



4D Insider version 6.7

ADDENDUM

Ce document présente les nouveautés apparues dans la version 6.7 de 4D Insider.

Création et installation des composants 4D

La version 6.7 de 4D Insider permet de générer des *composants* et de les installer dans des bases 4D 6.7. Cette nouvelle fonctionnalité a pour but de permettre aux développeurs de commercialiser du code 4D en toute sécurité.

Ce chapitre traite des sujets suivants :

- Présentation et caractéristiques des composants
- Développer et générer des composants
- Installer ou mettre à jour un composant
- Contraintes et conseils de développement

Présentation et caractéristiques des composants

Qu'est-ce qu'un composant ?

Un *composant* regroupe un ensemble d'objets 4D (tables, méthodes projet, formulaires, barres de menus, variables...) représentant une ou plusieurs fonctionnalités supplémentaires. Par exemple, vous pouvez développer un composant 4D de courrier électronique. Un composant est autonome, il doit pouvoir être installé dans tout fichier de structure 4D.

Les composants sont définis, générés et installés à l'aide de 4D Insider. L'analyse des références croisées effectuée par 4D Insider (objets cibles et objets sources) permet de regrouper la totalité des objets utilisés par un composant.

A la différence des bibliothèques et des groupes, les composants intègrent la notion de **protection** des objets qui les composent. Lors de la phase de développement du composant, chaque objet se voit attribuer le type "Public", "Protégé" ou "Privé". Cet attribut détermine si l'objet, une fois le composant installé dans la base 4D, sera visible ou modifiable dans 4^e Dimension et dans 4D Insider.

Principes de création

Comme les groupes, les composants sont créés dans 4D Insider par regroupement et ajout d'objets 4D provenant d'une base ou d'une bibliothèque. Chaque nouveau composant dispose automatiquement d'un ensemble de trois dossiers ("Public", "Protégé" et "Privé"), parmi lesquels le développeur doit répartir chaque objet.



Une fois la répartition terminée, le développeur génère le composant, qui devient alors un fichier stocké sur disque. Ce fichier pourra ensuite être installé dans une base 4D.

Principes d'installation

Vous pouvez installer autant de composants que vous voulez dans une base 4D.

L'installation d'un composant s'effectue par l'intermédiaire d'un fichier de composant, généré par 4D Insider et stocké sur disque.

La procédure d'installation de ce fichier dans une base 4D s'effectue également depuis 4D Insider. Ce fonctionnement facilite la diffusion et la mise à jour des composants 4D. 4D Insider permet en effet de mettre à jour des bases 4D lorsque de nouvelles versions des composants sont générées.

Enfin, il est possible de désinstaller un composant.


Définition des attributs

Lors de la création d'un composant avec 4D Insider, chaque objet inclus dans le composant reçoit l'un des trois attributs suivants :

Public, **Protégé** ou **Privé**. Ces attributs, à la base du système de protection des composants, indiquent si les objets concernés seront visibles et modifiables dans le mode Structure de 4^e Dimension et dans 4D Insider, une fois le composant installé.

Note Bien entendu, en phase de développement d'un composant, tous les objets restent visibles et déplaçables dans 4D Insider, quel que soit leur attribut.

- **Public** : les objets “publics” seront visibles et modifiables par les utilisateurs, toutefois ils ne pourront être ni renommés ni supprimés. Ce type d’objet peut être utile pour fournir des objets personnalisables par les utilisateurs.
Dans les éditeurs de 4D, les objets publics apparaissent comme tous les autres objets.
- **Protégé** : les objets “protégés” seront visibles mais ne pourront être ni modifiés ni supprimés par les utilisateurs. Une méthode protégée peut être appelée, mais son contenu ne peut être ni visualisé ni modifié (la zone de prévisualisation de l’Explorateur reste vide).
Dans les éditeurs de 4D, l’icône des objets protégés est barrée d’un trait rouge :

 MyMod_Prot_CallsTable1

- **Privé** : les objets “privés” ne seront ni visibles ni, par conséquent, modifiables par les utilisateurs des composants, aussi bien dans 4^e Dimension que dans 4D Insider.

Note Les tables et les champs ne peuvent avoir l’attribut “Privé”. En outre, en cas de désinstallation du composant, ils ne seront pas supprimés de la base. Reportez-vous au [paragraphe “Désinstaller un composant”, page 14](#).

Le tableau suivant résume les possibilités offertes dans 4D et 4D Insider par les objets contenus dans les composants, en fonction de leur attribut :

	Nom visible	Contenu visible	Contenu modifiable	Renommable ou supprimable
<i>Public</i>	Oui	Oui	Oui	Non
<i>Protégé</i>	Oui	Non	Non	Non
<i>Privé</i>	Non	Non	Non	Non

Note De manière générale, les attributs des objets des composants seront respectés par toutes les applications et plug-ins de l’environnement 4D, tels que 4D Write ou 4D Compiler.

Visualisation et utilisation des composants installés

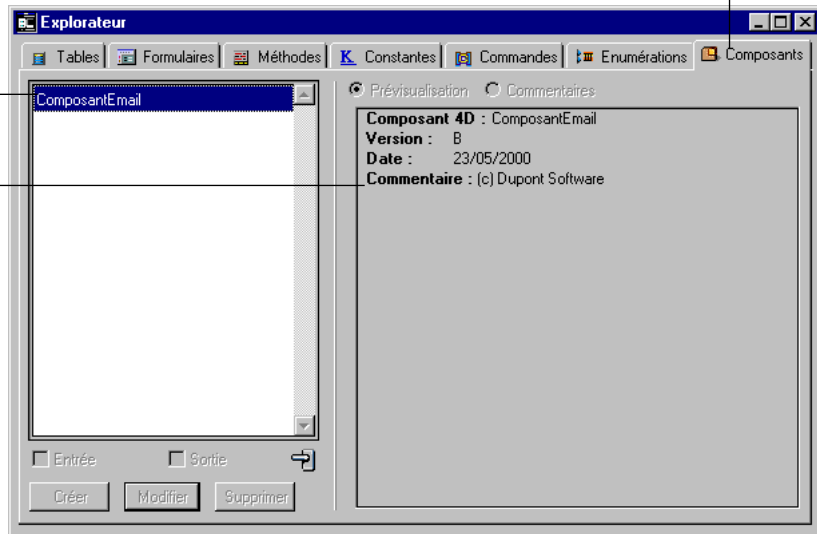
Lorsqu'un composant est installé dans une base 4D, tous les objets qu'il contient sont ajoutés dans la base et sont disponibles dans les éditeurs du mode Structure lorsque leur type est "Public" ou "Protégé" (cf. [paragraphe "Définition des attributs", page 2](#)).

Chaque composant installé dans la base est listé dans la page **Composants** de l'Explorateur de 4D :

Onglet Composants

Liste des composants installés dans la base

Propriétés du composant sélectionné (saisies dans 4D Insider)



Dans 4D Insider, un composant installé reste visible dans les listes. Le filtre **Groupes & composants** permet d'afficher les composants installés.

Toutefois, le dossier "Privé" n'apparaît pas et le contenu du dossier "Protégé" ne peut être visualisé. Aucun objet du composant ne peut être supprimé ou renommé. Les objets ne peuvent être dissociés. Enfin, le composant ne peut être déplacé dans une autre base ou librairie.

Développer et générer un composant

Le développement d'un nouveau composant 4D s'effectue en trois principales étapes :

1 Développement de la (des) fonctionnalité(s) du composant dans 4D.

Le développement de composants nécessite quelques précautions. Reportez-vous au [paragraphe "Contraintes et conseils de développement", page 14](#).

2 Création dans 4D Insider d'un nouveau composant regroupant l'ensemble des objets, et définition de l'attribut de chaque objet.

Cette étape est traitée dans le [paragraphe "Créer un nouveau composant", page 6](#).

3 Génération dans 4D Insider d'un "fichier de composant" destiné à l'installation du composant dans les bases 4D.

Cette étape est traitée dans le [paragraphe "Générer le fichier de composant", page 10](#).

L'ultime étape consiste à installer ou mettre à jour le composant dans les bases 4D. Reportez-vous au [paragraphe "Installer et mettre à jour un composant", page 11](#).

Types d'objets gérés

La plupart des objets identifiés par 4D Insider comme "objets déplaçables" peuvent être inclus dans un composant :

- Tables et champs
- Triggers
- Formulaires
- Méthodes formulaire, Méthodes objet et Méthodes projet
- Menus et Barres de menus
- Enumérations
- Info-bulles
- Ressources¹ (STR# et PICT)
- Images de la bibliothèque
- Formats et filtres
- Feuilles de style
- Commentaires

1. L'utilisation de ressources dans un composant nécessite des précautions d'emploi. Reportez-vous au [paragraphe "Gérer les numéros de ressources STR# et PICT", page 15](#).

Les objets de langage suivants peuvent également être inclus dans un composant :

- Variables locales, process et interprocess
- Ensembles
- Sélections
- Sémaphores

Note Par conséquent, notez en particulier que les méthodes base, plug-ins, groupes et composants ne peuvent pas être inclus dans un composant.

Créer un nouveau composant

La création d'un nouveau composant s'effectue à partir d'une librairie ou d'une base de données 4D ouverte par 4D Insider.

4D Insider vous propose deux options :

- Créer un composant vide, que vous remplirez par glisser-déposer.
- Créer un composant à partir de la sélection courante d'objets. Par défaut, les objets sélectionnés seront placés dans le dossier "Protégé". Ce composant peut être modifié par la suite.

► Pour créer un composant vide :

- 1 Ouvrez la structure d'une base ou d'une librairie avec 4D Insider.**
- 2 Sélectionnez la commande Nouveau... dans le menu Composants.**

La boîte de dialogue des propriétés du composant apparaît :

Choix du mode (désactivé lorsqu'aucun objet n'est sélectionné)

3 Saisissez les informations concernant le composant.

Ces informations seront affichées dans l'Explorateur de 4D, une fois le composant installé. Notez que les propriétés du composant peuvent être modifiées à tout moment. Pour plus d'informations, reportez-vous au [paragraphe "Propriétés du composant", page 8](#).

4 Si le bouton radio de choix du mode de création est désactivé (aucun objet n'était sélectionné lorsque vous avez utilisé la commande Nouveau...), cliquez sur le bouton OK.

OU

Si le bouton radio de choix du mode est actif, sélectionnez l'option "Créer un composant vide" et cliquez sur le bouton OK.

Le nouveau composant est immédiatement créé dans une nouvelle fenêtre. Il comporte trois dossiers vides, "Privé", "Protégé" et "Public".

5 Répartissez dans les dossiers chaque objet que vous souhaitez inclure dans le composant.

- Pour **ajouter** un ou plusieurs objet(s) dans un des dossiers du composant, sélectionnez-le(s) dans les autres fenêtres, faites-le(s) glisser et déposez-le(s) dans le dossier souhaité. Vous pouvez également **réorganiser** le contenu des dossiers du composant par glisser-déposer d'objets.

- Pour **supprimer** un ou plusieurs objet(s) d'un dossier du composant, sélectionnez-le(s) et faites un glisser-déposer de la sélection vers la fenêtre d'origine.

Bien entendu, seuls les objets déplaçables peuvent être inclus dans un composant (cf. [paragraphe "Types d'objets gérés", page 5](#)).

Les objets sont déplacés avec leurs dépendances, en respectant la cohérence des attributs. Par exemple, si un formulaire est placé dans le dossier "Privé", tous les objets qu'il référence seront également déplacés dans le même dossier. Pour plus d'informations sur ce point, reportez-vous au [paragraphe "Attributs des objets déplacés", page 9](#).

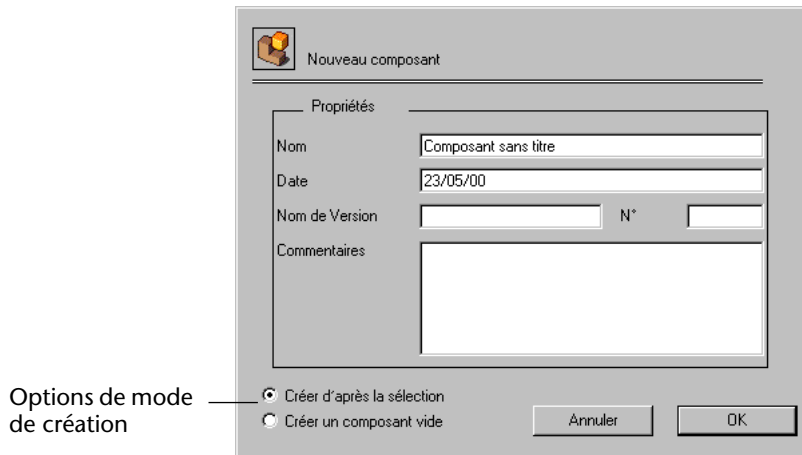
► Pour créer un composant à partir de la sélection courante d'objets :

1 Ouvrez la structure d'une base ou d'une librairie avec 4D Insider.**2 Sélectionnez, dans la liste Base, les objets que vous souhaitez inclure dans le composant.**

Seuls les objets déplaçables seront inclus dans le composant (cf. [paragraphe "Types d'objets gérés", page 5](#)).

3 Sélectionnez la commande Nouveau... dans le menu Composants.

La boîte de dialogue des propriétés du composant apparaît.



4 Saisissez les informations concernant le composant.

Ces informations seront affichées dans 4D, une fois le composant installé. Notez qu'en phase de développement, les propriétés du composant peuvent être modifiées à tout moment. Pour plus d'informations, reportez-vous au [paragraphe "Propriétés du composant", page 8](#).

5 Vérifiez que l'option "Créer d'après la sélection" est sélectionnée et cliquez sur le bouton OK.

Le composant est immédiatement créé dans une nouvelle fenêtre. Il comporte trois dossiers, "Privé", "Protégé" et "Public".

- Tous les objets sélectionnés sont automatiquement placés dans le dossier "Protégé".
- Toutes les dépendances (cibles) des objets sélectionnés sont automatiquement placées dans le dossier "Privé".

6 Si nécessaire, réorganisez le contenu des dossiers "Privé", "Protégé" et "Public" en utilisant le glisser-déposer (cf. étape n°5 de la création de composants vides).

Propriétés du composant

Les propriétés du composant apparaissent dans la zone de prévisualisation de l'Explorateur de 4D, lorsque celle-ci est déployée : Nom, Numéro de version, Nom de version, Date de création et Commentaires. Ces informations permettront d'identifier précisément tout composant installé dans une base. Vous pouvez afficher à tout moment les propriétés du composant sélectionné en choisissant la commande **Propriétés...** dans le menu **Composants**.

- **Nom** : Nom du composant, tel qu'il apparaîtra aux utilisateurs. Au sein d'une même base ou librairie, le nom d'un composant doit être unique.
Ce nom sera utilisé par 4D Insider au moment de l'intégration du composant pour déterminer si l'opération est une installation ou une mise à jour. Ce nom sera aussi utilisé par défaut pour le fichier de composant généré.
Le nom d'un composant peut comporter jusqu'à 32 caractères.

Note Vous pouvez renommer un composant à l'aide de la commande **Renommer** de 4D Insider.

- **Date** : Date de création du composant. Par défaut, la date du jour est affichée. Vous pouvez toutefois saisir toute date.
- **Numéro de version** : Numéro de version du composant, utile pour la maintenance et la mise à jour des composants. S'il s'agit d'une mise à jour, ce numéro est utilisé par 4D Insider pour déterminer si le composant en cours d'installation est plus récent que le composant déjà installé. Cette zone est obligatoire et le numéro de version doit être supérieur à 0.
- **Nom de version** : Indicateur du type de version du composant ("Démonstration", "Complète", "Macintosh", etc.). Le nom de version sera ajouté au nom du composant afin de fournir le nom par défaut du fichier de composant.
- **Commentaires** : Tout texte (255 caractères) que vous jugerez utile d'associer au composant (copyrights, conseils, etc.).

Attributs des objets déplacés

Lorsque vous déplacez des objets vers les dossiers des composants, 4D Insider déplace automatiquement leurs dépendances (objets cibles).

Les attributs par défaut des objets cibles sont déterminés de la manière suivante :

Objet placé dans le dossier	Attribut des dépendances de l'objet
Public	Public
Protégé	Privé
Privé	Privé

Si un même objet apparaît simultanément dans les dossiers "Public" et "Protégé" (par exemple, une variable utilisée à plusieurs emplacements de la base), il sera défini en "Public".

De manière générale, lorsqu'un même objet est référencé dans plus d'un dossier du composant, le niveau de protection le moins élevé est utilisé.

Opérations sur les composants en cours de création dans 4D Insider

Toutes les fonctions de 4D Insider continuent de s'appliquer aux objets placés dans les différents dossiers d'un nouveau composant.

Les composants s'affichent toujours à la suite des autres objets, quel que soit le type de filtre qui est appliqué. Le filtre **Grouper & composants** permet d'afficher les composants (en développement ou installés).

Dans le cas d'une sélection à laquelle vous avez appliqué un filtre, les dossiers du composant ouvert n'afficheront que les objets correspondant à ce filtre.

De même, lorsque vous effectuez une recherche, il se peut que certains des objets répondant aux critères de recherche se trouvent à l'intérieur d'un composant. Dans ce cas, le composant apparaîtra dans la liste issue de la recherche, mais n'affichera que les objets trouvés.

Générer le fichier de composant

Une fois que vous avez défini un composant, vous devez générer le fichier qui sera utilisé pour l'installation ou la mise à jour du composant. Un fichier de composant est multi-plate-forme : il peut être généré sur une plate-forme et installé sur une autre.

► Pour générer un fichier de composant :

1 Dans la liste Base de 4D Insider, sélectionnez le composant à générer.

2 Choisissez la commande Générer... dans le menu Composants.

Une boîte de dialogue standard d'enregistrement de fichiers apparaît. Par défaut, un fichier de composant reçoit le nom donné au composant suivi d'un espace puis du nom et du numéro de la version (par exemple "MonComposant Démo-2"). Sous Windows, l'extension ".4CP" lui est en outre associée et un fichier ".RSR" est généré.

3 Définissez le nom et l'emplacement du fichier et cliquez sur Enregistrer.

Le fichier de composant est alors généré sur le disque, prêt à être installé.

Installer et mettre à jour un composant

L'installation d'un nouveau composant ou d'une version de mise à jour au sein d'une base 4D s'effectue par l'intermédiaire de 4D Insider. Le principe consiste à installer le composant (généré par 4D Insider) dans une structure¹ préalablement ouverte par 4D Insider.

Le programme gère les éventuels conflits issus de l'installation et en informe l'utilisateur à mesure qu'ils sont détectés (cf. [paragraphe "Gestion des conflits d'installation"](#), page 12).

► Pour installer ou mettre à jour un composant :

1 Ouvrez avec 4D Insider la base dans laquelle vous souhaitez installer le composant.

2 Choisissez la commande Installer/Mettre à jour... dans le menu Composants.

Une boîte de dialogue standard d'ouverture de fichiers apparaît.

3 Sélectionnez le fichier du composant et cliquez sur le bouton Ouvrir.

4D Insider analyse le contenu du composant et l'intègre à la base ouverte.

4D Insider détecte si l'opération correspond à une installation ou à une mise à jour. Cette détection est basée sur le nom et le numéro de version du composant, définis dans les Propriétés du composant (cf. [paragraphe "Propriétés du composant"](#), page 8).

- en cas d'installation nouvelle, tous les objets du composant sont installés.
- en cas de mise à jour, 4D Insider compare le numéro de version du composant en cours d'installation avec celui du composant déjà présent. Si la date du "nouveau" composant est plus ancienne que celle du composant déjà installé, une boîte de dialogue d'alerte le signale à l'utilisateur — il peut alors **Poursuivre** ou **Annuler** la mise à jour.

4D Insider remplace les anciens objets par les objets plus récents (nom identique), ajoute les nouveaux objets et ne modifie pas les objets n'existant plus dans la nouvelle version du composant.

4D Insider tient compte des objets "publics" ayant été modifiés par l'utilisateur et propose à l'utilisateur de les conserver ou de les remplacer. En outre, si l'intégration provoque des conflits ou des interrogations, l'utilisateur en est informé.

1. Il n'est pas possible d'installer un composant dans une librairie.

Ces points sont traités dans le paragraphe suivant, “Gestion des conflits d’installation”.

4 Enregistrez la base à l’aide de 4D Insider.

Le composant est alors installé dans la base et est listé dans la page “Composants” de l’Explorateur de 4D (cf. [paragraphe “Visualisation et utilisation des composants installés”, page 4](#)).

Gestion des conflits d’installation

Lorsqu’un composant est installé ou mis à jour dans une base 4D, divers interrogations et conflits peuvent se produire :

- en cas de mise à jour, 4D Insider détecte qu’un ou plusieurs objets “publics” ont été modifiés par l’utilisateur après l’installation du composant.
- un ou plusieurs objets de même type et de même nom existent déjà dans la base.

4D Insider permet de détecter et de résoudre ces conflits au moment de l’installation :

■ Objets publics modifiés (mise à jour uniquement)

Dans ce cas, 4D Insider affiche une boîte de dialogue d’alerte et permet à l’utilisateur de choisir le mode de mise à jour :

- Remplacer l’objet
- Tout remplacer
- Ne pas remplacer l’objet
- Stopper l’installation

■ Conflits de noms

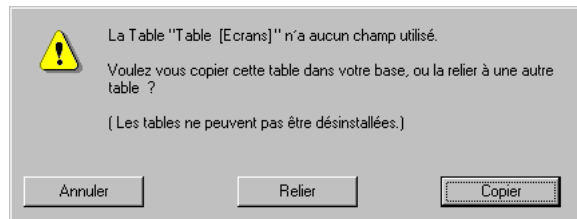
Dans ce cas, 4D Insider stoppe le processus d’installation du composant, affiche une boîte de dialogue d’alerte et enregistre la liste des objets en conflit. Cette liste est stockée sous forme de fichier texte dans le dossier de la base 4D. A noter que les noms de tous les objets mis en cause sont listés, y compris ceux des dossiers “privés” et “protégés”. Les conflits de noms entre les objets logiques tels que les variables sont gérés, de manière à permettre la compilation de la base et éviter les conflits inter-composants.

Pour pouvoir installer le composant, il pourra être nécessaire de renommer certains objets de la base ou du composant.

■ Cas particulier : Table d'interface des composants

Il est fréquent que les développeurs aient besoin de créer une table d'interface dans le but de pouvoir générer des formulaires utilisés comme boîtes de dialogue. Généralement, ces formulaires n'utilisent pas de champs, mais uniquement des variables.

Comme il n'est pas possible de supprimer une table dans 4^e Dimension, l'installation successive de composants dans une même base risque de provoquer une multiplication inutile des tables du type [Interface]. Pour résoudre cette question, 4D Insider propose d'utiliser une "table d'interface des composants". Lorsqu'une telle table est définie dans un composant, une boîte de dialogue spécifique est affichée au moment de l'installation du composant :



Le choix suivant est proposé :

■ **Copier** cette table dans la base

Dans ce cas, la table est créée dans la base 4D, mais devient une table 4D standard, n'appartenant pas au composant. Elle deviendra donc utilisable comme table d'interface lors de l'installation éventuelle d'autres composants.

Cette option peut être utilisée lors de l'installation d'un premier composant.

■ **Relier** la table à une table existante de la base

Dans ce cas, la liste des tables de la base est présentée, l'utilisateur doit sélectionner celle que le composant utilisera comme table d'interface.

Ce fonctionnement est rendu possible par le fait que les composants n'utilisent aucun des champs de la table d'interface ; elle peut être modifiée et utilisée sans risquer d'interagir avec le comportement du composant.

Pour qu'une table puisse être définie comme "table d'interface des composants", ses caractéristiques doivent être les suivantes :

- aucun des champs de la table n'est utilisé par le composant,
- l'attribut de la table est "public".

Désinstaller un composant

4D Insider propose une fonction permettant de désinstaller des composants préalablement installés dans des bases 4D.

- Pour désinstaller un composant d'une base 4D :
 - 1 Ouvrez avec 4D Insider la base contenant le composant à désinstaller.**
 - 2 Dans la liste Base, sélectionnez le nom du composant à supprimer.**
 - 3 Sélectionnez la commande Désinstaller... dans le menu Composants.**

Cette commande n'est active que lorsqu'un composant a été installé dans la base.

Une boîte de dialogue apparaît, vous permettant de confirmer ou d'annuler l'opération.
 - 4 Cliquez sur OK pour valider l'opération.**

Tous les objets du composant sont alors supprimés.

Les tables éventuellement créées par le composant ne sont pas supprimées. Ces objets deviennent "publics" et sont renommées [Table Supprimée]. Les champs ne sont pas modifiés.

Note Pour supprimer tous les objets d'un composant (tables et champs compris), vous pouvez également procéder de la manière suivante : sélectionnez tous les objets de la base contenant le composant et déplacez-les vers une nouvelle base vierge. Comme les objets d'un composant ne peuvent être déplacés, la nouvelle base ne contiendra pas les objets du composant.

Contraintes et conseils de développement

Le développement de composants destinés à être diffusés au sein de multiples bases 4D entraîne des contraintes spécifiques. Ce paragraphe dresse la liste de ces contraintes et fournit quelques conseils de développement.

Eviter les conflits de noms

Lors de l'installation d'un composant, 4D Insider préserve l'intégrité de la base en contrôlant les noms des objets. En cas d'installation d'un nouveau composant, si des objets de même nom sont détectés dans le composant et dans la base, l'installation est annulée.

Vous devez donc veiller à utiliser des noms d'objets originaux, par exemple leur attribuant des préfixes personnalisés.

Nous conseillons aux développeurs de composants d'utiliser la nomenclature suivante :

- saisir un identifiant unique comprenant 5 caractères au début de chaque nom d'objet (par exemple "mkdev"),
- faire suivre cet identifiant d'un trait de soulignement (" _"),
- insérer trois ou quatre caractères indiquant l'attribut de l'objet ("priv" ou "pub"),
- faire suivre cet attribut d'un autre trait de soulignement,
- enfin, saisir le nom de l'objet.

Exemples :

mkdev_Priv_MonFormulaire

DDcom_Pub_MaMéthode

Vous trouverez des informations supplémentaires sur les identifiants uniques et les règles de préfixage à l'adresse Internet suivante : <http://www.4D.com>.

Déclarer les variables

En vue de la compilation des bases ayant intégré un ou plusieurs composants, il est nécessaire de s'assurer qu'aucun conflit ou ambiguïté de typage ne puisse se produire, en particulier au sein de méthodes "privées" ou "protégées". Par défaut, 4D Compiler n'affiche pas les noms des méthodes privées. Toutefois, 4D Compiler listera les noms des méthodes en cause en cas de conflit de typage, quel que soit leur attribut.

Il est donc fortement conseillé de typer toutes les variables 4D utilisées dans les méthodes des composants, par exemple dans une méthode dont le nom débute par *COMPILER* (voir à ce sujet les commandes du thème "Compilateur" dans le manuel *Langage* de 4D).

Gérer les numéros de ressources STR# et PICT

Si votre composant utilise des ressources personnalisées de type PICT (incluses dans des formulaires) ou STR#, il les référence à l'aide de numéros spécifiques.

Or, lorsque le composant est installé dans une structure, il est possible que ces numéros soient déjà attribués, par exemple si un autre composant utilisant des ressources a déjà été installé. Dans ce cas, 4D renumérote automatiquement les nouvelles ressources lors de l'installation. Ce mécanisme garantit qu'aucune ressource n'est "écrasée" au cours de l'installation, mais peut s'avérer problématique si

le code de votre composant référence les ressources par des numéros "en dur".

Pour éviter ce problème, vous pouvez utiliser la nouvelle commande de 4D **Lire ID ressource composant**. Sur la base du nom du composant, du type de ressource et de son numéro original, cette commande retourne le numéro réellement utilisé de toute ressource dans la base.

Par exemple, le code ci-dessous ne permet pas de garantir que les appels de ressources seront correctement effectués :

 `Si les ressources sont renumérotées, cet appel sera incorrect

vNumRes := 15000

LISTE DE CHAINES VERS TABLEAU(vNumRes; tabChaînes; fichierRes)

Il est fortement conseillé de préférer le code suivant :

 `Cet appel sera correct dans tous les cas

vNumRes := **Lire ID ressource composant**("LeComposant"; "STR#"; 15000)

LISTE DE CHAINES VERS TABLEAU (vNumRes; tabChaînes; fichierRes)

De même, si vos formulaires exploitent des ressources PICT, vous pouvez utiliser Lire ID ressource composant avec les commandes de gestion des ressources PICT telles que LIRE RESSOURCE IMAGE.

Pour plus d'informations sur la commande Lire ID ressource composant, reportez-vous au manuel *Langage* de 4D.

Gérer les numéros d'images de la bibliothèque

A l'instar des ressources, si un composant 4D fait appel à des images stockées dans la bibliothèque d'images leurs numéros de référence seront modifiés lors de l'installation au cas où des images ayant le même numéro existent déjà dans la base.

Pour éviter ce problème, vous devez utiliser dans le composant la commande LIRE IMAGE DANS BIBLIOTHEQUE et lui passer comme premier paramètre un **nom** d'image et non un numéro (nouveau 4D 6.7). A la différence du numéro de référence, le nom de l'image ne pourra pas être modifié au cours de l'installation, et les appels seront corrects dans tous les cas.

En revanche, comme pour les autres objets, 4D Insider interdira l'installation d'un composant si une image de même nom existe déjà dans la bibliothèque.

Note La commande SUPPRIMER IMAGE DANS BIBLIOTHEQUE ne fait rien avec une image appartenant à un composant. La commande ECRIRE IMAGE

DANS BIBLIOTHEQUE ne fait rien avec une image appartenant à un composant et dont l'attribut est "privé" ou "protégé".

Protéger les composants Vous pouvez, si vous le souhaitez, lier chaque copie de votre composant à une seule application 4D installée, afin d'empêcher toute copie illicite. En effet, une clé unique est générée pour chaque application 4D installée. Sur la base de cette clé, vous pouvez contrôler que le composant est bien utilisé dans l'application pour laquelle la licence d'utilisation a été octroyée.

Ce système de protection est basé sur l'emploi de la commande de 4D **LIRE INFORMATIONS SERIALISATION** : un utilisateur souhaitant acquérir un composant fournit au développeur sa clé unique — générée à l'aide de la commande LIRE INFORMATIONS SERIALISATION. Cette opération peut, par exemple, être effectuée par l'intermédiaire d'un formulaire "Bon de commande" intégré à la version de démonstration du composant.

Le développeur du composant peut alors générer son propre numéro de série, en combinant la clé et l'algorithme de cryptage de son choix. Le composant livré comportera une fonction permettant de tester si les informations retournées par LIRE INFORMATIONS SERIALISATION correspondent bien à ce numéro de série. Dans le cas contraire, le composant sera rendu inutilisable.

Pour plus d'informations sur la commande LIRE INFORMATIONS SERIALISATION, reportez-vous à la description de cette commande dans le manuel *Langage* de 4D.