



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

# Composer Help

Projets et répertoires

# Projets et répertoires

## Contents

- 1 Projets et répertoires
  - 1.1 Types de projet Application vocale
  - 1.2 Types de projet Application routage
  - 1.3 Structure/répertoires de projet
  - 1.4 Dossiers créés quand mise à niveau projets et schémas
  - 1.5 Ajout de fichiers à un projet existant
  - 1.6 Autorisations de projet
  - 1.7 À l'aide de Composer partagé sous-routines
  - 1.8 Propriétés de projet

Un Composer *projet* est associé avec soit :

- Une application vocale de Genesys Voice Platform ou
- Une application de routage pour la plate-forme Orchestration.

En règle générale, un projet comprend plusieurs prédéfinis, structuré des fichiers et dossiers contenant toutes les ressources de l'application.

Pour plus d'informations sur les projets référencer autres projets, voir la figure dans la rubrique **boîte de dialogue Propriétés de projet**. Développer projet références.

## Types de projet Application vocale

**Les applications de voix** utiliser deux types de la Composer :

- **Java Composer Projects** --utilisation JSP et Java mettent en œuvre les blocs du côté serveur et la logique métier personnalisée. Ceux-ci peut être **déployé** serveurs Tomcat, JBoss ou IBM WebSphere ou d'autres serveurs d'applications web qui répondent aux exigences de la la Composer 8.1 Deployment Guide.
- **Projets de composer .NET** --utiliser ASP.NET et C# mettre blocs du côté serveur et la logique métier personnalisée. Le projet peut uniquement être **déployé** pour Microsoft IIS.

Consultez également **création des Applications vocales pour GVP**. **Remarque :** Projets .NET peuvent afficher cet avertissement dans l'affichage de la Console. "include\getWebServiceData.aspx(482) : AVERTISSEMENT AVERTISSEMENT CS0618 : "Microsoft.Web.Services3.SoapContext.Security' est obsolète : "SoapContext.Security est obsolète. Remplacez découlant SendSecurityFilter ou ReceiveSecurityFilter et créer des assertion de stratégie personnalisée qui génère ces filtres. " Cet avertissement peut être ignoré et aucun solutions ne sont nécessaires. Il apparaîtra pas comme une erreur ou un avertissement dans la vue des problèmes.

## Types de projet Application routage

**Routage applications** sont créés en tant que **Java Composer Projects**. Ils sont des applications SCXML avec le support complet pour le **Genesys Modules fonctionnels**. Une application de routage peut être déployée sur un serveur d'application web remplit les minimums prérequis décrits dans le *Composer 8.1 Deployment Guide*. Consultez également **créer un nouveau projet routage**.

## Structure/répertoires de projet

Un projet Composer (Java ou .NET) contiendra certains ou tous les sous-dossiers selon le type de projet :

- App\_Code --.NET Composer projets uniquement. Ce dossier sera vide par défaut comme groupés Composer toutes les C# classes dans le fichier ComposerBackend.dll. C# classes personnalisées seront également aller dans ce dossier.
- Corbeille --des bibliothèques utilisés dans un projet de Composer .NET aller ici.
- Callflows --dossier pour l'enregistrement de tous les schémas callflow (fichiers .callflow)
- DB -- **base de données connection.properties** et les fichiers .sql sont stockés ici.
- include --standard Composer fourni include des fichiers utilisés par **dorsal** blocs logiques.

Les fichiers JavaScript personnalisés (.js) peuvent être inclus dans une application de routage en plaçant le déroulement dans la include/utilisateur dossier. Une nouvelle génération du code pour tous les schémas IPD dans le projet est obligatoire, après avoir placé les fichiers. Les fonctions JavaScript dans le fichier spécifié .js peuvent ensuite être utilisées à partir de n'importe quel bloc de flux de travail qui les prend en charge les expressions par écrit par exemple les blocs attribuer, branchement et ECMAScript.

- META\_INF --créé lorsque vous créez un nouveau projet de Composer Java. Il est nécessaire pour Java et est inclus dans un fichier .war exportation à partir de Composer. Ne faites pas de modifications dans ce répertoire.
- WEB-INF/lib --Java Composer projets uniquement. Dossier pour bibliothèques de dépendance externe telles que les fichiers JAR. Remarque : Le serveur d'applications Tomcat doit être redémarré après avoir modifié les fichiers JAR dans ce dossier.
- Interaction processus --dossier pour l'enregistrement de toutes les **schémas de processus d'interaction** (fichiers .ixnprocess).
- Ressources --dossier pour les ressources audio et grammaire. Ressources/grammaires--dossier pour **grammaire Builder** (fichiers .gbuilder) et les fichiers GrXML.
  - Ressources/grammaires/<code langue> --places spécifique à la langue grammaires ici (telles que les dossiers US ou es MX).
  - Ressources/invites --dossier pour invite fichiers.
  - Ressources/invites/<code langue> --Place spécifique à la langue invite ici. Si la langue de l'application est modifié appel en cours à l'aide d'un bloc de définir la langue, les invites chemins de ressource audio dans ces dossiers de langue seront être convertis en langage en cours en cours d'exécution.
- Scripts --dossier pour ECMAScript écrit à l'utilisateur. Les fichiers JavaScript personnalisés (.js) peuvent être inclus dans une application vocale en plaçant le déroulement dans la Scripts dossier.
- src-gen --dossier pour les fichiers de VXML/SCXML de génération de code.
- upgradeReports --lorsque **migration stratégies IRD dans Composer**, dossier pour les rapports de migration. Également utilisé pour les rapports à suite de **mise à niveau projets et schémas**.
- SRC --dossier pour code personnalisé, telles que les pages de la logique dorsal écrit par l'utilisateur.
- **Flux de travail** --dossier pour l'enregistrement de tous les schémas de flux de travail (.workflow fichiers).

Code de statique VXML/SCXML est généré avec le nom du fichier de schéma de Composer. Le code est enregistré dans le dossier de src-gen sous le projet actif en cours. Les deux types de projet ont la nature de projet différente. Basé sur la nature de ces projet, des préférences, rédaction et constructeurs différents sont associés à projets. Par exemple, .NET Composer projets et Composer

Java ont des préférences différentes pour le déploiement, car ils sont déployés aux serveurs web/application différente.

## Dossiers créés quand mise à niveau projets et schémas

Les dossiers supplémentaires suivantes peuvent également être créés dans l'Explorateur de projet :

- Lors de la mise à niveau vers 8.1, une mise à niveau projet crée le dossier `./WEB-INF/lib`, copie des fichiers de `./lib` à `./WEB-INF/lib`, puis supprime le `./lib` dossier dans le projet de Composer Java.
- `archivage` --pour la passation aux contenu origine du projet Composer (créé pendant une mise à niveau).
- `upgradeReports` --pour les rapports de mise à niveau (créés pendant une mise à niveau).

## Ajout de fichiers à un projet existant

Composer recommande l'ajout de fichiers (c'est-à-dire, un fichier audio invites) à un projet existant au sein de Composer à l'aide des méthodes suivantes :

- Utilisez le **fichier > importation** capacité.
- Ajouter directement à partir de l'Explorateur Windows, puis actualisez la liste des ressources en appuyant sur F5 dans l'Explorateur de projet de Composer.
- Faites glisser les fichiers sur Explorateur de projet de Composer.

Pour les fichiers désynchronisés, reportez-vous à rubrique de dépannage **Workspace fichiers pas synchronisés**.

## Autorisations de projet

Processus génération composer projet mise à niveau et le code est nécessaire `current\launching` utilisateur la permission écriture pour les fichiers et répertoires de projet Composer. Si vous déplacez projets entre Windows et de système d'exploitation X, ces considérations peuvent s'appliquer :

ÉCRIRE l'autorisation de :

Dans le système d'exploitation Windows 7, projets créés à l'aide de système d'exploitation Mac doit à définir cette autorisation. Pour ce faire :

1. Ouvrez l'Explorateur Windows et accédez au répertoire projet Composer.
2. Le dossier de projet faites un clic droit et sélectionnez le **propriétés** option pour ouvrir la boîte de dialogue Propriétés.
3. Sélectionnez le **sécurité** onglet et cliquez sur le **avancé** bouton.

4. Dans la boîte de dialogue Propriétés avancées, sélectionnez le **autorisations effectives** onglet.

5. Dans l'onglet autorisations effectives, sélectionnez l'utilisateur en cours / groupes pour accorder l'autorisation complète.

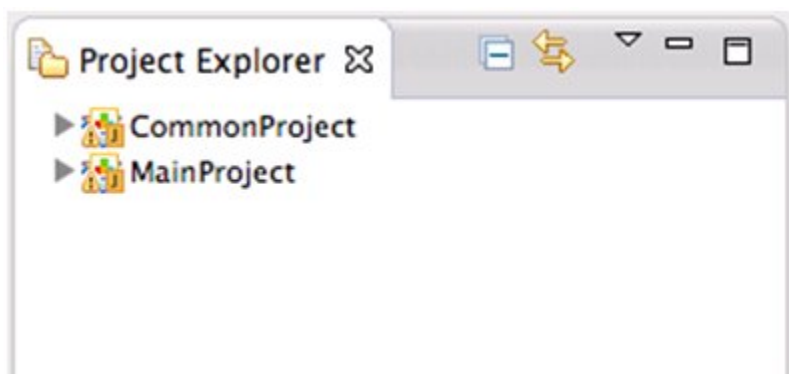
Également décocher la **en lecture seule** et **masqué** Propriétés dans le **générales** onglet pour les répertoires de projet et secondaire. Remarque : Lors de l'importation de Composer projets, si le **copier projets dans l'espace de travail** option est sélectionnée, les autorisations susmentionnées doit être défini pour le répertoire de projet copié séparément.

## À l'aide de Composer partagé sous-routines

Généralement, sous-routines font partie du projet dans lequel les autres schémas les appeler. Cela le projet autonome qui peut être déployé en tant qu'unité avec dépendances minimales sur d'autres projets. Toutefois, dans certains cas sous-routines peuvent être utilisés par plusieurs projets mais doivent être présents dans le seul emplacement dans l'espace de travail. Généralement, ce il faut stockés dans un projet unique dans l'espace de travail est déterminé par le besoin de déployer sur tous les sous-routines à un seul endroit à partir d'où ces sous-routines peuvent être référencées par plusieurs applications - similaires à la manière dont un service est exposé. Il est recommandé que sous-routines une partie du projet ils passent dans et pour activer cette "partage" par le biais d'un système SCM (par exemple, les liens symboliques ClearCase ; autre système va prendre en charge cette fonction différemment). Si ce n'est pas une option, sous-routines en Composer peuvent être placés dans un "courantes" projet, de sorte que plusieurs autres projets peuvent accéder et les réutiliser. REMARQUE : Afin de prendre en charge la substitution URL dans le "\$\$" variables, cette fonction nécessite Orchestration Server version 8.1.300.27 et ci-dessus.

Dans l'exemple, nous crée deux projets :

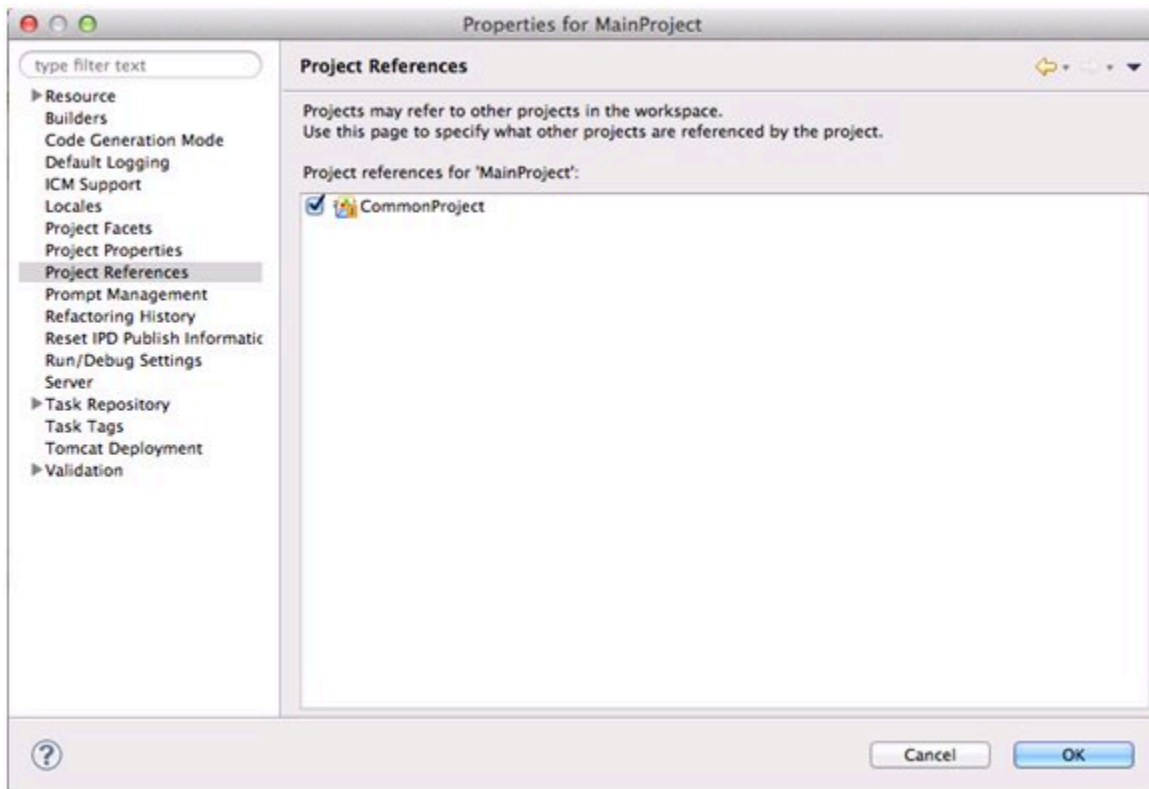
- CommonProject – le projet contenant sous-routines
- MainProject – le projet contenant les schémas principales, qui utilisent des programmes dans CommonProject



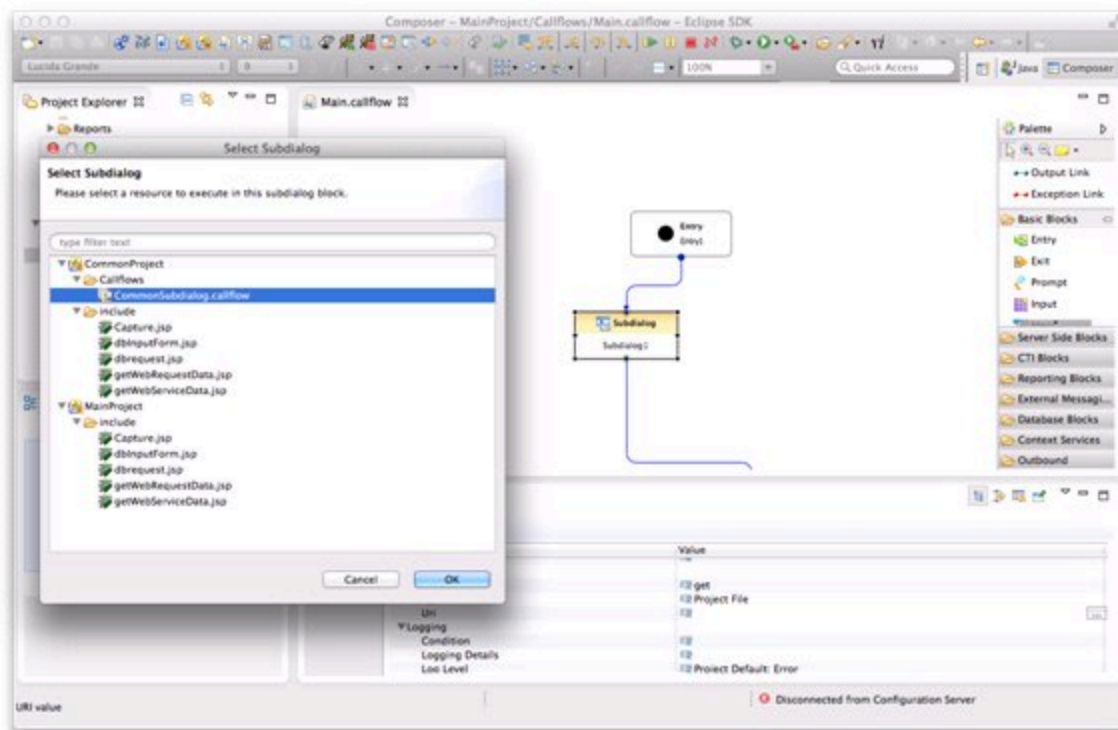
Après création sous-routines dans CommonProject, MainProject doit être défini comme référence à CommonProject. Cela signifie que MainProject pouvez utiliser les fichiers sous-routines dans CommonProject.

Pour ce faire, ouvrez la page de propriétés de projet de MainProject en cliquant et en sélectionnant

**propriétés.** Sélectionnez le **références projet** page sur le côté gauche et activez la case à cocher pour **CommonProject**:



Dans un callflow dans MainProject, vous pouvez créer un **bloc Subdialog** qui utilise un diagramme Subcallflow dans CommonProject:



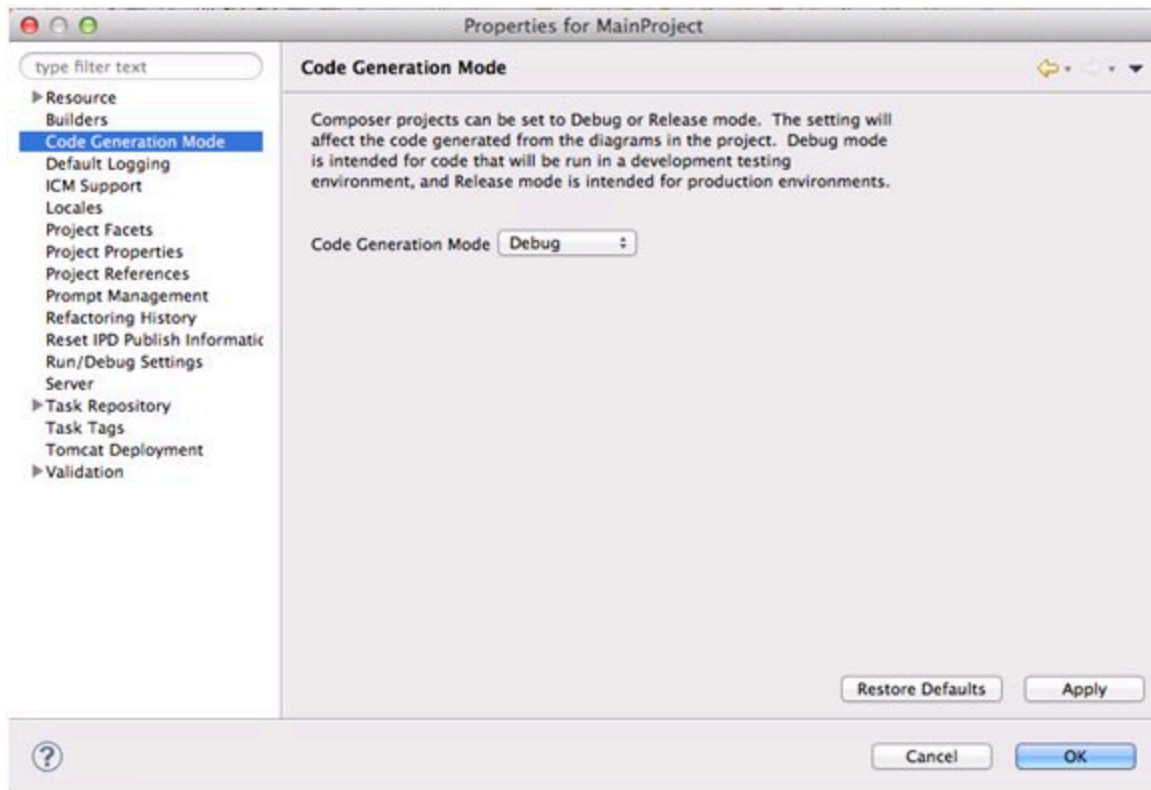
## Les Modes de diffusion et débogage

Lorsque vous utilisez sous-routines partagés, il peut être utile séparer le processus d'élaboration de la dernière application déployée. Pendant le processus d'élaboration, il est supposé que `CommonProject` résident dans l'espace de travail même comme `MainProject`. Toutefois, dans un environnement de production, un service plus complexe peut être nécessaire pour sous-routines de l'hôte.

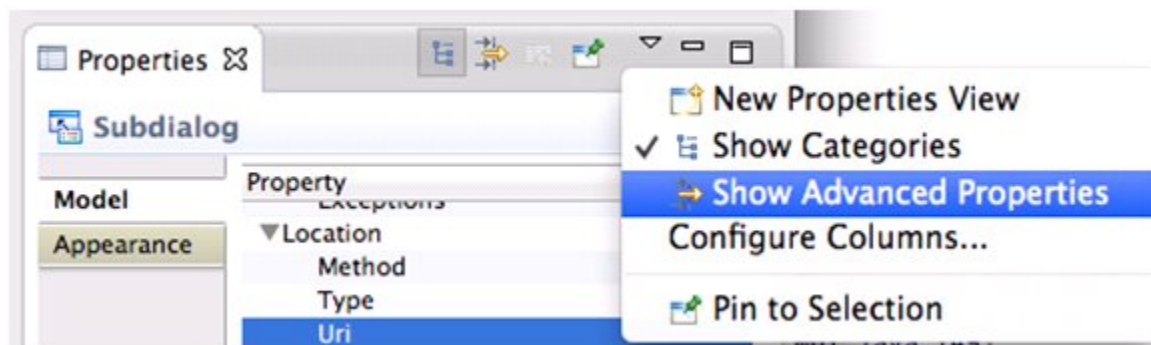
Composer prend en charge le concept de *débogage* et *version* génération de code mode. À l'aide de cet indicateur de mode, le même projet peut générer différent code approprié pour des tâches spécifiques. Si le mode est réglé sur débogage, la référence aux sous-routines sera généré tels qu'ils pointent subroutines en l'espace de travail local et donc aide pendant la phase de développement de débogage. Une fois le développement est terminé, le mode peut être activée pour **version** afin que le code généré à partir de ce point sur s'appeler sous-routines spécifiés par le biais des valeurs dynamiques collectivement doit résoudre vers une URL lors de l'exécution pointant vers un emplacement central hébergeant ces sous-routines.

Débogage et le mode de diffusion peuvent être définies par la boîte de dialogue Propriétés projet :





Pour appliquer **Mode de diffusion** partagés développement sous-routines, ouvrir la vue Propriétés du bloc Subdialog dans le la callflow pour MainProject. Activer le **afficher les propriétés avancées** option :

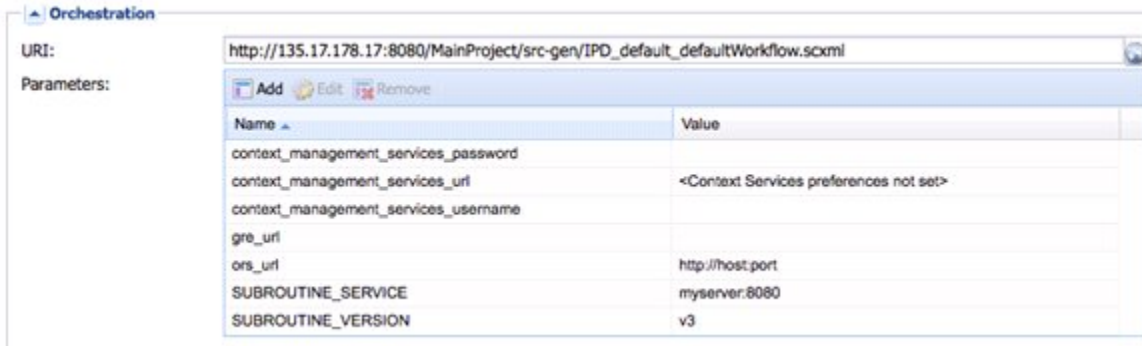


Cela fait apparaître une **la version Mode URI** propriété :

▼ Location	
Method	get
Release Mode Uri	http://\$\$SUBROUTINE_SERVICE\$\$/subroutines/\$\$SUBROUTINE_VERSION\$\$
Type	Project File
Uri	

Notez que n'importe quel élément délimité par « \$\$ » dans cette propriété peut remplacé lors de l'exécution. Dans cet exemple, SUBROUTINE\_SERVICE et SUBROUTINE\_VERSION peuvent être définies

comme des paramètres dans le EnhancedRoutingScript objet, ou selon les paramètres dans un IVR. Par exemple :



The screenshot shows a window titled "Orchestration". It has a "URI:" field with the value "http://135.17.178.17:8080/MainProject/src-gen/IPD\_default\_defaultWorkflow.scxml". Below it is a "Parameters:" section with a table. The table has columns "Name" and "Value".

Name	Value
context_management_services_password	
context_management_services_url	<Context Services preferences not set>
context_management_services_username	
gre_url	
ors_url	http://host:port
SUBROUTINE_SERVICE	myserver:8080
SUBROUTINE_VERSION	v3

Une fois que l'Application est prête à déployer, définissez les **Code générer Mode** du projet à **la version**. Cela génère un code qui utilise les **la version Mode URI** valeur de la propriété.

## Propriétés de projet

Sélection propriétés dans le projet menu s'ouvre une montrant boîte de dialogue la **propriétés** du projet sélectionné ou du projet qui contient la ressource sélectionnée.