

IBM Cúram Social Program Management



# Handbuch Cúram Development Compliancy

*Version 6.05*



IBM Cúram Social Program Management



# Handbuch Cúram Development Compliancy

*Version 6.05*

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 31 gelesen werden.

**Überarbeitung: Mai 2013**

Diese Ausgabe bezieht sich auf IBM Cúram Social Program Management v6.0.5 und alle nachfolgenden Releases, sofern nicht anderweitig in neuen Ausgaben angegeben.

Licensed Materials - Property of IBM.

© Copyright IBM Corporation 2012, 2013.

© Cúram Software Limited. 2011. All rights reserved.

---

# Inhaltsverzeichnis

**Tabellen. . . . . v**

**Kapitel 1. Einführung . . . . . 1**

1.1 Zweck . . . . . 1  
1.2 Angesprochene Zielgruppe . . . . . 1  
1.3 Voraussetzungen . . . . . 1

**Kapitel 2. Cúram-konform entwickeln . . . 3**

2.1 Übersicht . . . . . 3  
2.2 Neues Projekt starten . . . . . 3  
    2.2.1 Erläuterung der Entwicklungsverzeichnisstruktur . . . . . 3  
    2.2.2 Quellcodekontrolle . . . . . 3  
2.3 Serverquellenartefakte ändern . . . . . 4  
    2.3.1 Quellcode für neue Methoden und Klassen schreiben . . . . . 4  
    2.3.2 Cúram Express Rules (CER)-Regelwerke ändern . . . . . 4  
    2.3.3 Codetabellen erweitern . . . . . 5  
2.4 Quellcode und APIs. . . . . 5  
    2.4.1 Interne APIs. . . . . 5  
    2.4.2 Externe APIs . . . . . 6  
    2.4.3 Erweiterungsmechanismen . . . . . 6  
    2.4.4 Zusammenfassung . . . . . 8  
2.5 Grundsätzliche Konformitätsschwierigkeiten vermeiden . . . . . 9  
    2.5.1 Projektspezifische Präfixe in Artefaktnamen verwenden . . . . . 9  
    2.5.2 Numerische Kennungen in angepassten Anfangs- und Demonstrationsdaten verwenden . . . 10  
    2.5.3 Keine In-Place-Änderungen an Anwendungsdateien vornehmen. . . . . 10  
    2.5.4 Keine Abhängigkeiten von Muster- oder Demonstrationsartefakten schaffen. . . . . 11  
    2.5.5 Änderungen an dynamischen Artefakttypen an das Entwicklungssystem zurückmelden . . . 11

2.5.6 Keine neuen Abhängigkeiten von internen APIs herstellen . . . . . 12

**Anhang A. Details zur Komponentenkonformität . . . . . 13**

A.1 Einführung . . . . . 13

**Anhang B. Nicht empfohlene Erweiterungsmechanismen . . . . . 15**

B.1 Einführung . . . . . 15  
B.2 Erweiterungsklassen . . . . . 15  
    B.2.1 Entität . . . . . 15  
    B.2.2 Struct . . . . . 16  
    B.2.3 Prozess, Fassade, Webservice, WSInbound 17  
B.3 Unterklassifizieren und Ersetzen . . . . . 19  
    B.3.1 Entität . . . . . 19  
    B.3.2 Prozess, Fassade, Webservice, WSInbound 21  
B.4 Unterklassifizieren ohne Ersetzen . . . . . 23  
    B.4.1 Entität . . . . . 23  
    B.4.2 Prozess, Fassade, Webservice, WSInbound 24  
B.5 Domänenüberschreibung . . . . . 26  
    B.5.1 Domänendefinitionen . . . . . 26  
B.6 Beziehungen. . . . . 28  
    B.6.1 Zuweisungsfähigkeit . . . . . 28  
    B.6.2 Spaltenberechnungen (Aggregation) . . . 28  
    B.6.3 Fremdschlüssel . . . . . 28  
    B.6.4 Index . . . . . 28  
    B.6.5 Eindeutiger Index . . . . . 28  
B.7 Andere Mechanismen. . . . . 29  
    B.7.1 Ausschlüsse . . . . . 29

**Bemerkungen. . . . . 31**

Marken. . . . . 33



---

## Tabellen

1. Details zur Komponentenkonformität . . . . .	13	7. 'Unterklassifizieren ohne Ersetzen' zur Anwendung für Entitätsklassen . . . . .	23
2. Erweiterungsklassen zur Anwendung für Entitätsklassen . . . . .	15	8. 'Unterklassifizieren ohne Ersetzen' zur Anwendung für andere modellierte Klassen . . . . .	24
3. Erweiterungsklassen zur Anwendung für Struct-Klassen. . . . .	16	9. Domänendefinitionen überschreiben . . . . .	26
4. Erweiterungsklassen zur Anwendung für andere modellierte Klassen . . . . .	17	10. Zuweisbare Beziehungen . . . . .	28
5. 'Unterklassifizieren und Ersetzen' zur Anwendung für Entitätsklassen . . . . .	19	11. Aggregationen . . . . .	28
6. 'Unterklassifizieren und Ersetzen' zur Anwendung für andere modellierte Klassen . . . . .	21	12. Fremdschlüssel . . . . .	28
		13. Indizes . . . . .	28
		14. Eindeutige Indizes . . . . .	28
		15. Ausschlüsse . . . . .	29





---

# Kapitel 1. Einführung

---

## 1.1 Zweck

Dieses Dokument stellt Leitlinien zum konformen Erstellen angepasster Funktionalität bereit.

Bitte beachten Sie, dass sich mit Version 6.0.3 einige dieser Leitlinien geändert haben. Während für Kunden, die sie bereits verwendet haben, alle Anwendungs-Anpassungsmechanismen weiterhin unterstützt werden, werden einige von ihnen für neue Entwicklungen nicht mehr empfohlen.

---

## 1.2 Angesprochene Zielgruppe

Dieses Dokument ist für Designer und Entwickler in Projektteams gedacht, die mit dem Erstellen von Cúram-Anwendungen zu tun haben.

---

## 1.3 Voraussetzungen

Um aus diesem Dokument einen Nutzen zu ziehen, sind praktische Erfahrungen mit der Anwendungsentwicklungsumgebung erforderlich. Dieses Handbuch enthält viele Verweise auf relevante Entwicklungsdokumentation.



---

## Kapitel 2. Cúram-konform entwickeln

---

### 2.1 Übersicht

Dieses Kapitel enthält wichtige Aspekte, die für Cúram-konforme Entwicklungen zu beachten sind. Diese Aspekte sind unbedingt zu beachten, damit der Support den Kunden dabei unterstützen kann, eigene Anpassungen für die Anwendung auszuführen. Durch Beachtung dieser Punkte wird es dem Kunden auch erleichtert, auf künftige Versionen aufzupgraden.

---

### 2.2 Neues Projekt starten

Vor dem Start eines neuen Projekts sollte man sich einen Überblick über die Entwicklungsverzeichnisstruktur verschafft haben. Zudem ist es wichtig, diese unter Quellcodekontrolle zu stellen.

#### 2.2.1 Erläuterung der Entwicklungsverzeichnisstruktur

Die Kenntnis der Entwicklungsverzeichnisstruktur ist erforderlich, um zu wissen, wo die Entwicklungsartefakte positioniert sind, wie sie organisiert sind und wo Änderungen an diesen Artefakten gespeichert werden können. Beachten Sie, dass Sie zuerst die Entwicklungsversion der Anwendung installieren müssen, um Zugriff auf die Entwicklungsverzeichnisstruktur zu erhalten.

Die folgende Liste beschreibt die Verzeichnisse, in die die Kunden- und Serverentwicklungsartefakte installiert werden:

- Kundenentwicklungsartefakte werden in das Verzeichnis `webclient` installiert. Details zur konformen Entwicklung von Kundenanwendungen finden Sie im Dokument *Cúram Web Client Reference Manual*.
- Serverentwicklungsartefakte werden in das Verzeichnis `EJBServer` installiert.

Sowohl im Verzeichnis `webclient` als auch im Verzeichnis `EJBServer` gibt es ein Unterverzeichnis `namenscomponents` mit einem weiteren Unterverzeichnis `namenscustom`. Das Unterverzeichnis `custom` ist der Ort, an dem alle projektspezifischen Entwicklungsartefakte angeordnet werden sollen. Die anderen Unterverzeichnisse, `components`, enthalten alle Anwendungsentwicklungsartefakte, die mit dem Produkt geliefert wurden.

**Wichtig:** Der Ordner `custom` enthält eine Starterstruktur für die Erstanwendung und wird in der gesamten Entwicklerdokumentation als der Bereich genannt, an dem alle Artefakte entwickelt werden sollten. Es wird aber darauf hingewiesen, dass dies nicht zwingend erforderlich ist und projektabhängig gewählt werden kann, ob innerhalb dieser Komponente entwickelt werden oder eine neue benannte Komponente geschaffen werden soll, die dem Projekt entspricht.

In dem Verzeichnis `EJBServer\components\custom\model` sind eine Starter-Modelldatei und einige Modellfragmente enthalten.

#### 2.2.2 Quellcodekontrolle

Um alle Änderungen an Quellenartefakten verfolgen zu können, sollte die Entwicklungsverzeichnisstruktur unter Quellcodekontrolle gestellt werden. Nachdem dies erfolgt ist, sollten alle Entwicklungsartefakte mit Kennungen versehen werden. Achten Sie darauf, dass die Kennung auf die Version der Anwendung hinweist. Das ermöglicht es, mithilfe einer anderen Funktionalität von jedem beliebigen Punkt aus einen Bericht über alle Dateien zu erstellen, die hinzugefügt oder geändert wurden, um die Projektfunktionalität zu implementieren. Dieser Bericht erweist sich dann als hilfreich, wenn ein neues Release der Anwendung übernommen werden soll.

Bitte beachten Sie, dass es ab Version 6.0.3 Änderungen in der Art gibt, wie der Java™-Quellcode geliefert wird. Weitere Informationen dazu finden Sie in Abschnitt 4.

---

## 2.3 Serverquellenartefakte ändern

Es gibt eine Vielzahl von Serverartefakttypen, wobei es sich bei einigen von ihnen um Anwendungsklassen handelt. Einige von ihnen sind in einem Anwendungsmodell dargestellt. Andere Java-Schnittstellen sind "handgearbeitet". Für beide Klassenkategorien gibt es Anwendungsimplementierungen, deshalb muss zwischen ihnen unterschieden werden. Während es möglich ist, Aspekte einer modellierten Schnittstelle durch Ändern des Modells oder neu generierten Code zu verändern, ist das Ändern von "handgearbeiteten" Schnittstellen nicht möglich.

Modellierte Schnittstellen:

- Treten im Anwendungs-UML-Modell auf

"Handgearbeitete" Schnittstellen:

- Treten im Anwendungs-UML-Modell nicht auf
- Treten in den Komponentenverzeichnissen Ihrer Entwicklungsumgebung auf
- Nicht anpassbar
- Enthalten die `@ImplementedBy` Google Guice-Anmerkung zum Kennzeichnen der Anwendungsimplementierungsklasse

Einige Komponenten können Schnittstellen enthalten, die in keine der oben genannten Kategorien fallen. In den komponentenspezifischen Dokumentationen werden sie jedoch immer beschrieben. Sowohl die modellierten als auch die "handgearbeiteten" Anwendungsschnittstellen können anpassbare Implementierungen enthalten.

Details zur Implementierung von Quellenartefakten finden Sie im Dokument *Cúram Server Developer's Guide* im Abschnitt über die Implementierung von modellierten Schnittstellen sowie im Dokument *Persistence Cookbook* im Abschnitt über die Implementierung von "handgearbeiteten" Schnittstellen. Die Kategorie muss anhand der implementierten Schnittstelle bestimmt werden. Die Empfehlungen zum Ändern von Serverquellenartefakten haben sich mit Version 6.0.3 geändert. Bitte beachten Sie, dass die in diesem Dokument ("Cúram Development Compliance Guide") enthaltenen Empfehlungen verbindlich sind.

### 2.3.1 Quellcode für neue Methoden und Klassen schreiben

Neue kundenspezifische Klassen, Klassen mit eingeschlossenen bestehenden Klassen sowie - unter ganz bestimmten Umständen - neue Unterklassen von bestehenden Klassen sollten in neue Quelldateien geschrieben werden. Alle neuen Quelldateien sollten im Unterverzeichnis `source` des Verzeichnisses `EJBServer\components\custom` abgelegt werden.

Bei modellierten Klassen bestimmt die erstellte Klassenhierarchie die Paketstruktur für die neuen Quelldateien. Weitere Informationen zum Modellieren neuer Klassen und Ersetzen vorhandener Implementierungen finden Sie im Dokument *Cúram Server Developer's Guide*.

Bei den "handgearbeiteten" Implementierungen ist es Ihre Entscheidung, wie die neue Klasse paketiert werden soll. Informationen zum Konfigurieren neuer Unterklassen mithilfe von Google Guice finden Sie im Dokument *Persistence Cookbook*.

### 2.3.2 Cúram Express Rules (CER)-Regelwerke ändern

Die Regelwerke des CER-Editors werden in der Datenbank gespeichert, nicht im Dateisystem. Regelwerke, die mit der Kernkomponente geliefert wurden, dürfen NICHT angepasst werden. Andere Lösungen können für ihre Regelwerke eigene Konformitätserklärungen aufweisen.

### 2.3.3 Codetabellen erweitern

Bitte beachten Sie, dass es jetzt Dokumentationen gibt, die Auskunft darüber geben, welche Codetabellen sicher erweitert werden können und welche Codetabellen es vom Kunden erfordern, vor einer Anpassung den Support anzufragen. Eine Liste von Codetabellen, die sich nicht ohne Kontaktieren des Supports erweitern lassen, wird in der Projektdokumentationsverzeichnisstruktur für jede Installation in einem Ordner namens 'RestrictedCodeTables' bereitgestellt. Wenn Sie eine in dieser Liste aufgeführte Codetabelle anpassen möchten, sollten Sie den Support anfragen.

---

## 2.4 Quellcode und APIs

Die gesamte Java-Funktionalität der Anwendung wird ab jetzt in Form von vordefinierten JAR-Dateien geliefert. Für Organisationsmodule, die mit Version 5.0 eingeführt wurden (und für die der Quellcode auch bisher nicht mitgeliefert wurde), war dies schon immer der Fall. Nun gilt es für alle Bereiche. Die Anwendungsfunktionalität wird in einer Kundeninstallation nur noch dann neu generiert oder wiederhergestellt, wenn die Verwendung von Erweiterungsmechanismen des Kunden es erfordert. Das bedeutet, dass der Erstellungsprozess des Kunden nicht mehr die gesamte Java-Quellcodebasis wiederherstellen muss, sondern nur noch projektspezifischen Quellcode oder abhängigen wiederhergestellten Java-Quellcode.

Für eine begrenzte Anzahl wichtiger Funktionsbereiche wird ab Version 6.0.3 Java-Quellcode in keiner Form mehr geliefert. Es wird zwar für die restlichen Bereiche der Anwendung weiterhin Quellcode mitgeliefert, jedoch nur als "Muster" für Dokumentationszwecke. Dieser Code ist ab Version 6.0.3 nicht mehr direkt in den Erstellungsprozess mit eingebunden. Der Muster-Quellcode für die einzelnen Komponenten in Form von JAR-Dateien wird folgendermaßen geliefert: `EJBServer\components\\sample\src.zip`. Die erstellten Versionen der jeweiligen Komponenten sind an dem folgenden Ort zu finden: `EJBServer\components\\lib\.jar`.

Ebenso sind ab Version 6.0.3 Klassenoperationen durch Anmerkungen als 'intern' oder 'extern' markiert.

Externe Operationen bilden die offizielle API der künftigen Anwendung. Sie sind für die Verwendung durch den Kunden gedacht und werden von seinem eigenen Code aus aufgerufen.

**Wichtig:** Klassen ohne Anmerkung sind standardmäßig 'intern'.

### 2.4.1 Interne APIs

Auch wenn es möglich ist, interne APIs vom Kundencode aus aufzurufen oder zu unterklassifizieren, wird ab Version 6.0.3 davon abgeraten. Solche APIs sind mit der Anmerkung `@AccessLevel (INTERNAL)` versehen.

**Wichtig:** 'Abgeraten' bedeutet in diesem Zusammenhang, dass ihre Verwendung zwar weiterhin unterstützt wird, aber dass solche APIs in künftigen Releases geändert oder entfernt werden können, nachdem dies dem Kunden mit einer Mindestankündigungsfrist von einem Jahr mitgeteilt worden ist.

**Anmerkung:** Es gibt jedoch keine solche Ankündigungsfrist für jegliche APIs, die in Version 6.0.3 als 'intern' markiert sind. Das bedeutet, dass es gegenwärtig keine Pläne gibt, eine der in 6.0.3 als 'intern' markierten APIs zu ändern. Daher sollte dem Kunden genügend Zeit zur Verfügung stehen, solche Migrationen zu planen.

Bestehende Kundenreferenzen auf APIs, die von Version 6.0.3 an als 'intern' markiert sind, erfüllen die gleiche Funktion wie zuvor, nur mit der Ausnahme, dass in Eclipse-Projekten, die solche Abhängigkeiten enthalten, Warnungen generiert werden.

Es ist anzustreben, dass sich die Projekte mit der Zeit von solchen Abhängigkeiten von internen APIs wegbewegen und keine neuen Abhängigkeiten geschaffen werden, wobei das auch kurzfristig unvermeidbar sein kann, wenn sich ein Kundenprojekt in seiner Design- oder Entwicklungsphase befindet. Die

meisten jetzigen Kunden werden nach Übernahme von Version 6.0.3 oder späteren Versionen mit Warnungsmeldungen konfrontiert sein. Es wird nicht erwartet, dass sie diese im Rahmen der Übernahme sofort beseitigen. Wie bereits angedeutet, hat dies keinen Einfluss auf den Support-Anspruch des Kunden.

Beachten Sie, dass, wie auch bei früheren Versionen der Anwendung, einige interne APIs dahingehend konfiguriert sind, dass sie in Eclipse fehlerhafte Zugriffsbeschränkungen generieren, wenn sie referenziert werden. Diese APIs sind mit der Anmerkung `@AccessLevel(RESTRICTED)` versehen. Solche Verweise werden in Kundenprojekten nicht unterstützt. Diese APIs sind immer interne APIs gewesen und wurden auch bisher nicht für den Kundengebrauch unterstützt. Es ist deutlich zu erkennen, um welche Art der API es sich handelt: zugriffsbeschränkte APIs generieren Eclipse-Fehlermeldungen, nicht empfohlene APIs erzeugen Warnmeldungen in Eclipse.

## 2.4.2 Externe APIs

Es kann von Kundenprojekten aus direkt auf externe APIs verwiesen werden. Solche APIs sind mit der Anmerkung `@AccessLevel(EXTERNAL)` versehen. Für alle externen APIs wird für pro einzelne Komponente ein Javadoc zur Verfügung gestellt, das unter 'EJBServer\components\\doc\api.zip' zu finden ist. Bitte beachten Sie, dass es für Komponenten, die keine externen APIs besitzen, kein Javadoc gibt. Vom Kundencode aus sollten nur Klassen referenziert werden, die in einem Javadoc dokumentiert sind. Der Verweis auf andere Klassen führt zu Warnungsmeldungen oder fehlerhaften Zugriffseinschränkungen; diese Klassen werden daher nicht unterstützt.

Beachten Sie außerdem, dass wie alle APIs auch diejenigen, die als 'extern' markiert sind, sich mit der Zeit entwickeln, dabei jedoch abwärtskompatibel bleiben. Wenn Sie den Eindruck haben, dass Ihre Anforderungen auch über eine Kombination in der Verwendung von externen APIs und zulässigen Erweiterungsmechanismen nicht erfüllt werden, sollten Sie diesen Fall mithilfe des Supports klären. Wenn notwendig, kann eine neue API, ein Anpassungsankerpunkt, ein Strategiemuster oder ein konfigurationsbasierter Ansatz zur Verfügung gestellt werden. Solche neuen APIs können in Form von Feature-Packs geliefert werden. Alternativ dazu kann, wenn nötig, in einigen Fällen eine vorhandene interne API in eine externe umgewandelt werden.

## 2.4.3 Erweiterungsmechanismen

Das Entfernen von Quellcode aus den Bereichen der Kernfunktionalität, wie oben erwähnt, hat zu einer Änderung des empfohlenen Ansatzes geführt, wie Erweiterungsmechanismen für Kundenprojekte zu verwenden sind. Bisher war es so, dass der Kunde, wenn er die verschiedenen Anwendungserweiterungsmechanismen (z.B. Erweiterungsklassen, Unterklassifizieren mit oder ohne Ersetzen, oder Aggregation) nutzen wollte, in der gesamten Codebasis suchen konnte, wo und wie Zielklassen innerhalb des Anwendungscodes aufgerufen wurden. Daraufhin konnte er die funktionalen Auswirkungen der Erweiterung beurteilen, die in Erwägung gezogen wurde.

Ab Version 6.0.3 steht dem Kunden der Quellcode für einige Bereiche der Kernfunktionalität nicht mehr zur Verfügung. Zudem ist eine große Anzahl an APIs als 'intern' markiert worden. Im folgenden Abschnitt sind die Änderungen in den Empfehlungen bezüglich der Erweiterungsverfahren für Kundenprojekte zusammengefasst.

Bitte beachten Sie, dass sich dieser Abschnitt ausschließlich auf Einschränkungen bezüglich der Erweiterung von Anwendungsartefakten bezieht. Alle Erweiterungsmechanismen können weiterhin für kundendefinierte Klassen verwendet werden. Alle diese Artefakte können in ihrer Art selbstverständlich 'extern' sein und von jedem beliebigen Teil der Kundenimplementierung aus aufgerufen werden.

**Wichtig:** Dieser Abschnitt bietet nur eine Zusammenfassung auf höhere Ebene. Die vollständigen Angaben dazu, welche Mechanismen von Version 6.0.3 an für die jeweiligen Klassentypen zulässig sind, werden in dem Dokument Anhang B, „Nicht empfohlene Erweiterungsmechanismen“, auf Seite 15 gegeben. Wo von einem Mechanismus abgeraten wird, gibt dieser Anhang entsprechende Empfehlungen zu alternativen Mechanismen, die der Kunde verwenden kann.

### 2.4.3.1 Entitätsklassen

Von einigen Ausnahmen abgesehen, wird ab jetzt vom direkten Gebrauch und der Bearbeitung von Anwendungsentitätsklassen durch den Kunden abgeraten. In vielen Fällen besitzen Anwendungsentitätsklassen direkte Fassadenschicht-Äquivalente, die als 'extern' markiert sind. Diese können vom Kunden verwendet werden. Das Hinzufügen von stereotypen und nicht-stereotypen Operationen zu Anwendungsentitäten ist jedoch weiterhin zulässig, wie auch das Einstellen einer Anzahl von Entitätsoptionen.

Vor Version 6.0.3 konnten Attribute mithilfe einer Erweiterung zu Anwendungsentitätsklassen hinzugefügt werden. Nachdem jedoch der Quellcode für einige Bereiche der Hauptfunktionalität entfernt wurde, hat der Kunde keine Einsicht mehr, ob Attribute, die über Erweiterungsklassen hinzugefügt wurden, externen APIs zugewiesen werden. Aus diesem Grund wird nun vom Hinzufügen von Attributen zu Anwendungsentitätsklassen abgeraten.

Hat der Kunde den Bedarf, Anwendungsanzeigen Daten hinzuzufügen, so sollte er neue kundenspezifische Entitätsklassen hinzufügen und externe Anwendungspflege-Operationen in seine eigenen Prozessklassen einschließen, so dass beide Tabellen atomisch verwaltet werden können. Danach kann die Anwendungsanzeige dahingehend geändert werden, dass sie auf die neuen Prozessklassen verweist.

**Anmerkung:** Entitäten, die Angabentypen darstellen, bilden eine Ausnahme zu dieser Regel. Der Kunde kann weiterhin mithilfe einer Erweiterung Attribute zu solchen Anwendungsangabentitäten hinzufügen, so wie es vom Evidence Generator erfordert wird.

In Version 6.0.3 sind die Anwendungsangabentitäten fälschlicherweise als 'intern' markiert worden; dies wird im nachfolgenden Release korrigiert werden. Für den Augenblick gilt, dass ein Kunde, der eine Erweiterung für Angabentitäten nutzt, um Attribute hinzuzufügen, in Eclipse auf Warnungen bezüglich dieser Klassen stoßen können, dass diese speziellen Warnungen jedoch zu ignorieren sind.

Diese Anmerkung bezieht sich nur auf Entitäten, die Angabentypen darstellen, nicht auf andere Anwendungsentitätsklassen.

### 2.4.3.2 Domänendefinitionen

Grundsätzlich ist die Verwendung und das Überschreiben von Anwendungsdomänendefinitionen durch den Kunden weiterhin zulässig. Es wird jedoch ab jetzt davon abgeraten, den Grundtyp einer Domänendefinition und einer Anzahl von codetabellenbezogenen Optionen zu ändern.

### 2.4.3.3 Struct-Klassen

Anwendungs-Structklassen sind grundsätzlich 'extern', was bedeutet, dass sie in kundenspezifischer Funktionalität referenziert werden können.

Es wird dem Kunden davon abgeraten, Aggregationen direkt aus Anwendungs-Structs zu einem anderen Struct zu erstellen, da er nicht mehr die vollständige Einsicht darin hat, wo die Anwendungs-Structs gerade verwendet werden. Er kann jedoch weiterhin Aggregationen verwenden, um Anwendungs-Structs in seine eigenen projektspezifischen Structs mit einzubeziehen.

### 2.4.3.4 Andere modellierte Klassen

Für andere modellierte Klassen der Anwendung (wie Prozess, Fassade, WSInbound und Webservice) wird jetzt von der Verwendung jeglicher Erweiterungsmechanismen abgeraten.

Bis zur Version 6.0.3 war Unterklassifizieren und Ersetzen ein allgemein gebräuchlicher Mechanismus zum Hinzufügen und Ändern von Operationen bei Anwendungsprozess- und Fassadenklassen. Wie bei der Erweiterung von Anwendungsentitätsklassen, so gilt auch hier, dass dies ab jetzt möglicherweise ein unsicherer Weg ist, da der Kunde nicht mehr notwendigerweise die ganze Einsicht darin hat, wo diese Klassen verwendet werden.

Ähnlich wie bei den Entitätsklassen sollte der Kunde stattdessen seine eigenen Prozess-, Fassaden- oder WSInbound-Klassen modellieren und codieren und sie entweder in vorhandene externe APIs einschließen oder eine neue Funktionalität implementieren. Was Fassadenoperationen betrifft, können betroffene UIM-Seiten, sofern gewünscht, auf die neuen Fassadenoperationen ausgerichtet werden.

### 2.4.3.5 Nichtmodellerte Klassen

Einige Komponenten enthalten nichtmodellerte Klassen. Für jede dieser Klassen gibt es ein Javadoc, in dem die Verwendung der externen Schnittstelle oder Klasse beschrieben wird.

In einigen nichtmodellerten Klassen sind Eclipse-Zugriffsbeschränkungen vorhanden, so dass der Kunde eine Leitlinie hat, welche APIs er aufrufen oder anpassen kann. Bestimmte Klassen und Pakete sind als eingeschränkt markiert. Diese Klassen dürfen nicht verwendet werden, da es sich bei ihnen um interne Klassen handelt, die sich mit der Zeit verändern können. Zugriffsbeschränkungen sollten nicht aus der Datei 'Eclipse.classpath' entfernt werden, weil das zur Verarbeitung von eingeschränkten Klassen führen könnte, was wiederum Probleme beim Upgraden verursachen kann.

Einige nichtmodellerte Komponenten enthalten package-geschützte Klassen, die nicht mit einem Kundencode verwendet werden sollten. Der Kunde darf in dieselbe Package-Struktur keinen Kundencode einfügen, um package-geschützte Klassen aufzurufen oder zu referenzieren.

Viele nichtmodellerte APIs lassen sich nicht direkt anpassen. Nur Schnittstellen oder Klassen, die mit der Anmerkung `@Implementable` versehen sind, können erweitert oder implementiert werden. Solche Klassen verfügen über ein Javadoc, in dem ihre Anpassung oder Implementierung detailliert beschrieben wird. Nichtmodellerte Klassen, die nicht mit der Anmerkung `@Implementable` versehen sind, dürfen nicht erweitert oder implementiert werden, da mit der Zeit neue Operationen hinzugefügt werden könnten, die sich möglicherweise auf das Upgraden auswirken.

Für Klassentypen, die mit der Anmerkung `@Implementable` versehen sind, sind Ereignisse und Strategien die typischen Anpassungsmechanismen.

Ereignisse ermöglichen es dem Kunden, an unterschiedlichen Punkten der Anwendung angepasste Logik hinzuzufügen. Details dazu, wie man Ereignislistener hinzufügt, finden Sie im Dokument 'Persistence Cookbook'. Ereignisklassen sind typischerweise mit 'xxxEvent' benannt und lassen sich auf diese Weise leicht ermitteln.

Strategiemuster ermöglichen es dem Kunden, das Standardverhalten bestimmter Funktionen innerhalb der Anwendung zu ändern. Jede Strategiekategorie ist mit einer Standardimplementierung versehen; der Kunde kann jedoch wählen, ob er mithilfe von Guice-Bindungen die Standardimplementierung einer beliebigen Strategieoperation überschreiben will. Weitere Details dazu, wie Guice-Bindungen verwendet werden, finden Sie im Dokument 'Persistence Cookbook'. Strategieklassen sind typischerweise mit 'xxx-Strategy' benannt und lassen sich auf diese Weise leicht ermitteln.

**Anmerkung:** Weitere Details zur Konformität der einzelnen Komponenten finden Sie im Dokument Anhang A, „Details zur Komponentenkonformität“, auf Seite 13.

## 2.4.4 Zusammenfassung

Hier die zusammengefassten Aussagen:

Zulässigkeit von Verweisen auf eine Anwendungsklasse im Kundencode:

- Handelt es sich um eine externe Klasse, so ist ein Verweis zulässig.
- Handelt es sich um eine interne Klasse, so wird ein Verweis darauf in Ihrem vorhandenen Code zwar unterstützt; jedoch ist dies nicht empfehlenswert. In einem neuen Code sollte kein Verweis darauf enthalten sein.
- Handelt es sich um eine zugriffseingeschränkte Klasse, so wird ein Verweis darauf nicht unterstützt.



Zulässigkeit von Anpassungen einer Anwendungsklasse:

- Handelt es sich um eine modellierte Klasse, so finden Sie in Anhang B Details dazu, welche Anpassungen zulässig sind.
- Handelt es sich um eine nichtmodellerte Klasse, so finden Sie in ihrem Javadoc bzw. dem Konfigurations- oder Entwicklungshandbuch für ihre übergeordnete Komponente die Details für die Anpassungspunkte.

---

## 2.5 Grundsätzliche Konformitätsschwierigkeiten vermeiden

In diesem Abschnitt werden mögliche Konformitätsprobleme beschrieben und Faustregeln zur Vermeidung solcher Probleme gegeben. Diese Faustregeln können unschwer bereits während der ersten Phasen eines Projektes befolgt werden. Werden sie jedoch nicht befolgt, können sie später schwere Störungen verursachen, deren Behebung schwierig und kostenintensiv werden kann.

### 2.5.1 Projektspezifische Präfixe in Artefaktnamen verwenden

Alle neuen Quellenartefaktnamen (Modellklassen, Quelldateien, Nachrichten, Nachrichtendateien usw.) sollten mit Präfixen, entsprechenden Akronymen oder Abkürzungen versehen werden. Dabei sollte man durchgängig dasselbe Akronym oder dieselbe Abkürzung verwenden. Mit dem Fortschreiten des Projekts macht dies die Projektzusätze leichter von den Kernartefakten unterscheidbar. Diese Unterscheidungsmöglichkeit wird umso nützlicher, je mehr die Entwicklung anwächst. Im Allgemeinen werden die meisten Projekte mit einem Akronym bezeichnet. Dieses Akronym wäre dann eine gute Wahl für ein Präfix.

Die Verwendung eines projektspezifischen Präfix beugt Namenskonflikten zwischen den neuen, von Ihnen hinzugefügten Artefakten und den neuen Artefakten, die Cúram im Laufe der Zeit hinzufügt, vor. Namenskonflikte können zu hohen Kosten führen und sind nur schwer zu beseitigen.

Beispiel: Sie übernehmen ein Service-Pack und stellen fest, dass einer Ihrer benutzerdefinierten Datenbankfeldzusätze denselben Namen aufweist wie ein neues Anwendungsfeld mit abweichenden Geschäftsbedeutungen oder Datentypen. Anderes Beispiel: Sie übernehmen ein Service-Pack und stellen fest, dass ein neues Anwendungs-Codetablenelement hinzugefügt worden ist, das mit einem benutzerdefinierten Element desselben Namens, das Sie hinzugefügt haben und eine andere Bedeutung besitzt, kollidiert. Solche Kollisionen lassen sich dadurch vermeiden, dass Sie darauf achten, neue benutzerdefinierte Artefakte durchgehend mit einem konsistenten Präfix zu versehen.

Einige Artefakttypen besitzen mehr als eine Kennung, was bei ihrer Benennung berücksichtigt werden muss. Beispiele hierfür sind Entitätsklassen und Codetablenelemente. Eine angepasste Entitätsklasse hat einen Tabellennamen mit demselben unveränderten Namensbereich (dem Datenbankschema) wie die Anwendungstabellen und muss innerhalb dieses Namensbereichs einen eindeutigen Tabellennamen besitzen. Sie hat auch einen Java-Klassennamen, der über denselben hierarchischen Namensbereich (Paketstruktur) verfügt wie die Anwendungs-Java-Klassen. Ebenso hat ein angepasstes Codetablenelement sowohl einen Wert als auch eine Java-Kennung, die beide einen unveränderten Namensbereich mit Anwendungselementen in derselben Codetabelle besitzen.

Bitte nehmen Sie zur Kenntnis, dass die Verwendung von projektspezifischen Präfixen beim Überschreiben eines Anwendungsartefaktes keine Anwendung findet, da der Überschreibungsmechanismus gewöhnlich darauf basiert, dass Ihr angepasstes Artefakt exakt denselben Namen besitzt wie das Anwendungsartefakt, das überschrieben werden soll.

Weitere Aspekte:

- Kennungen gibt es in vielen Versionen, beispielsweise Dateinamen, XML-IDs, Java-Klassennamen oder eine Kombination daraus.
- Es ist ratsam, ein kurzes Präfix zu benutzen, da die Namenlängen an einigen Stellen eingeschränkt sind (z.B. bei bestimmten Typen von Datenbankkennungen).

**Anmerkung:** Nicht nur Quellenartefakte, auch Kennungswerte müssen berücksichtigt werden, da diese mit den von IBM verwendeten Werten kollidieren könnten.

Die APIs `TransactionInfo.setFacadeScopeObject` und `TransactionInfo.getFacadeScopeObject` ermöglichen es dem Entwickler, auf Objekte zuzugreifen, die der laufenden Transaktion zugeordnet sind. Bei Verwendung dieser API sollten Sie eine Zeichenfolge als Ihre Objektkennung verwenden, die mit einem passenden Präfix versehen ist, um sicher sein zu können, dass keine Ihrer Daten für die Transaktion mit den IBM zugehörigen Daten kollidieren.

## 2.5.2 Numerische Kennungen in angepassten Anfangs- und Demonstrationsdaten verwenden

Vordefinierte Anfangs- und Demonstrationsdaten werden mittels DMX-Dateien in eine Anwendungsdatenbank geladen. Diese Daten werden entweder bei der Erstinstallation oder während des Upgrades eines Systems in die Datenbank installiert. Ein Satz von Anfangs- und Demonstrationsdaten wird mit der Anwendung bereitgestellt. Der Kunde kann auch eigene Anfangs- oder Demonstrationsdaten hinzufügen.

Um Überschneidungen zwischen den Anfangs- und Demonstrationsdaten, die mit der Anwendung mitgeliefert werden, und den Daten, die vom Laufzeitsystem erzeugt werden, zu vermeiden, ist es wichtig, die Kennungen (z.B. Primärschlüssel) für die Kundenanfangs- und -demonstrationsdaten aus den reservierten Bereichen zu entnehmen. Für diesen Zweck wurde eine Gruppe von Bereichen für die Verwendung durch den Kunden reserviert.

### 2.5.2.1 Reservierte Bereiche

In Projekten sollte man Kennungen (Primärschlüssel) in ihren angepassten Initialen und Demonstrationsdaten verwenden, die aus den folgenden reservierten Bereichen bezogen werden:

- Nicht menschlich lesbare Primärschlüssel: 45.000 bis 49.999 (inklusive)
- Menschlich lesbare Primärschlüssel: 11.521 bis 12.799 (inklusive)
- Regelsätze: 4.500 bis 4.999 (inklusive)

Kunden, die bereits Kennungen von außerhalb dieser Bereiche verwendet haben, erhalten die Möglichkeit, sich vor der Durchführung ihres nächsten Upgrades damit auseinanderzusetzen.

### 2.5.2.2 Große Datensätze

Zuweilen besteht die Notwendigkeit, überaus große Datensätze zu generieren. Das kann beispielsweise für einen Belastungstest erforderlich sein. In solchen Fällen kann die Anzahl der erforderlichen Datensätze weit über die hier dokumentierten zugewiesenen Schlüsselbereiche hinausgehen. In einer solchen Situation sollte man nach einem anderen Lösungsansatz suchen.

Anstatt die Schlüssel aus den zugeordneten Bereichen zu verwenden, sollte der Schlüsselservers dazu verwendet werden, die erforderlichen Schlüsselwerte zu generieren. Werden diese Daten in eine erneuerte Datenbank importiert, sollte der Endwert des Schlüsselsatzes ebenfalls extrahiert und in die Schlüsselsatz-tabelle geladen werden, wodurch der anfänglich mit der Anwendung gelieferte Schlüsselsatzwert ersetzt wird. Bei Fragen zu diesem Vorgang kontaktieren Sie den Support.

### 2.5.2.3 Ausnahmebedingung für Codetabellen

Bitte beachten Sie, dass die obige Erklärung nicht für Codetabellen gilt.

## 2.5.3 Keine In-Place-Änderungen an Anwendungsdateien vornehmen

Service-Packs und Notfall-Patches müssen in der Lage sein, Anwendungsdateien sicher zu verschieben, umzustrukturieren oder zu überschreiben. Wenn Sie diese Dateien verändern, können Service-Packs oder Notfall-Patches sie ohne Benachrichtigungsfrist überschreiben. Es gibt keine Garantie, dass solche Änderungen mit den von Ihnen vorgenommen Modifikationen kompatibel sind. Möglicherweise lassen sich Ihre In-Place-Änderungen hinterher nicht mehr anwenden.

Es gibt eine sehr geringe Anzahl an Ausnahmen von dieser Regel. Diese sind nachstehend aufgeführt:

- EJBServer
  - /project/config/datamanager\_config.xml
  - /project/config/deployment\_packaging.xml
  - /project/properties/Bootstrap.properties
  - .classpath
  - .project
- Webclient
  - /JavaSource/curam/omega3/ApplicationConfiguration.properties
  - /JavaSource/curam/omega3/il8n/CDEJResources.properties
  - .classpath
  - .project

## 2.5.4 Keine Abhängigkeiten von Muster- oder Demonstrationsartefakten schaffen

Muster- und Demoartefakte sind für kundenspezifische Abhängigkeiten, z.B. APIs, auf die verwiesen wird, UIM-Dateien, Codetabellen, Nachrichtendateien usw. in Komponenten mit Kundencode auf keinen Fall zulässig. Diese Artefakte können ohne Benachrichtigungsfrist jederzeit Änderungen unterworfen sein.

Da in den verschiedenen Cúram-Produktbereichen die Artefakte auf unterschiedliche Weise als intern, Muster oder Demos markiert sind, kann dieses Handbuch keine genauen Angaben darüber machen, wie sie zu identifizieren sind. Es gibt jedoch einige zuverlässige Faustregeln:

- Artefakte, deren Name, Codepaket, Modellpaket oder Dateipfad die Worte 'intern', 'Muster' oder 'Demo' oder offensichtliche Ableitungen davon enthalten.

In Zweifelsfällen kontaktieren Sie bitte den Support.

**Wichtig:** Der Ordner CPMSample ist intern. Alle Codes und Artefakte innerhalb dieses Ordners können jederzeit ohne vorherige Benachrichtigung geändert werden. Möchte der Kunde Funktionalitäten innerhalb von CPMSample nutzen, muss er sie in ihrer Codebasis duplizieren.

## 2.5.5 Änderungen an dynamischen Artefakttypen an das Entwicklungssystem zurückmelden

Wenn Sie dynamische Artefakttypen an Produktions- oder Testsystemen modifizieren, sollten Sie stets darauf achten, dass diese Modifikationen an das Entwicklungssystem zurückgemeldet werden.

Es sind in der Anwendung unterschiedliche 'dynamische' Entwicklungsartefakte vorhanden, die in Laufzeit an einem Produktions- oder Testsystem geändert werden können (z.B. Codetabellen, Workflows usw.). Laufzeitänderungen an diesen Artefakten sollten immer an die Entwicklungscodetabelle zurücksynchronisiert werden, so dass parallel entstehende Entwicklungsänderungen vor der Bereitstellung in diese Laufzeitänderungen integriert werden können.

Solche parallelen Änderungen an den Artefakten können während einer routinemäßigen Projektmeilenstein-Entwicklung entstehen, bei der Übernahme von Service-Packs oder beim Durchführen von Major oder Minor Upgrades. In jedem Fall muss die Entwicklungscodetabelle der zentrale Ort sein, an dem die parallel entstandenen Änderungen zusammengefügt und geprüft werden. Das Kerndatensystem für diese Artefakte ist die Entwicklungscodetabelle.

## **2.5.6 Keine neuen Abhängigkeiten von internen APIs herstellen**

Ab Version 6.0.3 sollte es der Kunde vermeiden, als 'intern' markierte Anwendungsklassen und -operationen aufzurufen oder anzupassen, da sich solche APIs in künftigen Versionen der Anwendung ändern können.

---

# Anhang A. Details zur Komponentenkonformität

---

## A.1 Einführung

Dieser Abschnitt enthält Informationen zur Konformität der einzelnen Komponenten.

**Wichtig:** Soweit nicht anders angegeben, kann für alle, auch für die hier nicht aufgeführten Komponenten, davon ausgegangen werden, dass die folgenden allgemeinen Konformitätserklärungen Anwendung finden:

Zulässigkeit von Verweisen auf eine Anwendungsklasse mit Kundencode:

- Handelt es sich um eine externe Klasse, so ist ein Verweis zulässig.
- Handelt es sich um eine interne Klasse, so wird ein Verweis darauf in Ihrem vorhandenen Code zwar unterstützt; jedoch ist dies nicht empfehlenswert. In einem neuen Code sollte kein Verweis darauf enthalten sein.
- Handelt es sich um eine zugriffseingeschränkte Klasse, so wird ein Verweis darauf nicht unterstützt.

Zulässigkeit von Anpassungen einer Anwendungsklasse mit Kundencode:

- Handelt es sich um eine modellierte Klasse, so finden Sie in Anhang B Informationen dazu, was zulässig ist.
- Handelt es sich um eine nichtmodellerte Klasse, so finden Sie in ihrem Javadoc (EJBServer\components\\doc\api.zip) Informationen dazu, was zulässig ist.

*Tabelle 1. Details zur Komponentenkonformität.*

In dieser Tabelle sind die Komponenten mit nichtmodellierten APIs aufgeführt.

Komponente	Details
Cúram Client Development Environment	Weitere Hinweise zur Anpassung und Verwendung dieser Komponente finden Sie im Handbuch <i>Cúram Web Client-Reference</i> . Bitte beachten Sie, dass Dateien aus dem CuramCDEJ-Ordner während des Antragerstellungsprozesses in temporäre Erstellungsordner kopiert werden. Das bloße Vorhandensein solcher Dateien außerhalb des CuramCDEJ-Ordners macht sie nicht für Anpassungen verfügbar.
Cúram Server Development Environment	<p>Im Javadoc dieser Komponente sind alle Anpassungspunkte und externen APIs im Detail aufgeführt. Vom Kundencode aus sollten nur Klassen referenziert werden, die in einem Javadoc dokumentiert sind. Der Verweis auf andere Klassen führt zu Warnungsmeldungen oder fehlerhaften Zugriffseinschränkungen; diese Klassen werden daher nicht unterstützt(Die Verschlüsselungsfunktionalität von Cúram wird für die Kundenverwendung über die dokumentierte Verwendung ih Handbuch <i>Cúram Server Developer's Guide</i> und <i>Cúram Security Handbook</i> hinaus nicht unterstützt ).</p> <p>Das bin-Verzeichnis dieser Komponente enthält Apache Ant-Erstellungsscripts, die nicht direkt modifiziert werden dürfen. Aktualisierungen zu diesen Scripts können durch Erstellen neuer angepasster Ant-Scripts und unter Verwendung der Ant-Übernahmefunktion erfolgen.</p> <p>Der Treiberordner zu dieser Komponente enthält Datenbanktreiber, die den Zugriff auf die Anwendungsdatenbank ermöglichen. Wenn notwendig, können diese Treiber durch geeignete Treiber für die jeweilige Datenbank ersetzt werden, vorausgesetzt, es handelt sich bei der Datenbank um eine unterstützte Datenbankversion wie im Handbuch <i>Cúram Supported Prerequisites</i> mit den unterstützten Voraussetzungen aufgeführt.</p> <p><b>Anmerkung:</b> Tritt ein Problem mit einem Treiber auf, der nicht im Lieferumfang des Produkts enthalten war (d.h. der nicht für den Gebrauch mit dieser Anwendung getestet und verifiziert wurde), kann der Kunde aufgefordert werden, den Treiber durch eine bereits getestete Version zu ersetzen und das Problem mit dem Anbieter dieses Produkts zu erörtern. Bitte beachten Sie, dass Dateien aus dem CuramSDEJ- Ordner während des Antragerstellungsprozesses in temporäre Erstellungsordner kopiert werden. Das bloße Vorhandensein solcher Dateien außerhalb des CuramSDEJ- Ordners macht sie nicht für Anpassungen verfügbar.</p>
Cúram Administration Suite	Bitte beachten Sie, dass ab Version 6.0.3 die Konformitätserklärung für Klassen in der Cúram Administration Suite nicht von denen für andere Komponenten abweicht. Externe APIs in der Administration Suite können mit dem Kundencode eingeschlossen und aufgerufen werden.

Tabelle 1. Details zur Komponentenkonformität (Forts.).

In dieser Tabelle sind die Komponenten mit nichtmodellierten APIs aufgeführt.

Komponente	Details
Persistenzinfrastruktur	Die Persistenzinfrastruktur ist nicht anpassbar. Der Kunde darf keine Kundencodes in Programmisolutions-Codepaketen anordnen (curam.util.persistence sowie alle Unterpakete). Weitere Informationen zur Verwendung dieser APIs finden Sie im <i>Persistence Cookbook</i> .
CER-Infrastruktur	Die Konformitätserklärung für die CER-Infrastruktur finden Sie im Dokument <i>Cúram Express Rules Reference Manual</i> . CER-Entitäten (d.h. jede Entität mit dem Präfix 'Creole') müssen als intern angesehen werden und können Änderungen unterliegen. Sie sollten vom Kunden ausschließlich über CER-APIs oder DMX-Dateien aktualisiert oder abgefragt werden.
Abhängigkeitsmanager	Der Abhängigkeitsmanager umfasst alle Server-Artefakte des curam.dependency-Codepakets sowie alle dessen Unterpakete.  Das Codepaket des Abhängigkeitsmanagers besteht aus den folgenden Komponenten: <ul style="list-style-type: none"> <li>• der CER-Infrastruktur sowie</li> <li>• der Basisanwendung.</li> </ul> Der Abhängigkeitsmanager kann in keiner Weise angepasst werden. Alle Abhängigkeitsmanager-APIs sind ausschließlich für den internen Gebrauch bestimmt. Die Konformitätserklärung für den Abhängigkeitsmanager finden Sie im Dokument <i>Cúram Express Rules Reference Manual</i> .
Anspruchsberechtigung und Leistungshöhenengine-API	Anleitungen zur Konfiguration und Anpassung dieser Komponente finden Sie im Dokument <i>Inside Cúram Eligibility and Entitlement Using Cúram Express Rules Guide</i> (Handbuch zu den Grundlagen der Anspruchsberechtigung und Leistungshöhe unter Verwendung von Cúram Express Rules).
Evidence Generator	Der Evidence Generator gehört zur Anwendungsinfrastruktur, die als Teil der Toolverzeichnisstruktur (EGTools) mitgeliefert wird. Weitere Informationen zur Verwendung des Evidence Generator finden Sie im Dokument <i>Cúram Evidence Generator Specification</i> .
DocMaker- und Onlinehilfe-Tools	Kein Teil des DocMaker- oder des Onlinehilfe-Tools darf angepasst werden, es sei denn, es wird im Dokument <i>Cúram Online Help System Development Guide</i> so vorgegeben.
Pod-Infrastruktur	Die Pod-Infrastruktur wird in den Dateien 'widget-inf.jar' und 'widget-utility.jar' mitgeliefert. Sie ist nicht anpassbar. Pod-Ladeprogramme sind ebenfalls nicht anpassbar. Weitere Informationen zur Entwicklung von Pods finden Sie im Dokument <i>Cúram Pod Developer's Guide</i> .
Fondsprogrammverwaltung	Anleitungen zur Anpassung dieser Komponente finden Sie im Dokument <i>Funded Program Management Developer Guide</i> sowie dem Javadoc der Komponente.
Cúram-Vorfälle (Incidents)	Anweisungen zur Anpassung von Incident-Entitäten oder zum Ersetzen von Incident-Implementierungen finden Sie im Dokument <i>Persistence Cookbook</i> sowie dem Javadoc der Komponente.
Cúram Citizen Context Viewer	Weitere Anweisungen zur Anpassung dieser Komponente finden Sie im Handbuch <i>Cúram Citizen Context Viewer - Konfiguration</i> sowie dem Javadoc der Komponente.
Cúram-Berater	Folgende Komponenten werden mit Cúram-Berater geliefert: Berater.
Cúram Common Intake	Folgende Serverkomponenten werden mit Cúram Common Intake geliefert: Intake, PCR, CREOLEProgramRRecommendation, ReferralsLite und CPMReferralsLite.
Posteingang	Anleitungen zur Konfigurierung und Anpassung dieser Komponente finden Sie in Teil IV des Dokuments <i>Cúram Workflow Reference Guide</i> .
Cúram-Warteliste	Anleitungen zur Anpassung dieser Komponente finden Sie im Dokument <i>Cúram Waitlist Customization Guide</i> und dem Javadoc der Komponente.
IBM Cúram Business Intelligence and Analytics	Anleitungen zur Anpassung dieser Komponente finden Sie im Dokument <i>Cúram Business Intelligence Reporting Developer Guide</i> .
IBM Cúram Social Enterprise Collaboration	Die folgenden Serverkomponenten werden mit Social Enterprise Collaboration geliefert: SocialEnterpriseCollaboration, CaseParticipantIndex und ClientAccess.
IBM Cúram Universal Access	Weitere Anleitungen zur Anpassung dieser Komponente finden Sie im Dokument <i>Cúram Universal Access Developers Guide</i> und dem Javadoc der Komponente.
IBM Cúram Outcome Management	Folgende Serverkomponenten werden mit Cúram Outcome Management geliefert: AssessmentPlanning, AssessmentPlanningCPM, DecisionAssistAssessments und SimpleOutcomeManagement.
IBM Cúram Provider Management	Anleitungen zur Anpassung dieser Komponente finden Sie im Dokument <i>Cúram Provider Management Developer Guide</i> und dem Javadoc der Komponente.
IBM Cúram Youth Services(CYS)	Anleitungen zur Anpassung von CYS-Entitäten oder zum Ersetzen von CYS-Implementierungen finden Sie im Dokument <i>Persistence Cookbook</i> und dem Javadoc der Komponente.
IBM Cúram Child Care (CCC)	Anleitungen zur Anpassung von CCC-Entitäten oder zum Ersetzen von CCC-Implementierungen finden Sie im Dokument <i>Persistence Cookbook</i> und dem Javadoc der Komponente.

---

# Anhang B. Nicht empfohlene Erweiterungsmechanismen

---

## B.1 Einführung

Wie bereits beschrieben, wird jetzt von vielen Mechanismen abgeraten, die bisher (in den Versionen vor 6.0.3) als Mittel zur Erweiterung oder zum Ersetzen von Anwendungsklassen empfohlen wurden. Dieser Anhang enthält detaillierte Angaben dazu, welche Mechanismen für welche Klassentypen zulässig bzw. nicht mehr zu empfehlen sind. Außerdem enthält er Vorschläge für zu ergreifende Maßnahmen, wenn Sie feststellen, dass für einen Mechanismus oder eine Klassentypenkombination, die Sie einsetzen wollen, eine Warnung erscheint.

---

## B.2 Erweiterungsklassen

### B.2.1 Entität

Tabelle 2. Erweiterungsklassen zur Anwendung für Entitätsklassen

Aktion	Modelloption	Nicht empfohlen?	Alternative
Stereotype Entitätsoperation hinzufügen  (e.g. <<ns>>, <<nsreadmulti>>)	Keine	Nicht empfohlen	Statt eine <<extension>>-Klasse zu verwenden, fügen Sie die stereotype Operation durch Verwenden von 'Unterklassifizieren ohne Ersetzen' hinzu.
Eine Entitätsoperation ändern (z.B. Parameter)	Keine	Nicht empfohlen	Eine neue stereotype Operation mit der gewünschten Struktur mithilfe von 'Unterklassifizieren ohne Ersetzen' erstellen.  Wenn Sie es für begründet halten, die Struktur einer Anwendungsentitätsoperation zu ändern, wenden Sie sich bitte an den Support.
Entitätsoperationsoption ändern	Automatisches ID-Feld  Automatischer ID-Schlüssel  Kein generiertes SQL  Optimistisches Sperren  Anordnen nach SQL  Wobei gilt:	Nicht empfohlen	Eine neue stereotype Operation mithilfe von 'Unterklassifizieren ohne Ersetzen' erstellen.  Wenn Sie es für begründet halten, diese Optionen für Anwendungsentitätsoperation zu ändern, wenden Sie sich bitte an den Support.
	Prüfung auf Datenbanktabellen-Ebene	Nicht empfohlen	Diese Option ist über Laufzeiteigenschaften konfigurierbar. Weitere Informationen zur Vorgehensweise finden Sie in Abschnitt 12.3.6 des Dokuments "Cúram Server Modeling Guide".

Tabelle 2. Erweiterungsklassen zur Anwendung für Entitätsklassen (Forts.)

Aktion	Modelloption	Nicht empfohlen?	Alternative
	Vorgang bei Fehler Vorgang nach Datenzugriff Vorgang vor Datenzugriff 'Readmulti Max' wie 'Informational' behandeln Ausnahmebedingung 'Readmulti Max'-Datensätze zurückgegeben	Nicht empfohlen	Der Kunde sollte diese Optionen für Anwendungsentitätsoperationen nur durch 'Unterklassifizieren und Ersetzen' ändern.
Entitätsklassenoption ändern	Validierung aktivieren	Nicht empfohlen	Der Kunde sollte diese Option für Anwendungsentitätsoperationen nur durch 'Unterklassifizieren und Ersetzen' ändern.
	Abstrakt Optimistisches Sperren zulassen Kein generiertes SQL Oberklasse ersetzen	Nicht empfohlen	Wenn Sie es für begründet halten, diese Optionen für Anwendungsentitätsoperation zu ändern, wenden Sie sich bitte an den Support.
	Überprüfungsfelder Zuletzt aktualisiertes Feld	Zulässig	Derzeit nur über Erweiterungsklassen unterstützt, auch ab Version 6.0.3
Entitätsattribut hinzufügen	Keine	Nicht empfohlen	Hat ein Kunde den Wunsch, zu Anwendungsanzeigen Daten hinzuzufügen, so sollte er neue kundenspezifische Entitätsklassen hinzufügen und Cúram CRUD-Operationen in die eigenen Prozessklassen einschließen, so dass beide Tabellen atomisch verwaltet werden können. Daraufhin können Cúram-Anzeigen dahingehend geändert werden, dass sie auf die neuen Prozessklassen verweisen.
Entitätsattributoption ändern	Nullen zulassen	Nicht empfohlen	Wenn Sie es für begründet halten, diese Option für Anwendungsentitätsattribute zu ändern, wenden Sie sich bitte an den Support.

## B.2.2 Struct

Tabelle 3. Erweiterungsklassen zur Anwendung für Struct-Klassen

Aktion	Modelloption	Nicht empfohlen?	Alternative
Attribut einem Struct hinzufügen	Keine	Nicht empfohlen	Erstellen Sie ein projektspezifisches Struct und aggregieren Sie das Anwendungs-Struct aus dem projektspezifischen Struct zu dem Anwendungs-Struct (nicht umgekehrt).  Verwenden Sie das neue 'Verbund'-Struct in der erforderlichen kundenspezifischen Funktionalität.
Struct-Attribut ändern	Keine	Nicht empfohlen	Erstellen Sie ein projektspezifisches Struct und aggregieren Sie das Anwendungs-Struct aus dem projektspezifischen Struct zu dem Anwendungs-Struct (nicht umgekehrt).  Verwenden Sie das neue 'Verbund'-Struct in der erforderlichen kundenspezifischen Funktionalität.  Wenn Sie den Eindruck haben, es sei begründet, ein Attribut eines Anwendungs-Structs zu ändern, wenden Sie sich bitte an den Support.



Tabelle 3. Erweiterungsklassen zur Anwendung für Struct-Klassen (Forts.)

Aktion	Modelloption	Nicht empfohlen?	Alternative
Struct-Option ändern	Überprüfungsfelder	Nicht empfohlen	Um Überprüfungsfelder über eine Anzeige aus einer Entität zu replizieren, müssen dazu neue stereotype Operationen erstellt werden, die die Überprüfungsfelder verwalten. Außerdem muss eine neue Fassade erstellt werden, die die bestehenden Entitäts-CRUD-Operationen einschließt und die neuen stereotypen Operationen aufruft. Bei Bedarf müssen die Benutzerschnittstellenseiten aktualisiert werden.

## B.2.3 Prozess, Fassade, Webservice, WSInbound

Tabelle 4. Erweiterungsklassen zur Anwendung für andere modellierte Klassen

Aktion	Modelloption	Nicht empfohlen?	Alternative
Klassenoption ändern	Abstrakt FIDs erstellen Oberklasse ersetzen Web-Services-Bindungsstil WS ist XML-Dokument Dokumenttyp Fassaden-Bean erstellen Anbietername Anforderungshandler Antworthandler Anforderung validieren XML-Dokument XML-Schema	Nicht empfohlen	Wenn Sie es für begründet halten, diese Optionen für eine Anwendungsprozess, -fassade, -WebService oder für -WSInbound-Klassen zu ändern, wenden Sie sich bitte an den Support.
Operation hinzufügen	Keine	Nicht empfohlen	Dies wurde über Erweiterungsklassen auch bisher nicht empfohlen, weil es vom Kunden erfordert hätte, eine In-Place-Modifikation eines Java-Anwendungscode durchzuführen.  Wenn Sie einer Prozess-, Fassaden-, Webservice- oder WSInboundklasse eine Operation hinzufügen wollen, schließen Sie die Klasse und die Operation in Ihre eigene projektspezifische Klasse und Operation mit ein.
Operation ändern (z.B. Anzeige der Operation)	Keine	Nicht empfohlen	Erstellen Sie eine neue Operation in einer projektspezifischen Klasse und schließen Sie dabei externe APIs der Anwendungsklassenfunktionalität ein, sofern geeignet.  Sind keine geeigneten Erweiterungspunkte vorhanden, aber Sie halten es für begründet, die Funktion oder Struktur einer Anwendungsoperation zu ändern, wenden Sie sich bitte an den Support.
Operationsoption ändern	Business-Intelligence-Prüfaufrufe	Nicht empfohlen	Diese Option ist über Laufzeiteigenschaften konfigurierbar. Weitere Informationen zur Vorgehensweise finden Sie in Abschnitt 12.3.6 des Dokuments "Cúram Server Modeling Guide".

Tabelle 4. Erweiterungsklassen zur Anwendung für andere modellierte Klassen (Forts.)

Aktion	Modelloption	Nicht empfohlen?	Alternative
	<p>Geschäfts-Datumsfeld</p> <p>Byte-Zeichensatz für Nachrichten-codierung</p> <p>Sicherheitsfunktion erstellen</p> <p>Ist XA transaktionsorientiert</p> <p>Nachrichtentyp</p> <p>Warteschlange für Connector-Factory JNDI-Name</p> <p>Antwortwarteschlange für JNDI-Name</p> <p>Zeitlimitüberschreitung für Antwortnachricht</p> <p>Schattentyp</p> <p>Transaktionsorientiert</p> <p>Übertragungswarteschlange für JNDI-Name</p>	Nicht empfohlen	<p>Wenn Sie es für begründet halten, eine dieser Optionen für einen Anwendungsprozess, eine -fassade, einen -WebService oder für -WSInbound-Klassen zu ändern, wenden Sie sich bitte an den Support.</p>
	Sichere Felder	Nicht empfohlen	<p>Möchte der Kunde die Einstellung ändern, welche Felder einer Anwendungsoperation als sicher behandelt werden sollen, muss er die Operation in die eigene Fassadenklasse und Operation einschließen und die Option 'Sichere Felder' für diese neue Operation auf die gewünschte Einstellung setzen.</p> <p>Betroffene Benutzerschnittstellen-Anzeigendefinitionen müssen auf die neue Operation verweisen, falls erforderlich.</p>
Operationsparameter-Option ändern	Pflichtfelder	Nicht empfohlen	<p>Möchte der Kunde die Einstellung ändern, welche Felder einer Anwendungsoperation als Pflichtfelder behandelt werden sollen, muss er die Operation in die eigene Fassadenklasse und Operation einschließen und die Option 'Pflichtfelder' für diese neue Operation auf die gewünschte Einstellung setzen.</p> <p>Betroffene Benutzerschnittstellen-Anzeigendefinitionen müssen auf die neue Operation verweisen, falls erforderlich.</p>

## B.3 Unterklassifizieren und Ersetzen

### B.3.1 Entität

Tabelle 5. 'Unterklassifizieren und Ersetzen' zur Anwendung für Entitätsklassen

Aktion	Modelloption	Nicht empfohlen?	Alternative
Stereotype Entitätsoperation hinzufügen (z.B. <<ns>>, <<nsreadmulti>> usw.)	Keine	Nicht empfohlen	Statt 'Unterklassifizierung und Ersetzen' zu verwenden, fügen Sie die stereotype Operation durch Verwenden von 'Unterklassifizieren ohne Ersetzen' hinzu. Dadurch können Sie sicher sein, dass Ihre Unterklasse (und damit Ihre neuen stereotypen Operationen) als 'extern' behandelt werden und Sie in Eclipse keine Warnungen erhalten, sobald Sie diese referenzieren.  Beachten Sie, dass Sie weiterhin Warnungen erhalten, wenn Sie stereotype Operationen in der Basisentität direkt referenzieren, da diese 'intern' sind. Dies ist bewusst so konzipiert.
Nichtstereotype Entitätsoperation hinzufügen oder ändern	Keine	Nicht empfohlen	Statt 'Unterklassifizieren und Ersetzen' zu verwenden, fügen Sie eine nichtstereotype Operation durch Verwenden von 'Unterklassifizieren ohne Ersetzen' hinzu. Dies sorgt dafür, dass Ihre Unterklasse (und damit Ihre neuen nichtstereotypen Operationen) als 'extern' behandelt werden und Sie in Eclipse keine Warnungen erhalten, sobald Sie diese referenzieren.  Beachten Sie, dass Sie weiterhin Warnungen erhalten, wenn Sie Operationen in der Basisentität direkt referenzieren, da diese 'intern' sind. Dies ist bewusst so konzipiert.  Es wird dem Kunden davon abgeraten, neue Implementierungen für nichtstereotype Anwendungsentitätsoperationen bereitzustellen.
Struktur einer Entitätsoperation ändern	Keine	Nicht empfohlen	Eine neue stereotype Operation mithilfe von 'Unterklassifizieren ohne Ersetzen' erstellen.  Wenn Sie es für begründet halten, die Struktur einer Anwendungsentitätsoperation zu ändern, wenden Sie sich bitte an den Support.
Entitätsoperationsoption ändern	Automatisches ID-Feld  Automatischer ID-Schlüssel  Kein generiertes SQL  Optimistisches Sperren  Anordnen nach SQL  Wobei gilt:	Nicht empfohlen	Eine neue stereotype Operation mithilfe von 'Unterklassifizieren ohne Ersetzen' erstellen.  Wenn Sie es für begründet halten, diese Optionen für Anwendungsentitätsoperation zu ändern, wenden Sie sich bitte an den Support.
	Überprüfung auf Datenbanktabellen-Ebene	Nicht empfohlen	Diese Option ist über Laufzeiteigenschaften konfigurierbar. Weitere Informationen zur Vorgehensweise finden Sie in Abschnitt 12.3.6 des Dokuments "Cúram Server Modeling Guide".
	Vorgang bei Fehler  Vorgang nach Datenzugriff  Vorgang vor Datenzugriff	Zulässig (teilweise)	Es bleibt für den Kunden zulässig, Ausstiegspunkte für die Anwendungsentität zu implementieren.  Will der Kunde eine Verarbeitung in Ausstiegspunkten durchführen, für die es eine Standardimplementierung gibt, muss die Standardimplementierung zu Anfang der Ausstiegspunktimplementierung des Kunden aufgerufen werden (d.h. es muss ein Aufruf für 'super()' am Anfang stehen).  Es ist für den Kunden nicht zulässig, Anwendungsausstiegspunktimplementierungen auszuschalten.

Tabelle 5. 'Unterklassifizieren und Ersetzen' zur Anwendung für Entitätsklassen (Forts.)

Aktion	Modelloption	Nicht empfohlen?	Alternative
	'Readmulti Max' wie 'Informational' behandeln  Ausnahmebedingung  'Readmulti Max'-Datensätze zurückgeben	Zulässig	
Entitätsklassenoption ändern	Validierung aktivieren	Zulässig (teilweise)	Es bleibt für den Kunden zulässig, Ausstiegspunkte für die Anwendungsentität zu implementieren.  Will der Kunde eine Verarbeitung in Ausstiegspunkten durchführen, für die es eine Standardimplementierung gibt, muss die Standardimplementierung zu Anfang der Ausstiegspunktimplementierung des Kunden aufgerufen werden (d.h. es muss ein Aufruf für 'super()' am Anfang stehen).
	Abstrakt  Optimistisches Sperren zulassen  Kein generiertes SQL	Nicht empfohlen	Wenn Sie es für begründet halten, diese Optionen für Anwendungsentitätsoperation zu ändern, wenden Sie sich bitte an den Support.
	Überprüfungsfelder  Zuletzt aktualisiertes Feld	Nicht empfohlen	Verwenden Sie Erweiterungsklassen, um diese Optionen für eine Anwendungsentitätsklasse zu überschreiben.
	Oberklasse ersetzen	Zulässig (teilweise)	Es ist vorbehaltlos zulässig, die anderen 'zulässigen' Aktionen, die in diesem Abschnitt beschrieben sind, zu unterstützen.

## B.3.2 Prozess, Fassade, Webservice, WSInbound

Tabelle 6. 'Unterklassifizieren und Ersetzen' zur Anwendung für andere modellierte Klassen

Aktion	Modelloption	Nicht empfohlen?	Alternative
Klassensoption ändern	Abstrakt FIDs erstellen Oberklasse ersetzen Web-Services-Bindungsstil WS ist XML-Dokument Dokumenttyp Fassaden-Bean erstellen Anbietername Anforderungshandler Antworthandler Anforderung validieren XML-Dokument XML-Schema	Nicht empfohlen	<p>Erstellen Sie eine neue Operation in einer projektspezifischen Klasse, schließen Sie dabei externe APIs der Anwendungsklassenfunktionalität mit ein, sofern geeignet, und stellen Sie die entsprechenden Optionen für die neue Klasse ein.</p> <p>Wenn Sie es für begründet halten, diese Optionen für einen Anwendungsprozess, -fassade, -WebService oder für -WSInbound-Klassen zu ändern, wenden Sie sich bitte an den Support.</p>
Operation hinzufügen	Keine	Nicht empfohlen	Erstellen Sie eine neue Operation in einer projektspezifischen Klasse und schließen Sie dabei externe APIs der Anwendungsklassenfunktionalität mit ein, sofern geeignet.
Operation ändern	Keine	Nicht empfohlen	<p>Erstellen Sie eine neue Operation in einer projektspezifischen Klasse und schließen Sie dabei externe APIs der Anwendungsklassenfunktionalität mit ein, sofern geeignet.</p> <p>Wenn Sie es für begründet halten, die Struktur von Operationen für einen Anwendungsprozess, -fassade, -WebService oder für -WSInbound-Klassen zu ändern, wenden Sie sich bitte an den Support.</p>
Operationsoption ändern	Business-Intelligence-Prüfaufufe	Nicht empfohlen	Diese Option ist über Laufzeiteigenschaften konfigurierbar. Weitere Informationen zur Vorgehensweise finden Sie in Abschnitt 12.3.6 des Dokuments "Cúram Server Modeling Guide".

Tabelle 6. 'Unterklassifizieren und Ersetzen' zur Anwendung für andere modellierte Klassen (Forts.)

Aktion	Modelloption	Nicht empfohlen?	Alternative
	<p>Geschäfts-Datumsfeld</p> <p>Byte-Zeichensatz für Nachrichtencodierung</p> <p>Sicherheitsfunktion erstellen</p> <p>Ist XA transaktionsorientiert</p> <p>Nachrichtentyp</p> <p>Warteschlange für Connector-Factory JNDI-Name</p> <p>Antwortwarteschlange für JNDI-Name</p> <p>Zeitlimitüberschreitung für Antwortnachricht</p> <p>Schattentyp</p> <p>Transaktionsorientiert</p> <p>Übertragungswarteschlange für JNDI-Name</p>	Nicht empfohlen	<p>Wenn Sie es für begründet halten, eine dieser Optionen für einen Anwendungsprozess, eine -fassade, einen -WebService oder für -WSInbound-Klassen zu ändern, wenden Sie sich bitte an den Support.</p>
	Sichere Felder	Nicht empfohlen	<p>Möchte der Kunde die Einstellung ändern, welche Felder einer Anwendungsoperation als sicher behandelt werden sollen, muss er die Operation in die eigene Operation einschließen und die Option 'Sichere Felder' für die neue Operation auf die gewünschte Einstellung setzen.</p> <p>Betroffene Benutzerschnittstellen-Anzeigendefinitionen müssen auf die neue Operation verweisen, falls erforderlich.</p>
Operationsparameter-Option ändern	Pflichtfelder	Nicht empfohlen	<p>Möchte der Kunde die Einstellung ändern, welche Felder einer Anwendungsoperation als Pflichtfelder behandelt werden sollen, muss er die Operation in die eigene Operation einschließen und die Option 'Pflichtfelder' für diese neue Operation auf die gewünschte Einstellung setzen.</p> <p>Betroffene Benutzerschnittstellen-Anzeigendefinitionen müssen auf die neue Operation verweisen, falls erforderlich.</p>

## B.4 Unterklassifizieren ohne Ersetzen

### B.4.1 Entität

Tabelle 7. 'Unterklassifizieren ohne Ersetzen' zur Anwendung für Entitätsklassen

Aktion	Modelloption	Nicht empfohlen?	Alternative
Stereotype Entitätsoperation hinzufügen (z.B. <<ns>>, <<nsreadmulti>> usw.)	Keine	Zulässig	Statt 'Unterklassifizieren und Ersetzen' zu verwenden, fügen Sie die stereotype Operation durch Verwenden von 'Unterklassifizieren ohne Ersetzen' hinzu. Dadurch können Sie sicher sein, dass Ihre Unterklasse (und damit Ihre neuen stereotypen Operationen) als 'extern' behandelt werden und Sie in Eclipse keine Warnungen erhalten, sobald Sie diese referenzieren.  Beachten Sie, dass Sie weiterhin Warnungen erhalten, wenn Sie Operationen in der Basisentität direkt referenzieren, da diese 'intern' sind. Dies ist bewusst so konzipiert.
Nichtstereotype Entitätsoperation hinzufügen	Keine	Zulässig	Statt 'Unterklassifizieren und Ersetzen' zu verwenden, fügen Sie die nichtstereotype Operation durch Verwenden von 'Unterklassifizieren ohne Ersetzen' hinzu. Dies sorgt dafür, dass Ihre Unterklasse (und damit Ihre neuen nichtstereotypen Operationen) als 'extern' behandelt werden und Sie in Eclipse keine Warnungen erhalten, sobald Sie diese referenzieren.  Beachten Sie, dass Sie weiterhin Warnungen erhalten, wenn Sie Operationen in der Basisentität direkt referenzieren, da diese 'intern' sind. Dies ist bewusst so konzipiert.
Struktur einer Entitätsoperation ändern	Keine	Nicht empfohlen	Eine neue stereotype Operation mithilfe von 'Unterklassifizieren ohne Ersetzen' erstellen.
Entitätsoperationsoption ändern	Automatisches ID-Feld Automatischer ID-Schlüssel Kein generiertes SQL Optimistisches Sperren Anordnen nach SQL Wobei gilt:	Nicht empfohlen	Eine neue stereotype Operation mithilfe von 'Unterklassifizieren ohne Ersetzen' erstellen.
	Überprüfung auf Datenbanktabellen-Ebene	Nicht empfohlen	Diese Option ist über Laufzeiteigenschaften konfigurierbar, wenn Sie das Verhalten von Anwendungsoptionen ändern möchten.  Andernfalls erstellen Sie mithilfe von 'Unterklassifizieren ohne Ersetzen' eine neue stereotype Operation, um die erforderliche Funktionalität zu implementieren.
	Vorgang bei Fehler Vorgang nach Datenzugriff Vorgang vor Datenzugriff 'Readmulti Max' wie 'Informational' behandeln 'Readmulti Max'-Datensätze zurückgegeben	Nicht empfohlen	Verwenden Sie 'Unterklassifizieren und Ersetzen', um diese Optionen für eine Anwendungsentitätsklasse zu überschreiben.  Andernfalls erstellen Sie mithilfe von 'Unterklassifizieren ohne Ersetzen' eine neue stereotype Operation, um die erforderliche Funktionalität zu implementieren.

Tabelle 7. 'Unterklassifizieren ohne Ersetzen' zur Anwendung für Entitätsklassen (Forts.)

Aktion	Modelloption	Nicht empfohlen?	Alternative
Entitätsklassenoption ändern	Validierung aktivieren	Nicht empfohlen	Verwenden Sie 'Unterklassifizieren und Ersetzen', um diese Optionen für eine Anwendungsentitätsklasse zu überschreiben.  Andernfalls erstellen Sie mithilfe von 'Unterklassifizieren ohne Ersetzen' eine neue stereotype Operation, um die erforderliche Funktionalität zu implementieren.
	Abstrakt  Optimistisches Sperren zulassen  Kein generiertes SQL	Nicht empfohlen	Eine neue stereotype Operation mithilfe von 'Unterklassifizieren ohne Ersetzen' erstellen.
	Überprüfungsfelder  Zuletzt aktualisiertes Feld	Nicht empfohlen	Verwenden Sie Erweiterungsklassen, um diese Optionen für eine Anwendungsentitätsklasse zu überschreiben.

## B.4.2 Prozess, Fassade, Webservice, WSInbound

Tabelle 8. 'Unterklassifizieren ohne Ersetzen' zur Anwendung für andere modellierte Klassen

Aktion	Modelloption	Nicht empfohlen?	Alternative
Klassenoption ändern	Abstrakt  FIDs erstellen  Oberklasse ersetzen  Web-Services-Bindungsstil  WS ist XML-Dokument  Dokumenttyp  Fassaden-Bean erstellen  Anbietername  Anforderungshandler  Antworthandler  Anforderung validieren  XML-Dokument  XML-Schema	Nicht empfohlen	Erstellen Sie eine neue Operation in einer projektspezifischen Klasse, schließen Sie dabei externe APIs der Anwendungsklassenfunktionalität mit ein, sofern geeignet, und stellen Sie die entsprechenden Optionen für die neue Klasse ein.
Operation hinzufügen	Keine	Nicht empfohlen	Erstellen Sie eine neue Operation in einer projektspezifischen Klasse, schließen Sie dabei externe APIs der Anwendungsklassenfunktionalität mit ein, sofern geeignet, und stellen Sie die entsprechenden Optionen für die neue Klassenoperation ein.



Tabelle 8. 'Unterklassifizieren ohne Ersetzen' zur Anwendung für andere modellierte Klassen (Forts.)

Aktion	Modelloption	Nicht empfohlen?	Alternative
Operation ändern	Keine	Nicht empfohlen	Erstellen Sie eine neue Operation in einer projektspezifischen Klasse und schließen Sie dabei externe APIs der Anwendungsklassenfunktionalität mit ein, sofern geeignet.
Operationsoption ändern	Business-Intelligence-Prüfaufrufe Geschäftsdatumsfeld Byte-Zeichensatz für Nachrichtencodierung Sicherheitsfunktion erstellen Ist XA transaktionsorientiert Nachrichtentyp Warteschlange für Connector-Factory JNDI-Name Antwortwarteschlange für JNDI-Name Zeitlimitüberschreitung für Antwortnachricht Schattentyp Transaktionsorientiert Übertragungswarteschlange für JNDI-Name Sichere Felder	Nicht empfohlen	Erstellen Sie eine neue Operation in einer projektspezifischen Klasse, schließen Sie dabei externe APIs der Anwendungsklassenfunktionalität mit ein, sofern geeignet, und stellen Sie die entsprechenden Optionen für die neue Klassenoperation ein.
Operationsparameter-Option ändern	Pflichtfelder	Nicht empfohlen	Erstellen Sie eine neue Operation in einer projektspezifischen Klasse, schließen Sie dabei externe APIs der Anwendungsklassenfunktionalität mit ein, sofern geeignet, und stellen Sie die entsprechenden Optionen für die neue Klasse ein.

## B.5 Domänenüberschreibung

### B.5.1 Domänendefinitionen

Tabelle 9. Domänendefinitionen überschreiben

Aktion	Modell-option	Nicht empfohlen?	Alternative
Eine bestimmte Anwendungsdomänendefinition ändern		Nicht empfohlen (teilweise)	Die Anpassung der folgenden Anwendungsdomänendefinitionen durch den Kunden ist nicht zulässig: TRUNCATED_NOTE_TEXT NOTE_TEXT INCIDENT_DESCRIPTION SERVICE_DELIVERY_NOTE_TEXT_SMALL INJURY_DESCRIPTION ACTION_TAKEN CITIZEN_ACCOUNT_RICH_TEXT CW_RICH_STRING VIEW_LIFE_EVENTS_POST_SUBMIT_XML_DATA RICH_TEXT_EDITOR_WIDGET SEC_RICH_TEXT_VIEW_WIDGET RICH_TEXT PROGRESS_RICH_TEXT_SMALL
Domänendefinitions-option ändern	Name der Codetabelle  Code-tabellen-Stammebene	Nicht empfohlen	Eine neue Domänendefinition mit entsprechendem Codetabellennamen und entsprechender Stammebene erstellen und in ihre eigene Verarbeitung einschließen.  Es ist für den Kunden nicht zulässig, diese Optionen für Anwendungsdomänendefinitionen zu ändern.

Tabelle 9. Domänendefinitionen überschreiben (Forts.)

Aktion	Modelloption	Nicht empfohlen?	Alternative
	Eingebettete Leerzeichen komprimieren In Großbuchstaben umwandeln Angepasster Validierungsfunktionsname Standard Höchstwert Mindestgröße Mindestwert Mustererkennung Führende Leerzeichen entfernen Nachfolgende Leerzeichen entfernen Speichertyp	Zulässig	
	Maximalgröße	Zulässig (teilweise)	Nur zum Vergrößern zulässig. Zum Verkleinern der Größe einer Anwendungsdomänendefinition wenden Sie sich bitte an den Support. Nicht zur Verwendung beim Ändern der Maximalgröße der Domänendefinition 'USERNAME'.
Den Typ einer Domänendefinition ändern	Keine	Nicht empfohlen	Eine neue Domänendefinition mit entsprechendem Typ erstellen und in Ihre eigene Verarbeitung einschließen. Es ist für den Kunden nicht zulässig, die Grundtypen von Anwendungsdomänendefinitionen zu ändern.
Zeichenfolgelänge einer Domänendefinition ändern	Keine	Zulässig (teilweise)	Nur zum Vergrößern zulässig. Zum Verkleinern der Größe einer Anwendungsdomänendefinition wenden Sie sich bitte an den Support. Nicht zur Verwendung beim Ändern der Zeichenfolgelänge der Domänendefinition 'USERNAME'.
Eine neue Domänendefinition basierend auf einer Anwendungsdomänendefinition erstellen	Keine	Zulässig	

## B.6 Beziehungen

### B.6.1 Zuweisungsfähigkeit

Table 10. Zuweisbare Beziehungen

Aktion	Nicht empfohlen?
Ein vom Kunden bereitgestelltes Struct für ein Anwendungs-Struct oder eine Entität zuweisbar machen	Zulässig
Ein Anwendungs-Struct für ein anderes Anwendungs-Struct oder eine Entität zuweisbar machen	Nicht empfohlen

### B.6.2 Spaltenberechnungen (Aggregation)

Table 11. Aggregationen

Aktion	Nicht empfohlen?
Ein Anwendungs-Struct in einem vom Kunden bereitgestellten Struct aggregieren (d.h. ein Kunden-Struct erstellen, das ein Anwendungs-Struct 'enthält')	Zulässig
Ein vom Kunden zur Verfügung gestelltes oder ein Anwendungs-Struct in einem Anwendungs-Struct aggregieren (d.h. ein beliebiges Struct durch Aggregation einem Anwendungs-Struct hinzufügen)	Nicht empfohlen

### B.6.3 Fremdschlüssel

Table 12. Fremdschlüssel

Aktion	Nicht empfohlen?
Einen neuen Fremdschlüssel erstellen, dem die vom Kunden bereitgestellte Entität untergeordnet ist	Zulässig
Einen neuen Fremdschlüssel erstellen, dem die Anwendungsentität untergeordnet ist	Nicht empfohlen

### B.6.4 Index

Table 13. Indizes

Aktion	Nicht empfohlen?
Neuen Index erstellen (für die Anwendung oder für eine vom Kunden bereitgestellte Entität) mithilfe eines vom Kunden bereitgestellten Structs	Zulässig
Neuen Index erstellen (für eine Anwendung oder für eine vom Kunden bereitgestellte Entität) mithilfe eines Anwendungs-Structs	Nicht empfohlen

### B.6.5 Eindeutiger Index

Table 14. Eindeutige Indizes

Aktion	Nicht empfohlen?
Neuen eindeutigen Index zu einer Anwendungsentität erstellen	Nicht empfohlen
Neuen eindeutigen Index zu einer vom Kunden bereitgestellten Entität mithilfe eines Anwendungs-Structs erstellen	Nicht empfohlen
Neuen eindeutigen Index zu einer vom Kunden bereitgestellten Entität mithilfe eines vom Kunden bereitgestellten Structs erstellen	Zulässig

---

## B.7 Andere Mechanismen

### B.7.1 Ausschlüsse

*Tabelle 15. Ausschlüsse*

Aktion	Nicht empfohlen?
Verwenden Sie Ausschlüsse für den Versuch, Klassen aus einem Server-Build auszuschließen	Nicht empfohlen - ab Version 6.0.3 werden Anwendungsklassen nicht mehr mit jedem Mal neu erstellt



---

## Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden. Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden. Für die in diesem Handbuch beschriebenen Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing  
IBM Europe, Middle East & Africa  
Tour Descartes  
2, avenue Gambetta  
92066 Paris La Defense Cedex  
France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht. Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation  
Dept F6, Bldg 1  
294 Route 100  
Somers NY 10589-3216  
U.S.A.

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen.

IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Alle von IBM angegebenen Preise sind empfohlene Richtpreise und können jederzeit ohne weitere Mitteilung geändert werden. Händlerpreise können u. U. von den hier genannten Preisen abweichen.

Diese Veröffentlichung dient nur zu Planungszwecken. Die in dieser Veröffentlichung enthaltenen Informationen können geändert werden, bevor die beschriebenen Produkte verfügbar sind.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

#### COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielprogramme entstehen.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihres Unternehmens) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corporation abgeleitet.

© Copyright IBM Corp. \_Jahreszahl oder Jahreszahlen eingeben\_. Alle Rechte vorbehalten.



---

## Marken

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corp in den USA und/oder anderen Ländern. Weitere Produkt- oder Servicenamen können Marken von IBM oder anderen Herstellern sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Website "Copyright and trademark information" unter <http://www.ibm.com/legal/us/en/copytrade.shtml>.

Sonstige Namen können Marken der jeweiligen Rechtsinhaber sein. Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken oder Dienstleistungsmarken anderer Unternehmen sein.





