue « Opération » : case à cocher « Const ».
GUID d'opération	Le GUID unique pour l'opération en cours.
opHasSelfRefParam	Analyse la liste des paramètres dans l'opération en cours, renvoyant « T » si un type est la référence de classe (cela peut être ClassA* ou ClassA&, selon la valeur de la macro de substitution de champ d'option de génération de code genOptCOperationRefParamStyle).
opImplMacros	Une liste séparée par des espaces de macros définies dans l'implémentation de cette opération.
opIsRequête	Boîte de dialogue « Opération » : case à cocher « IsQuery ».
macros opérationnelles	Une liste séparée par des espaces de macros définies dans la déclaration pour cette opération.
nom de l'opération	dialogue « Opération » : Nom.
Notes d'opération	dialogue « Opération » : Notes .
opPure	dialogue « Opération » : case à cocher « Pure ».
opReturnArray	Boîte de dialogue « Opération » : case à cocher « Renvoyer le tableau ».
opReturnClassifierGUID	Le GUID unique pour le classificateur de l'opération en cours.
opReturnQualType	Le type de retour de l'opération qualifié par le chemin de l'espace de noms (en cas de génération d'espaces de noms) et le chemin du classificateur (délimité par des points). Si le classificateur de type de retour n'a pas été défini, il est équivalent à la



	macro opReturnType.
opReturnType	dialogue « Opération » : Type de retour .
portée opérationnelle	dialogue « Opération » : Portée.
opStatique	dialogue « Opération » : case à cocher « Statique ».
opStéréotype	dialogue « Opération » : Stéréotype.
opSynchronisé	dialogue « Opération » : case à cocher « Synchronisé ».

## Macros de substitution de champs de Paquetage

Ce tableau répertorie les macros de substitution de champ Paquetage .

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Paquetage de macros

Nom de la macro	Description
packageRésumé	dialogue ' Paquetage ' : Résumé.
Alias de paquet	dialogue ' Paquetage ' : Alias.
packageAuteur	dialogue ' Paquetage ' : Auteur.
Complexité du package	dialogue ' Paquetage ' : Complexité.
packageGUID	Le GUID unique pour le Paquetage actuel.
packageMots-clés	dialogue ' Paquetage ' : Mots-clés.
packageLangue	dialogue ' Paquetage ' : Langue.
nom du package	dialogue ' Paquetage ' : Nom.
chemin du paquet	La string représentant la hiérarchie des Paquetages , pour la classe concernée. Chaque nom Paquetage est séparé par un point (.).
paquetPhase	dialogue ' Paquetage ' : Phase.
portée du paquet	dialogue ' Paquetage ' : Portée.
Statut du paquet	dialogue ' Paquetage ' : Statut.
paquetStéréotype	dialogue ' Paquetage ' : Stéréotype.
Version du paquet	dialogue ' Paquetage ' : Version.

## Macros de substitution de champs de paramètres

Ce tableau répertorie chacune des macros de substitution de champ de paramètre.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros de paramètres

Nom de la macro	Description
paramètreClassifierGUID	Le GUID unique pour le classificateur du paramètre actuel.
paramètre par défaut	Opération dialogue « Paramètres » : champ « Par défaut ».
paramètreFixé	Boîte de dialogue « Paramètres » : case à cocher « Fixe ».
paramètreGUID	Le GUID unique pour le paramètre actuel.
paramètreIsEnum	Vrai, si le paramètre utilise le mot-clé enum (C++).
paramètreKind	Opération dialogue « Paramètres » : champ « Type ».
nom_paramètre	Opération dialogue « Paramètres » : champ « Nom ».
Notes de paramètre	Opération dialogue « Paramètres » : champ « Notes ».
paramètreQualType	Le type de paramètre qualifié par le chemin de l'espace de noms (en cas de génération d'espaces de noms) et le chemin du classificateur (délimité par des points). Si le classificateur de paramètres n'a pas été défini, est équivalent à la macro paramType.
paramètreType	Opération dialogue « Paramètres » : champ « Type ».

## Macros de substitution de champs problématiques

Ce tableau répertorie chacune des macros de substitution de champ Problème.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Problèmes de macros

Nom de la macro	Description
problèmeTerminé par	dialogue « Maintenance », onglet « Problèmes d'éléments » : Terminé par.
problèmeCompletedDate	dialogue « Maintenance », onglet « Problèmes d'élément » : Terminé.
problèmeHistoire	dialogue « Maintenance », onglet « Problèmes d'éléments » : Historique.
nom du problème	dialogue « Maintenance », onglet « Problèmes d'élément » : Nom.
ProblèmeNotes	dialogue « Maintenance », onglet « Problèmes d'éléments » : Description.
problèmePriorité	dialogue « Maintenance », onglet « Problèmes d'éléments » : Priorité.
problèmeSoulevé par	dialogue « Maintenance », onglet « Problèmes d'élément » : soulevé par.
problèmeRaisedDate	dialogue « Maintenance », onglet « Problèmes d'élément » : surélevé.
Statut du problème	dialogue « Maintenance », onglet « Problèmes d'élément » : Statut.
problèmeVersion	dialogue « Maintenance », onglet « Problèmes d'élément » : Version.

## Macros de substitution de champs d'exigences

Ce tableau répertorie chacune des macros de substitution de champ d'exigence avec une description du résultat.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros d'exigences

Nom de la macro	Description
exigenceDifficulté	dialogue ' Propriétés ' : Onglet 'Exiger' : Difficulté.
exigenceDernière mise à jour	dialogue ' Propriétés ' : Onglet 'Exiger' : Dernière mise à jour.
exigenceNom	dialogue ' Propriétés ' : Onglet 'Exiger' : Brève description.
ExigenceRemarques	dialogue ' Propriétés ' : Onglet 'Exiger' : Notes .
exigencePriorité	dialogue ' Propriétés ' : Onglet 'Exiger' : Priorité.
exigenceStatut	dialogue ' Propriétés ' : Onglet 'Exiger' : Statut.
exigenceType	Boîte dialogue ' Propriétés ' : Onglet 'Exiger' : Type .

## Macros de substitution de champs de ressources

Ce tableau répertorie chacune des macros de substitution de champ de ressource.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros de ressources

Nom de la macro	Description
ressourceAllocatedTime	Fenêtre Allocation des ressources : Alloué Time.
Date de fin de la ressource	Fenêtre d'allocation des ressources : date de fin.
ressourceExpectedTime	Fenêtre d'allocation des ressources : temps prévu.
ressourceTemps dépensé	Fenêtre d'allocation des ressources : temps dépensé.
Histoire des ressources	Fenêtre d'allocation des ressources : Historique.
nom de la ressource	Fenêtre d'allocation des ressources : Ressource.
Notes sur les ressources	Fenêtre d'allocation des ressources : description.
ressourcePourcentage terminé	Fenêtre d'allocation des ressources : Terminée (%).
rôle de ressource	Fenêtre d'allocation des ressources : Rôle.
Date de début de la ressource	Fenêtre Allocation des ressources : Date Démarrer .

## Macros de substitution de champs de risque

Ce tableau répertorie chacune des macros de substitution de champ de risque.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macro-risques

Nom de la macro	Description
nom du risque	Fenêtre Risques : Risque.
Notes sur les risques	Fenêtre Risques : ( Notes ).
Type de risque	Fenêtre Risques : Type .
risquePoids	Fenêtre Risques : Poids.

## Macros de substitution de champs de scénario

Ce tableau répertorie chacune des macros de substitution de champ de scénario avec une description du résultat.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Macros de scénario

Nom de la macro	Description
scénarioGUID	L' ID unique d'un scénario. Identifie le scénario sans ambiguïté dans un modèle.
nom du scénario	dialogue ' Propriétés ', onglet 'Scénario' : Scénario.
Notes de scénario	Boîte dialogue ' Propriétés ', onglet 'Scénario' : ( Notes ).
Type de scénario	Boîte dialogue ' Propriétés ', onglet 'Scénario' : Type .



# Macros de substitution Valeur Étiquetée

Les macros Valeur Étiquetée sont une forme spéciale de macros de substitution de champ, qui donnent accès aux étiquettes d'éléments et aux Valeur Étiquetés correspondantes. Ils peuvent être utilisés de deux manières :

- Substitution directe
- Substitution conditionnelle

## Substitution directe

Cette forme de macro remplace directement la valeur de l' étiquette nommée dans la sortie.

Structure : %<macroName>:"<tagName>"%

<macroName> peut être l'un des suivants :

- Étiquette d'att
- Étiquette de classe
- connecteurDestElemTag
- connecteurDestTag
- connecteurSourceElemTag
- connecteurSourceTag
- connecteurTag
- lienAttTag
- lienTag
- balise d'op
- PaquetTag
- paramètreTag

Cela correspond aux étiquettes pour les attributs, les classes, les opérations, Paquetages , les paramètres, les connecteurs avec les deux extrémités, les éléments aux deux extrémités des connecteurs et les connecteurs incluant l'extrémité de l'attribut.

<tagName> est une string représentant le nom étiquette spécifique.

## Exemple

```
%opTag:"attribut"%
```

## Substitution conditionnelle

Cette forme de macro imite la substitution conditionnelle définie pour les macros de substitution de champ.

Structure : %<macroName>:"<tagName> " (== "<test> " ) ? <subTrue> (: <subFalse>) %

Note :

- <macroName> et <tagName> sont tels que définis ici
- (<texte>) indique que <texte> est facultatif
- <test> est une string représentant une valeur possible pour la macro

- `<subTrue>` et `<subFalse>` peuvent être une combinaison de chaînes entre guillemets et du mot-clé valeur ; lorsque la valeur est utilisée, elle est remplacée par la valeur de la macro dans la sortie

## Exemples

```
%opTag : "opInline" ? "en ligne" : " "%
```

```
%opTag:"opInline" ? "inline"%
```

```
%classTag:"unsafe" == "true" ? "unsafe" : " "%
```

```
%classTag:"unsafe" == "true" ? "unsafe"%
```

Les macros Valeur Étiquetée utilisent la même convention de dénomination que les macros de substitution de champs.

## Macros de substitution de paramètres Gabarit

Si vous souhaitez donner accès dans un gabarit de transformation aux données concernant la transformation de la substitution de paramètres de liaison d'un connecteur Gabarit Binding dans le modèle, vous pouvez utiliser les macros de substitution de paramètres Gabarit . Les noms des macros sont en majuscules Camel. Les macros qui représentent des cases à cocher renvoient une valeur de 'T' si la case est sélectionnée ; sinon, la valeur est vide.

### Macros de substitution de paramètres Gabarit

Nom de la macro	Description
paramètreSubstitutionFormelle	dialogue « Propriétés de liaison Gabarit , onglet « Paramètre de liaison », panneau « Substitution(s) de paramètres » : Nom formel du paramètre Gabarit .
paramètreSubstitutionActuel	dialogue « Propriétés de liaison Gabarit , onglet « Paramètre de liaison », panneau « Substitution(s) de paramètre » : nom/expression du paramètre réel.
paramètreSubstitutionActuelClassifier	dialogue ' Gabarit Binding Propriétés ', onglet 'Paramètre de liaison', panneau 'Substitution(s) de paramètres' : classificateur de paramètres réel.

## Macros de substitution de champs Test

Ce tableau répertorie chacune des macros de substitution de champ Test avec une description du résultat.

Les macros de substitution de champs sont nommées selon la casse Camel. Les macros qui représentent des cases à cocher renvoient une valeur de « T » si la case est sélectionnée. Sinon, la valeur est vide.

### Test les macros

Nom de la macro	Description
Critères d'acceptation du test	dialogue Tester : Critères d'acceptation.
testVérifié par	Fenêtre Cas Test : Vérifié par.
Date d'exécution du test	Fenêtre Cas Test : Dernier Exécuter .
classe de test	Fenêtre Cas Test : Classe Test (le type de test défini : Unité, Intégration, Système, Acceptation, Inspection, Scénario)
testEntrée	dialogue Tester : Entrée.
nom du test	Fenêtre Cas Test : Test .
Notes de test	Fenêtre Cas Test : Description.
Résultats du test	dialogue Tester : Résultats.
testExécuté par	Fenêtre Cas Test : Exécuter par. (Les valeurs sont dérivées des définitions de l'auteur du projet dans la dialogue « Personnes » - « Paramètres > Données de référence > Types Modèle > Personnes > Auteurs du projet ».)
Statut du test	Fenêtre Cas Test : État.
Type de test	Fenêtre Cas Test : Type .

## Macros de fonctions

Les macros de fonction sont un moyen pratique de manipuler et de formater divers éléments de données. Chaque macro de fonction renvoie une string de résultat. Il existe deux manières principales d'utiliser les résultats des macros de fonction :

- Substitution directe de la string renvoyée dans la sortie, par exemple : `%TO_LOWER(attName)%`
- Stockage de la string renvoyée dans le cadre d'une définition de variable telle que : `$name = %TO_LOWER(attName)%`

Les macros de fonction peuvent prendre des paramètres, qui peuvent être transmis aux macros comme suit :

- Littéraux String , placés entre guillemets doubles
- Macros de substitution directe sans les signes de pourcentage qui les entourent
- Références variables
- Littéraux numériques

Plusieurs paramètres sont transmis à l'aide d'une liste séparée par des virgules.

Les macros de fonction sont nommées selon le style Tout en majuscules, comme dans :

`%CONVERT_SCOPE(opScope)%`

Les macros de fonctions disponibles sont décrites ici. Les paramètres sont indiqués par des crochets, comme dans :

`FONCTION_NOM([param]).`

### CONVERTIR\_SCOPE([umlScope])

À utiliser avec les langues prises en charge pour convertir [umlScope] en mot-clé de portée approprié pour la langue en cours de génération. Ce tableau montre la conversion de [umlScope] par rapport à la langue donnée.

Langue	Conversions
C++	Paquetage ==> public Public ==> public Privé ==> privé Protégé ==> protégé
C#	Paquetage ==> interne Public ==> public Privé ==> privé Protégé ==> protégé
Delphes	Paquetage ==> protégé Public ==> public Privé ==> privé Protégé ==> protégé
Java	Paquetage ==> {vide} Public ==> public Privé ==> privé Protégé ==> protégé

PHP	Paquetage ==> public Public ==> public Privé ==> privé Protégé ==> protégé
VB	Paquetage ==> Protégé Public ==> Public Privé ==> Privé Protégé ==> Protégé
VB .Net	Paquetage ==> Ami Public ==> Public Privé ==> Privé Protégé ==> Protégé

## COLLECTION\_CLASS([langue])

Donne la classe de collection appropriée pour la langue spécifiée pour l'attribut lié actuel.

## CSTYLE\_COMMENT([longueur\_enveloppe])

Convertit les notes de l'élément actuellement dans la portée en commentaires simples de style C, en utilisant /\* et \*/.

## DELPHI\_PROPERTIES([portée], [separator], [retrait])

Génère une propriété Delphi.

## DELPHI\_COMMENT([wrap\_length])

Convertit les notes de l'élément actuellement concerné en commentaires Delphi.

## EXEC\_ADD\_IN([nom\_fonction],, ...)

Appelle une fonction Add-In Enterprise Architect, qui peut renvoyer une string de résultat.

[addin\_name] et [function\_name] spécifient les noms du Add-In et de la fonction à appeler.

Les paramètres de la fonction Add-In peuvent être spécifiés via les paramètres [prm\_1] à [prm\_n].

```
$result = %EXEC_ADD_IN("MonAddin", "ProcessOperation", classGUID, opGUID)%
```

Toute fonction appelée par la macro EXEC\_ADD\_IN doit avoir deux paramètres : un objet EA.Repository et un tableau Variant contenant tous les paramètres supplémentaires de l'appel EXEC\_ADD\_IN. Le type de retour doit être Variant.

Fonction publique ProcessOperation( Référentiel As EA.Repository, args As Variant) As Variant

## RECHERCHER([src], [sous-chaîne])

Position de la première instance de [subString] dans [src] ; -1 si aucune.

## OBTENIR\_ALIGNEMENT()

Renvoie une string dans laquelle tout le texte de la ligne de sortie actuelle est converti en espaces et tabulations.

## JAVADOC\_COMMENT([wrap\_length])

Convertit les notes de l'élément actuellement dans la portée en commentaires de style javadoc.

## GAUCHE([src], [compte])

Les [count] premiers caractères de [src].

## LONGUEUR([src])

Longueur de [src]. Renvoie une string .

## MATH\_ADD(x,y) MATH\_MULT(x,y) et MATH\_SUB(x,y)

Dans un gabarit de code ou gabarit DDL, ces trois macros exécutent respectivement les fonctions mathématiques de :

- Addition ( $x+y$ )
- Multiplication ( $x*y$ ) et
- Soustraction ( $xy$ )

Les arguments  $x$  et  $y$  peuvent être des entiers ou des variables, ou une combinaison des deux. Considérez ces exemples, tels qu'ils sont utilisés dans un gabarit « Class » pour la génération de code C++ :

- $\$a = \%MATH\_ADD(3,4)\%$
- $\$b = \%MATH\_SUB(10,3)\%$
- $\$c = \%MATH\_MULT(2,3)\%$
- $\$d = \%MATH\_ADD(\$a,\$b)\%$
- $\$e = \%MATH\_SUB(\$b,\$c)\%$
- $\$f = \%MATH\_MULT(\$a,\$b)\%$
- $\$g = \%MATH\_MULT(\$a,10)\%$
- $\$h = \%MATH\_MULT(10,\$b)\%$

Ceux-ci calculent, dans la même séquence, à :

- $a = 3 + 4 = \$a$

- $b = 10 - 3 = \$b$
- $c = 2 * 3 = \$c$
- $d = a + b = \$d$
- $e = b - c = \$e$
- $f = a * b = \$f$
- $g = a * 10 = \$g$
- $h = 10 * b = \$h$

Lorsque le code est généré, le fichier .h (pour C++) contient ces chaînes correspondantes :

- $a = 3 + 4 = 7$
- $b = 10 - 3 = 7$
- $c = 2 * 3 = 6$
- $d = a + b = 14$
- $e = b - c = 1$
- $f = a * b = 49$
- $g = a * 10 = 70$
- $h = 10 * b = 70$

## MID([src], [début]) MID([src], [début], [nombre])

Sous-chaîne de [src] commençant à [start] et incluant [count] caractères. Lorsque [count] est omis, le reste de la string est inclus.

## PI([option], [ valeur ], {[option], [ valeur ]})

Ensembles le PI pour le gabarit actuel à [ valeur ]. Les valeurs valides pour [ valeur ] sont :

- "\n"
- " "
- « «
- « »

<option> contrôle le moment où le nouveau PI prend effet. Les valeurs valides pour <option> sont :

- I, Immédiat : le nouveau PI est généré avant la prochaine ligne gabarit non vide
- N, Suivant : le nouveau PI est généré après la prochaine ligne gabarit non vide

Plusieurs paires d'options sont autorisées dans un appel. Un exemple de situation où cette fonction serait utilisée est celle où un mot clé est toujours sur une nouvelle ligne, comme illustré ici :

```
%PI=" "%
%classAbstract ? "abstrait"%
%si classTag:"macro" != " "%
%PI("Je", "\n", "N", " ")%
%classTag:"macro"%
%finSi%
classe
```



%Nom de classe%

Pour plus de détails, voir *la macro Instruction de traitement (PI)* .

## **PROCESS\_END\_OBJECT([nom\_modèle])**

Permet aux classes qui sont une classe plus éloignée de la classe de base, d'être transformées en objets (tels que des attributs, des opérations, Paquetages , des paramètres et des colonnes) de la classe de base. [template\_name] fait référence au gabarit de travail qui stocke temporairement les données.

## **SUPPRIMER LES DUPLICATS([source], [separator] )**

Où [source] est une liste séparée par [separator] ; cela supprime toutes les chaînes en double ou vides.

## **REEMPLACER([ string ], [ancien], [nouveau])**

Remplace toutes les occurrences de [old] par [new] dans la string donnée <string>.

## **RESOLVE\_OP\_NAME()**

Résout les conflits dans les noms d'interface lorsque deux interfaces de méthode ont le même nom.

## **RESOLVE\_QUALIFIED\_TYPE() RESOLVE\_QUALIFIED\_TYPE( [separator] ) RESOLVE\_QUALIFIED\_TYPE( [separator] , [par défaut])**

Génère un type qualifié pour l'attribut actuel, l'attribut lié, le parent lié, l'opération ou le paramètre. Permet la spécification d'un séparateur autre que . et une valeur par défaut lorsque certaines valeur sont requises.

## **DROITE([src], [compte])**

Les [count] derniers caractères de [src].

## **TO\_LOWER([ string ])**

Convertit [ string ] en minuscules.

## **TO\_UPPER([ string ])**

Convertit [ string ] en majuscules.

## **TRIM([ string ]) TRIM([ string ], [trimChars])**

Supprime les espaces blancs de début et de fin de [ string ]. Si [trimChars] est spécifié, tous les caractères de début et de fin de l'ensemble de <trimChars> sont supprimés.

## **TRIM\_LEFT([ string ]) TRIM\_LEFT([ string ], [trimChars])**

Supprime les caractères de début spécifiés de <string>.

## **TRIM\_RIGHT([ string ]) TRIM\_RIGHT([ string ], [trimChars])**

Supprime les caractères de fin spécifiés de <string>.

## **VB\_COMMENT([longueur\_d'enveloppement])**

Convertit les notes de l'élément actuellement dans la portée en commentaires de style Visual Basic.

## **WRAP\_COMMENT([commentaire], [wrap\_length], [indent], [start\_string])**

Encapsule le texte [commentaire] à une largeur [wrap\_length] en plaçant [indent] et [start\_string] au début de chaque ligne.

\$comportement = %WRAP\_COMMENT(opBehavior, "40", " ", " // " )%

<wrap\_length> doit toujours être passé sous forme de string , même si WRAP\_COMMENT traite ce paramètre comme un integer .

## **WRAP\_LINES([texte], [longueur\_enveloppe], [chaîne\_début] {, [chaîne\_fin] })**

Encapsule [texte] comme indiqué pour être [wrap\_length], en ajoutant [start\_string] au début de chaque ligne et [end\_string] à la fin de la ligne si cela est spécifié.

## **XML\_COMMENT([wrap\_length])**

Convertit les notes de l'élément actuellement concerné en commentaires de style XML.

## Contrôle des macros

Les macros de contrôle sont utilisées pour contrôler le traitement et le formatage des gabarits . Les types de macros de contrôle de base incluent :

- La macro de liste, pour générer plusieurs fonctionnalités d'éléments, telles que des attributs et des opérations
- Les macros de ramification, qui forment des constructions if-then-else pour exécuter conditionnellement des parties d'un gabarit
- La macro PI pour formater les nouvelles lignes dans la sortie, qui prend effet à partir de la prochaine ligne non vide
- Une macro de fonction PI qui permet de définir PI sur une variable et ajoute la possibilité de définir le PI qui est généré avant la ligne suivante
- Les macros de synchronisation

En général, les macros de contrôle sont nommées selon la casse Camel.

## Liste des macros

Si vous devez parcourir en boucle ou itérer sur un ensemble d'objets contenus dans ou sous l'objet actuel, vous pouvez le faire à l'aide de la macro `%list`. Cette macro effectue un passage itératif sur tous les objets dans la portée du gabarit actuel et appelle un autre gabarit pour traiter chacun d'eux.

La structure de base est la suivante :

```
%list=<Nom du modèle> @separator=<chaîne> @indent=<chaîne> (<conditions>) %
```

où `<string>` est une string littérale entre guillemets et `<TemplateName>` peut être l'un de ces noms gabarit :

- Attribut
- AttributImpl
- Classe
- Base de classe
- ClasseImpl
- Initialiseur de classe
- Interface de classe
- Contrainte
- Gabarit personnalisé ( gabarits personnalisés vous permettent de définir vos propres gabarits )
- Effort
- Classe intérieure
- Implémentation de la classe intérieure
- Fichier lié
- Métrique
- Namespace
- Opération
- OpérationImpl
- Paramètre
- Problème
- Exigence
- Ressource
- Risque
- Scénario
- Test

`<conditions>` est facultatif et ressemble aux conditions des instructions « if » et « elsif ».

### Exemple

Dans une transformation de classe, la classe peut contenir plusieurs attributs ; cet exemple appelle la transformation d'attribut et génère le résultat du traitement de la transformation pour chaque attribut de la classe concernée. La liste résultante sépare ses éléments par une seule nouvelle ligne et les indente respectivement de deux espaces. Si la classe concernée avait des attributs stéréotypés, ils seraient générés à l'aide du gabarit spécialisé approprié.

```
%list="Attribut" @separator="\n" @indent="  "%
```

L'attribut séparateur, désigné par `@separator`, spécifie l'espace qui doit être utilisé entre les éléments de la liste, à l'exclusion du dernier élément de la liste.

L'attribut indent, indiqué par @indent, spécifie l'espace par lequel chaque ligne de la sortie générée doit être indentée.

## Cas particuliers

Il y a quelques cas particuliers à prendre en compte lors de l'utilisation de la macro %list :

- Si le gabarit d'attribut est utilisé comme argument de la macro %list, cela génère également des attributs dérivés des associations en exécutant le gabarit LinkedAttribute approprié
- Si le gabarit ClassBase est utilisé comme argument de la macro %list, cela génère également des bases de classe dérivées des liens dans le modèle en exécutant le gabarit LinkedClassBase approprié
- Si le gabarit ClassInterface est utilisé comme argument de la macro %list, cela génère également des bases de classe dérivées des liens dans le modèle en exécutant le gabarit LinkedClassInterface approprié
- Si InnerClass ou InnerClassImpl est utilisé comme argument de la macro %list, ces classes sont générées à l'aide des gabarits Class et ClassImpl respectivement ; ces arguments indiquent que les gabarits doivent être traités en fonction des classes internes de la classe concernée

## Branchement des macros

Les macros de ramification fournissent des constructions if-then-else. Le CTF supporte une forme limitée de ramification via ces macros :

- si
- sinonSi
- autre
- finSi
- endTemplate (qui sort du gabarit actuel)

La structure de base des macros if et elseIf est :

```
%si <test> <opérateur> <test>%
```

où <opérateur> peut être l'un des éléments suivants :

- ==
- !=
- < (comparaison mathématique, inférieur à)
- > (comparaison mathématique, supérieur à)
- <= (comparaison mathématique, inférieur ou égal à)
- >= (comparaison mathématique, supérieur ou égal à)

et <test> peut être l'un des suivants :

- une string littérale, entourée de guillemets doubles
- une macro de substitution directe, sans les signes de pourcentage qui l'entourent
- une référence variable

Note que si vous utilisez l'un des opérateurs de comparaison mathématiques, <test> doit être un nombre décimal au format string .

Les branches peuvent être imbriquées et plusieurs conditions peuvent être spécifiées à l'aide de l'une des méthodes suivantes :

- et, ou
- ou

Lors de la spécification de plusieurs conditions, « et » et « ou » ont le même ordre de priorité et les conditions sont traitées de gauche à droite.

Si les instructions conditionnelles sur les chaînes sont sensibles à la casse, « une String » n'est pas égale à « une CHAÎNE ». Par conséquent, dans certaines situations, il est préférable de définir la variable \$str=TO\_LOWER(variable) ou TO\_UPPER(variable) puis de comparer à un cas spécifique.

Les macros ne sont pas prises en charge dans les instructions conditionnelles. Il est préférable d'affecter les résultats d'une macro ( string ) à une variable, puis d'utiliser la variable dans la comparaison.

```
$fldType = % TO_LOWER ($paramètre1)%
```

```
$COMMENT = "Utilisez les 4 premiers caractères pour les types de champs Date et Heure"
```

```
$fldType4 = % GAUCHE ($fldType, 4)%
```

```
%si $fldType4 == "date"%
```

```
Date et heure
```

```
%endif%
```

Cela prend un paramètre de valeur « Datetime », « DATETIME » ou « Date » et renvoie « Datetime ».

Les macros endif ou endTemplate doivent être utilisées pour signifier la fin d'une branche. De plus, la macro endTemplate provoque le retour immédiat du gabarit , si la branche correspondante est en cours d'exécution.

## Exemple 1

```
%si elemType == "Interface"%  
;  
%autre%  
%OpérationBody%  
%finSi%
```

Dans ce cas:

- Si l'élément Type est « Interface », un point-virgule est renvoyé
- Si l'elemType n'est pas "Interface", un gabarit appelé Operation Body est appelé

## Exemple 2

```
$bases="ClassBase"  
$interfaces=""%  
%si $bases != " " et $interfaces != ""%  
:$bases, $interfaces  
%elseif $bases != ""%  
:$bases  
%elseif $interfaces != ""%  
:$interfaces  
%finSi%
```

Dans ce cas, le texte renvoyé est ':ClassBase'.

## Conditions utilisant une valeur booléenne

Lors de la configuration d'une ramification à l'aide de conditions impliquant une case à cocher système (champs booléens), comme Attribute.Static (attStatic), l'instruction conditionnelle serait écrite comme suit :

```
%si attStatic == "T"%
```

Par exemple:

```
% si attCollection == "T" ou attOrderedMultiplicity == "T" %  
% fin du modèle %
```

## Macros de synchronisation

Les macros de synchronisation sont utilisées pour fournir des indications de formatage à Enterprise Architect lors de l'insertion de nouvelles sections dans le code source, lors de la synchronisation en aval. Les valeurs des macros de synchronisation doivent être définies dans les gabarits du fichier.

La structure de configuration des macros de synchronisation est :

%<nom>=<valeur>%

où <nom> peut être l'une des macros répertoriées ici et <valeur> est une string littérale entourée de guillemets doubles.

### Macros de synchronisation

Nom de la macro	Description
Espace de synchronisationNewClassNotes	Espace à ajouter à une nouvelle note de cours. valeur par défaut : \n.
synchNewAttributeNotesSpace	Espace à ajouter à une nouvelle note d'attribut. valeur par défaut : \n.
synchNewOperationNotesSpace	Espace à ajouter à une nouvelle note d'opération. valeur par défaut : \n.
synchNouvelleOpérationBodySpace	Espace à ajouter au corps d'une nouvelle opération. valeur par défaut : \n.
synchNamespaceBodyIndent	Indentation appliquée aux classes dans les espaces de noms non globaux. valeur par défaut : \t.



## La macro d'instructions de traitement (PI)

La macro PI (Processing Instruction) fournit un moyen de définir le texte séparateur à insérer entre les morceaux de code (qui représentent des entités) générés à l'aide d'un gabarit .

La structure de définition de l'instruction de traitement est :

```
%PI=<valeur>%
```

Dans cette structure, <value> est une string littérale entourée de guillemets doubles, avec ces options :

- "\n" - Nouvelle ligne (par défaut)
- " " - Espace
- "\" - Onglet
- "" - Nul

Par défaut, le PI est configuré pour générer une nouvelle ligne (\n) pour chaque substitution non vide, ce comportement peut être modifié en réinitialisant la macro PI. Par exemple, la déclaration d'attribut d'une classe dans un code VB simple serait générée en une seule ligne (sans nouvelle ligne). Ces propriétés sont dérivées des propriétés d'attribut de classe dans le modèle pour générer, par exemple :

Format d'impression privé constant comme String = "Portrait"

Le gabarit pour générer ceci commence par le PI défini sur un espace plutôt que sur une nouvelle ligne :

```
% PI = " " %
```

```
% CONVERT_SCOPE (attScope)%
```

```
%finSi %
```

```
% si attConst == "T" %
```

```
Const
```

```
%finSi %
```

Lors de la transformation, attscope renvoie le mot-clé VB « Private » et attConst renvoie « Const » sur la même ligne espacée d'un seul espace (conformément à l'exemple de définition VB Class.Attribute précédent).

Alternativement, lors de la génération d'une classe, vous pouvez souhaiter que la déclaration de classe, les notes et le corps de la classe soient tous séparés par des lignes doubles. Dans ce cas, le %PI est défini sur /n/n' pour renvoyer un espacement de ligne double :

```
% PI = "\n\n" %
```

```
% Déclaration de classe %
```

```
% Notes de cours %
```

```
% Corps de classe %
```

### Caractéristiques de l'IP

- Les lignes vides n'ont aucun effet sur la sortie
- Toute ligne contenant une macro qui produit un résultat vide n'entraîne pas de séparateur PI (espace/nouvelle ligne)
- La dernière entrée ne renvoie pas de PI ; par exemple, %Classbody% n'a pas de double ligne ajoutée après le corps

# Macros de génération de code pour Statemachines Exécutables

Les gabarits répertoriés ici sont disponibles via l'éditeur de code Gabarit (l'option de ruban « Développer > Code source > Options > Modifier le code Gabarits »); sélectionnez « STM\_C++\_Structured » dans le champ « Langue ».

Les gabarits sont structurés comme indiqué :

StmContextStateMachineEnum

StmStateMachineEnum

StmContextStateEnum

StmAllStateEnum

StmContextTransitionEnum

StmTransitionEnum

StmContextEntryEnum

StmAllEntryEnum

Chaîne de machine StmContextStateToEnum

Chaîne de machine StmStateToEnum

StmContextStateEnumToString

StmStateEnumToString

StmContextTransitionEnumToString

StmTransitionEnumToString

Nom de l'état du contexte Stm vers Guid

Nom de l'état Stm vers Guid

Nom de la transition StmContextVersGuid

Nom de transition Stm vers Guid

Définition du contexte Stm

StmStateMachineEnum

StmAllStateEnum

StmTransitionEnum

StmAllEntryEnum

Initialisation de la variable StmAllRegion

État de Stm avec événement différé

Événement différé Stm  
Mappage de processus de transition Stm  
Procédure de transition Stm  
Sortie de transition Stm  
Entrée de transition Stm  
Transition sortante de la cible Stm  
État de la sous-machine parent de la cible Stm  
Mappage de StmStateProc  
Procédure d'état de Stm  
Entrée d'état Stm  
Transition sortante Stm  
Entrée de référence StmConnectionPoint  
StmParameterizedInitial  
StmSubMachineInitial  
StmRegionInitial  
StmRegionDésactiver  
Procédure de sortie de l'état de Stm  
Transition d'état Stm  
Événement StmState  
Transition déclenchée par StmState  
StmStateCompletionTransition  
État de la transition entrante de StmState  
État de transition sortante de StmState  
Événement StmSubmachineStateExit  
Transition sortante StmVertex  
Événement de sortie de référence de point de connexion Stm  
Événement StmStateExit  
Transition sortante StmVertex  
Variable StmAllRegion  
Chaîne de machine StmStateToEnum  
Exécution de StmStateMachine  
Données initiales de l'état Stm  
Entrée de machine à état stm  
Transition sortante Stm  
StmStateMachineRunInitial  
StmStateMachineInitial  
Exécution de StmStateMachine

Gestionnaire de contexte Stm

Gestionnaire de simulation Stm  
Déclaration d'instance de contexte Stm

Instance de contexte Stm

État d'exécution de la variable de contexte Stm

Association d'instances de contexte Stm

StmContextInstanceClear

Proxy d'événement Stm

Énumération StmSignal

StmContextJoinEventEnum

Énumération d'événements StmJoin

Énumération d'événements Stm

Définition de StmSignal

Attribution d'attributs de signal Stm

Attribut StmSignal

Initialisation du signal Stm

Chaîne d'événements StmEventStringToEnum

StmEventEnumToString

Nom de l'événement Stm à Guid

Gestionnaire de console Stm

Déclaration d'instance de contexte Stm

Instance de contexte Stm

État d'exécution de la variable de contexte Stm

Association d'instances de contexte Stm

StmContextInstanceClear

StmStateMachineStrongToEnum

StmInitialForTransition

Transition sortante StmVertex

Événement StmSend

StmBroadcastEvent

Référence de contexte Stm

## Signal et événement

Nom de la macro	Description
stmEventEnum	Le nom de l'événement avec le préfixe « ENUM_ », tout en majuscules.

Guide des événements Stm	Le GUID de l'événement.
Nom de l'événement stm	Le nom de l'événement avec les espaces et les astérisques supprimés.
Variable d'événement stm	Le nom de l'événement avec le préfixe « m_ » en minuscule.
Événement stmIsSignal	Est « T » si l'élément est un SignalEvent.
stmSignalEnum	Le nom du signal avec le préfixe « ENUM_ », tout en majuscules.
stmSignalFirstEvent	Le nom de l'événement avec le préfixe « ENUM_ », tout en majuscules.
stmSignalGuid	Le GUID du signal.
stmSignalNom	Le nom du signal avec les espaces et les astérisques supprimés.
Variable de signal stm	Le nom du signal avec le préfixe « m_ » en minuscule.
stmTriggerName	Transition Propriétés : Le nom du Déclencheur .
Spécifications de stmTrigger	Propriétés de Transition : La spécification du Déclencheur .
Type de déclencheur stm	Propriétés de Transition : Le type du Déclencheur .

## Contexte

Nom de la macro	Description
stmContextName	Le nom de la classe avec les espaces et les astérisques supprimés.
stmContextQualName	Le nom qualifié de la classe pour laquelle le code est généré.
stmContextVariableName	
stmContextFileName	Le nom du fichier de sortie pour la classe pour laquelle le code est généré.

## Écriture de l'état d'exécution Object dans l'initialisation Statemachine

Nom de la macro	Description
stmContextVariableRunstateName	
stmContextVariableRunstat	

eValue	
stmContextHasState machine	Est « T » si le contexte actuel possède une ou plusieurs State machines .
stmHasHistoryPattern	Est « T » si la State machine a un Motif d'historique.
stmHasTerminatePattern	Est « T » si la State machine a un Motif de terminaison.
stmHasDeferredEventPatte rn	Est « T » si la State machine a un Motif d'événement différé.
stmHasSubmachinePattern	Est « T » si la State machine a un Motif de sous-machine.
stmHasOrthogonalPattern	Est « T » si la State machine a un Motif orthogonal.

## State machine

Nom de la macro	Description
stmState machineName	Le nom de la State machine avec les astérisques et les espaces supprimés.
stmState machineEnum	Le nom de la State machine plus « ENUM_ » plus le nom de la State machine en majuscules.
stmState machineGuid	Le GUID de l'élément State machine .
stmState Count	Le nombre d'éléments State dans la State machine .
stmSub machineInitial Count	Le nombre d'éléments initiaux dans l'élément State de la sous-machine.
stmState machineHasSub machineState	Est « T » si la State machine possède au moins un State de sous-machine.
stmState machineInitial Count	Le nombre d'éléments initiaux dans la State machine .

## Région

Nom de la macro	Description
stmRegionEnum	Le nom de la région State plus « ENUM_ » plus le nom de la région State en majuscules.
stmRegionFQName	Le nom complet de la région State .

stmRegionName	Le nom de la région State avec les espaces et les astérisques supprimés.
Variable de région stm	Le nom de la région State avec le préfixe « m_ » en minuscule.
Variable stmRegionFQ	Le nom complet de la région State avec le préfixe « m_ » en minuscule.
stmRegionGuid	Le GUID de la région.
stmRegionInitial	
stmRegionIsOwnedByState Machine	Est « T » si la région appartient à une Statemachine .

## Transition

Nom de la macro	Description
stmTransitionEnum	Le nom de la Transition avec le préfixe « ENUM_ », plus le nom de la Transition en majuscules.
Guide de transition stm	Le GUID de la Transition.
stmTransitionName	Le nom de la Transition avec les espaces et les astérisques supprimés.
stmTransitionSourceGuid	Le GUID de l'élément source dans la transition.
stmTransitionTargetGuid	Le GUID de l'élément cible dans la transition.
Variable de transition stm	Le nom de la Transition avec le préfixe 'm_' en minuscule.
Variable source de transition stm	
Variable cible de transition stm	
Variable stmTransitionFQ	
stmSourceVertexEnum	Le nom du sommet source de la transition plus « _ENUM » plus le nom du sommet source de la transition en majuscules.
stmTargetVertexEnum	Le nom du sommet cible de la transition plus « _ENUM » plus le nom du sommet cible de la transition en majuscules.
stmSourceIsInitial	Est « T » si la source de la transition est une initiale.
stmSourceIsState	Est « T » si la source de la transition est un State .

stmSourceIsEntryPoint	La valeur est « T » si la source de la transition est un point d'entrée.
stmSourceIsExitPoint	La valeur est « T » si la source de la transition est un point de sortie.
stmSourceIsFork	C'est « T » si la source de la transition est un Fork.
stmSourceIsJoin	Est « T » si la source de la transition est un élément Join.
stmTargetIsFinalState	Est « T » si la cible de la transition est un élément State final.
stmTargetIsExitPoint	Est « T » si la cible de la transition est un élément de point de sortie.
stmTargetIsState	Est « T » si la cible de la transition est un élément State .
stmTargetIsChoice	La valeur est « T » si la cible de la transition est un élément de choix.
stmTargetIsJunction	Est « T » si la cible de la transition est un élément de jonction.
stmTargetIsEntryPoint	Est « T » si la cible de la transition est un élément de point d'entrée.
stmTargetIsConnectionPointReference	Est « T » si la cible de la transition est un élément de référence de point de connexion.
stmTargetIsFork	Est « T » si la cible de la transition est un élément Fork.
stmTargetIsJoin	Est « T » si la cible de la transition est un élément de jointure.
Effet de transition stm	L'effet de la transition.
stmTransitionGuard	La Garde de la Transition.
Type de transition stm	Le type ou la sorte de transition.
stmTargetInitialTransition	
stmTargetIsSubmachineState	Est « T » si la cible de la transition est un State de sous-machine.
stmSourceStateEnum	Le nom de l'état source de la transition avec le préfixe « _ENUM » en majuscule.
stmTargetStateEnum	Le nom de l'état cible de la transition, avec le préfixe « _ENUM » en majuscule.
stmTargetVertexFQName	Le nom entièrement qualifié du sommet cible de la transition.
stmTargetIsDeepHistory	Est « T » si la cible de la transition est un State d'histoire profonde.
stmTargetIsShallowHistory	Est « T » si la cible de la transition est un State d'histoire superficielle.
stmTargetIsTerminate	Est « T » si la cible de la transition est un élément Terminate.
stmParentIsStateMachine	Est « T » si le sommet est un point d'entrée ou un point de sortie, ou si le conteneur est une StateMachine .



stmSourceParentStateEnum	
stmTargetParentStateEnum	
stmTargetSubmachineEnum	
Index de la région cible stm	
stmIsSelfTransition	Est « T » si la source de la transition est la même que sa cible.
stmHistoriquePropriétaireRégionInitialTransition	
stmDefaultHistoryTransition	

## Sommet et State

Nom de la macro	Description
stmVertexName	Le nom du Vertex.
stmStateName	Le nom de l' State .
stmVertexGuid	Le GUID du sommet.
stmVertexFQName	Le nom complet du Vertex.
stmStateFQName	Le nom complet de l' State .
stmVertexType	Le type du sommet ; l'un des éléments suivants : « State », « FinalState », « Pseudostate », « ConnectionPointReference » ou « » (vide).
stmPseudo-étatKind	Le type de pseudo-état ; l'un des éléments suivants : « initial », « deepHistory », « shallowHistory », « join », « fork », « junction », « choice », « entryPoint », « exitPoint » ou « terminate ».
stmPseudo-étatNom	Le nom du pseudo-État.
Variable stmPseudostate	Le nom du pseudo-État avec le préfixe « m_ » en minuscule.
stmPseudostateÉtatNom de la machine	Le nom de la Statemachine pseudo-état.
stmPseudostateStateMachineVariable	Le nom de la Statemachine pseudo-état avec le préfixe « m_ » en minuscule.

Variable stmVertex	Le nom du Vertex avec le préfixe 'm_' en minuscule.
stmVertexEnum	Le nom du sommet plus '_ENUM' plus le nom du sommet en majuscules.
stmStateEnum	Le nom de l' State plus « _ENUM » plus le nom de l' State en majuscules.
stmConnectionPointReferenceStateName	Le nom de la référence du point de connexion.
Variable d'état de référence du point de connexion stm	Le nom de la référence du point de connexion avec le préfixe « m_ » en minuscules.
stmConnectionPointReferenceEntryCount	
stmParameterizedInitialCount	
stmInitialCountForTransition	
Variable d'état stm	Le nom de l' State avec le préfixe « m_ » en minuscule.
comportement de stmStateEntry	Le comportement défini pour une opération Action « entrée » pour un State (le texte de l'onglet « Comportement » pour l'opération Action « entrée » dans la fenêtre Fonctionnalités de l'élément).
Code d'entrée stmState	Le code initial défini pour une opération Action « entrée » pour un State (le texte de l'opération Action « entrée » dans l'onglet « Code » du comportement).
stmStateDoBehavior	Le comportement défini pour une opération Action « do » pour un State (le texte de l'onglet « Comportement » pour l'opération Action « do » dans la fenêtre Fonctionnalités de l'élément).
stmStateDoCode	Le code initial défini pour une opération Action « do » pour un State (le texte de l'opération Action « do » dans l'onglet « Code » du comportement).
comportement de sortie stmState	Le comportement défini pour une opération Action « quitter » pour un State (le texte de l'onglet « Comportement » pour l'opération Action « quitter » dans la fenêtre Fonctionnalités de l'élément).
stmStateExitCode	Le code initial défini pour une opération Action « sortie » pour un State (le texte de l'opération Action « sortie » dans l'onglet « Code » du comportement).
stmStateSubmachineName	Le nom de la sous-machine.
Variable de sous-machine stmState	Le nom de la sous-machine avec le préfixe « m_ » en minuscule.
stmStateIsFinal	Est « T » si l' State est un FinalState.
stmStateIsSubmachineState	Est 'T' si l' State est un State de sous-machine (page ' Propriétés '   Avancé   propriété 'isSubmachineState').

stmSubMachineEnum	Le nom de la sous-machine suivi de « _ENUM » plus le nom de la sous-machine en majuscules.
stmStateHasChildrenToJoin	
stmStateIsTransitionTarget	
stmCeciEstLaSource	
stmThisIsSourceState	
stmStateParentIsSubmachine	Est « T » si le conteneur de State est une Statemachine .
stmStateContainerMatchTransitionContainer	
stmVertexRegionIndex	
stmÉtatRégionNombre	Le nombre de régions dans l' State .
stmStateInitialCount	Le nombre d'éléments initiaux dans la Statemachine .
Variable stmVertexContainer	
stmVertexParentEnum	
stmStateHasUnGuardedCompletionTransition	
stmStateEventHasUnGuardedTransition	
stmInitialTransition	

## Association d'instances

Nom de la macro	Description
Nom de l'instance source stm	
stmTargetInstanceName	
stmSourceRoleName	
stmTargetRoleName	



## Macros de génération de code EASL

Enterprise Architect fournit un certain nombre de macros de génération de code Enterprise Architect Simulation Bibliothèque (EASL) pour générer du code à partir de modèles comportementaux. Il s'agit de :

- EASL\_INIT
- EASL\_OBTENIR
- Liste EASLListe et
- EASL\_FIN

### EASL\_INIT

La macro EASL\_INIT permet d'initialiser un modèle de comportement EASL. La génération du code du modèle de comportement dépend de ce modèle.

Aspect	Description
Syntaxe	<pre>%EASL_INIT(&lt;&lt;GUID&gt;&gt;)%</pre> <p>où:</p> <ul style="list-style-type: none"> <li>• &lt;&lt;GUID&gt;&gt; est le GUID de l' Object (généralement un élément de classe) qui est le propriétaire du modèle de comportement</li> </ul>

### EASL\_OBTENIR

La macro EASL\_GET permet de récupérer une propriété ou une collection d'un objet EASL. Les objets EASL et les propriétés et collections de chaque objet sont identifiés dans les rubriques *Collections EASL* et *Propriétés EASL* .

Aspect	Description
Syntaxe	<pre>\$result = %EASL_GET(&lt;&lt;Propriété&gt;&gt;, &lt;&lt;ID du propriétaire&gt;&gt;, &lt;&lt;Nom&gt;&gt;)%</pre> <p>où:</p> <ul style="list-style-type: none"> <li>• &lt;&lt;Property&gt;&gt; est l'un des éléments suivants : « Property », « Collection », « At », « Count » ou « IndexOf »</li> <li>• &lt;&lt;OwnerID&gt;&gt; est l' ID de l' objet propriétaire pour lequel la propriété/collection doit être récupérée</li> <li>• &lt;&lt;Nom&gt;&gt; est le nom de la propriété ou de la collection à laquelle on accède</li> <li>• \$result est la valeur renvoyée ; c'est « » si ce n'est pas une propriété valide</li> </ul> <p>Si &lt;&lt;Propriété&gt;&gt; est :</p> <ul style="list-style-type: none"> <li>• « At », alors &lt;&lt;OwnerID&gt;&gt; est l' ID d'une collection et &lt;&lt;Name&gt;&gt; est l'index dans la collection pour laquelle l'élément doit être récupéré</li> <li>• « Count », puis &lt;&lt; Owner ID &gt;&gt; est l' ID d'une collection et &lt;&lt; Name &gt;&gt; n'est pas utilisé ; il récupérera le numéro d'élément dans la collection</li> <li>• "IndexOf", puis &lt;&lt;Owner ID&gt;&gt; est l' ID d'une collection et &lt;&lt;Name&gt;&gt; est l' ID de l'élément dans la collection ; il récupérera l'index (format string ) de l'élément dans la collection</li> </ul>
Exemple	<pre>\$sPropName = %EASL_GET("Propriété", \$context, "Nom")%</pre>

## Liste EASL

La macro EASLList est utilisée pour restituer chaque objet d'une collection EASL en utilisant le gabarit approprié.

Aspect	Description
Syntaxe	<pre>\$result = %EASLList=&lt;&lt;Nom du modèle&gt;&gt; @separator=&lt;&lt;Séparateur&gt;&gt; @indent=&lt;&lt;indent&gt;&gt; @owner=&lt;&lt;Identifiant du propriétaire&gt;&gt; @collection=&lt;&lt;NomDeLaCollection&gt;&gt; @option1=&lt;&lt;OPTION1&gt;&gt; @option2=&lt;&lt;OPTION2&gt;&gt;..... @optionN=&lt;&lt;OPTIONN&gt;&gt;%</pre> <p>où:</p> <ul style="list-style-type: none"> <li>• &lt;&lt;TemplateName&gt;&gt; est le nom de tout gabarit de modèle comportemental ou gabarit personnalisé</li> <li>• &lt;&lt;Separator&gt;&gt; est un séparateur de liste (tel que « \n »)</li> <li>• &lt;&lt;indent&gt;&gt; est une indentation à appliquer au résultat</li> <li>• &lt;&lt;OwnedID&gt;&gt; est l' ID de l' objet qui contient la collection requise</li> <li>• &lt;&lt;CollectionName&gt;&gt; est le nom de la collection requise</li> <li>• &lt;&lt;OPTION1&gt;...&lt;&lt;OPTION99&gt;&gt; sont des options diverses qui peuvent être passées sur le gabarit ; chaque option est donnée comme paramètre d'entrée supplémentaire au gabarit</li> <li>• \$result est la valeur résultante ; c'est « » s'il ne s'agit pas d'une collection valide</li> </ul>
Exemple	<pre>\$sStates = %EASLList=" State " @separator="\n" @indent="\t" @owner=\$StateMachineGUID @collection=" States " @option=\$sOption%</pre>

## EASL\_FIN

La macro EASL\_END est utilisée pour libérer le modèle de comportement EASL.

Aspect	Description
Syntaxe	%EASL_END%

## Comportementale Modèle Gabarits

- Action
- Affectation Action
- Pause Action
- Appel Action
- Action Créer
- Action Détruire
- Action si

- Boucle Action
- Action Opaque
- Action parallèle
- Action RaiseEvent
- Action RaiseException
- Interrupteur Action
- Comportement
- Comportement corporel
- Déclaration de comportement
- Paramètre de comportement
- Appeler l'argument
- Décision Action
- Condition Décision
- Logique Décision
- Tableau de Décision
- Garde
- Déclaration de propriété
- Notes sur la propriété
- Object de propriété
- State
- Rappel State
- State Dénombrer
- State Nom énuméré
- Statemachine
- Historique Statemachine
- Transition
- Effet de transition
- Déclencheur

## Collections EASL

Cette rubrique répertorie les collections EASL pour chacun des objets EASL, tels que récupérés par la macro de génération de code [EASL Code Generation Macros](#).

### Action

Nom de la collection	Description
Arguments	Les arguments de Action .
Sous-actions	Les sous-actions de l' Action .

### Comportement

Nom de la collection	Description
Actes	Les actions du comportement.
Nœuds	Les nœuds du comportement.
Paramètres	Les paramètres du comportement.
Variables	Les variables du comportement.

### Classificateur

Nom de la collection	Description
Toutes les machines d'état	Toutes Statemachines pour le classificateur.
Propriétés asynchrones	Les propriétés asynchrones du classificateur.
Déclencheurs asynchrones	Les déclencheurs asynchrones du Classificateur.
Comportements	Les comportements du Classificateur.
Propriétés	Les propriétés du classificateur.
Propriétés chronométrées	Les propriétés chronométrées du classificateur.
Déclencheurs temporisés	Les déclencheurs temporisés du Classifier.



Déclencheurs	Tous déclencheurs du Classificateur.
--------------	--------------------------------------

## Construction

Nom de la collection	Description
Tous les enfants	Les enfants du Construct.
Dépendances du client	Les dépendances du client sur le Construct.
Séréotypes	Les stéréotypes du Construct.
Dépendances des fournisseurs	Les dépendances du fournisseur sur le Construct.

## Nœud

Nom de la collection	Description
Bords entrants	Les bords entrants du nœud.
Bords sortants	Les bords sortants du nœud.
Sous-nœuds	Les sous-nœuds du nœud.

## State

Nom de la collection	Description
Comportements à faire	Les comportements de State .
Comportements d'entrée	Les comportements d'entrée de State .
Comportements de sortie	Les comportements de sortie de State .

## Statemachine

Nom de la collection	Description
----------------------	-------------

Tous les États finaux	Les States finaux de Statemachine .
Tous les États	Tous States au sein de la Statemachine , y compris ceux au sein States de sous-machine.
Transitions dérivées	Les transitions dérivées de la Statemachine avec l'effet valide associé.
States	Les States dans la Statemachine .
Transitions	Les transitions au sein de la Statemachine .
Sommets	Les sommets de la Statemachine .

## Transition

Nom de la collection	Description
Effets	Les effets de la transition.
Gardes	Les gardes de la Transition.
Déclencheurs	Les déclencheurs de la Transition .

## Déclencheur

Nom de la collection	Description
Transitions déclenchées	Les transitions déclenchées associées au Déclencheur .

## Sommet

Nom de la collection	Description
Transitions sortantes dérivées	Les transitions sortantes dérivées du sommet après avoir traversé les pseudo-nœuds.
Transitions entrantes	Les transitions entrantes du Vertex.
Transitions sortantes	Les transitions sortantes du Vertex.



## Propriétés EASL

Cette rubrique répertorie les propriétés EASL pour chacun des objets EASL, telles que récupérées par la macro de génération de code [EASL Code Generation Macros](#).

### Action

Nom de la propriété	Description
Comportement	Le comportement associé à Action ( Action de comportement d'appel ou Action d'opération d'appel).
Corps	Le corps de Action .
Contexte	Le contexte de l' Action .
Garde	La garde de Action .
EstFinal	Une vérification pour savoir si l'action est une Action finale.
Est gardé	Une vérification pour savoir si l'action est une Action gardée.
EstInitial	Une vérification pour savoir si l'action est une Action initiale.
Gentil	Le genre d' Action .
Suivant	L'action suivante de Action .
Nœud	Le nœud associé à Action dans le graphique.

### Argument

Nom de la propriété	Description
Paramètre	L' ID du paramètre associé à l'argument.
Valeur	La valeur par défaut de l'argument.

### Comportement

Nom de la propriété	Description
Action initiale	L'action initiale du comportement.

est en lecture seule	Le isReadOnly du comportement.
estSingleExecution	L'exécution est unique du comportement.
Gentil	Le type de comportement.
Type de retour	Le type de retour du comportement.
Spécification	La spécification du comportement.

## Appeler l'événement

Nom de la propriété	Description
Opération	Le fonctionnement du CallEvent.

## ChangeEvent

Nom de la propriété	Description
Changer d'expression	L'expression de changement du ChangeEvent.

## Classificateur

Nom de la propriété	Description
A des comportements	Une vérification pour savoir si le classificateur possède des modèles comportementaux (activité et interaction).
Langue	Le langage du classificateur.
Statemachine	La Statemachine du classificateur.

## Condition

Nom de la propriété	Description

Expression	L'expression de la condition.
Inférieur	La valeur inférieure de la Condition.
Supérieur	La valeur supérieure de la Condition.

## Construction

Nom de la propriété	Description
Obtenir une valeur marquée	La Valeur Étiquetée de la Propriété.
Le stéréotype est-il appliqué ?	Une vérification pour savoir si un stéréotype particulier est appliqué à la propriété.
Notes	Notes sur la propriété.
Type UML	Le type UML de la propriété.
Visibilité	La visibilité de la Propriété.

## Bord

Nom de la propriété	Description
Depuis	L' ID du nœud à partir duquel l'Edge provient.
À	L' ID du nœud sur lequel l'Edge est ciblé.

## Objet événement

Nom de la propriété	Description
Type d'événement	Le type d'événement de l' Object événement.

## Exemple

Nom de la propriété	Description
Classificateur	Le classificateur de l'instance.
Valeur	La valeur de l'instance.

## Paramètre

Nom de la propriété	Description
Direction	La direction du paramètre.
Type	Le type du paramètre.
Valeur	La valeur du paramètre.

## Primitif

Nom de la propriété	Description
Nom FQ	Le nom FQ du Primitif.
ID	L' ID du Primitif.
Nom	Le nom du Primitif.
ObjectType	Le type object du Primitive.
Mère	L'IDParent du Primitive.

## Objet de propriété

Nom de la propriété	Description
Taille de la limite	La taille limite du PropertyObject (s'il s'agit d'une collection).
ClassificateurStereoType	Le stéréotype du classificateur du PropertyObject.
Est-ce queAsynchProp	Une vérification pour savoir si le PropertyObject est une propriété asynchrone.
EstCollection	Une vérification pour savoir si le PropertyObject est une collection.

IsOrdered	Une vérification pour savoir si le PropertyObject est ordonné (s'il s'agit d'une collection).
EstTimedProp	Une vérification pour savoir si le PropertyObject est une propriété temporisée.
Gentil	Le type de PropertyObject.
Valeur inférieure	La valeur inférieure du PropertyObject (s'il s'agit d'une collection).
Type	Le type de PropertyObject.
Valeur supérieure	La valeur supérieure du PropertyObject (s'il s'agit d'une collection).
Valeur	La valeur de PropertyObject.

## SignalEvent

Nom de la propriété	Description
Signal	Le signal du SignalEvent.

## State

Nom de la propriété	Description
A une sous-machine	Une vérification pour savoir si l' State est un état de sous-machine.
Est-ce que l'état final	Un contrôle pour savoir si l' State est un état final.
Sous-machine	Obtenir l' ID de la sous-machine contenue dans l' State (le cas échéant).

## Statemachine

Nom de la propriété	Description
A un état de sous-machine	Une vérification pour savoir si la Statemachine possède un état de sous-machine.
État initial	L'état initial de la Statemachine .
État de la sous-machine	L' State de la sous-machine de Statemachine .



## Événement temporel

Nom de la propriété	Description
Quand	La propriété « quand » de TimeEvent.

## Transition

Nom de la propriété	Description
A un effet	Une vérification pour savoir si la transition a un effet valide.
IsDerived	Une vérification pour savoir si la transition est une transition dérivée.
Est-ce que Transcend	Vérification permettant de savoir si la transition transcende d'une Statemachine ( State de sous-machine) à une autre.
Est déclenché	Une vérification pour savoir si la transition est déclenchée.
Source	La source de la Transition.
Cible	L'objectif de la Transition.

## Déclencheur

Nom de la propriété	Description
État de destination asynchrone	L'état de destination asynchrone du Déclencheur (s'il s'agit d'un déclencheur asynchrone).
Propriété dépendante	L' ID de la propriété associée au Déclencheur .
Événement	L'événement du Déclencheur .
Nom	Le nom du Déclencheur .
Type	Le type du Déclencheur .

## Sommet

Nom de la propriété	Description
EstHistoire	Une vérification pour savoir si le sommet est un état d'historique.
Est-ce un pseudo-état ?	Une vérification pour savoir si le sommet est un pseudo-état.
Type d'état pseudo	Le type de pseudo-état du Vertex.

## Appelez Gabarits depuis Gabarits

En utilisant des appels de fonction avec des paramètres, vous pouvez appeler gabarits à partir d'autres gabarits, qu'il s'agisse de gabarits standards ou de gabarits définis par l'utilisateur créés dans votre projet. De plus, des gabarits appelés peuvent renvoyer une valeur et peuvent être appelés de manière récursive.

### Exemples

Une instruction d'appel renvoyant un paramètre à une variable :

```
$Source = %StateEnumeratedName($Source)%
```

Une instruction d'appel à un gabarit qui a des paramètres :

```
%RuleTask($GUID, $index)%
```

En utilisant l'instruction \$parameter dans le gabarit appelé :

```
$GUID = $paramètre1
```

```
$index = $paramètre2
```

Gabarits supportent les appels récursifs, tels que cet appel récursif sur le gabarit RuleTask :

```
$GUID = $paramètre1
```

```
$index = $paramètre2
```

```
% PI = " " %
```

```
$nul = "Initialiser object condition et action"
```

```
$count = %BR_GET("Nombre de règles")%
```

```
% si $count == " " ou $count == $index %
```

```
%ComputeRulet($GUID)%
```

```
\n
```

```
% fin du modèle %
```

```
%Règle($index)%
```

```
\n
```

```
$index = %MATH_ADD($index, "1")%
```

```
%RuleTask($GUID, $index)%
```

# L'éditeur Code Gabarit dans le développement des OMD

Ces rubriques décrivent comment utiliser la fenêtre de l'éditeur Code Gabarit pour créer gabarits personnalisés :

- [Create Custom Templates](#)
- [Customize Base Templates](#)
- [Add New Stereotyped Templates](#)

L'éditeur de code Gabarit fournit les facilités de l'Éditeur de Code Common, notamment Intelli-sense pour les macros gabarit de génération de code. Pour plus d'informations sur Intelli-sense et l'Éditeur de Code Common, consultez la rubrique *Édition du code source* .

## Créer Gabarits personnalisés

Enterprise Architect propose une large gamme de gabarits qui définissent la manière dont les éléments de code sont générés. Si ceux-ci ne sont pas suffisants pour vos besoins (par exemple, si vous souhaitez générer du code dans un langage non pris en charge par Enterprise Architect), vous pouvez créer gabarits personnalisés entièrement nouveaux. Vous pouvez également ajouter des remplacements de stéréotypes à vos gabarits personnalisés ; par exemple, vous pouvez répertorier tous vos paramètres et leurs notes dans vos notes de méthode.

### Accéder

Ruban	Développer > Code source > Options > Modifier le code Gabarits Conception > Paquetage > Transformation > Transformation Gabarits
Raccourcis Clavier	Ctrl+Maj+P ( gabarits de génération de code) Ctrl+Alt+H ( gabarits de transformation MDA )

### Créer gabarits personnalisés à l'aide de l'éditeur Code Gabarits

Étape	Description
1	Dans le champ « Langue », cliquez sur la flèche déroulante et sélectionnez le langage de programmation approprié.
2	Cliquez sur le bouton Ajouter un nouveau Gabarit personnalisé. La dialogue « Créer un nouveau Gabarit personnalisé » s'affiche.
3	Dans le champ « Type Gabarit », cliquez sur la flèche déroulante et sélectionnez l' object modélisation approprié. L'option '<None>' nécessite un traitement spécial ; elle permet la définition d'une macro de fonction qui ne s'applique réellement à aucun des types, mais doit être appelée en tant que fonction pour définir les variables \$parameter1, \$parameter2 et ainsi de suite pour chaque valeur transmise.
4	Dans le champ « Nom Gabarit », saisissez un nom approprié. Cliquez sur le bouton OK .
5	Dans l'onglet 'Code Gabarits Editor', le nouveau gabarit est inclus dans la liste ' Gabarits ', avec la valeur 'Oui' dans le champ 'Modifié'. Le gabarit est appelé < Gabarit Type>__< Gabarit Name>. Note le double caractère de soulignement entre le type gabarit et le nom gabarit .
6	Sélectionnez le gabarit dans la liste Gabarits et modifiez le contenu du champ Gabarit pour répondre à vos besoins.
7	Cliquez sur le bouton Enregistrer. Ceci stocke le nouveau gabarit , qui est maintenant disponible dans la liste des gabarits à utiliser. Vous

	pouvez également ajouter une substitution de stéréotype au gabarit , si nécessaire.
--	---

## Notes

- Pour une langue personnalisée, vous devez définir le gabarit du fichier afin qu'il puisse appeler les gabarits de la section d'importation, Namespace et de la classe, ainsi que tout autre gabarits que vous jugez applicable.

## Personnaliser Gabarits de base

Enterprise Architect propose une large gamme de gabarits qui définissent la manière dont les éléments de code sont générés. Si vous souhaitez modifier la manière dont un élément de code est généré, vous pouvez personnaliser les gabarits fournis par le système. Vos modifications peuvent concerner l'effet du gabarit lui-même ou ses appels à d'autres gabarits. Vous pouvez également ajouter des remplacements de stéréotypes à vos gabarits personnalisés ; par exemple, vous pouvez répertorier tous vos paramètres et leurs notes dans vos notes de méthode.

Lorsque vous personnalisez un gabarit (de base) fourni par le système, vous créez en réalité une copie du gabarit qui est utilisée de préférence à l'original. Toutes les modifications ultérieures sont apportées à cette copie et le gabarit de base d'origine est masqué. Si vous supprimez ensuite la copie, elle ne peut plus remplacer l'original, qui est alors réutilisé.

### Accéder

Ruban	Développer > Code source > Options > Modifier le code Gabarits
Raccourcis Clavier	Ctrl+Maj+P

### Personnaliser un gabarit de base

Étape	Description
1	Dans l'éditeur de code Gabarit, dans le champ « Langue », cliquez sur la flèche déroulante et sélectionnez le langage de programmation pour lequel vous souhaitez personnaliser les gabarits de base.
2	Dans la liste Gabarits, cliquez sur le gabarit de base à éditer.
3	Mettre à jour le gabarit.
4	Cliquez sur le bouton Enregistrer pour enregistrer vos modifications.
5	Répétez les étapes 2 à 4 pour chacun des gabarits de base pertinents que vous souhaitez personnaliser.
6	Si vous préférez, ajoutez une ou plusieurs substitutions de stéréotypes à l'un des gabarits.

## Ajouter de nouveaux Gabarits stéréotypés

Il est parfois utile de définir un gabarit de génération de code spécifique à utiliser avec des éléments d'un stéréotype donné. Cela permet de générer un code différent pour les éléments, en fonction de leur stéréotype. Enterprise Architect fournit des gabarits par défaut, qui ont été spécialisés pour les stéréotypes couramment utilisés dans les langages pris en charge. Par exemple, le gabarit « Operation Body » pour C# a été spécialisé pour le stéréotype de propriété, de sorte qu'il génère automatiquement ses méthodes constitutives « get » et « set ». Vous pouvez remplacer les gabarits stéréotypés par défaut comme décrit dans la rubrique *Remplacer Gabarits par défaut* . De plus, vous pouvez définir gabarits pour vos propres stéréotypes, comme décrit ici.

### Accéder

Ruban	Développer > Code source > Options > Modifier le code Gabarits
Raccourcis Clavier	Ctrl+Maj+P

### Ajouter un nouveau gabarit stéréotypé à l'aide de l'éditeur Code Gabarit

Étape	Description
1	Sélectionnez la langue appropriée dans la liste Langue.
2	Sélectionnez l'un des gabarits de base, dans la liste Gabarits .
3	Cliquez sur le bouton « Ajouter un nouveau remplacement stéréotypé ». La dialogue « Nouveau remplacement Gabarit » s'affiche.
4	Sélectionnez la Fonctionnalité et/ou le stéréotype de Classe requis. Cliquez sur le bouton OK .
5	Le nouveau gabarit stéréotypé s'affiche dans la liste des remplacements de stéréotypes, marqué comme modifié.
6	Effectuez les modifications requises dans l'éditeur de code Gabarits .
7	Cliquez sur le bouton Enregistrer pour stocker le nouveau gabarit stéréotypé dans le fichier projet. Enterprise Architect peut maintenant utiliser le gabarit stéréotypé lors de la génération de code pour les éléments de ce stéréotype.

### Notes

- Les stéréotypes de classe et fonctionnalité peuvent être combinés pour fournir un niveau de spécialisation supplémentaire pour fonctionnalités ; par exemple, si les propriétés doivent être générées différemment lorsque la



classe a un stéréotype MyStereotype, alors la propriété et MyStereotype doivent être spécifiés dans la dialogue New Gabarit Override

## Remplacer Gabarits par défaut

Enterprise Architect dispose d'un ensemble de gabarits de génération de code intégrés ou par défaut. L'éditeur Gabarits de code vous permet de modifier ces gabarits par défaut, personnalisant ainsi la manière dont Enterprise Architect génère le code. Vous pouvez choisir de modifier tout ou partie des gabarits de base pour obtenir le style de codage requis.

Tous gabarits que vous avez remplacés sont stockés dans le fichier projet. Lors de la génération de code, Enterprise Architect vérifie d'abord si un gabarit a été modifié et, si tel est le cas, utilise ce gabarit. Sinon, le gabarit par défaut approprié est utilisé.

### Accéder

Ruban	Développer > Code source > Options > Modifier le code Gabarits
Raccourcis Clavier	Ctrl+Maj+P

### Référence

Remplacez un gabarit de génération de code par défaut à l'aide de l'éditeur Gabarits de code.

Lors de la génération de code, Enterprise Architect utilise maintenant le gabarit de remplacement au lieu du gabarit par défaut.

Champ/Bouton	Description
Langue	Sélectionnez la langue appropriée dans la liste.
Gabarits	Sélectionnez l'un des gabarits de base dans la liste.
Les stéréotypes supplantent	Si le gabarit de base comporte des remplacements stéréotypés, vous pouvez en sélectionner un dans la liste.
<Autres domaines>	Apportez toutes les autres modifications nécessaires.
Sauvegarder	Cliquez sur ce bouton pour enregistrer la version modifiée du gabarit dans le fichier projet. Le gabarit est marqué comme modifié.

## Cadre grammatical

Enterprise Architect fournit support de l'ingénierie inverse pour un certain nombre de langages de programmation courants. Cependant, si le langage que vous utilisez n'est pas pris en charge, vous pouvez écrire votre propre grammaire à l'aide de l'éditeur de grammaire intégré. Vous pouvez ensuite incorporer la grammaire dans une MDG Technologie pour fournir à la fois support de l'ingénierie inverse et de la synchronisation de code pour votre langage cible.

Le framework permettant d'écrire une grammaire et de l'importer dans Enterprise Architect est le complément direct du framework Code Gabarit . Alors que gabarits de code servent à convertir un modèle en une forme textuelle, les grammaires sont nécessaires pour convertir un texte en modèle. Les deux sont nécessaires pour synchroniser les modifications dans vos fichiers sources.

Un exemple de fichier source de langue et un exemple de grammaire pour cette langue sont fournis dans le répertoire Exemples de code, auquel vous pouvez accéder à partir de votre répertoire d'installation (l'emplacement par défaut est C:\Program Files\Sparx Systems\EA). Deux autres fichiers de grammaire sont également fournis, illustrant des aspects spécifiques du développement de grammaires.

## Composants

Composant	Description
Grammaire Syntaxe	<p>Les grammaires définissent la manière dont un texte doit être découpé en une structure, ce qui est nécessaire lorsque vous convertissez du code en une représentation UML . Au niveau le plus simple, une grammaire est constituée d'instructions permettant de découper une entrée pour former une structure.</p> <p>Enterprise Architect utilise une variante du formulaire Backus-Naur (nBNF) pour inclure des instructions de traitement, dont l'exécution renvoie des informations structurées à partir des résultats analysés sous la forme d'un arbre de syntaxe abstraite (AST), qui est utilisé pour générer une représentation UML .</p>
Éditeur de grammaire	L'éditeur de grammaire est un éditeur intégré que vous pouvez utiliser pour ouvrir, modifier, valider et enregistrer des fichiers de grammaire.
Débogage de la grammaire	<p>Vous pouvez déboguer les fichiers de grammaire que vous créez en utilisant deux facilités :</p> <ul style="list-style-type: none"> <li>• L' Parser , qui génère l'AST pour la grammaire</li> <li>• Le profileur, qui analyse également la grammaire et génère l'AST mais qui expose le chemin de profilage pour montrer exactement ce qui s'est passé à chaque étape du processus</li> </ul>

# Grammaire Syntaxe

Les grammaires définissent la manière dont un texte doit être décomposé en une structure, ce qui est exactement ce dont vous avez besoin lorsque vous convertissez du code en une représentation UML . Au niveau le plus simple, une grammaire n'est qu'une série d'instructions permettant de décomposer une entrée pour former une structure. Enterprise Architect utilise une variante de la forme Backus-Naur (BNF) pour exprimer une grammaire de manière à lui permettre de convertir le texte en une représentation UML . Ce que la grammaire d' Enterprise Architect offre par rapport à une BNF pure est l'ajout d'instructions de traitement, qui permettent de renvoyer des informations structurées à partir des résultats analysés sous la forme d'un arbre de syntaxe abstrait (AST). Une fois l'AST terminé, Enterprise Architect le traitera pour produire un modèle UML .

## Syntaxe

Syntaxe	Détail
Commentaires	<p>Les commentaires ont la même forme que dans de nombreux langages de programmation.</p> <p>// Vous pouvez commenter la fin d'une ligne en ajoutant deux /.</p> <p>/* Vous pouvez commenter plusieurs lignes en ajoutant un / suivi d'un *.</p> <p>Le commentaire se termine lorsque vous ajoutez un * suivi d'un /. */</p>
Instructions	<p>Les instructions précisent les détails clés du fonctionnement de la grammaire. Elles sont généralement incluses en haut de la grammaire et ressemblent aux appels de fonction dans la plupart des langages de programmation.</p>
Règles	<p>Les règles constituent le corps d'une grammaire. Une règle peut avoir une ou plusieurs définitions séparées par des délimiteurs de type pipe ( ).</p> <p>Pour qu'une règle soit validée, une seule définition complète doit être validée. Les règles se terminent par le caractère point-virgule (;).</p>
Définitions	<p>Une définition est l'un des chemins que peut emprunter une règle. Chaque définition est composée d'un ou plusieurs termes.</p>
Listes de définitions	<p>Une liste de définitions correspond à un ou plusieurs ensembles de termes. Ceux-ci seront évalués dans l'ordre jusqu'à ce que l'un d'eux réussisse. Si aucun ne réussit, la règle qui le contient échoue. Chaque paire de définitions est séparée par un caractère  .</p> <p>Il s'agit d'une règle simple comportant trois définitions :</p> <pre>&lt;greeting&gt; ::= "bonjour"   "salut"   ["bonjour"] "matin";</pre>
Termes	<p>Un terme peut être une référence à une règle, une valeur spécifique, une plage de valeurs, une sous-règle ou une commande.</p>
Commandes	<p>Comme les instructions, les commandes ressemblent à des appels de fonction. Elles remplissent deux fonctions principales :</p> <ul style="list-style-type: none"> <li>• Traiter les jetons d'une manière spécifique ou</li> <li>• Pour fournir un résultat à l'appelant</li> </ul>



# Instructions de grammaire

Les instructions précisent les détails clés du fonctionnement de la grammaire. Elles sont généralement incluses en haut de la grammaire et ressemblent aux appels de fonction dans la plupart des langages de programmation.

## Instructions

Instruction	Description
<code>sensible aux majuscules et minuscules()</code>	L'une de ces deux instructions doit spécifier si la correspondance des jetons doit être sensible à la casse ou non. Par exemple, les langages de la famille BASIC ne sont pas sensibles à la casse, tandis que les langages de la famille C le sont.
<code>caseInsensitive()</code>	
<code>délimiteurs(DelimiterRule: Expression)</code>	L'instruction <code>delimiters</code> indique à l'analyseur lexical quelle règle utiliser pour la découverte des délimiteurs. Les délimiteurs sont utilisés lors de l'analyse des mots-clés et peuvent être définis comme les caractères qui peuvent être utilisés immédiatement avant ou après les mots-clés de la langue.
<code>lex(TokenRule: Expression)</code>	L'instruction <code>lex</code> indique à l'analyseur lexical le nom de la règle racine à utiliser pour son analyse.
<code>analyser(RootRule: Expression)</code> <code>analyser(RootRule : Expression, SkipRule : Expression)</code>	L'instruction d'analyse indique à l'analyseur le nom de la règle racine à utiliser pour son traitement. Le deuxième argument facultatif spécifie une règle de saut (ou d'échappement), qui est généralement utilisée pour gérer les commentaires.

# Règles de grammaire

Les règles sont exécutées pour décomposer le texte en structures. Une règle est composée d'une ou plusieurs définitions, chacune d'elles étant composée d'un ou plusieurs termes.

## Types de règles

Règle	Description
Règles nommées	Un nom suivi d'une liste de définitions. Par exemple : <code>&lt;règle&gt; ::= &lt;terme1&gt; &lt;terme2&gt;   "-" &lt;terme1&gt;;</code>
Règles en ligne	À l'intérieur d'une définition, une règle est définie entre parenthèses. Celles-ci agissent exactement de la même manière que s'il s'agissait d'une règle nommée appelée par un terme. Par exemple : <code>&lt;règle&gt; ::= (&lt;en ligne&gt;);</code>
Règles facultatives	À l'intérieur d'une définition, une règle définie entre crochets. Cette règle réussit même si le contenu échoue. Par exemple : <code>&lt;règle&gt; ::= [&lt;inline&gt;];</code>
Règles répétitives	À l'intérieur d'une définition, un terme suivi d'un signe plus. Cette règle correspond à la règle interne une ou plusieurs fois. Par exemple : <code>&lt;règle&gt; ::= &lt;en ligne&gt;;</code> <code>règle ::= (&lt;terme1&gt; &lt;terme2&gt;)+;</code>
Règles répétitives facultatives	À l'intérieur d'une définition, une règle suivie d'une étoile. Cette règle correspond à la règle interne zéro fois ou plus, ce qui signifie qu'elle réussit même si la règle interne ne réussit jamais. Par exemple : <code>&lt;règle&gt; ::= &lt;inline&gt;;</code> <code>règle ::= (&lt;terme1&gt; &lt;terme2&gt;)*;</code>

# Termes de grammaire

Les termes identifient où les jetons sont consommés.

## Types de termes

Type	Description
Termes concrets	Chaînes entre guillemets. Par exemple, « classe »
Caractères Unicode	Un terme réservé au lexer, ayant le préfixe U+0x suivi d'un nombre hexadécimal. Par exemple : U+0x1234
Gammes	Un terme réservé à l'analyseur lexical, correspondant à n'importe quel caractère entre les deux caractères spécifiés. Par exemple, « a »..« z » ou U+0x1234..U+2345
Références	Le nom d'une autre règle, entre crochets angulaires. Le jeton correspondra si cette règle réussit. Par exemple, <anotherRule>
Commandes	Un appel à une commande spécifique.



## Commandes de grammaire

Les commandes, comme les instructions, ressemblent à des appels de fonction. Elles remplissent deux fonctions principales :

- Traiter les jetons d'une manière spécifique ou
- Pour fournir un résultat à l'appelant

### Commandes

Commande	Description
attribut(Nom : String , valeur : expression)	<p>Crée un attribut sur le nœud AST actuel. L'attribut sera créé avec le nom spécifié dans la source de grammaire et recevra la valeur de tous les jetons consommés dans le cadre de l'exécution de l'expression de valeur.</p> <p>Cette commande produit les attributs du nœud AST sur lesquels Enterprise Architect opère dans l'ingénierie de code.</p>
attributEx(Nom : String ) attributEx(Nom : String , Valeur : String )	<p>Crée un attribut sur le nœud AST actuel sans consommer de jetons. L'attribut sera créé avec le même nom que celui spécifié dans la source de grammaire et avec une valeur vide ou la valeur spécifiée par l'argument Value facultatif.</p> <p>Cette commande produit les attributs du nœud AST sur lesquels Enterprise Architect opère dans l'ingénierie de code.</p>
except(Cible : Expression, Exception : Expression)	<p>Consomme les données d'entrée qui correspondent à l'expression cible, mais échoue sur les données qui correspondent à l'expression d'exception. Son fonctionnement est assez similaire à celui de la commande skip, mais exactement à l'opposé de celle-ci.</p>
échouer()	<p>Fait échouer l'analyseur sur la règle actuelle, y compris toutes les définitions restantes.</p>
mots-clés()	<p>Correspond à n'importe quelle string littérale utilisée comme terme de grammaire ; c'est-à-dire que si vous entrez une string explicite que vous recherchez, elle devient un mot clé.</p>
mapLeft(Cible : Expression, Commun : Expression)	<p>Consomme les données d'entrée qui correspondent à l'expression cible, puis consomme les données d'entrée qui correspondent à l'expression commune.</p> <p>Les nœuds et Attributs AST générés à partir de l'expression commune sont ensuite copiés dans chacun des nœuds de niveau supérieur générés par l'expression cible.</p> <p>Les expressions cibles et communes doivent être analysées avec succès pour que cette commande réussisse.</p>
mapRight(Commun : Expression, Cible : Expression)	<p>Consomme les données d'entrée qui correspondent à l'expression commune, puis consomme les données d'entrée qui correspondent à l'expression cible.</p> <p>Les nœuds et Attributs AST générés à partir de l'expression commune sont ensuite copiés dans chacun des nœuds de niveau supérieur générés par l'expression cible.</p>

	Les expressions communes et cibles doivent être analysées avec succès pour que cette commande réussisse.
nœud (Nom : String , cible : expression)	Crée un nœud AST sous le nœud AST actuel (les nœuds sur lesquels Enterprise Architect opère dans l'ingénierie de code). Le nœud sera créé avec le nom spécifié dans la source de grammaire.
preProcess(Cible : Expression)	Évalue une expression et utilise ces données prétraitées dans plusieurs définitions. Cette fonction est particulièrement utile dans l'analyse d'expressions, où la même expression de gauche sera évaluée par rapport à un certain nombre d'opérateurs. Cette commande réduit le travail que l'analyseur doit effectuer pour que cela se produise.
sauter(Cible : Expression) skip(Cible : Expression, Échap : Expression)	Consomme des données d'entrée (caractères lors de l'analyse lexicale et jetons lors de l'analyse) jusqu'à ce que l'expression « Target » soit mise en correspondance. L'expression facultative « Escape » peut être utilisée pour gérer des instances telles que des guillemets échappés dans des chaînes.
skipBalanced(Origine : Expression, Cible : Expression) skipBalanced(Origine : Expression, Cible : Expression, Échappement : Expression)	Consomme des données d'entrée (caractères ou jetons) jusqu'à ce que l'expression « Target » corresponde et que le niveau d'imbrication atteigne zéro. Si l'expression « Origin » correspond pendant ce processus, le niveau d'imbrication est augmenté. Si l'expression « Target » correspond, le niveau d'imbrication est diminué. Lorsque le niveau d'imbrication atteint zéro, la commande se termine avec succès. Une expression « Escape » facultative peut être fournie.
sauterEOF()	Consomme toutes les données restantes (caractères ou jetons) jusqu'à la fin du fichier.
jeton (cible : expression)	Crée un jeton lors de l'analyse lexicale pour le traitement lors de l'analyse. La valeur du jeton sera la valeur de tous les caractères consommés suite à l'exécution de l'expression cible.
avertissement()	Insère un avertissement dans l'AST résultant.

## Nœuds AST

Lors de la définition d'une grammaire, vous utiliserez des nœuds AST et des attributs de nœud AST qui peuvent être reconnus dans l'ingénierie de code dans Enterprise Architect, dans les résultats AST renvoyés par les commandes `attribute`, `attributeEx` et `node`. Les nœuds et les attributs sont identifiés dans ces tableaux. Tous les autres seront ignorés dans l'ingénierie de code.

### Noeud FILE

Le nœud FILE représente un fichier. Il n'est associé à rien, mais contient toutes les informations requises.

Multiplicité / Nœuds	Description
0..* / PAQUETAGE	Voir le nœud <i>PACKAGE</i> .
0..* / CLASSE	Voir le nœud <i>CLASS</i> .
0..* / IMPORT	Le nœud qui représente l'espace de noms/ Paquetage ou équivalent importé. L'attribut 'NAME' du nœud sera le nom de l'espace de noms/ Paquetage ou équivalent importé.
0..* / COMMENT	Les étiquettes de champ faisant partie d'une règle de saut seront au niveau racine ; le générateur de code recherche les commentaires de ce type par position par rapport au nœud.
0..1 / INSÉRER_POSITION	Cela donne la position où les nouvelles classes, Paquetages et implémentations de méthodes peuvent être insérés dans le fichier. S'il n'est pas trouvé, le générateur de code insère automatiquement les nouveaux éléments immédiatement après que le dernier ait été trouvé dans le code.

### Noeud PACKAGE

Le nœud PACKAGE correspond à un espace de noms ou équivalent dans le fichier. Lors de l'importation avec « paquetage par espace de noms », Enterprise Architect crée un Paquetage directement sous l'importation pour cela et place toutes les classes à l'intérieur. Lorsque vous n'importez pas d'espaces de noms, Enterprise Architect recherche des classes sous ce point, mais ne fait rien avec ce nœud.

De plus, si vous générez avec les espaces de noms activés (voir les rubriques d'aide sur *les options de code* pour les langages génériques), une classe générée ne correspondra pas à une classe dans le code, sauf si elles se trouvent sous la même structure Paquetage.

Contenu dans les nœuds : FICHER

Multiplicité / Nœuds	Description
1 / NOM	Voir le nœud <i>NAME</i> .
0..* / CLASSE	Voir le nœud <i>CLASS</i> .
0..* / PAQUETAGE	Le nœud Paquetage enfant.
0..1 /	Donne la position où le corps Paquetage s'ouvre. Cela peut également être utilisé

POSITION_OUVERTE	comme position d'insertion.
0..1 / INSÉRER_POSITION	Indique la position à laquelle les nouvelles classes et Paquetages peuvent être insérés dans le fichier. S'il n'est pas trouvé, le générateur de code insère automatiquement les nouveaux éléments immédiatement après que le dernier ait été trouvé dans le code.
0..1 / SUPPRIMER	Empêche l'indentation lors de l'insertion dans ce Paquetage .

## Noeud CLASSE/INTERFACE

Le noeud CLASS (ou INTERFACE) est le plus important dans la génération de code. Il est introduit sous forme d'objets Class (ou Interface).

Voir la classe *DECLARATION* et la classe *BODY* .

Contenu dans les noeuds : FILE, PACKAGE, Class BODY

## Déclaration de CLASSE

Contenu dans les noeuds : CLASSE/INTERFACE

Multiplicité / Noeuds	Description
1 / NOM	Voir le noeud <i>NOM</i> .
0..* / PARENT	Voir le noeud <i>PARENT</i> .
0..* / ÉTIQUETTE	Voir le noeud <i>TAG</i> .
0..1 / DESCRIPTION	Voir le noeud <i>DESCRIPTION</i> .
1 / NOM	Le nom de la classe. S'il existe un noeud <i>NAME</i> , celui-ci écrasera cet attribut.
0..1 / PORTÉE	La portée UML de la classe - publique, privée, protégée ou Paquetage .
0..1 / RÉSUMÉ	Si présent, indique qu'il s'agit d'une classe abstraite.
0..1 / VERSION	La version de la classe.
0..1 / STÉRÉOTYPE	Le stéréotype qu'Enterprise Architect doit attribuer à la classe. Cela ne prend pas support les stéréotypes multiples.
0..1 / FEUILLE D'ÎLE	Si présent, indique qu'il s'agit d'une classe feuille/finale/scellée qui ne peut être héritée par aucune sous-classe.
0..1 / MULTIPLICITÉ	Si présent, représente la multiplicité de la classe.
0..1 / LANGUE	En règle générale, vous n'avez pas besoin de définir cette option.

0..1 / REMARQUE	Généralement non utilisé car il est abordé par les commentaires au-dessus de la classe.
0..1 / ALIAS	S'il est présent, représente l'alias de tout identifiant, tel qu'un Namespace , une classe ou une variable.
0..* / MACRO	Ajoute une Valeur Étiquetée numérotée qu'Enterprise Architect peut utiliser pour round -retour entre les macros.

## Noeud BODY de classe

Contenu dans les nœuds : CLASSE/INTERFACE

Multiplicité / Nœuds	Description
0..* / MÉTHODE	Voir le nœud <i>METHODE</i> .
0..* / ATTRIBUT	Voir le nœud <i>ATTRIBUTE</i> .
0..* / CHAMP	Voir le nœud <i>FIELD</i> .
0..* / CLASSE	Voir le nœud <i>CLASS</i> .
0..* / PORTÉE	Voir le nœud <i>SCOPE</i> .
0..* / PROPRIÉTÉ	Ce nœud représente la définition de la propriété dans le corps de la classe.
0..* / ÉTIQUETTE	Voir le nœud <i>TAG</i> .
0..* / PARENT	Voir le nœud <i>PARENT</i> .
0..1 / POSITION_OUVERTE	Indique la position à laquelle le corps de la classe s'ouvre. Cela peut également être utilisé comme position d'insertion.
0..1 / INSÉRER_POSITION	Indique la position à laquelle les nouveaux membres de la classe peuvent être insérés dans le fichier. S'il n'est pas trouvé, le générateur de code insère automatiquement les nouveaux éléments immédiatement après que le dernier ait été trouvé dans le code.

## Noeud SCOPE

Il s'agit d'une fonctionnalité facultative pour les langages ressemblant à C++ qui ont des blocs qui spécifient la portée des éléments. Le langage doit avoir un nom spécifié qui est utilisé pour la portée de tous les éléments du Bloc . À tous les autres égards, il se comporte de manière identique au nœud Class BODY.

Contenu dans les nœuds : classe BODY

Multiplicité / Nœuds	Description

1 / NOM	Utilisé comme portée pour toutes les méthodes et attributs contenus dans la portée.
---------	---

## Noeud METHODE

Contenu dans les nœuds : classe BODY, SCOPE

Multiplicité / Nœuds	Description
1 / Méthode DECLARATION	Voir la méthode <i>DECLARATION Node</i> .

## Méthode DECLARATION Node

Contenu dans les nœuds : METHODE

Multiplicité / Nœuds	Description
0..1 / TYPE	Voir <i>TYPE Node</i> .
0..* / PARAMÈTRE	Voir le nœud <i>PARAMETER</i> .
0..* / ÉTIQUETTE	Voir <i>TAG NODE</i> .
0..1 / DESCRIPTION	Voir le nœud <i>DESCRIPTION</i> .
0..1 / MULTI PARAMÈTRE	Prend en charge le style de déclaration de liste de paramètres de Delphi. C'est l'équivalent de <i>FIELD</i> .
1 / NOM	Le nom de la méthode.
0..1 / TYPE	Le type de retour de la méthode.
0..1 / PORTÉE	La portée UML de la méthode - Public, Privé, Protégé ou Paquetage .
0..1 / RÉSUMÉ	Si présent, indique que la méthode est abstraite.
0..1 / STÉRÉOTYPE	Le stéréotype qu'Enterprise Architect doit attribuer à la méthode. Cette option ne prend pas support les stéréotypes multiples.
0..1 / STATIQUE	Si présent, indique que la méthode est statique.
0..1 / CONST ou CONSTANTE	Si présent, indique que la méthode est constante.
0..1 / PUR	Si présent, indique que la méthode est une méthode pure.
0..1 / QUESTIONNAIRE	Si présent, indique que la méthode est en mode requête/lecture seule.

0..1 / TABLEAU	Si présent, indique que le type de méthode (type de retour) est un tableau.
0..1 / SYNCHRONISÉ	Si présent, indique que la méthode est une méthode synchronisée.
0..* / MACRO	La macro spécifiée dans la déclaration de méthode.
0..1 / CSHARPIMPLEMENTS	Spécifie un comportement spécial pour C# .
0..1 / COMPORTEMENT	Fournit support pour Aspect J, en utilisant le comportement.
0..1 / AFFICHAGE DU COMPORTEMENT	Fournit support pour Aspect J, en utilisant le comportement, et affiche le comportement rétro-conçu sur le diagramme .

## Nœud ATTRIBUT

Contenu dans les nœuds : classe BODY, SCOPE

Multiplicité / Nœuds	Description
1 / TYPE	Voir <i>TYPE Node</i> .
0..* / ÉTIQUETTE	Voir le nœud <i>TAG</i> .
0..1 / DESCRIPTION	Voir le nœud <i>DESCRIPTION</i> .
1 / NOM	Le nom de l'attribut.
0..1 / TYPE	Le type de l'attribut.
0..1 / PORTÉE	La portée UML de l'attribut - Public, Privé, Protégé ou Paquetage .
0..1 / PAR DÉFAUT	La valeur par défaut de l'attribut.
0..1 / CONTAINER ou ARRAY	Si présent, indique le conteneur de l'attribut.
0..1 / CONFINEMENT	Référence ou valeur .
0..1 / STÉRÉOTYPE	Le stéréotype qu'Enterprise Architect doit attribuer à l'attribut. Cette option ne prend pas support les stéréotypes multiples.
0..1 / STATIQUE	Si présent, indique qu'il s'agit d'un attribut statique.
0..1 / CONST ou CONSTANTE	Si présent, indique qu'il s'agit d'un attribut constant.
0..1 / COMMANDÉ	Si présent, indique que l'attribut ( valeur ) est ordonné.
0..1 / LIMITE BASSE	Si présent, représente la bordure inférieure de la valeur de l'attribut.

0..1 / LIMITE HAUTE	Si présent, représente la bordure supérieure de la valeur de l'attribut.
0..1 / TRANSITOIRE ou VOLATILE	Si présent, indique que l'attribut est transitoire ou volatil.

## Nœud FIELD

Un champ correspond à plusieurs déclarations d'attributs en une seule. Tout ce qui n'est pas défini dans les déclarateurs mais défini dans le champ lui-même sera défini pour chaque déclarateur. Tout ce qui est pris en charge dans un attribut est pris en charge dans le champ. Si aucun déclarateur n'est trouvé, cela fonctionne de la même manière qu'un attribut.

Contenu dans les nœuds : classe BODY, SCOPE

Multiplicité / Nœuds	Description
0..* / DÉCLARATEUR	Voir le nœud <i>ATTRIBUTE</i> .

## Nœud PARAMETRE

Contenu dans les nœuds : méthode DECLARATION, TEMPLATE

Multiplicité / Nœuds	Description
1 / TYPE	Voir <i>TYPE Node</i> .
0..* / ÉTIQUETTE	Voir le nœud <i>TAG</i> .
0..1 / DESCRIPTION	Voir le nœud <i>DESCRIPTION</i> .
0..1 / NOM	Le nom du paramètre.
0..1 / TYPE	Le type du paramètre.
0..1 / TYPE	Devrait être dedans, dehors, dehors ou revenir.
0..1 / PAR DÉFAUT	La valeur par défaut du paramètre.
0..1 / FIXE	Si présent, indique que le paramètre est fixe/constant.
0..1 / TABLEAU	Si présent, indique que le type de paramètre est un tableau.

## Nœud NOM

Contenu dans les nœuds : PACKAGE, classe DECLARATION



Multiplicité / Noeuds	Description
1 / NOM	La partie nom.
0..* / QUALIFICATION	La partie qualificative.
0..* / PARTIE DU NOM	Une alternative à l'utilisation de NAME et QUALIFIER. Une string de valeurs, toutes sauf la dernière étant considérées comme des qualificateurs. La dernière est considérée comme le nom.

## Nœud TYPE

Contenu dans les nœuds : méthode DECLARATION, ATTRIBUTE, PARAMETER

Multiplicité / Noeuds	Description
0..1 / MODÈLE	L'ensemble du texte du gabarit est le nom du type. Utilisé uniquement si NAME n'est pas défini. Voir le nœud <i>TEMPLATE</i> .
1 / NOM	La partie nom.
0..* / QUALIFICATION	La partie qualificative.
0..* / PARTIE DU NOM	Une alternative à l'utilisation de NAME et QUALIFIER. Une string de valeurs, toutes sauf la dernière étant considérées comme des qualificateurs. La dernière est considérée comme le nom.

## Nœud MODÈLE

Contenu dans les nœuds : TYPE

Multiplicité / Noeuds	Description
0..* / PARAMÈTRE	Voir le nœud <i>PARAMETER</i> .
1 / NOM	

## Nœud PARENT

Contenu dans les nœuds : classe DÉCLARATION

Multiplicité / Noeuds	Description
0..1 / TYPE	A la valeur Parent, Implements ou VirtualP.

1 / NOM	La partie nom du parent.
0..* / QUALIFICATION	La partie qualificative du parent.
0..* / PARTIE DU NOM	Une alternative à l'utilisation de NAME et QUALIFIER. Une string de valeurs, toutes sauf la dernière étant considérées comme des qualificateurs. La dernière est considérée comme le nom.
0..1 / INSTANTIATION	Si présent, indique l'instanciation d'un paramètre gabarit .

## Nœud TAG

Contenu dans les nœuds : classe DECLARATION, méthode DECLARATION, ATTRIBUT, PARAMETER

Multiplicité / Nœuds	Description
1 / NOM	Le nom de la Valeur Étiquetée (l' Étiquette ).
0..* / VALEUR	La valeur de la Valeur Étiquetée .
0..1 / MÉMO	S'il est présent, indique que le type de la Valeur Étiquetée est <memo>.
0..1 / NOMÉMO	Si présent, indique que le type de la Valeur Étiquetée n'est pas <memo>.
0..1 / GROUP	S'il est présent, indique que la valeur est un groupe Valeur Étiquetée .

## DESCRIPTION Nœud

Contenu dans les nœuds : classe DECLARATION, méthode DECLARATION, ATTRIBUT, PARAMETER

Multiplicité / Nœuds	Description
0..* / VALEUR	Le texte qu'Enterprise Architect doit attribuer à la Note .

## Édition des grammaires

Si vous devez écrire et modifier une grammaire pour du code importé dans un nouveau langage de programmation, vous pouvez le faire à l'aide de l'éditeur de grammaire intégré.

### Accéder

Ruban	Développer > Code source > Éditeur de grammaire
-------	---

### Créer et modifier la grammaire

Champ/Bouton	Action
Grammaire ouverte	Affichez un navigateur via lequel vous pouvez localiser et ouvrir le fichier contenant la grammaire que vous souhaitez modifier.
Récent	Les grammaires récemment utilisées peuvent être rapidement accessibles à l'aide de cette zone de liste déroulante.
Sauvegarder	Enregistrer le fichier actuel.
Enregistrer sous	Enregistre une copie du fichier actuel
Valider la grammaire	La validation grammaticale va exécuter une série de tests sur la grammaire actuelle pour s'assurer de sa validité. Des erreurs et des avertissements seront affichés vous informant à la fois des erreurs qui rendront la grammaire inutilisable et des conditions dans lesquelles vous pourriez obtenir des résultats inattendus.
Aide	Afficher cette rubrique d'aide.

### Options Menu Contexte

Champ/Bouton	Action
Ouvrir le fichier	Affichez un navigateur via lequel vous pouvez localiser et ouvrir le fichier contenant la grammaire que vous souhaitez modifier.
Valider	La validation grammaticale va exécuter une série de tests sur la grammaire actuelle pour s'assurer de sa validité. Des erreurs et des avertissements seront affichés vous informant à la fois des erreurs qui rendront la grammaire inutilisable et des conditions dans lesquelles vous pourriez obtenir des résultats inattendus.
Langue	L'éditeur de grammaire utilise par défaut la forme Backus-Naur normale (nBNF).

---

	L'option mBNF est également disponible.
Numéros de ligne	Activez ou désactivez les numéros de ligne dans l'éditeur de grammaire.

## Analyse des résultats AST

L'arbre de syntaxe abstraite (AST) est le code qu'Enterprise Architect voit lorsqu'il traite une grammaire.

Vous analysez le texte dans la moitié inférieure de la fenêtre de l'éditeur de grammaire et révision ce qui est affiché comme résultat. Vous pouvez soit ouvrir un fichier, soit coller du texte. Si vous avez collé du texte qui correspond à quelque chose qui ne peut pas apparaître au niveau du fichier (comme les paramètres d'opération), vous pouvez sélectionner une règle alternative à utiliser comme point de départ. L'analyse commencera alors à partir de cette règle.

### Accéder

Ruban	Développer > Code source > Éditeur de grammaire > Débogueur de grammaire > Résultats AST
-------	--

### Options de la barre d'outils

Option	Action
Ouvrir le fichier	Ouvrez un exemple de fichier d'entrée pour effectuer un test.
Récent	Les fichiers sources récemment ouverts peuvent être sélectionnés dans cette zone de liste déroulante.
Analyser	Effectuez l'opération d'analyse. Si l'analyse réussit, l'onglet « Résultats AST » contiendra l'AST résultant.
Sélectionner une règle	Cette liste déroulante vous permet de sélectionner une règle racine alternative pour le traitement de votre source d'échantillon.
Aide	Afficher cette rubrique d'aide.

## Profilage de l'analyse grammaticale

Lorsque vous analysez une grammaire que vous avez créée, elle peut présenter des erreurs que vous ne pouvez pas diagnostiquer immédiatement. Pour vous aider à résoudre ces erreurs, vous pouvez réviser le processus suivi par l'analyseur pour générer l'AST que vous pouvez voir, à l'aide du profileur de grammaire.

Vous analysez à nouveau le texte dans la moitié inférieure de la fenêtre de l'éditeur de grammaire, mais cette fois, l'arborescence affiche chaque règle que l'analyseur a tenté d'appliquer, où il est arrivé et s'il a réussi ou non. Les règles d'ouverture d'un fichier, de collage d'un fichier et de définition de la règle de départ restent les mêmes.

### Accéder

Ruban	Développer > Code source > Éditeur de grammaire > Débogueur de grammaire > Résultats du profileur
-------	---

### Options de la barre d'outils

Option	Action
Ouvrir le fichier	Affichez un navigateur via lequel vous pouvez localiser et ouvrir le fichier contenant la grammaire que vous souhaitez modifier.
Analyser	Effectuez l'opération d'analyse. Si l'analyse réussit, l'onglet « Résultats AST » contiendra l'AST résultant et l'onglet « Résultats du profil » contiendra des informations de débogage concernant le chemin emprunté par l'analyseur dans votre grammaire. Les données de profil sont extrêmement utiles lors du débogage d'une nouvelle grammaire.
Sélectionner une règle	Si vous souhaitez utiliser une règle racine différente pour traiter votre source d'échantillon, cliquez sur la flèche déroulante et sélectionnez la règle alternative.
Aide	Afficher cette rubrique d'aide.

### Notes

- Étant donné que le profilage peut prendre beaucoup de temps pour les fichiers volumineux, l'onglet « Résultats du profil » n'est pas rempli si vous n'affichez pas cet onglet lorsque vous commencez l'analyse

## Éditeur de macros

L'éditeur de macros permet à l'utilisateur de compléter la grammaire avec une liste de mots-clés et de règles pour exclure les macros lors des opérations d'analyse grammaticale. La liste de définitions de macros est particulièrement utile lors du développement de grammaires pour des langages qui supportent les macros tels que C++. Elle évite la nécessité de décrire ces règles dans la grammaire elle-même et peut être utilisée avec plusieurs grammaires.

Cette fonctionnalité est disponible à partir de la version 14.1 Enterprise Architect .

### Accéder

Ruban	Développer > Code source > Éditeur de grammaire > Éditeur de macros
-------	---

### Édition des macros

Ouvrir le fichier	Ouvrir une liste de définitions de macros existante
Récent	Les listes de définitions de macros récemment ouvertes peuvent être sélectionnées à partir de cette zone de liste déroulante
Sauvegarder	Enregistre les modifications apportées à la liste de définitions de macros ouverte
Enregistrer sous	Enregistre une copie de la liste de définitions de macros existante
Valider	Valide la grammaire de la liste de définitions de macro

## Exemples de grammaires

Le répertoire Exemples de code configuré par le programme d'installation Enterprise Architect contient un exemple de grammaire que vous pouvez charger dans l'éditeur de grammaire pour révision , et dans le Débogueur de grammaire pour analyser et profiler.

L'exemple de grammaire se compose de deux fichiers :

- test.ssl - un fichier source de langage d'exemple simple, dans le style de C, et
- ssl.nbnf - une grammaire pour le langage d'exemple simple

L'exemple illustre :

- Tokenisation (en utilisant le Lexer)
- Création d'un Paquetage
- Création d'une classe ou d'une interface
- Création d'un attribut
- Création d'une opération (avec paramètres)
- Importer des commentaires

Le répertoire Exemples de code contient également deux autres fichiers de grammaire que vous pouvez examiner :

- Exemple d'expressions.nBNF - cela illustre comment l'analyse des expressions est configurée et traitée, avec un texte de commentaire détaillé fournissant des explications
- Exemple CSV.nBNF - un exemple de grammaire pour le traitement des fichiers CSV



