

Business Rule Framework plus (BRFplus)

Inhalt

1	Business Rule Framework plus (BRFplus)	4
1.1	Getting Started	6
	Die Benutzungsoberfläche	7
	Workbench-Personalisierung	12
	Typischer Workflow	19
	Workflow for Analytical Functions	21
	Quick Start Tutorial	29
1.2	Konzepte	29
	Anwendung	30
	Ablagetypen	35
	Namenskonventionen für BRFplus	36
	Verwaltungsdaten	39
	Kontext und Ergebnis	46
	Objekte anlegen	48
	Objekte suchen	52
	Objekte löschen	57
	Datenobjekte und Datenobjekttypen	61
	Vergleichsoperationen	70
	Ausdrücke	80
	Ausdrucksart	129
	Aktionen	132
	Dynamischer Datenbank-View (DDBV)	142
	Funktion	146
	Partnermodulintegration	164
	Regel	167
	Regelsatz	168
	Katalog	171
	Objektfilter	174
	Versionierung	176
	Berechtigungen	183
1.3	Aufgaben	185
	Anwendungen modellieren	186
	Funktionen bauen	190
	Ausdrücke anlegen	196
	Ausdrucksarten anlegen	235
	Datenobjekte anlegen	236

Regel anlegen.	243
Regelsatz anlegen.	245
Mit Katalogen arbeiten.	247
Aktionsarten anlegen.	252
Konfiguration der Benutzereinstellungen.	258
Objekte transportieren.	259
Versionen festlegen.	260
Aktuelle Objekte zu Favoriten hinzufügen.	260
Verwendung von Objekten überprüfen.	261
Objekte in einer Anwendung löschen.	261
Mit Tools arbeiten.	262

1 Business Rule Framework plus (BRFplus)

Verwendung

Business Rule Framework plus (BRFplus) bietet eine umfangreiche Programmierschnittstelle (API) und eine Benutzungsoberfläche (UI) zur Definition und Verarbeitung von Geschäftsregeln. Es ermöglicht eine intuitive Modellierung von Regeln sowie deren Wiederverwendung in anderen Anwendungen.

Im Folgenden finden Sie einige Beispielszenarien für den Gebrauch von BRFplus:

- Datenvalidierung und Ermittlung ungültiger Daten und Status
- Abgleichen von Verantwortlichkeiten, passenden Produkten und Einsatzorten
- Berechnung von Kosten, Aufwänden und Risiken
- BRFplus als technische Konfigurations-Engine

Die wichtigsten BRFplus-Komponenten schließen Anwendung, Funktion, Katalog, Ausdruck, Aktion und Datenobjekt ein.

Die Schnittstelle zwischen einer mit BRFplus modellierten Geschäftsregel und einer Anwendung, die diese Geschäftsregel verwendet, wird durch eine BRFplus-Funktion bereitgestellt. Die Funktion dient als Container für die gesamte Geschäftslogik einer Regel, unabhängig von ihrer Komplexität. Regeln werden als Ausdrücke implementiert und einer Funktion zugeordnet. Die Regeleingabe wird als Kontext und die Regelausgabe als Ergebnis bezeichnet. Kontext und Ergebnis bestehen aus einem der folgenden Datenobjekttypen: Element, Struktur und Tabelle

BRFplus unterstützt Funktionen, wie Simulation, Trace, Transport, Export und Import von XML-Daten.

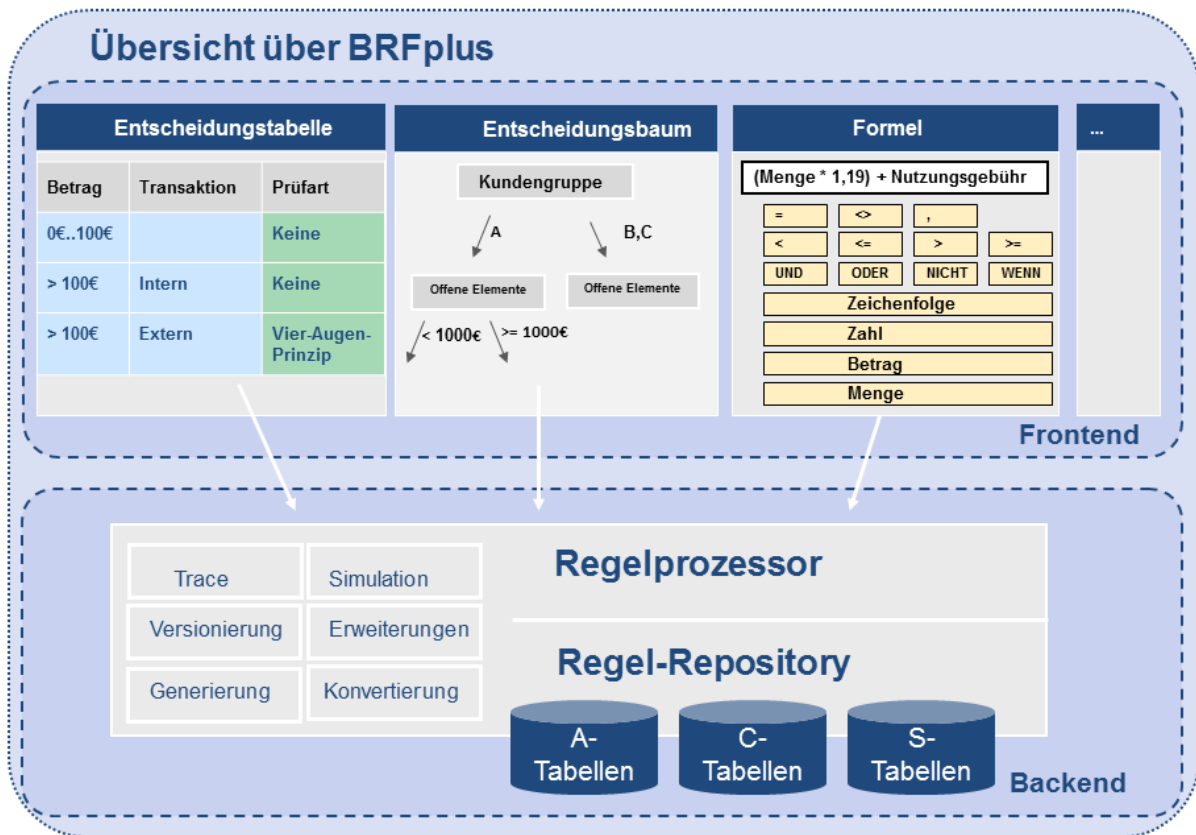


Abbildung 1: Übersicht über BRFplus

Einführungshinweise

BRFplus ist ein ABAP-basiertes Framework und ist daher bestens für eine Integration in eine ABAP-basierte Systemumgebung geeignet, beispielsweise als Erweiterung für eine bestehende Komponente der SAP Business Suite. Wenn Ihre Systemumgebung jedoch hauptsächlich auf Java basiert oder Sie planen, Geschäftsregeln für eine auf einer serviceorientierten Architektur (SOA) basierenden Systemlandschaft zu implementieren, ist es empfehlenswert, die Java-basierte Anwendung SAP Business Rules Management (BRM) zu verwenden. SAP BRM ist mit SAP NetWeaver 7.1 EHP1 oder höher verfügbar.

Es ist ebenfalls möglich, ein Szenario zu implementieren, in dem beide Komponenten gleichzeitig verwendet werden: In einer gemischten Umgebung bestehend aus ABAP- und Java- Elementen können beide Engines in ihrem entsprechenden Bereich verwendet werden. BRFplus und SAP BRM können sich gegenseitig aufrufen. Das bedeutet beispielsweise, dass Sie in SAP BRM Regeln pflegen können und diese per Fernaufruf in BRFplus verwenden können und umgekehrt.

Auch in einer SOA-Umgebung ist es empfehlenswert, die Web-Services von BRFplus zu verwenden. Dies gilt auch für Szenarien, in denen ein Großteil der zu verarbeitenden Daten in einem ABAP-Backend-System abgelegt sind. In diesem Fall bringt BRFplus Vorteile hinsichtlich Performance, Sizing sowie der Verfügbarkeit der integrierten Workbench.

1.1 Getting Started

Use

In order to use the extended BRFplus capabilities provided by SAP NetWeaver Decision Service Management (DSM), ensure that DSM is installed and set up. For information, see:

- [SAP NetWeaver Decision Service Management Installation Guide](#) .
-

This section provides introductory information to help you get started with BRFplus.

[Understanding the User Interface \[page 7\]](#)

Introduces the user interface (UI) of the BRFplus workbench.

[Typical Workflow \[page 19\]](#)

Provides an overview of all the stages involved to model and implement business rules in BRFplus.

[Workflow for Analytical Functions \[page 21\]](#)

Provides an overview of the tasks required to model and implement analytical functions in BRFplus.

[Quick Start Tutorials \[page 29\]](#)

Provides step-by-step lessons that guide you through the process of modeling an application, from start to finish, to get you quickly up and running with BRFplus.

Note

Before you begin working with BRFplus, ensure that the following SICF core services are active:

- /sap/public/bc/icons
- /sap/public/bc/icons_rtl
- /sap/public/bc/its/mimes
- /sap/public/bc/pictograms
- /sap/public/bc/ur
- /sap/public/bc/webdynpro/*
- /sap/public/bc/webicons
- /sap/public/icman
- /sap/public/myssocntl
- /sap/public/ping
- /sap/bc/echo/*
- /sap/bc/error/*

1.1.1 Die Benutzungsoberfläche

Verwendung

Die Benutzungsoberfläche (UI) von BRFplus enthält eine Workbench, die es Ihnen ermöglicht, Geschäftsregeln zu verwalten. Sie können Objekte anlegen, ändern und auch löschen. Die Workbench stellt Zugriff auf alle im System verfügbaren BRFplus-Objekte zur Verfügung. Die Workbench ermöglicht nicht nur die Gestaltung der BRFplus-spezifischen Regelinhalte, sondern stellt auch Werkzeuge zur Integration einer BRFplus-Anwendung in Ihre Host-Anwendung zur Verfügung, von der Sie BRFplus aufrufen. Hierzu implementieren Sie die jeweiligen Aufrufe in den Quellcode Ihrer Anwendung. Die Workbench unterstützt Sie dabei, indem sie Web-Services, Funktionsmodule und Code-Snippets generiert, die Sie nutzen können, um Ihren ABAP-Code mit einer bestimmten BRFplus-Funktion zu verknüpfen.

Sie können die BRFplus-Workbench folgendermaßen starten:

- Im Backend-System mithilfe der Transaktion **BRFPLUS**.
- Im Web-Browser mithilfe der BRFplus-Workbench-URL, die für Ihre Systemlandschaft gültig ist.

Hinweis

Die BRFplus-Workbench basiert auf der WebDynpro-ABAP-Technologie von SAP NetWeaver. Somit können Sie generische Modifikatoren und Parameter auf die Browser-URL anwenden, um das Verhalten der Workbench anzupassen. Sie können zum Beispiel den Parameter **sap-accessibility=X** zur URL hinzufügen, um Unterstützung eines Bildschirmleseprogramms im UI zu aktivieren.

Voraussetzungen

Die Rolle *BRFplus-Administrator* (**SAP_BC_FDT_ADMINISTRATOR**) wurde Ihrem Benutzerprofil hinzugefügt. Sie können natürlich ebenso eine benutzerdefinierte Rolle mit einer Teilmenge an Berechtigungen nutzen, die in **SAP_BC_FDT_ADMINISTRATOR** enthalten sind.

Funktionsumfang

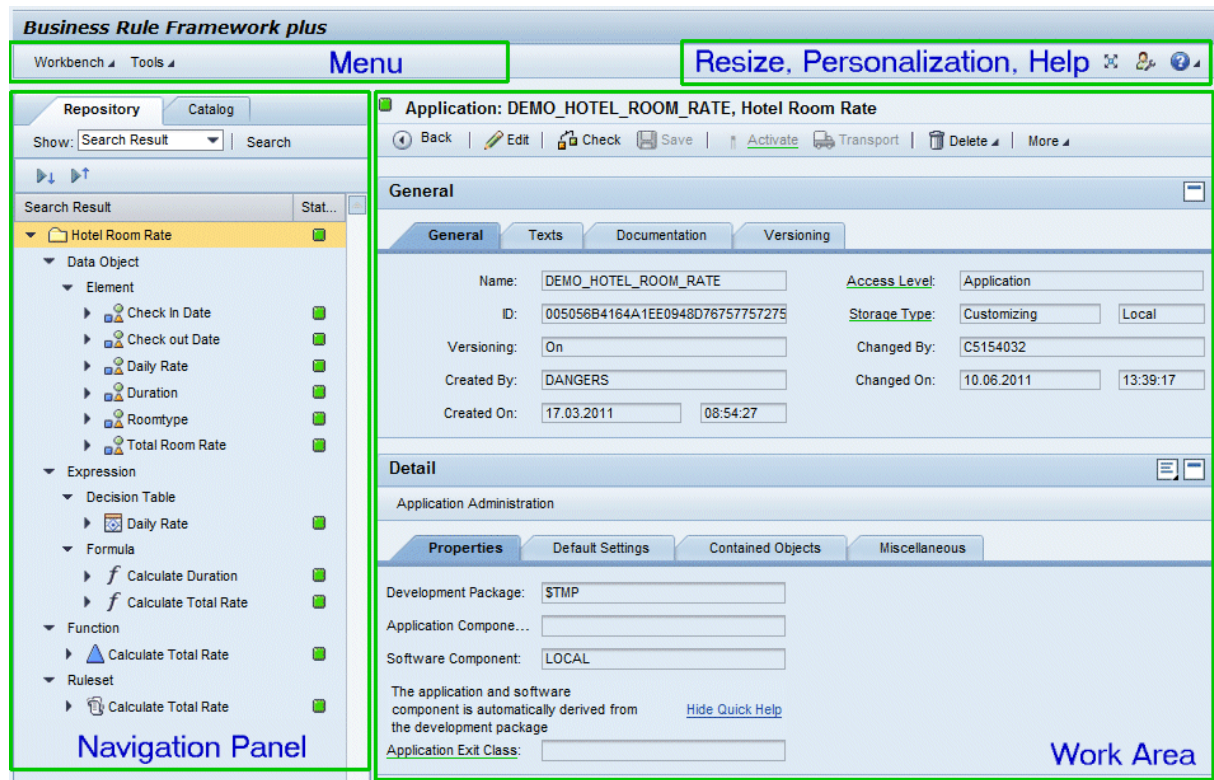
Die BRFplus-Workbench besteht aus zwei Hauptbereichen: den *Navigationsbereich* auf der linken Seite und dem *Arbeitsbereich* auf der rechten Seite. Die Workbench stellt ebenfalls einige allgemeine Funktionen über die *Menüleiste* zur Verfügung. Das Dialogfenster *Systemstatus*, das Sie unter *Hilfe* finden, zeigt das Backend-System und den Client, auf dem Sie aktuell angemeldet sind, den angemeldeten Benutzer und die Sprache sowie den aktuellen UI-Modus.

Die Workbench hat zwei UI-Modi:

- Einfacher Modus
Dieser UI-Modus wird für Benutzer im Unternehmen empfohlen. Dieser Modus beschränkt sich auf unternehmensrelevante Einstellungen und blendet einen Großteil der technischen Details aus, die nur für administrative Aufgaben gebraucht werden.

- **Expertenmodus**
Dieser UI-Modus wird für Geschäftsexperten oder Entwickler empfohlen. Im *Expertenmodus* haben Sie Zugriff auf einige zusätzliche Werkzeuge, die für Verwaltungs- und Bereinigungsaufgaben gebraucht werden. Darüber hinaus sind einige technische Funktionen nur im *Expertenmodus* verfügbar (zum Beispiel die Funktion *Zur aktiven Version zurückkehren* mit der Änderungen an einem inaktiven Objekt verworfen werden können).

Sie können im Dialogfenster *Personalisierung* zwischen den zwei Modi wechseln.



Menü

Das *Menü* besteht aus folgenden Bestandteilen:

- *Workbench*
Sie können Workbench-Operationen ausführen, wie eine Anwendung anlegen, die aktuellen Objekte im Repository anzeigen, das aktuell in die Workbench geladene Objekt mit dem Repository-Baum synchronisieren sowie das *aktuelle Objekt als Favoriten hinzufügen* [Seite 260].
- *Werkzeuge*
Die Verfügbarkeit der Werkzeuge hängt vom gewählten UI-Modus ab. Im einfachen Modus ist nur das Simulationstool aktiviert. Weitere Werkzeugkomponenten sind nur im Expertenmodus verfügbar.
 - *Anwendungsverwaltung*
Sie können mithilfe des Werkzeuges zur Anwendungsverwaltung *Nicht verwendete Objekte löschen* [Seite 269]. Sie können ebenfalls alte Versionen von Objekten, die nicht mehr gebraucht werden, verwerfen, zum Löschen vorgemerkte Objekte löschen sowie Objekte neu organisieren.
 - *Anwendungsverwendung*
Sie können das Werkzeug für Anwendungsverwendung verwenden, um alle Anwendungen abzufragen, die von der ausgewählten Anwendung verwendet werden oder andere Anwendungen abzufragen, die die ausgewählte Anwendung verwenden.

- [Lean Trace](#)
Dieses Werkzeug ermöglicht es Ihnen, das Laufzeitverhalten einer BRFplus-Funktion allgemein zu prüfen und zu analysieren. Der Name "Lean Trace" zeigt an, dass dieses Werkzeug auf einer relaxierten Granularitätsebene arbeitet, so dass bestimmte Ausführungsschritte aggregiert oder vollständig ausgeblendet sind.
- [Massenänderung](#)
Mit diesem [Werkzeug \[Seite 278\]](#) können Sie eine Auswahl an BRFplus-Objekten basierend auf verschiedenen Kriterien anlegen und dann eine bestimmte Änderung auf alle gewählten Objekte gleichzeitig anwenden.
- [Simulation](#)
Sie können Eingaben für die Kontextdatenobjekte einer Funktion bereitstellen und die Ausführung zu Test- und Debugging-Zwecken simulieren. [Simulation \[Seite 270\]](#) ist ähnlich zum Testen von Funktionsbausteinen.
- [Transportanalyse](#)
In einem [Transportauftrag \[Seite 259\]](#) können Sie die Konsistenz aller BRFplus-Objekte prüfen und analysieren.
- [Web-Service-Generierung](#) und [RFC-Generierung](#)
Externe Benutzer können die in BRFplus definierten Regeln mithilfe des [Werkzeugs Web-Service-Generierung \[Seite 272\]](#) verarbeiten. Ähnlich kann vorgegangen werden, wenn eine SAP-Systemlandschaft installiert ist und Sie auf eine BRFplus-Anwendung aus einem System heraus zugreifen wollen, während sich die Anwendung jedoch in einem anderen System befindet. Um die erforderliche Umgebung einzurichten, können Sie das [Werkzeug RFC-Generierung](#) verwenden.
- [XML-Export](#) und [XML-Import](#)
Sie können aktive Versionen von BRFplus-Objekten mithilfe der BRFplus-spezifischen Konvertierung zu XML bequem von einem System in ein anderes transportieren (vorausgesetzt beide Systeme unterstützen BRFplus). Weitere Informationen finden Sie unter [Export und Import von XML-Daten \[Seite 274\]](#).
- [Hilfe](#)
Unter [Hilfe](#) können Sie direkt auf die kontextspezifische [Hilfe zur Anwendung](#) zugreifen, die an den aktuell bearbeiteten Objekttyp angehängt ist.
Sie können ebenso den Bereich [Help Center](#) am rechten Rand des Fensters öffnen. Dort finden Sie einen Link, der Sie auf die Startseite der BRFplus-Anwendungshilfe führt, von der aus Sie auf verschiedene Hilfethemen zugreifen können. Darüber hinaus finden Sie im Hilfemenü eine Legende, in der die wichtigsten Symbole der Workbench sowie die Bedeutung der in der Benutzungsoberfläche verwendeten typographischen Konventionen erklärt werden. Abschließend können Sie die [Systeminformationen](#) hier finden.

➔ Empfehlung

Wenn Sie noch nicht mit der BRFplus-Workbench gearbeitet haben, empfehlen wir, einen Rechtsklick auf dem UI-Hintergrund auszuführen und [Soforthilfe anzeigen](#) aus dem Kontextmenü auszuwählen. Es werden Ihnen zusätzliche Hilfetexte auf dem UI angezeigt (entweder statische Texte oder Texte, die nur sichtbar sind, wenn Sie mit der Maus über einen grün unterstrichenen Text fahren). Diese Hilfetexte helfen Ihnen, den Zweck eines bestimmten Bildelements oder das Konzept einer bestimmten Funktion zu verstehen. Sobald Sie sich mit der Workbench vertraut gemacht haben, können Sie diese Hilfetexte ausblenden, indem Sie [Soforthilfe ausblenden](#) aus demselben Kontextmenü auswählen.

Navigationbereich

Sie können das BRFplus-Objekt-Repository mithilfe des [Navigationsbereichs](#) auf der linken Seite des Workbench-Fensters durchsuchen. In den Einstellungen unter [Personalisierung](#) können Sie festlegen, ob die Objekte mit dem technischen Namen oder mit dem Kurztext angezeigt werden sollen.

Die im Navigationsbereich angezeigten Objekte sind keine einfachen Einträge in einer flachen Liste. Vielmehr ist jedes Objekt der Einstiegspunkt für eine Hierarchie von Unterelementen und Verwendungen. Sie können durch die Objekthierarchie navigieren, indem Sie auf das Dreieckssymbol links von jedem Objekt klicken.

Ein Anwendungseintrag im Navigationsbereich dient als Einstiegspunkt für die verschiedenen Objekttypen, aus denen die Anwendung besteht, alle Objektinstanzen für jeden Typ, deren Verwendung in anderen Objekten, die wiederum als Ausgangspunkt für einen weiteren Drilldown dienen.

Das System liefert für jedes Objekt eine visuelle Kennzeichnung des Objektstatus.

- Objektart: Das Symbol links neben dem Objektnamen zeigt den Objekttypen an (Anwendung, verschiedene Ausdrucksarten, Funktionen usw.).
- Schriftart: In **Fettschrift** dargestellte Objekte sind transportierbar (anderenfalls sind sie lokal im aktuellen System).

Hinweis

Diese Einstellung wird von der Anwendung, zu der ein Objekt gehört, geerbt. Ob eine Anwendung transportierbar ist oder nicht, hängt vom Entwicklungspaket ab, dem die Anwendung zugeordnet ist.

- Ampel Objekte mit einer grünen Ampel sind aktiv und können in einer Anwendung verwendet werden.

Die BRFplus-Objekte im [Navigationsbereich](#) können in verschiedenen Sichten angeordnet werden. In den Einstellungen unter [Personalisierung](#) können Sie für jede Sicht festlegen, ob sie angezeigt werden soll oder nicht. Darüber hinaus können Sie festlegen, welche der verschiedenen Sichten beim Starten des Systems angezeigt werden soll. Die folgenden Sichten sind verfügbar:

- [Repository](#)
Im Repository können Sie auf alle Objekte von allen Anwendungen zugreifen. Die [Repository](#)-Sicht ermöglicht eine detaillierte Analyse eines Objekts mittels eines rekursiven Drilldowns in die Unterobjekte der Objekte sowie in den Verwendungsnachweis des Objekts.
- [Zuletzt verwendet](#)
Sie können auf einfache Weise auf Ihre zuletzt bearbeiteten Objekte zugreifen. In der Liste können bis zu 20 Objekte gespeichert werden. Sie wird automatisch vom System nach dem „first in, first out“-Prinzip gepflegt.
- [Favoriten](#)
Sie können alle Objekte, die Sie Ihren Favoriten hinzugefügt haben, anzeigen. Dies ist nützlich, wenn Sie eine dauerhafte Liste an Objekten führen wollen, die für Sie interessant sind, jedoch nicht täglich bearbeitet werden. Um ein Objekt der Sicht [Favoriten](#) hinzuzufügen, suchen Sie das jeweilige Objekt zunächst in der Sicht [Repository](#) und wählen dann [Zu Favoriten hinzufügen](#) aus dem Kontextmenü des Objekts.
- [Katalog](#)
Sie können Kataloge als Container für eine Untermenge an Objekten definieren und darauf aus dem Repository heraus zugreifen.

Hinweis

Die Sichten *Zuletzt verwendet* und *Favoriten* liefern eine flache Liste von Objekten ohne die zusätzlichen Drilldown-Optionen der *Repository*-Sicht. Sie können die Drilldown-Funktion auf einfache Weise nutzen, indem Sie *In Repository-Sicht zeigen* aus dem Kontextmenü des jeweiligen Objekts wählen.

Arbeitsbereich

Im Arbeitsbereich können Sie Objekte anzeigen und bearbeiten. Am oberen Bildrand zeigt das System den Abschnitt *Allgemein* mit den Kopfdaten des Objekts an. Dieser Abschnitt ist generisch und für alle von BRFPplus unterstützten Objekttypen verfügbar. Die für einen Objekttypen spezifischen Einstellungen werden im Abschnitt *Details* unterhalb des Abschnitts *Allgemein* angezeigt.

Prüfergebnisse

Unter bestimmten Bedingungen kann der Arbeitsbereich einen dritten Abschnitt enthalten: *Prüfergebnisse*. Für jeden in BRFPplus verfügbaren Objekttyp bietet das System objekttypspezifische Konsistenzprüfungen, die bei Aktivierung des Objekts implizit ausgeführt werden. Sie können die Prüfungen ebenso manuell starten, indem Sie die Drucktaste *Prüfen* in der Workbench wählen. Sobald durch die Prüfung ein Problem festgestellt wurde (sei es schwerwiegend oder nicht), zeigt das System den Abschnitt *Prüfergebnisse* am oberen Rand des Arbeitsbereichs an und listet alle während der Prüfung generierten Nachrichten auf. Sie können dann mit der Untersuchung der Probleme beginnen. Die Liste der Nachrichten wird angezeigt, bis Sie den Abschnitt *Prüfergebnisse* explizit mit der Drucktaste *Schließen* schließen.

Manchmal liefern die Prüfungen eine große Zahl an Nachrichten, abhängig von der Komplexität des geprüften Objekts und den Verknüpfungen mit anderen Objekten. In so einem Fall kann das Navigieren durch die große Zahl an Nachrichten in einem relativ kleinen Abschnitt sehr mühsam sein. Um dieses Problem zu lösen, bietet der Abschnitt *Prüfergebnisse* eine Downloadfunktion, die es ermöglicht, die gesamte Liste der Nachrichten in einer Microsoft Excel-Datei zu speichern. Auf diese Weise können Sie sich einfach einen Überblick verschaffen. Sie können die Tabellenkalkulationsdatei als Arbeitsvorrat nutzen, um nachzuverfolgen, welche Probleme bereits gelöst wurden und welche noch bearbeitet werden müssen.

Systeminformationen

Das Dialogfenster *Systeminformationen*, das über das *Hilfe*-Menü geöffnet werden kann, liefert die folgenden Verwaltungsdaten zur laufenden Sitzung:

- System
 - System ID
 - Systemmandant, in dem Sie angemeldet sind
 - Systemmodus (entweder regulärer Modus oder Verteilungsmodus; der Verteilungsmodus ist nur für Business ByDesign-Systeme relevant)
- Benutzer
 - Benutzername
 - Anmeldesprache
 - Aktueller UI-Modus (*einfacher Modus* vs. *Expertenmodus*, siehe Abschnitt oben)
 - Für Objektzeitstempel verwendete Zeitzone (Ortszeit des eingeloggten Benutzers, oder UTC) Diese Einstellung kann im Dialogfenster *Personalisierung* geändert werden.

Tastaturunterstützung

Die Benutzung der Maus ist nicht immer der schnellste Weg, um die Workbench zu bedienen. Sie können auf die *Workbench* sowie auf das Menü *Werkzeuge* jeweils mithilfe der Tastenkombination Alt + W zugreifen. Es


ist ebenfalls eine Liste von Benutzeraktionen verfügbar, die Sie mithilfe von Tastaturkürzeln ausführen können. Diese Tastaturkürzel können im Dialogfenster [Workbench-Personalisierung](#) angepasst werden.

Weitere Informationen

[Workbench-Personalisierung \[Seite 12\]](#)

1.1.2 Workbench-Personalisierung

Definition

Die BRFplus-Workbench ist der zentrale Ort zum Modellieren und Testen von Geschäftsregeln. BRFplus bietet zahlreiche Möglichkeiten für die Konfiguration der Workbench-Benutzeroberfläche, um diese optimal gemäß Ihren Anforderungen zu gestalten. Für den Zugriff auf die Personalisierungseinstellungen wählen Sie  in der rechten oberen Ecke des Workbench-Fensters.

Hinweis

Die meisten Änderungen in den Personalisierungseinstellungen werden unmittelbar wirksam, sobald Sie das Dialogfenster [Personalisierung](#) schließen und zu einem anderen Objekt navigieren. Für einige der Personalisierungseinstellungen ist es jedoch erforderlich, dass Sie die Workbench neu starten, damit die Änderungen wirksam werden.

Funktionen

Benutzermodus

BRFplus eignet sich sowohl für Anwendungsbenutzer als auch für Geschäftsexperten und Entwickler. Da beide Anwendertypen verschiedene Anforderungen an ihre Arbeitsumgebung haben und manche Anwender beide Rollen innehaben, bietet BRFplus die Möglichkeit, die BRFplus-Workbench individuell an die jeweiligen Anforderungen anzupassen. Dies umfasst auch die Möglichkeit, die Benutzeroberfläche der Workbench auf zwei rollenorientierte Modi pro Anwender einzustellen, den einfachen Modus und den Expertenmodus.

Die Einstellung für den Anwendermodus berücksichtigt zwei Aspekte:

- Umfang der Personalisierungseinstellungen
Die geänderten Personalisierungseinstellungen werden im System mit Bezug zum aktuell ausgewählten Anwendermodus protokolliert. Wenn Sie also z. B. die Einstellungen für die [Sichten](#) im einfachen Modus ändern, wirkt sich dies nicht aus auf die Einstellungen der [Sichten](#) im Expertenmodus und umgekehrt.
- Auswahl des Anwendermodus
Wenn Sie das Dialogfenster Personalisierung öffnen, zeigt das System den Anwendermodus, der aktuell in der Workbench ausgeführt wird. Um den Anwendermodus zu wechseln, nehmen Sie die gewünschte

Auswahl vor und wählen Sie [Sichern](#). Das System passt das Workbench-Rendering gemäß den Einstellungen an, die für den neu ausgewählten Anwendermodus definiert wurden.

Hinweis

Als Systemadministrator können Sie zusätzliche Modi anlegen, die den Anforderungen verschiedener Benutzer oder Rollen in Ihrem Unternehmen entsprechen. Standardmäßig erben diese benutzerdefinierten UI-Modi ihre Einstellungen vom einfachen Modus, es sei denn, Sie ändern ausdrücklich eine bestimmte Einstellung. Außerdem können Sie mit benutzerdefinierten UI-Modi zusätzliche Tools, Web-Services oder Webseiten in die BRFplus-Workbench integrieren.

Kategorien

Die Personalisierungseinstellungen werden in folgende Kategorien gruppiert:

- Allgemein
Einstellungen für bereichsübergreifende Aspekte und das Startverhalten des Systems
- Sichten
Einstellungen zur Steuerung der Verfügbarkeit verschiedener Workbench-Sichten
- Repository
Einstellungen zur Steuerung des Verhaltens der [Repository](#)-Sicht.
- Kataloge
Einstellungen zur Steuerung des Verhaltens der [Katalog](#)-Sicht.
- Ausdruck
Einstellungen für spezifische Ausdrucksarten
- Filter
Einstellungen zum Definieren von Objektfiltern

Für alle Personalisierungseinstellungen verwendet das System vordefinierte Standardwerte. Falls Sie aufgrund der geänderten Einstellungen auf Probleme stoßen, klicken Sie einfach auf [Auf Standardeinstellungen zurücksetzen](#), um die Standardeinstellungen wiederherzustellen.

Allgemein

- [Maximale Anzahl der Abfrageergebnisse](#)
Mit dieser Einstellung können Sie eine Obergrenze für die Anzahl der übereinstimmenden Elemente festlegen, die dann vom System als Ergebnis der Abfrage angezeigt werden. Falls Ihr System viele Objekte enthält und die Systemperformance sehr langsam ist, ist es empfehlenswert, diese Anzahl zu verringern. Diese Einstellung wirkt sich auf die Objektlisten aus, die bei der Suche nach einem vorhandenen Objekt vom System angezeigt werden. Ist die Anzahl der übereinstimmenden Objekte für eine Abfrage höher als die hier definierte Einstellung, erhalten Sie eine entsprechende Systemnachricht. Sie können entscheiden, ob Sie diesen Wert erhöhen oder die Suchkriterien verfeinern möchten.
Standard: 200.

Hinweis

Die Anzahl der in der [Repository](#)-Sicht aufgelisteten Objekte ist von dieser Einstellung **nicht** betroffen. Hier zeigt das System immer alle Objekte an, die während der letzten Suche als Ergebnis zurückgegeben wurden.

- [Technische Namen anzeigen](#)
Mit dieser Einstellung können Sie definieren, ob das System den technischen Namen oder den Kurztext für die Anzeige des Objekts verwenden soll. Üblicherweise ist die Anzeige des Kurztextes angenehmer, da

dieser Text in natürlicher Sprache verfasst und ggf. auch in verschiedene Sprachen übersetzt ist. Der technische Name hingegen bietet den Vorteil, dass er unabhängig von der Anmeldesprache immer derselbe ist. Dies vereinfacht die Kommunikation zwischen Personen, die sich mit unterschiedlichen Sprachen am System anmelden.

Diese Einstellung betrifft das Anzeigeverhalten im Navigationsbereich der Workbench sowie in der Kopfzeile des Arbeitsbereichs eines derzeit bearbeiteten Objekts und alle Referenzen auf abhängige Objekte (z. B. alle Regeln eines Regelsatzes, Datenobjekte, die von einem Ausdruck verwendet werden usw.).

- **Technische Aspekte anzeigen**

Mit dieser Einstellung steuern Sie, ob das System diese Werkzeuge im Menü **Werkzeuge** anzeigt, die eher technischer Natur sind und für Systemadministratoren und Programmierer gedacht sind, wie z. B. **Anwendungsverwaltung** oder **XML-Export**. Für Geschäftsbutzer empfehlen wir diese Einstellung nicht zu verwenden, um Verwechslungen zu vermeiden.

- **Technische Attribute anzeigen**

Mit dieser Einstellung definieren Sie, ob das System zusätzliche technische Daten für einige der BRFplus-Objekte anzeigen soll. Beispielsweise werden dann interne IDs eines Entscheidungsbaum- oder Katalogknotens angezeigt. Diese Daten werden insbesondere für technische Untersuchungen und zu Debugging-Zwecken benötigt. Für Geschäftsbutzer sind diese Daten nicht erforderlich.

Weiterhin steuert diese Einstellung, ob die Anzahl der Dezimalstellen eines Elementardatenobjekts mit Untertyp `Amount` angezeigt wird. Diese Informationen werden sowieso ignoriert, da während der Laufzeit die Anzahl an Dezimalstellen eines Betrags automatisch von den zentralen Einstellungen in den Währungstabellen abgeleitet wird. Anders gesagt, die Anzeige der Anzahl an Dezimalstellen für ein Betrag-Datenobjekt hat keinerlei Einfluss auf das Systemverhalten.

Hinweis

Allgemein gilt für alle Daten, die über das Kennzeichen **Technische Attribute anzeigen** gesteuert werden, dass die Einstellung dieses Kennzeichens zu keinerlei funktionalen Einschränkungen für den Benutzer führt. Dies ist der Hauptunterschied zur bereits vorhin genannten Einstellung **Technische Aspekte anzeigen**, die dazu dient, wichtige Teile der Werkzeugfunktionen für den Benutzer auszublenden.

- **XML-Schema anzeigen**

Mit dieser Einstellung definieren Sie, ob der Benutzer während des XML-Exports entscheiden kann, welches XML-Schema für die zu exportierenden Daten zu verwenden ist. Wenn Sie diese Einstellung deaktivieren, hat der Benutzer keinen Einfluss auf die Schemaauswahl. In diesem Fall verwendet das System standardmäßig das interne Schema.

- **Objektzeitstempel basierend auf UTC-Referenzzeit**

Mit dieser Einstellung entscheiden Sie, wie das System die Zeitstempel für Objekte darstellt. Diese Einstellung wirkt sich aus auf das Anlegedatum und die Uhrzeit sowie auf das Änderungsdatum und -uhrzeit eines Objekts, die Zeitstempel von Objektversionen und den Gültigkeitszeitraum von Regelobjekten. Dies hat folgende Auswirkungen:

- **Ein**

Alle angezeigten Zeitstempel werden in ihr UTC-Äquivalent (UTC = Universal Time Coordinated) transformiert. Dadurch erhalten Sie eine standardisierte Zeitbasis für alle BRFplus-Objekte im System, unabhängig davon, wann und von wem die Objekte angelegt oder geändert wurden. Diese Einstellung wird empfohlen, wenn Sie Objekte basierend auf Zeitstempeln vergleichen oder bearbeiten müssen.

- **Aus**

Alle angezeigten Zeitstempel werden gemäß den Zeitzoneneinstellungen des Benutzers angezeigt, der das Objekt angelegt oder gesichert hat. Diese Einstellung wird von den Benutzern üblicherweise

als praktisch erachtet. Sie kann jedoch zu Verwechslungen führen, wenn verschiedene Benutzer in unterschiedlichen Zeitzonen Objekte im selben System bearbeiten und Sie eine Übersicht über verschiedene Objekte basierend auf den jeweiligen Zeitstempeln benötigen.

Generell ist die Verwendung von UTC-Zeitstempeln empfehlenswert für administrative Aufgaben, bei denen viele Objekte zu verarbeiten sind. Lokale Zeitstempel hingegen geben dem Benutzer hilfreiche Informationen darüber, wann ein bestimmtes Objekt angelegt oder geändert wurde.

i Hinweis

Die Zeitstemeleinstellung hat keinen Einfluss auf Datums- und Uhrzeitwerte, die Sie in Datenobjekten mit Typ Zeitpunkt verwenden. Dasselbe gilt auch für Datum und Zeitfunktionen in Formelausdrücken. In folgenden Fällen erwartet das System immer UTC-Zeitstempel.

- **Allgemeiner Bereich erweitert**

Mit dieser Einstellung entscheiden Sie, ob das System administrative Daten eines Objekts standardmäßig anzeigt oder nicht. Wenn Sie diese Einstellung aktivieren, erweitert das System den Bereich **Allgemein** im oberen Arbeitsbereich, sobald Sie zu einem BRFplus-Objekt navigieren. Dadurch erhalten Sie eine umfangreiche Übersicht über alle Daten, die für ein Objekt erfasst wurden. Allerdings benötigt die Anzeige dieser Daten viel Platz auf der Benutzungsoberfläche. Wenn Sie diese Einstellung deaktivieren, werden diese Informationen ausgeblendet und Sie haben deutlich mehr Platz, um effektiv mit den Objekteinstellungen zu arbeiten.

Sichten

In diesem Register definieren Sie, welche der unterschiedlichen Sichten im Workbench-Navigationsbereich angeboten werden und welche der ausgewählten Sichten beim Start der Workbench angezeigt wird. Diese Einstellung umfasst einige integrierte Konsistenzprüfungen, um sicherzustellen, dass mindestens eine Sicht ausgewählt wird und dass die Standardsicht in den ausgewählten Sichten enthalten ist.

Repository

- **In Repository einbinden**

Mit dieser Einstellung definieren Sie, welche Objekte in der Workbench-Sicht **Repository** angezeigt werden sollen, abhängig vom jeweiligen Speichertyp. Beispielsweise können Sie festlegen, dass Systemobjekte ausgeblendet werden, da Sie diese sowieso nicht ändern werden.

Weiterhin können Sie steuern, ob eine dedizierte Spalte für das Transportverhalten der einzelnen Objekte im Navigationsbereich angezeigt werden soll (**Transportspalte anzeigen**). Da das Transportverhalten eines Objekts jedoch abhängig ist von anwendungsweiten Einstellungen, können Sie diese Spalte auch ausblenden, um zusätzlichen Platz zur Anzeige von Objektnamen zu bieten.

Dasselbe gilt auch für die **Spalte Änderungsdatum anzeigen**. Wenn Sie sich dafür entscheiden, diese Spalte in der Baumstruktur des Repositorys anzuzeigen, geht das System wie folgt vor:

- Die Spalte **Änderungsdatum** wird zur Baumstruktur hinzugefügt und zeigt an, wann das Objekt letztmalig geändert wurde.
- Wenn Sie den Objektknoten erweitern, wird die zusätzliche Kategorie **Letzte Änderung (<Anzahl der geänderten Objekte>)** angezeigt. Diese Kategorie wird dann in verschiedene vordefinierte Periodenkategorien (z. B. **Heute**, **Letzte Woche** usw.) unterteilt und Sie sehen hier die abhängigen Objekte des auf oberster Ebene befindlichen Objekts, gruppiert nach Datum der letzten Änderung. Beachten Sie, dass die Periodennamen (letzte Woche, letzter Monat usw.) als Kennzeichen für die Länge der Periode zu verstehen sind und **nicht** als Kalenderdaten. Die Periodenkategorie **Letzte Woche** bedeutet beispielsweise "geändert in den letzten 7 Tagen" und **nicht** "geändert in der vorherigen Kalenderwoche".

- In der Dropdown-Liste *Zuletzt geänderte Objekte anzeigen* können Sie die Anzahl von Objekten festlegen, die Sie in der Kategorie *Letzte Änderung* anzeigen möchten. Wenn Sie diese Anzahl gering halten, können Sie die Systemperformance verbessern. Liegt jedoch die von Ihnen festgelegte Anzahl unter der Anzahl der Objekte, die in dieser Kategorie vorhanden sind, werden Sie vom System darüber informiert, indem die Anzeigeart zu *Letzte Änderung (<Anzahl der geänderten Objekte>/<Gesamtanzahl an Objekten>)* geändert wird.
- *Gelöschte Objekte*
BRFplus bietet eine mehrstufige Verdrängungsstrategie, die dazu verwendet wird, Objekte über einen festgelegten Zeitraum im System verfügbar zu machen, obwohl diese aus Unternehmenssicht als veraltet oder nicht mehr erforderlich erachtet werden. Mit dieser Einstellung entscheiden Sie, ob solche Objekte in der *Repository*-Sicht angezeigt werden.
- *Objekte in Anwendungen gruppieren*
Sie können festlegen, wie das System die Objekte, die zu einer Anwendung gehören, anzeigen soll. Sie können aus folgenden Alternativen wählen:
 - *Typen und Untertypen*: Sofern anwendbar, werden Objekte nach ihren Typen und Untertypen gruppiert. Mit dieser Anwendungseinstellung können Sie z. B. einen Eintrag Ausdrücke (= Objekttyp) finden, gefolgt von den verschiedenen Ausdruckstypen (= Untertyp), gefolgt von den Objekten dieses Typs. Diese Einstellung ist hilfreich für Anwendungen, die eine hohe Anzahl von Objekten enthalten, da sie eine bessere Übersicht bietet und die Systemperformance verbessert. Dies liegt daran, dass die Anzahl von Objekten, die über die Datenbank gelesen werden, reduziert wird. Auf der anderen Seite erfordert das Navigieren zu einem Objekt mehr Mausklicks und die zusätzlichen Baumstrukturen bergen das Risiko, dass Objektnamen abgeschnitten werden.
 - *Typen*: Diese Einstellung bietet eine Objektgruppierung auf einer Ebene und dient somit als Kompromiss für die Einstellungen *Typen und Untertypen* und *Flache Liste*.
 - *Flache Liste*: Alle Objekte einer Anwendung werden in alphabetischer Reihenfolge aufgeführt, unabhängig vom Objekttyp. Diese Einstellung eignet sich als Schnellübersicht für kleine Anwendungen. Wenn sie jedoch für Anwendungen verwendet wird, die eine hohe Anzahl von Objekten enthalten, kann diese Einstellung zu massiven Leistungseinschränkungen führen.

Kataloge

Mit dieser Einstellung definieren Sie, welche Objektkataloge in der *Katalog*-Sicht des Workbench-Navigationsbereichs verfügbar sind und welche standardmäßig angezeigt werden sollen.

Kataloge eignen sich, die Komplexität zu reduzieren und dem Benutzer die Unübersichtlichkeit zahlreicher Objekte zu ersparen, die möglicherweise im System vorhanden sind. Dieses Prinzip der reduzierten Komplexität wird zudem durch das Konzept der Katalogpersonalisierung erzwungen. Im Gegensatz zur *Repository*-Sicht, in der alle Objekte einer Anwendung angezeigt werden oder die Teil eines Suchergebnisses sind, werden **nicht** alle vorhandenen Kataloge in der *Katalog*-Sicht angezeigt. Vielmehr müssen Sie bei Verwendung der *Katalog*-Sicht eine der folgenden Optionen auswählen:

- Wählen Sie in der *Repository*-Sicht die Option *In Katalogsicht anzeigen* aus dem Kontextmenü eines Katalogobjekts.
- Konfigurieren Sie die Sicht im Dialogfenster *Personalisierung*. Diese Konfiguration erfolgt in zwei Schritten:
 1. Klicken Sie auf *Hinzufügen* und wählen Sie aus der Liste aller im System vorhandenen Kataloge diejenigen aus, die in der *Katalog*-Sicht verfügbar sein sollen.
 2. Legen Sie einen der im vorherigem Schritt ausgewählten Kataloge als Standardkatalog fest, indem Sie in die Spalte *Ist Standard* klicken.

Nachdem Sie die Konfiguration der Sicht abgeschlossen haben, können Sie die *Katalog*-Sicht verwenden. Wenn Sie die *Katalog*-Sicht öffnen, zeigt das System den Standardkatalog an. Alle anderen Kataloge, die

Sie für Ihre persönliche Katalogliste ausgewählt haben, sind über [Zu anderem Katalog wechseln](#) in der [Katalog](#)-Sicht verfügbar.

Hinweis



Es besteht die Möglichkeit, die oben beschriebenen vorbereitenden Schritte für Kataloge zu umgehen. Wählen Sie in der [Repository](#)-Sicht die Option [In Katalogsicht anzeigen](#) aus dem Kontextmenü eines Katalogobjekts. Das System wechselt von der [Repository](#)-Sicht zur [Katalog](#)-Sicht und der ausgewählte Katalog kann verwendet werden. Obwohl dieser Mechanismus für den täglichen Gebrauch nicht zu empfehlen ist, ist er dennoch hilfreich für Testzwecke, wenn Sie einen neuen Katalog definieren müssen und eine schnelle Übersicht über das Ergebnis wünschen.

Im Gegensatz zu allen anderen Einstellungen im Dialogfenster [Personalisierung](#) unterscheidet das System bei Katalogen **nicht** zwischen den Benutzermodi. Anders gesagt, es gibt nur einen Satz von Katalogeinstellungen, die sowohl für den [Einfachen Modus](#) als auch für den [Expertenmodus](#) wirksam sind.

Ausdruck

In diesem Register finden Sie zusätzliche Einstellungen für einige der komplexeren Objekttypen, die in BRFplus verfügbar sind.

Entscheidungstabelle

- [Tabelleneinstellungen anzeigen](#)
Wenn Sie eine Entscheidungstabelle (Ausdruck) bearbeiten, blendet das System zunächst das detaillierte Definitionsbild für Spalten- und Zeileneinstellungen aus. Wenn Sie diese Definition ändern möchten, klicken Sie zunächst auf die Schaltfläche , die ganz rechts auf der Symbolleiste im Bereich [Details](#) angezeigt wird. Mit der Einstellung [Tabelleneinstellungen anzeigen](#) definieren Sie, ob diese Schaltfläche angezeigt wird. Wenn Sie sich dafür entscheiden, die Schaltfläche  auszublenden, hat der Benutzer keinen Zugriff auf die Zeilen- und Spaltendefinition. Er kann jedoch trotzdem Regeln in der Tabelle anlegen und pflegen. Diese Einstellung ist hilfreich, um äußerst komplexe Entscheidungstabellen vor ungewollten Änderungen zu schützen.
- [Zeilennummern anzeigen](#)
Diese Einstellung legt fest, ob das System eine zusätzliche Spalte für die Zeilennummern anzeigt. Die Anzeige von Zeilennummern ist hilfreich, um sich in Tabellen mit vielen Zeilen zurechtzufinden. Diskutieren mehrere Mitarbeiter über bestimmte Tabelleninhalte ist diese Einstellung sogar unerlässlich. Andererseits benötigt die Anzeige der Spalte Zeilennummern viel Platz, der ansonsten für Datenspalten verfügbar wäre.

Hinweis

Wenn Sie sich dafür entscheiden, dass die Spalte mit den Zeilennummern angezeigt wird, können Sie diese Spalte per Drag&Drop an eine beliebige Position verschieben. Es ist **nicht** vorgesehen, dass in allen Fällen Spaltennummer 1 belegt wird. Wenn Sie die Zeile weiter nach rechts ziehen, erhalten Sie mehr Platz für Datenspalten und Sie haben immer noch die Möglichkeit, die aktuelle Zeilennummer zu suchen.

- [Anzahl sichtbarer Zeilen](#)
Mit dieser Einstellung schränken Sie die Anzahl der Zeilen ein, die gleichzeitig für eine Entscheidungstabelle angezeigt werden. Dies ist hilfreich für umfangreiche Entscheidungstabellen mit mehreren Tausend Zeilen, die andernfalls zu deutlichen Leistungseinbußen führen würden. Natürlich können Sie nach unten blättern, um die anderen Zeilen anzuzeigen. Das System liest dann die erforderlichen Tabellenzeilen aus der Datenbank und zeigt sie entsprechend an.

- **Mindestanzahl an Spalten für horizontale Blätterleiste**

Mit dieser Einstellung definieren Sie die geringste Anzahl von Spalten in einer Entscheidungstabelle, die das System in einer horizontalen Blätterleiste anzeigen kann. Tabellen mit komplexer Struktur und vielen Feldern haben häufig einen breiten horizontalen Blätterbereich, sodass der Benutzer keine vollständige Übersicht über die gesamte Tabellenzeile erhält. Wenn Sie diese Einstellung erhöhen, reduziert das System die Breite aller Spalten, damit die vorgegebene Anzahl von Spalten angezeigt werden kann. Nur wenn die Anzahl der Spalten nach wie vor diesen Grenzwert überschreitet, zeigt das System eine Blätterleiste an. Wenn Sie die Anzahl der Spalten erhöhen, wird die Spaltenbreite reduziert. Dies führt ggf. zwar dazu, dass Zelleninhalte abgeschnitten werden (obwohl sie über den Zellen-tooltip immer noch verfügbar sind), dennoch profitieren Sie von einer besseren Übersicht über die Tabelleninhalte.

i Hinweis

Das System berücksichtigt für die Berechnung der Spaltenbreite alle Tabellenspalten, die aktuell angezeigt werden. Dies umfasst sowohl Ergebnisdatenspalten als auch Spalten mit Zeilennummern, die nach Bedarf angezeigt werden. Spalten, die in den Tabelleneinstellungen ausgeblendet wurden, werden hingegen ignoriert.

Regelsatz

- **Anzahl an erweiterten Regeln**

Mit dieser Einstellung schränken Sie die Anzahl der Regeln ein, die für einen erweiterten Regelsatz angezeigt werden. Dies ist hilfreich für umfangreiche Regelsätze mit Dutzenden von Regeln, die andernfalls zu deutlichen Leistungseinbußen führen würden. Die Regeln, die den hier eingegebenen Wert betreffen, werden anfangs mit nur einer Zeile pro Regel angezeigt. Sie müssen diese manuell erweitern, um Detailinformationen anzuzeigen.

- **Kopfzeile des Regelsatzes anzeigen**

Mit dieser Einstellung steuern Sie, ob ein Regelsatzobjekt, das in die Workbench geladen wird, im erweiterten Kopfdatenbereich für Regelsätze angezeigt wird oder nicht. Die Kopfdaten enthalten administrative Daten, die den Regelsatz beschreiben, sowie zusätzliche Informationen über die Regelsatzvariablen. Je nach Komplexität des Regelsatzes, benötigt insbesondere die Auflistung von Variablen viel Platz auf dem Bildschirm. Daher ist es ggf. besser, diese Option zu deaktivieren, damit der verfügbare Platz zur Anzeige von Regeln genutzt werden kann, ohne dass ein Blättern erforderlich wird. Diese Option ist standardmäßig **nicht** aktiviert.
Anzeigen

Filter

Auf dieser Registerkarte können Sie die Zuordnung von Objektfiltern zu Anwendungen pflegen. Mit der Schaltfläche **Hinzufügen** können Sie eine neue Filterzuordnung anlegen. Für jede Zuordnung können Sie wählen, ob der Filter für alle Anwendungen im System angewandt werden soll. Falls nein, können Sie zusätzliche Zeilen pro Filter anlegen, um die einzelnen Anwendungen festzulegen, für die der jeweilige Filter angewendet werden soll. Wenn Sie die Einstellung **Alle Anwendungen** von **Nein** zu **Ja** ändern, gehen alle einzelnen Zuordnungen, die bereits definiert wurden, verloren.

i Hinweis

Die Pflege von Filterzuordnungen ist eine administrative Aufgabe und sollte üblicherweise systemweit und konsistent gehandhabt werden. Diese Vorgehensweise ist besser als individuelle Präferenzen von einzelnen Benutzern. Dieses Register dient hauptsächlich Informationszwecken und wird den meisten Benutzern nur im Lesemodus angezeigt.

Hotkeys

Auf dieser Registerkarte finden Sie eine vordefinierte Liste mit Benutzeraktionen, die üblicherweise für alle Arten von BRFplus-Objekttypen verwendet werden, z. B. Sichern, Löschen, Überprüfen usw. Für alle diese Aktionen können Sie einen individuellen Hotkey aus einer Liste der möglichen Hotkeys festlegen, die folgende Bereiche umfassen:

- Strg + 0 ... Strg + 9
- Strg + A ... Strg + Z
- Strg + F2 ... Strg + F12

Hinweis

Falls eine Benutzeraktion, die mit einem Hotkey verknüpft ist, nicht verfügbar ist (z. B. ist die Schaltfläche [Zurück](#) deaktiviert, wenn noch kein vorheriger Navigationsschritt vorhanden ist), wird der Hotkey an den Webbrowser weitergegeben und löst ggf. eine Aktion aus, die vom Browser mit diesem Hotkey verknüpft wurde. Die Aktion [Zurück](#) ist z. B. standardmäßig mit dem Hotkey Strg + B verknüpft. Ist die Schaltfläche [Zurück](#) deaktiviert, löst die Kombination Strg + B stattdessen die im Microsoft Internet Explorer definierte Funktion [Favoriten verwalten](#) aus, oder es wird in Mozilla Firefox die Seitenleiste [Lesezeichen](#) geöffnet.

Im Gegensatz zu dem zuvor genannten Verhalten wird der Hotkey Strg + F4 direkt vom Betriebssystem ausgeführt. Wenn Sie diese Taste drücken, wird das aktuelle Browserregister sofort geschlossen, ohne dass der Benutzer eingreifen kann. Da BRFplus dieses Verhalten nicht überschreiben kann, wird der Hotkey Strg + F4 **nicht** in der Liste der zuordenbaren Hotkeys angeboten.

1.1.3 Typischer Workflow

Verwendung

Die nachstehenden Schritte beschreiben einen typischen Workflow zum Integrieren einer BRFplus-Funktion in Ihre Anwendung.

Hinweis

In der folgenden Übersicht wird der Begriff „Anwendung“ sowohl im ersten als auch im letzten Schritt verwendet, hat aber zwei unterschiedliche Bedeutungen:

- Im ersten Schritt [Anwendung anlegen](#) bezieht sich „Anwendung“ auf die BRFplus-Elementart „Anwendung“, welche als übergeordnetes Objekt zum Sammeln aller Objekte dient, mit denen die Geschäftslogik modelliert wird, die Sie in Ihre Lösung implementieren möchten.
- Im letzten Schritt [In Anwendung integrieren](#) bezieht sich „Anwendung“ auf Ihre Lösung, die Sie um die in BRFplus-Regeln gekapselte Geschäftslogik erweitern möchten.

Vorgehensweise

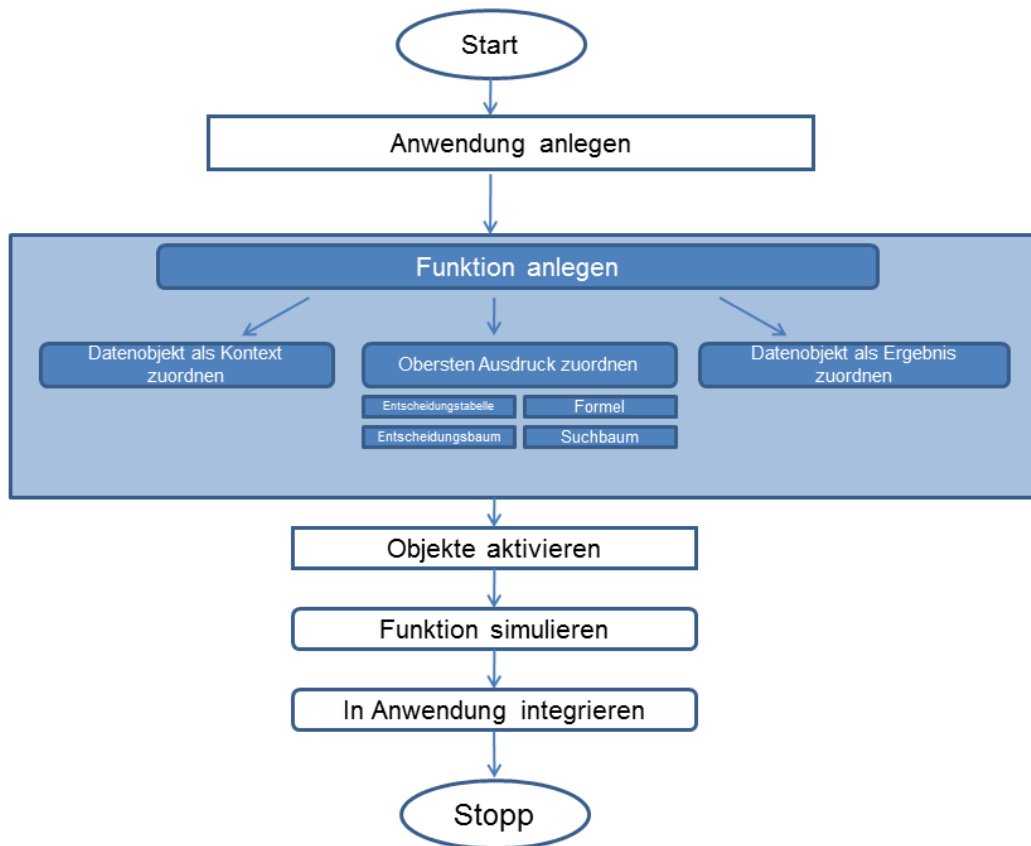


Abbildung 2: Workflow für Geschäftsregeln

1. Legen Sie eine Anwendung an.
Weitere Informationen finden Sie unter [Anwendung anlegen \[Seite 186\]](#).
2. Legen Sie eine Funktion an.
Weitere Informationen finden Sie unter [Funktion anlegen \[Seite 191\]](#).
 - Weisen Sie der Funktion Kontext und Ergebnis zu.
Weitere Informationen finden Sie unter [Einer Funktion Kontext und Ergebnis zuweisen \[Seite 193\]](#).
 - Weisen Sie der Funktion einen obersten Ausdruck zu.
Weitere Informationen finden Sie unter [Einer Funktion einen obersten Ausdruck zuweisen \[Seite 194\]](#).
3. Aktivieren Sie die Objekte.
4. Simulieren Sie die Funktion.
Weitere Informationen finden Sie unter [Funktion simulieren \[Seite 270\]](#).
5. Integrieren Sie die BRFplus-Funktion in Ihre Anwendung.

Weitere Informationen


[Kurzanleitung-Tutorial \[Seite 29\]](#)

1.1.4 Workflow for Analytical Functions

Use BRFplus to create analytical functions with rules that are based on persisted data in CDS Views. These analytical functions use a database context for the query and a table object for the results.

Prerequisites

To use all the functionality described in this workflow, the following landscape is required:

- SAP S/4HANA Cloud 1705
- SAP Decision Service Management 1.0 SP04. For information about DSM installation, see the DSM Installation Guide.
- SAP HANA Rules Framework 1.0 SP10 or higher. For more information, see [SAP HANA Rules Framework Installation Guide](#) .

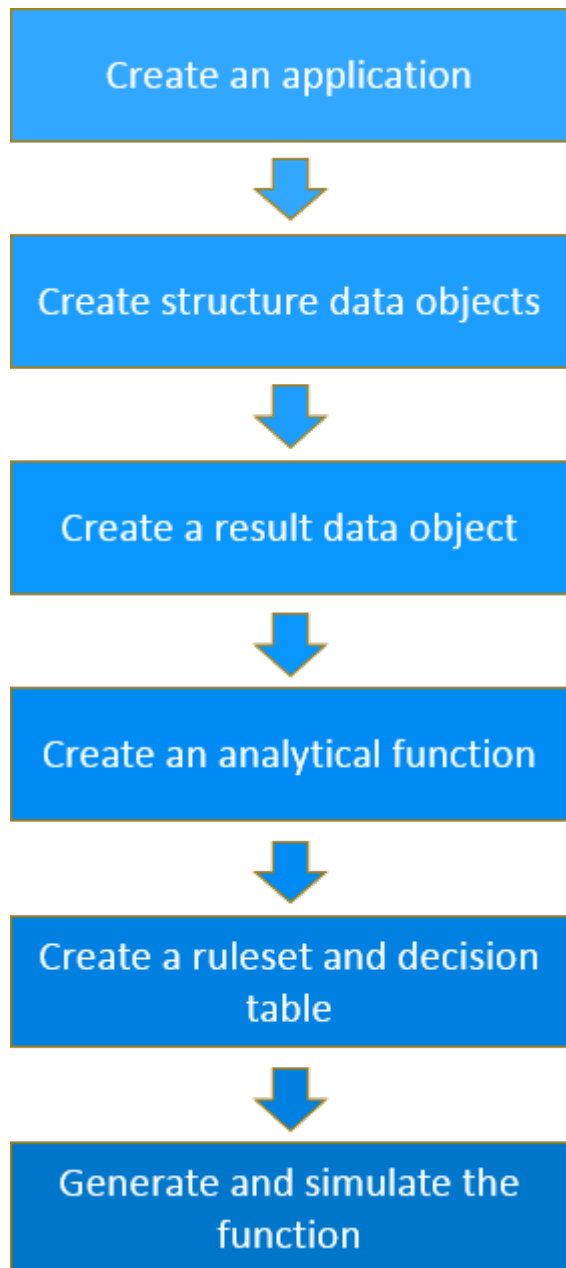
Note

For information about available functionality when working with earlier SP releases of SAP NetWeaver 7.51 or SAP S/4HANA, and SAP HANA Rules Framework, see [Landscapes for Working with Analytical Rules \[page 27\]](#).

For information about the setup configuration that is required to work with analytical functions, see [Setup for Working with Analytical Mode \[page 25\]](#).

Before you begin to create analytical functions ensure that you can connect to the CDS Views that you want to use.

Workflow



- [#unique_8/unique_8_Connect_42_subsection-im1 \[page 23\]](#)
- [#unique_8/unique_8_Connect_42_subsection-im2 \[page 23\]](#)
- [#unique_8/unique_8_Connect_42_subsection-im3 \[page 23\]](#)
- [#unique_8/unique_8_Connect_42_subsection-im4 \[page 23\]](#)
- [#unique_8/unique_8_Connect_42_subsection-im5 \[page 24\]](#)
- [#unique_8/unique_8_Connect_42_subsection-im6 \[page 24\]](#)

Click each tile for more information.

Create an Application

A BRFplus application serves as a super-ordinate object for collecting all objects used to model the business logic that you want to integrate into your solution.

Create an application and choose [Create and navigate to object](#).

For more information, see [Creating an Application \[page 186\]](#).

Create Structure Data Objects

In the application, create structure data objects.

- To use persisted data, bind the structure to a CDS View using the [Bind to CDS View](#) option.
- When you bind to a CDS view, BRFplus creates an element for each attribute in the CDS View based on the attribute definition. These elements can be used in rules in BRFplus. You do not need to add more data objects to the structure.
- In some cases, you can refer the element to a value list that was provided by an OData service.
- When the CDS view contains parameters, all the parameters are added to the data object.
- It is recommended to maintain descriptions in all your data object definitions. The descriptions are used in the rule expression language to enable user-friendly language in rules.

For more information, see:

- [Creating an Element Data Object \[page 236\]](#)
- [Struktur-Datenobjekt anlegen \[page 241\]](#)

Create a Result Data Object

In the application, create a result data object. You can create a structure data object or a table data object for the results.

You do not need to define binding for the result object. The result object can use existing elements or you can add data objects, if required.

Alternatively, you can create and add the result data object when you create the function.

For more information, see [Ein Tabellen-Datenobjekt anlegen \[page 242\]](#).

Create an Analytical Function

Create the function and assign context and result to the function.

1. Create a new function and choose [Create and navigate to object](#).
2. Set the function properties.
In the [Mode](#) field, choose [Analytical Mode](#) to define an analytical function, which will contain rules based on persisted data,.
3. Add the context for the analytical function.
 - In the [Database Context](#) tab, add the required structure data objects that you created for the analytical function.
Only data objects that are bound to a CDS view can be added to the database context.
When the data object is bound to a CDS view that contains parameters, the parameters are added to the [Signature](#) tab.
 - You can add data objects from other applications subject to access levels.
4. Assign a result for the function.
 - In the [Signature](#) tab, assign the result data object.

The result data object for an analytical function must be a table as it collects the results of all the assigned decision tables. In the [Result Data Object](#) section, choose the required option from the [Data Object](#) drop-down list.

- You can add data objects from other applications subject to access levels.

5. Save and activate the function.

For more information, see [Eine Funktion anlegen \[page 191\]](#).

Create a Ruleset and Decision Table

1. In the [Assigned Rulesets](#) tab, create a new ruleset without rules.
2. Add a decision table to the ruleset.
 1. Open the ruleset
 2. In the [Decision Tables](#) section, choose [Create](#) from the drop-down list.
 3. Define a decision table and choose [Create and navigate to object](#).

The decision table for an analytical function is defined in an embedded SAPUI5 Rule Builder control, based on OData services and OData interfaces.
3. Define the decision table's settings.
 1. Choose [Settings](#).
 2. Choose the hit policy - [First Match](#) or [All Matches](#).
 3. Define the conditions columns of the decision table by adding attributes from the vocabulary and optionally, an operator. Each row in the settings defines a column in the decision table. Choose [+](#) to add a row.

The auto-complete functionality filters your choices as you enter values.

The decision table automatically uses the result table (also known as signature) that was defined for the function.
 4. Click [Apply](#).
4. Define conditions in the decision table.
 1. In each row, set values for the condition columns and the result columns.
 2. Save and activate the decision table.
5. Activate the ruleset.

➔ Tip

Alternative ways to define and edit a decision table:

- Create an SAP Fiori app and embed the Rule Builder control in which you can define and edit the rules.
- Use OData APIs for analytical rules. For more information, see .

For information about using the SAP rule expression language in decision tables, see [Rule Expression Language in Decision Tables](#) in the *SAP Rule Expression Language Guide*.

Generate and Simulate the Function

1. In the function's [Code Generation](#) tab, choose [Generate Function](#) [Without Trace](#).
2. Simulate the function:
 1. Choose [Simulation](#).
 2. Choose [Continue](#).
 3. Choose [Execute](#).

For more information, see [Simulating a Function \[page 270\]](#).

1.1.4.1 Setup for Working with Analytical Mode

You can work with analytical mode in BRFplus to perform a business rules evaluation of a large set of data persisted in a SAP HANA database.

Prerequisites

- SAP NetWeaver 7.51 SP00 and higher, or SAP NetWeaver 7.65 and higher
- SAP Decision Service Management 1.0 SP04
- SAP HANA Rules Framework 1.0 SP08 or higher. For more information, see the relevant version of the *SAP HANA Rules Framework Installation Guide* on the SAP Help Portal at https://help.sap.com/viewer/p/SAP_HANA_RULES_FRAMEWORK.

Context

The following steps are required before you can use analytical rules in BRFplus workbench.

Setup for Using BRFplus with SAP HANA Rules Framework

Procedure

1. Create a SAP HANA user.
2. Assign the role `HRF_Tech_Role_ABAP_ONLY.hdbrole` to the SAP HANA user.
3. Create an RFC connection to the SAP HANA XS server using transaction code SM59.
4. Create a new entry under `HTTP Connections to External Server` with the following settings.

Table 1:

Setting	Tab	Description
RFC Destination		Name of the RFC destination to the external server
Target Host	Technical Settings	SAP HANA XS server host name
Service No.	Technical Settings	Service number in the format 80<INSTANCE>, for example, 8000
User Name	Logon & Security	SAP HANA user that you created

Setting	Tab	Description
Password	Logon & Security	SAP HANA user password
Status of Secure Protocol	Logon & Security	Security settings

5. Create and run a report to register the RFC connection to BRFplus that you created. (You can use the following code sample.)

Sample Code

```
"set connection for client
  cl_fdt_setting=>set_xs_compiler_rfc_conn(
    EXPORTING
      iv_rfc_connection = '<RFC Connection Name>'    " Logical Destination
      (Specified in Function Call)
      iv_client         = '<client number>' ).
```

6. Test the connection by running transaction FDT_TEST_HRF_CMP_CON.

On successful setup, the following message is displayed:

```
HRF server communication using RFC destination <RFC_name> is valid.
```

Activating OData Services

Context

These steps are only relevant for on-premise installations and should be performed in the back-end server.

Procedure

1. Activate the Rule OData service.
 - a. Start transaction /IWFND/MAINT_SERVICE.
 - b. Choose [Add Service](#) and set the following parameters:
 - *System Alias* - **LOCAL**, or the name of your system alias
 - *Technical Service Name* - **RULE_SRV**
 - *External Service Name* - **RULE_SRV**
 - c. Choose [Get Services](#).
 - d. In the [Select Backend Services](#) list, select the service you just created and choose [Add Selected Services](#).
 - e. In the [Add Service](#) dialog, set *Package Assignment* as **\$TMP** or choose [Local Object](#).
 - f. Choose [Confirm](#).
2. Activate the Vocabulary OData service.

-
- a. Start transaction /IWFND/MAINT_SERVICE.
 - b. Choose *Add Service* and set the following parameters:
 - *System Alias* - **LOCAL**, or the name of your system alias
 - *Technical Service Name* - **VOCABULARY_SRV**
 - *External Service Name* - **VOCABULARY_SRV**
 - c. Choose *Get Services*.
 - d. In the *Select Backend Services* list, select the service you just created and choose *Add Selected Services*.
 - e. In the *Add Service* dialog, set *Package Assignment* as **\$TMP** or choose *Local Object*.
 - f. Choose *Confirm*.

Weitere Informationen

[Workflow for Analytical Functions \[page 21\]](#)

1.1.4.2 Landscapes for Working with Analytical Rules

This table describes the landscapes required to work with analytical rules and the changes in documentation for updated functionality for each landscape.

Table 2:

Landscape	Changes
<ul style="list-style-type: none"> • S/4HANA Cloud 1705 • SAP Decision Service Management 1.0 SP04 • SAP HANA Rules Framework 1.0 SP10 	<p>May 5, 2017</p> <p>The following new functionality is available when working with analytical functions:</p> <ul style="list-style-type: none"> • When a structure is bound to a CDS view, you can refer to existing value help lists and you can create a new list of values using OData services. • The descriptions that are defined in your data objects are used in the rule expression language to enable user-friendly language in rules. It is recommended to maintain descriptions in all your data object definitions. • When a data object is bound to a CDS view that contains parameters, all the parameters are added to the data object. • Additional OData APIs for managing rules in an analytical workflow. <p>For more information, see:</p> <ul style="list-style-type: none"> • Workflow for Analytical Functions [page 21] • Data Binding [page 66] • Creating an Element Data Object [page 236]
<ul style="list-style-type: none"> • SAP NetWeaver 7.51 SP01 or SP02, or S/4HANA Cloud 1611 • SAP Decision Service Management 1.0 SP04 • SAP HANA Rules Framework 1.0 SP09 	<p>January 30, 2017</p> <p>The following new functionality is available:</p> <p>When working with analytical functions, you can add data objects from other applications subject to access levels. See Workflow for Analytical Functions [page 21].</p>
<ul style="list-style-type: none"> • SAP NetWeaver 7.51 SP00 or S/4HANA Cloud 1608 • SAP Decision Service Management 1.0 SP04 • SAP HANA Rules Framework 1.0 SP08 	<p>November 14, 2016</p> <p>The following new options are now available when working with analytical functions:</p> <p>Bind to CDS View:</p> <ul style="list-style-type: none"> • Struktur-Datenobjekt anlegen [page 241] • Data Binding [page 66] <p>Analytical mode of operation:</p> <ul style="list-style-type: none"> • Eigenschaften festlegen [page 192] • Operationsmodi [page 151] <p>A new section was added about working with analytical functions. See Workflow for Analytical Functions [page 21].</p>

1.1.5 Quick Start Tutorial

Use

This section gives access to a tutorial that will help you learn to use BRFplus. The tutorial gives a quick introduction to the different features so that you can learn the basics of working with BRFplus objects and how to model applications. At the same time, the tutorial highlights the integration of BRFplus with SAP Decision Service Management, a separately licensed add-on that extends the functional capabilities of BRFplus and makes business rules management easy even in complex and distributed system landscapes.

In the tutorial, we show how to set up a managed system and assign a sample application to it (exercise 1). This application derives the price of a product (exercises 2 and 3), adds shipping costs of 5% (but at least \$2 if the price is under \$100) (exercise 4), and sends a notification mail if a product is not found in the price list (exercise 4). Then we simulate the application on the managed system (exercise 5) before we deploy it and call it from the managed system (exercise 6). Finally, we make a delta deployment using an activation time stamp in the future (exercise 7). We offer this tutorial in two different flavors on the SAP Community Network (SCN):

- [Script \(with many screenshots\)](#) 📄
- [Video](#) 📺

Note

SAP Decision Service Management is an add-on to SAP NetWeaver and has to be licensed separately. It is **not** included in a standard SAP NetWeaver license.

1.2 Konzepte

Verwendung

In diesem Abschnitt werden die grundlegenden Konzepte für das Definieren und Verarbeiten von Regeln erläutert. Es werden unter anderem folgende Themen behandelt:

- [Anwendungen \[Seite 30\]](#)
- [Ablagetypen \[Seite 35\]](#)
- [Namenskonventionen für BRFplus \[Seite 36\]](#)
- [Verwaltungsdaten \[Seite 39\]](#)
- [Kontext und Ergebnis \[Seite 46\]](#)
- [Datenobjekte und Datenobjekttypen \[Seite 61\]](#)
- [Bindung \[Seite 66\]](#)
- [Vergleichsoperationen \[Seite 70\]](#)
- [Ausdrücke \[Seite 80\]](#)
- [Aktionstypen \[Seite 132\]](#)
- [Funktion \[Seite 146\]](#)

- [Operationsmodi \[Seite 132\]](#)
- [Regel \[Seite 167\]](#)
- [Regelsatz \[Seite 168\]](#)
- [Katalog \[Seite 171\]](#)
- [Objektfilter \[Seite 174\]](#)
- [Versionierung \[Seite 176\]](#)
- [Objekte anlegen \[Seite 48\]](#)
- [Objekte löschen \[Seite 57\]](#)
- [Export und Import von XML-Daten \[Seite 274\]](#)

1.2.1 Anwendung

Verwendung

Ein BRFplus-Anwendungsobjekt ist ein Container für verschiedene BRFplus-Objekte. Eine BRFplus-Anwendung spiegelt prinzipiell eine bestimmte Systemfunktionalität wider, die Sie mithilfe einiger mit BRFplus modellierter Geschäftsregeln verbessern möchten. In diesem Fall ist es empfehlenswert, alle BRFplus-Objekte, die zu dieser gegebenen Systemfunktionalität beitragen, in einer BRFplus-Anwendung zusammenzufassen.

Funktionsumfang

Objekthierarchie und Anwendungszuordnung

Der Anwendungsobjekttyp befindet sich auf der höchsten Ebene der Klassifizierungshierarchie für Objekte. Nichtsdestotrotz können Sie beliebig viele Anwendungen definieren. Alle Objekte müssen einer Anwendung zugeordnet sein. Das System bietet Ihnen hierzu zwei Möglichkeiten an, ein Objekt einer Anwendung zuzuordnen:

- Sie können neue Objekte über den Kontext einer gegebenen Anwendung anlegen. Wählen Sie hierzu in den Details der Anwendung auf der Registerkarte [Enthaltene Objekte](#) die Drucktaste [Objekt anlegen](#). Alle auf diese Weise angelegten Objekte werden automatisch der Anwendung zugeordnet, in der sie angelegt wurden.
- Sie können neue Objekte auch über die Navigationsleiste der BRFplus-Workbench anlegen. Hierzu verwenden Sie das Kontextmenü eines der bereits vorhandenen Objekte. Das Objekt wird dann automatisch derselben Anwendung zugeordnet wie das Objekt, von dem aus Sie das Kontextmenü aufgerufen haben.

Hinweis

Wenn in einem System kein BRFplus-Objekt als Ausgangspunkt vorhanden ist, können Sie in der BRFplus-Workbench selbst eine Anwendung anlegen. Wählen Sie hierzu ► [Workbench](#) ► [Anwendung anlegen](#) ►.

Ablagetypen

Eine Anwendung kann einen der folgenden drei Ablagetypen aufweisen:

- System (Standard)
- Stammdaten
- Customizing

Der Ablagetyp bestimmt das Verhalten der Anwendung bei Systemtransporten. System- und Customizing-Anwendungen können lokal sein, Stammdatenanwendungen sind immer lokal. Der Ablagetyp sowie die Lokalitätseinstellung einer Anwendung werden von allen Objekten geerbt, die dieser Anwendung zugeordnet sind. Alle drei Anwendungsarten können einem Entwicklungspaket zugeordnet werden.

Von SAP ausgelieferte Customizing-Anwendungen werden immer auf Kundensystemen im Mandanten 000 installiert. Sie können eine solche Anwendung in einen Mandanten kopieren, der für die Entwicklung verwendet wird, indem Sie einen Customizing-Transportauftrag anlegen. Weitere Informationen finden Sie in SAP Hinweis [1908227](#).

Hinweis

Der Ablagetyp muss beim Anlegen einer Anwendung festgelegt werden. Diese Einstellung kann anschließend nicht mehr geändert werden.

Versionierung

Wie für alle Objekttypen können Sie auch für Anwendungen entscheiden, ob das System frühere Versionen der Anwendung in der Datenbank behalten soll, nachdem Sie geändert wurde. Außerdem können Sie eine Standard-Versionierungseinstellung für neu angelegte Objekte in der Liste [Enthaltene Objekte](#) festlegen. Neu angelegte Objekte erben die auf Anwendungsebene definierte Versionierungseinstellung.

Hinweis

Diese geerbte Versionierungseinstellung ist lediglich eine Standardeinstellung und kann zu jeder Zeit für jedes Objekt geändert werden.

Beachten Sie, dass die Standard-Versionierungseinstellung von Objekten, die der Anwendung zugeordnet sind, das Versionierungsverhalten der Anwendung selbst **nicht** beeinflussen. Die Versionierungseinstellung für ein Anwendungsobjekt wird auf die gleiche Art gesteuert wie für alle anderen Objekttypen.

Hinweis

Unter bestimmten Umständen kann die Verwendung von Objektversionen potenzielle „Nebenwirkungen“ haben, die zu schwer behebbaren Fehlern führen können. Aus diesem Grund hat SAP entschieden, die Standard-Versionierungsmodi [Versionierung ein](#) und [Versionierung erzwungen](#) als „nicht empfohlen“ zu definieren. Weitere Informationen finden Sie unter [Versionierung: Vor- und Nachteile \[Seite 182\]](#).

Weitere Einstellungen

Eine Anwendung stellt auch Attribute für die Anwendungskomponente und die Softwarekomponente bereit. Diese müssen mit der Anwendungs- und Softwarekomponente übereinstimmen, die für das Entwicklungssystem definiert wurden. Die Anwendungs- und Softwarekomponente werden automatisch vom Entwicklungspaket übernommen, wenn sie nicht explizit gesetzt werden.

Mithilfe der Zugriffsebene können Sie in einer Anwendung auch die Wiederverwendbarkeit, Änderbarkeit und Sichtbarkeit steuern.

➔ Empfehlung

Wir empfehlen Ihnen, für die Definition des Entwicklungspakets, der Software- und der Anwendungskomponente dieselben Werte zu verwenden wie für die Softwarelösung, die Sie durch die BRFplus-Anwendung erweitern möchten. Dies erleichtert alle Aktivitäten in Bezug auf die Softwarestruktur, vor allem Transporte.

Einschränkungen und Abhängigkeiten

Die Einstellungen für das Entwicklungspaket, die Anwendungs- und die Softwarekomponente einer Anwendung (und die dazugehörigen Objekte) hängen teilweise voneinander ab. Dabei gibt es folgende Einschränkungen:

- Das der Anwendung zugehörige Entwicklungspaket bestimmt, ob sie transportierbar oder lokal ist.
- Die Definition einer Anwendung als transportierbar oder lokal lässt sich nicht mehr rückgängig machen. Wenn eine Anwendung zum ersten Mal gesichert wurde, kann diese Einstellung nicht mehr geändert werden. Sie können eine lokale Anwendung keinem transportierbaren Paket zuordnen und umgekehrt. Bei einem entsprechenden Versuch wird eine Fehlermeldung ausgegeben.
- Für eine lokale Anwendung ist es nicht notwendig, eine Softwarekomponente festzulegen. Umgekehrt ist es für eine transportierbare Anwendung zwingend erforderlich, eine Softwarekomponente festzulegen, da das SAP-Transportsystem auf diesem Attribut basiert.

Klasse für Anwendungs-Exits

Wenn Sie für Ihre Anwendung besondere Verarbeitungsroutinen (beispielsweise für Hilfsberechnungen, Validierungen usw.) benötigen, können Sie diese zusätzlichen Funktionalitäten über Methoden einer ABAP-Klasse implementieren. Diese Methoden können Sie anschließend verwenden, um beispielsweise die Funktionalität eines Formelausdrucks zu erweitern, indem Sie zusätzliche Formelfunktionen einführen. Die entsprechende Exit-Klasse muss für die jeweilige Anwendung registriert sein. Tragen Sie hierzu den Namen der Klasse in das Feld *Klasse für Anwendungs-Exits* ein.

Folgendes ist bei Anwendungs-Exit-Klassen zu beachten:

- Sie können nur eine Exit-Klasse pro BRFplus-Anwendung definieren.
- Eine Anwendungs-Exit-Klasse muss das Interface **IF_FDT_APPLICATION_SETTINGS** implementieren.
- Sobald Sie die Funktionalitäten einer Exit-Klasse verwenden, empfehlen wir Ihnen, die Zuordnung zu dieser Exit-Klasse **nicht** mehr zu ändern. Sollten Sie Änderungen an der Funktionalität vornehmen wollen, versuchen Sie diese innerhalb der bereits zugeordneten Klasse zu implementieren.

i Hinweis

Wenn Sie die Zuordnung einer Exit-Klasse zu einer Anwendung dennoch ändern oder löschen möchten, müssen Sie manuell überprüfen, ob die implementierte Funktionalität bereits verwendet wurde. Für jede Verwendung müssen Sie entweder eine Behelfslösung bereit stellen oder die Verwendung rückgängig machen. Beachten Sie, dass es unter Umständen schwierig ist, solche Verwendungen zu finden (beispielsweise, wenn ein Formelausdruck eine Funktion verwendet, die früher von der zu ändernden Exit-Klasse bereitgestellt wurde).

Objekttypspezifische Einstellungen

Während alle bisher genannten Einstellungen direkt die Anwendung selbst betreffen, gibt es eine Reihe an Einstellungen, mit deren Hilfe Sie die Eigenschaften und das Verhalten von bestimmten Objekttypen steuern

können, die innerhalb des Anwendungsumfangs verwendet werden. Folgende Einstellungen können Sie auf der Registerkarte *Sonstiges* vornehmen:

Tabelle 3:

Einstellung	Kommentar
<i>Wiederaufnahme von Regelsatz aktiviert</i>	<p>Regelsätze können in zwei verschiedenen Modi verarbeitet werden:</p> <ul style="list-style-type: none"> • Standardmodus: Alle Regeln innerhalb eines Regelsatzes werden nacheinander ausgeführt. Die Verarbeitung wird beendet, nachdem die letzte Regel in einem Regelsatz verarbeitet wurde oder wenn eine Exit-Bedingung erfüllt ist. • Verzögerter Modus: Wie der Standardmodus, allerdings gibt es hier die Möglichkeit, die Regelverarbeitung an dem Punkt fortzusetzen, an dem sie aufgrund einer Exit-Bedingung beendet wurde. <p>Wenn Sie diese Einstellung ausgewählt haben, wird die Regelverarbeitung im verzögerten Modus unterstützt.</p>
<i>Sekundäre Datenbankverbindung</i>	<p>Ausdrücke der Art <i>Prozeduraufruf</i> (Aufruftyp <i>Datenbankprozedur</i>) und <i>dynamischer Datenbank-View</i> können nicht nur auf Datenbankobjekte angewendet werden, die sich in der lokalen Datenbank befinden, sondern auch auf alternative Datenbanken, die sich in einem anderen System befinden, oder sogar auf eine eigenständige Datenbank, die nicht Teil eines SAP-Systems ist. Mit dieser Einstellung können Sie eine Datenbank aus der Liste von Datenbankverbindungen auswählen, die im lokalen System definiert wurden. Das System stellt alle Einträge bereit, die in der Backend-Transaktion DBCO gepflegt wurden. Wenn für eine Anwendung eine sekundäre Datenbank gepflegt wurde, verweisen die Ausdrücke <i>Prozeduraufruf</i> und <i>dynamischer Datenbank-View</i> in dieser Anwendung automatisch auf die sekundäre Datenbank.</p>

Standardeinstellungen für neue Objekte

Alle BRFplus-Objekte haben einige administrative Datenfelder gemeinsam. Es ist wahrscheinlich, dass die Mehrzahl der Objekte in einer Anwendung in manchen dieser Felder jeweils dieselben Inhalte hat. Aus diesem Grund können Sie für diese Felder auf Anwendungsebene Standardwerte festlegen, die für jedes neue Objekt übernommen werden. Diese Werte können allerdings auch wieder geändert werden.

Tabelle 4:

Einstellung	Kommentar
<i>Anwendungsprotokollobjekt, Anwendungsprotokoll-Unterobjekt</i>	Legt die Anwendungsprotokollobjekte fest, die für das Aufzeichnen von Protokolleinträgen von Objekten dieser Anwendung verwendet werden (zum Beispiel Aktionen der Art <i>Nachricht protokollieren</i> . Weitere Informationen finden Sie unter <i>Nachricht protokollieren (Aktion)</i> [Seite 133].
<i>Protokoll sichern</i>	Legt fest, ob Protokolldaten permanent in der Datenbank gespeichert werden oder nicht. Wenn die Protokolldaten nicht in der Datenbank gespeichert werden, werden sie nur zur Laufzeit im Speicher behalten und gehen nach der Sitzung verloren.
<i>Standarderzwingung</i>	Legt fest, ob und bis zu welchem Grad die Objekte einer Anwendung den anwendungsweiten Standardeinstellungen zum Anwendungsprotokoll folgen.
<i>Zulässige Meldungstypen</i>	Legt die Meldungstypen fest, die Sie in einer Anwendung verwenden können, wenn Sie eine Aktion <i>Nachricht protokollieren</i> anlegen.
<i>Versionierungsmodus</i>	<p>Legt fest, ob neu angelegte Objekte standardmäßig versioniert werden oder nicht.</p> <div> <p>i Hinweis</p> <p>Diese Einstellung betrifft alle Objekttypen außer Katalogen. Kataloge werden immer mit ausgeschalteter Versionierung angelegt.</p> </div>
<i>Spracheinstellungen</i>	Legt fest, ob Texte und Dokumentationen neu angelegter Objekte abhängig von Sprache, Version, Sprache und Version, oder weder Sprache noch Version gepflegt werden sollen. Objektnamen werden jedoch wie technische Namen behandelt und sind daher von diesen Einstellungen nicht betroffen.

Weitere Informationen

[Ablagetypen \[Seite 35\]](#)

[Versionierung \[Seite 176\]](#)

[Eine Anwendung anlegen \[Seite 186\]](#)

[Eigenschaften festlegen \[Seite 188\]](#)

[Objekte hinzufügen \[Seite 188\]](#)

[Die Zugriffsebene festlegen \[Seite 189\]](#)

[Versionen festlegen \[Seite 260\]](#)

1.2.2 Ablagetypen

Verwendung

BRFplus stellt verschiedene Ablagetypen für zu speichernde Objekte bereit. Die Ablagekategorie wird auf Grundlage der beabsichtigten Verwendung von Anwendungsobjekten ausgewählt. Der Ablagetyp wird als Eigenschaft einer BRFplus-Anwendung definiert. Wird ein neues Objekt angelegt, erbt es den Ablagetyp von der Anwendung, der es zugeordnet ist. Diese geerbte Einstellung kann nicht übersteuert werden.

Hinweis

Der Ablagetyp wird beim Anlegen einer Anwendung festgelegt. Nachdem die Anwendung zum ersten Mal gesichert wurde, ist diese Einstellung unumkehrbar. Es kann zu keinem späteren Zeitpunkt mehr geändert werden.

Funktionsumfang

- **Systemdaten**
Als Systemobjekte definierte Objekte sollten nicht geändert werden. Die Metadaten für Systemobjekte werden in mandantenunabhängigen Tabellen der Auslieferungsklasse S abgelegt. Diese Metadaten können nur mit einem Workbench-Transport transportiert werden. Alle Datenbank-Tabellennamen von Systemobjekten enden mit dem Buchstaben S. Systemobjekte können nicht auf Customizing- oder Stammdatenobjekte verweisen. Umgekehrt können jedoch Customizing- oder Stammdatenobjekte auf Systemobjekte verweisen.
- **Customizing-Daten**
Customizing-Objekte erlauben es Ihnen, in Ihrer Systemumgebung Änderungen vorzunehmen. Die Metadaten für Customizing-Objekte werden immer in mandantenabhängigen Tabellen der Auslieferungsklasse C abgelegt. Customizing-Objekte können nur auf Systemobjekte verweisen.
- **Stammdaten**
Stammdatenobjekte sind mandantenabhängige Objekte, die standardmäßig lokal sind und nicht transportiert werden können. Die Metadaten für Stammdatenobjekte werden in mandantenabhängigen Tabellen der Auslieferungsklasse A abgelegt. Stammdatenobjekte können auf System- und Customizing-Objekte verweisen.

In der folgenden Tabelle ist dargestellt, wie der Ablagetyt einer Anwendung die Verwendung ihrer enthaltenen Objekte beeinflusst:

Tabelle 5:

Ablagetyt	Mandant	Transport	Anwendungsübergreifende Verwendung
System	mandantenunabhängig	transportierbar oder lokal	kann Systemobjekte verwenden
Customizing	mandantenabhängig	transportierbar oder lokal	kann System- und Customizing-Objekte verwenden
Stammdaten	mandantenabhängig	lokal	kann System-, Customizing- und Stammdatenobjekte verwenden

Weitere Informationen

[Anwendung \[Seite 30\]](#)

1.2.3 Namenskonventionen für BRFplus

Verwendung

Die Namen der Objekte, die Sie in BRFplus anlegen und mit denen Sie arbeiten, werden wie technische Namen verwendet. Im Gegensatz zu den Feldern *Text* und *Kurztext* eines Objekts ist der *Name* immer sprachenunabhängig. Für globale Unternehmen, in denen Mitarbeiter mit unterschiedlichen Muttersprachen miteinander kommunizieren müssen, ist es daher ratsam, den Namen von Objekten zu verwenden. Da dieser sprachenunabhängig ist, können somit Missverständnisse vermieden werden.

In BRFplus gelten folgende Namenskonventionen:

- Der Name einer Anwendung muss innerhalb eines Systems eindeutig sein.

Hinweis

Wenn das System als Kundensystem eingerichtet wurde, reicht es aus, dass der Anwendungsname innerhalb des Systemmandanten eindeutig ist.

- Der Name einer Funktion muss innerhalb einer Anwendung eindeutig sein.
- Ein gültiger Name darf nur Zeichen aus folgendem Zeichensatz enthalten: "**0123456789ABCDEFGHIJKLMN0PQRSTUVWXYZ_/_**"
- BRFplus unterscheidet nicht zwischen Groß- und Kleinschreibung. Alle Objektnamen werden automatisch in Großbuchstaben konvertiert.

- Leerzeichen oder Bindestriche sind in Objektnamen nicht zulässig. BRFplus ersetzt Leerzeichen und Bindestriche (" ", "-") automatisch durch Unterstriche ("_") ersetzt.
- Das erste Zeichen muss ein Buchstabe oder ein Schrägstrich sein (und keine Zahl). Schrägstriche sind nur erlaubt, um ein Objekt einem Namensraum zuzuordnen. Weitere Informationen finden Sie weiter unten im Abschnitt *Präfix-Namensräume verwenden*.
- BRFplus-Objekte, die von Kunden angelegt werden, müssen mit einem der reservierten Buchstaben **Y** oder **Z** beginnen. Dies ist die übliche Vorgehensweise für Kundenobjekte im ABAP Dictionary.
- Der Name **TABLE_LINE** ist für kein BRFplus-Objekt zulässig, da es sich hierbei um ein reserviertes Wort handelt.

i Hinweis

Sie können in BRFplus auch unbenannte Objekte anlegen. Dies ist jedoch nur für Objekte möglich, die von anderen Objekten abhängen (beispielsweise ein neues Datenobjekt, das Sie als Ergebnisdatenobjekt einer Anwendung anlegen). Unbenannte Objekte können nicht wiederverwendet werden.

Als Ausnahme zu den oben genannten Regeln kann der Name eines Knotens in einem Katalog frei definiert werden. Für diese Elemente gibt es keine Prüfung auf den oben genannten Zeichensatz.

Präfix-Namensräume verwenden

BRFplus unterstützt einen informellen Ansatz, Objektnamen nach Namensräumen zu verwenden. Informell bedeutet, dass ein Namensraum in BRFplus kein eigenständiger Elementtyp ist. Vielmehr handelt es sich dabei lediglich um ein Präfix, das dem Objektnamen vorangestellt wird und das Objekt dadurch dem gegebenen Namensraum zuweist. Auch wenn dies ein sehr simples Konzept ist, können Sie es zur semantischen Gruppierung von Objekten verwenden und dadurch in großen Projekten die Komplexität verringern.

Wenn ein Objektname mit einem Schrägstrich beginnt, wird dies vom System als Beginn einer Namensraumbezeichnung interpretiert. Hieraus entstehen weitere Vorgaben für die formale Korrektheit von Objektnamen:

- Das erste Zeichen des Namens muss ein Schrägstrich sein. Dies gibt an, dass ein Namensraum verwendet wird.
- Für den Namensraum muss mindestens ein weiterer Schrägstrich verwendet werden, der entweder das Ende des Namensraums angibt oder, bei Namensräumen mit mehreren Ebenen, das Ende einer Ebene im Namensraum.
- Eine leere Zeichenkette als Namensraum ist nicht zulässig ("//").
- Auf den letzten Schrägstrich in einem Objektnamen muss mindestens ein weiteres Zeichen folgen (Der Objektname selbst).

i Hinweis

Je nach den Systemeinstellungen in Ihrem System kann die Verwendung von Präfix-Namensräumen beschränkt sein. Wenn Sie bei der Verwendung von Präfix-Namensräumen auf Probleme stoßen sollten, lesen Sie den SAP-Hinweis 150451.

Beispiel

Die folgende Tabelle enthält Beispiele gültiger und ungültiger Objektnamen:

Tabelle 6:

Name	Gültig	Kommentar
MY_OBJECT	ja	
ZMY_OBJECT_1	ja	Der Objektnamen ist im Kundennamensraum Z .
1_MY_OBJECT	nein	Der Name darf nicht mit einer Zahl beginnen.
/APP/MY_OBJECT	ja	
APP/MY/OBJECT	nein	Wenn Namensräume verwendet werden, muss das erste Zeichen ein Schrägstrich sein.
/APP/MY/OBJECT	ja	
/APP_MY_OBJECT	nein	Wenn Namensräume verwendet werden, sind mindestens zwei Schrägstriche notwendig.
/APP/FINANCIALS/COSTING/MY_OBJECT	ja	Ein gültiger Namensraum mit drei Ebenen.

Objekte umbenennen

Der Objektnamen wird beim Anlegen des Objekts definiert. Aufgrund des technischen Charakters der Eigenschaft *Name* ist es ratsam, diesen Namen beizubehalten. Dadurch beugen Sie vielen Problemen vor, die andernfalls schwierig ausfindig zu machen wären. In der BRFplus-Workbench werden Objektnamen daher immer nur im Lesemodus angezeigt.

Es ist jedoch trotzdem möglich, den Namen eines Objekts zu ändern, da BRFplus Objekte über Ihre ID identifiziert, nicht über den Namen. Innerhalb des Geltungsbereichs von BRFplus ist das Ändern des technischen Namens daher unkritisch. Allerdings ist es möglich, dass eine Objekt von einem ABAP-Backend-Programm, das BRFplus aufruft, direkt über seinen Namen referenziert wird. Sie müssen daher sicherstellen, dass alle externen, auf Namen basierten Verweise auf BRFplus-Objekte mit den Änderungen innerhalb von BRFplus übereinstimmen.

Wenn Sie einen Objektnamen dennoch ändern möchten, bietet BRFplus hierzu folgende Möglichkeiten an:

- Einzelne Objekte umbenennen
Um ein einzelnes BRFplus-Objekt umzubenennen, öffnen Sie die Verwaltungsdaten des Objekts im Abschnitt *Allgemein*. Wählen Sie auf der Registerkarte *Allgemein* die Option *Umbenennen*. Es wird ein Dialogfenster geöffnet, in dem Sie den Objektnamen ändern können.
- Mehrere Objekte umbenennen

Um mehrere Objekte umzubenennen, öffnen Sie das Massenänderungswerkzeug. Wählen Sie die Objekte aus, die Sie umbenennen möchten und wählen Sie die Drucktaste [Umbenennen](#). Es wird ein Dialogfenster geöffnet, in dem Sie ein Präfix oder ein Suffix definieren können, das den bestehenden Namen angehängt wird.

Weitere Informationen

[Verwaltungsdaten \[Seite 39\]](#)

[Massenänderung \[Seite 278\]](#)


1.2.4 Verwaltungsdaten

Definition

Alle Objekte, die Sie in BRFplus anlegen können, besitzen zusätzlich zu ihren typspezifischen Eigenschaften eine Reihe von gemeinsamen Eigenschaften, die unabhängig vom jeweiligen Objekttyp sind. Diese Eigenschaften werden als Verwaltungsdaten bezeichnet.

Verwendung

Mithilfe dieser Eigenschaften können Sie im System verfügbare BRFplus-Objekte identifizieren, suchen oder zusammenfassen. Sie finden die Verwaltungsdaten in der BRFplus-Workbench am oberen Rand des Pflegebildschirms für jedes Objekt unter dem Abschnitt [Allgemein](#). Sie können entscheiden, ob dieser

Abschnitt angezeigt werden soll, indem Sie am rechten Rand der Abschnittsüberschrift auf  klicken. Wenn Sie sich den Inhalt in diesem Abschnitt anzeigen lassen, können Sie sich einen kurzen Überblick über die Verwaltungsdaten des Objekts verschaffen. Wenn Sie sich den Inhalt nicht anzeigen lassen, haben Sie mehr Platz zur Anzeige der Objekteinstellungen.

Funktionsumfang

Übersicht

Wenn Sie ein BRFplus-Objekt zur Anzeige oder Bearbeitung öffnen, zeigt das System im Arbeitsbereich der Oberfläche die jeweiligen Objektdaten an. Ob die Verwaltungsdaten standardmäßig angezeigt werden oder nicht, hängt von den Einstellungen unter [Personalisierung](#) ab.

Details

Die Detailsicht der Verwaltungsdaten umfasst folgende Registerkarten:

Allgemein

Hier wird eine Reihe von Einstellungen angezeigt, die entweder beim Anlegen des Objekts definiert wurden (beispielsweise die Anwendung, zu der ein Objekt gehört), oder dem System direkt entstammen (wie das Anlege- und Änderungsdatum). Daher werden diese Einstellungen nur im Anzeigemodus ausgegeben. Die einzige Ausnahme stellt die [Zugriffsebene](#) dar. Sie kann vom Benutzer geändert werden. Weitere Informationen zu Zugriffsebenen finden Sie unter [Eine Zugriffsebene einstellen \[Seite 189\]](#).


Texte

In den Texteeinstellungen können Sie bestimmen, wie der beschreibende Name (nicht der technische Name) eines Objekts auf der Benutzungsoberfläche dargestellt wird. Die vom System angebotenen Optionen hängen dabei von der [Quelle](#) ab, die Sie für die Objekttexte auswählen:

- [Freie Texteingabe](#)
Bei dieser Option müssen Kurz- und Langtext manuell vom Benutzer eingegeben werden.
- [Text-Symbol](#)
Bei dieser Option wird der Text als Objekttext verwendet, der für ein gegebenes Text-Symbol (auch Textelement genannt) eines gegebenen ABAP-Reports im Backend-System definiert wurde. Im Gegensatz zu anderen Optionen unterstützt das [Text-Symbol](#) lediglich ein Textfeld. Aus diesem Grund kann hier nicht zwischen Kurz- und Langtexten unterschieden werden.
- [Exit-Klasse](#)
Bei dieser Option werden Kurz- und Langtext der angegebenen Exit-Klasse als Objekttexte verwendet.

Der folgende Abschnitt beschreibt die Textoptionen je nach ausgewählter [Quelle](#):

- [Freie Texteingabe](#)
 - [Abhängigkeit](#)
Sie können für manuell einzugebende Texte festlegen, ob diese durch alle Sprachen und Versionen hinweg konstant verwendet werden sollen, oder ob sich Texte in unterschiedlichen Sprachen und/oder Versionen ändern sollen.
 - Sprachabhängigkeit
Bei sprachabhängigen Texten wird ein zusätzliches Feld angezeigt, über das Sie die gewünschte Sprache wählen können. Dieses Feld bietet Ihnen alle im System verfügbaren Sprachen zur Auswahl. Sobald Sie die gewünschte Sprache gewählt haben, können Sie die Texte für diese Sprache pflegen. Diese Option ist hilfreich und kann sogar rechtlich verpflichtend sein, wenn Sie vorhaben, eine BRFplus-Anwendung in Ländern mit unterschiedlichen Sprachen zu verwenden.
 - Versionsabhängigkeit
Sie können auch für unter Versionsverwaltung stehende Objekte definieren, dass deren Texte versionsabhängig sein sollen. Bei dieser Einstellung enthält jede neue Version eines Objekts eine Kopie der Objekttexte aus der vorhergehenden Version. Diese Texte können dann in der neuen Version überschrieben werden, ohne die Texte in der vorhergehenden Version zu beeinflussen. Das kann nützlich sein, wenn es bei einem komplexen Objekt wie einem Formelausdruck von einer Version zur nächsten wesentliche Änderungen gab oder wenn ein Text den Namen einer externen Entität richtig wiedergeben soll, die im Laufe der Zeit umbenannt wurde.
 - [Sprache](#)
Wählen Sie eine der verfügbaren Sprachen aus und pflegen Sie Kurz- und Langtext für diese Sprache. Dieses Feld ist nur verfügbar, wenn Sie das Objekt als sprachabhängig definiert haben.
 - [Kurztext, Text](#)
Geben Sie die Texte für das aktuelle Objekt ein. Kurztexte können eine Länge von maximal 20 Zeichen haben, Texte bis zu 80 Zeichen. Von der BRFplus-Workbench wird der Kurztext zur Anzeige des Objekts verwendet, es sei denn Sie möchten, dass lediglich die technischen Namen angezeigt werden.

Um die Anzeigeeigenschaften festzulegen, wählen Sie  und dann unter der Registerkarte **Allgemein** **Technische Namen anzeigen**.

Hinweis

Das System besitzt einen Ersatzmechanismus, sollte der Text, der laut den Personalisierungseinstellungen angezeigt werden soll, nicht verfügbar sein (wenn beispielsweise der benötigte Text für die aktuelle Anmeldesprache nicht gepflegt wurde oder das Objekt keinen Namen besitzt). Es wird in diesem Fall versucht, einen der Texte in folgender Reihenfolge anzuzeigen:

1. Kurztext
2. Langtext
3. Dynamische String-Repräsentation bedeutender Elemente der Objektdefinition (beispielsweise den Wert einer Konstante als String)

- **Text-Symbol**

- **Programmname**

- Geben Sie den technischen Namen des ABAP-Reports ein, für den ein Text-Symbol definiert ist, das Sie als Text für das aktuelle Objekt verwenden möchten.

- **Schlüssel**

- Geben Sie die ID des gewünschten Text-Symbols ein, das für den oben genannten Report definiert ist. Drücken Sie anschließend die Eingabetaste. Der für das Text-Symbol gepflegte String wird nun als **Text** angezeigt.

- **Sprache**

- Wählen Sie eine der im System verfügbaren Sprachen aus, um zu überprüfen, welcher Text angezeigt wird, wenn ein Benutzer in dieser Sprache angemeldet ist.

Hinweis

Das Auswahlfeld **Sprache** stellt lediglich eine Hilfe dar, um herauszufinden, welche verschiedenen Sprachversionen für das ausgewählte Textsymbol existieren. Es kann **nicht** verwendet werden, um zusätzliche Sprachversionen zu pflegen. Dies kann nur über die jeweiligen Übersetzungstools für die Textelemente gemacht werden, die dem referenzierten Report zugeordnet sind.

- **Exit-Klasse**

Bei dieser Option werden Kurz- und Langtext der angegebenen Exit-Klasse als Objekttexte verwendet.

Hinweis

Die Exit-Klasse, die Sie hier eingeben, muss das Interface `IF_FDT_TEXT_DOCU_EXIT` implementieren. Die Workbench verwendet Methoden dieses Interfaces, um Texte für das Objekt abzurufen. Die Workbench unterstützt Sie dabei, eine passende Klasse zu finden. Hierzu werden in der Werthilfe lediglich die Klassen angezeigt, die diese Voraussetzung erfüllen.

Beachten Sie, dass diese Klasseneinstellung **nicht** mit der Anwendungs-Exit-Klasse zusammenhängt.

Dokumentation

BRFplus gibt Ihnen die Möglichkeit, für jedes Einzelobjekt, das Sie anlegen, wie Funktionen, Ausdrücke, Datenobjekte usw., eine Dokumentation zu pflegen. Dies ist besonders hilfreich für komplexe Objekte wie Formeln oder Entscheidungsbäume, die die Art und Weise, wie das System einen Business Case handhabt, direkt beeinflussen, manchmal sind sie jedoch schwer verständlich.

Zur Dokumentation von Objekten sind sehr ähnliche Optionen vorhanden wie für Texte. Daher werden im folgenden Absatz lediglich die dokumentationsspezifischen Optionen beschrieben. Die Optionen, die für Texte und Dokumentation gleich sind, werden nicht aufgeführt. Mehr Informationen zu diesen Optionen finden Sie oben im Abschnitt *Texte*.

- **Freie Texteingabe**
Bei dieser Option können Sie manuell einen Dokumentationstext von beliebiger Länge eingeben.
- **SAPscript-Objekt**
Bei dieser Option wird der im angegebenen SAPscript-Dokument enthaltene Text als Dokumentation für das Objekt referenziert.
- **Exit-Klasse**
Bei dieser Option wird die Dokumentation der gegebenen Exit-Klasse als Objektdokumentation verwendet.

Der folgende Abschnitt beschreibt die Textoptionen je nach ausgewählter *Quelle*:

- **Freie Texteingabe**
 - **Abhängigkeit, Sprache**
Siehe oben den Abschnitt *Texte*.
- **SAPscript-Objekt**
 - **ID**
Dokumentklassen-ID des SAPscript-Dokuments, das Sie als Objektdokumentation verwenden möchten. Mögliche Werte sind **TX** für allgemeine Texte, **RE** für Reportdokumentationen, **DT** für Dialogtexte, usw.
 - **Objekt**
Technischer Name des SAPscript-Dokuments, das Sie als Objektdokumentation verwenden möchten.
- **Exit-Klasse**
 - **Exit-Klasse**
Geben Sie hier den Namen der Exit-Klasse ein, die Sie zur Bereitstellung der Objektdokumentation verwenden möchten.

i Hinweis

Die Exit-Klasse, die Sie hier eingeben, muss das Interface `IF_FDT_TEXT_DOCU_EXIT` implementieren. Die Workbench verwendet Methoden dieses Interfaces, um Dokumentationen für das Objekt abzurufen. Die Workbench unterstützt Sie dabei, eine passende Klasse zu finden. Hierzu werden in der Worthilfe lediglich die Klassen angezeigt, die diese Voraussetzung erfüllen.

Beachten Sie, dass diese Klasseneinstellung **nicht** mit der Anwendungs-Exit-Klasse zusammenhängt.

- **Dokumentation**
In diesem Feld zeigt die BRFplus-Workbench den Dokumentationstext der Exit-Klasse an, die Sie unter **Exit-Klasse** eingegeben haben.

Wenn die vordefinierte Funktionalität der dynamischen Bereitstellung von Objektdokumentation mithilfe einer Exit-Klasse für Ihre Zwecke noch nicht flexibel genug ist, können Sie die Entwicklung individueller Erweiterungen zu diesem Konzept in Betracht ziehen. Wenn Sie beispielsweise eine Lizenz für das SAP NetWeaver Knowledge Warehouse besitzen, wäre es möglich, zusätzliche Dokumentation im Knowledge Warehouse zu erstellen und über die von der Exit-Klasse implementierte Interface-Methode das Topic zu einem bestimmten BRFplus-Objekt anzeigen zu lassen. Solche Erweiterungen sind jedoch völlig kundenspezifisch und werden von SAP nicht unterstützt.

Hinweis

Bei Datenobjekten, die an ein DDIC-Objekt gebunden wurden, erben die Verwaltungsdaten unter [Texte](#) und [Dokumentation](#) automatisch die jeweiligen Daten aus dem DDIC für das gebundene Objekt. Dies beinhaltet auch alle Sprachversionen, die möglicherweise im DDIC verfügbar sind. Text und Dokumentation werden auf folgende Art unterschiedlich vom System behandelt:

- Die vererbten Texte werden als Standardwerte für ein gebundenes Datenobjekt in das BRFplus-Repository importiert, können jedoch bei Bedarf überschrieben werden.
- Die vererbte Dokumentation wird als Referenz in das BRFplus-Repository importiert und kann nicht geändert werden.

Weitere Informationen finden Sie unter [Datenbindung \[Seite 66\]](#).

Versionen

Sie können auf alle Versionen eines gegebenen Objekts zugreifen, die im System gespeichert sind. Die Versionsliste enthält immer mindestens einen Eintrag (die aktuelle Version). Für jede Version zeigt das System folgende Informationen an:

Tabelle 7:

Eigenschaft	Beschreibung
Version	Versionsnummer. Bei komplexen Objekttypen enthält diese Spalte auch ein Dreieckssymbol, über das Sie die Objektstruktur in Spalte Objekttyp aufklappen können (siehe unten).
Benutzername	Anmeldename des Benutzers, der die Objektversion gesichert hat
Zeitstempel	Datum und Uhrzeit des Sicherns der Objektversion
Verarbeitungsrelevante Änderungen	Zeigt für eine Version eine Liste von Veränderungen an, die verarbeitungsrelevant sind und den Prozessablauf beeinflussen können. Einige Beispiele für solche Veränderungen sind: <ul style="list-style-type: none">• Der Wert einer Konstante wurde geändert• Ein Formelausdruck wurde geändert• Ein konstanter Ausdruck wurde durch eine Entscheidungstabelle ersetzt
Nicht verarbeitungsrelevante Änderungen	Zeigt für eine Version eine Liste von Veränderungen an, die nicht verarbeitungsrelevant sind und den Prozessablauf nicht beeinflussen. Einige Beispiele für solche Veränderungen sind: <ul style="list-style-type: none">• Der Kurztext wurde geändert• Die Versionierung wurde ausgeschaltet• Für die Dokumentation wurde die Abhängigkeit geändert

Eigenschaft	Beschreibung
Versionierung	Gibt an, ob die Versionierung eingeschaltet ist und, gegebenenfalls, welcher Versionierungsmodus aktiv ist. Diese Einstellung kann folgende Werte haben: <i>Ein</i> , <i>Transportiert mit <Transport-ID></i> oder <i>Nicht transportiert</i> für Vorgängerversionen der aktuellen Version. Für die aktuelle Version kann die Versionierung entweder ein- oder ausgeschaltet sein.
Objektyp	Typ des aktuellen Objekts. Bei komplexen Objekten zeigt diese Spalte die aufgeklappte Objektstruktur an. Ein Datenobjekt vom Typ Tabelle kann beispielsweise aufgeklappt werden, um seine Struktur und die in der jeweiligen Version vorhandenen Elemente anzuzeigen.

Je nach Definition des jeweiligen Objekts wird in der Spalte *Name* vor dem Objektnamen ein kleines dreieckiges Symbol angezeigt. Dies zeigt an, dass die aktuelle Objektversion eingebettete Objekte enthält, die ebenfalls versioniert sind. Wenn Sie die Liste eingebetteter Objekte aufklappen, können Sie direkt auf die Version des eingebetteten Objekts zugreifen.

Zusatzinformationen

Die Zusatzinformationen unter dieser Registerkarte helfen Ihnen dabei, den Ursprung eines BRFplus-Objekts zu identifizieren. Hier finden Sie den Namen des Systems, in dem ein Objekt erstellt wurde, die ID des Transportauftrags, mit dem das Objekt in das aktuelle System importiert wurde und die Mandantennummer im Quellsystem, aus dem das Objekt exportiert wurde.

Wenn ein nicht-lokales Objekt geändert wurde, in einen Transportauftrag aufgenommen wurde und auf Freigabe wartet, wird der entsprechende Transportauftrag hier angezeigt. Die Transport-ID ist in diesem Fall ein Link. Wenn Sie auf die Transport-ID klicken, öffnet sich das Transportanalysewerkzeug mit detaillierten Informationen zu diesem Transportauftrag.

Die Anzahl der gefüllten Felder gibt Aufschluss über das Verhältnis zwischen dem aktuellen System und dem aktuellen Objekt: Wenn das aktuelle System das Originalsystem des Objekts ist, enthält nur das Systemfeld einen Wert, während die anderen Felder leer bleiben (da in diesem Fall kein Transport nötig war).

Weitere Informationen

[Versionierung \[Seite 176\]](#)

1.2.4.1 Benutzerdefinierte Attribute

Erläutert, wie Sie für die Objekte einer BRFplus-Anwendung benutzerspezifische Metadaten bereitstellen können (zusätzlich zu dem vordefinierten Satz an administrativen Datenfeldern).

Verwendung

BRFplus bietet für jedes Objekt, das Sie in einer BRFplus-Anwendung anlegen, eine Vielzahl von Metadatenfeldern an. Einige dieser Felder werden automatisch vom System gepflegt, während andere vom Benutzer gepflegt werden können. Trotz der Menge an Informationen, die in diesen Feldern gespeichert werden kann, gibt es sicherlich Projekte oder Anwendungsszenarien, für die spezifische Objektattribute benötigt werden, die SAP nicht vorhersehen kann.

Beispiel

In einem Regelentwicklungsprojekt hat ein Abteilungsleiter eine neue Anwendung angefordert, bei der es möglich sein soll, dass jedem zur Anwendung gehörenden BRFplus-Objekt ein Verantwortlicher zugeordnet wird. In den Standard-Metadaten von BRFplus-Objekten ist jedoch nur erfasst, wer das Objekt angelegt oder zuletzt geändert hat. Mit benutzerdefinierten Attributen können Sie ein Datenobjekt des Typs Text in Ihrer Anwendung verwenden, um diese Anforderung zu realisieren. Erweitern Sie mithilfe dieses Textes die Metadaten jedes Objekts um ein zusätzliches Feld, das den Namen oder die Benutzer-ID der verantwortlichen Person enthält.

Mithilfe von benutzerdefinierten Attributen können Sie eine beliebige Anzahl von Attributen definieren, die Sie innerhalb der Anwendung verwenden können, um (zusätzlich zu den unter [Verwaltungsdaten \[Seite 39\]](#) beschriebenen Standardattributen) weitere Details zu den Objekten innerhalb dieser Anwendung anzugeben.

Benutzerdefinierte Attribute werden technisch umgesetzt, indem Element-Datenobjekte des Typs Text einer Anwendung zugeordnet werden, die dann automatisch für alle Objekte innerhalb dieser Anwendung verfügbar sind.

Vorgehensweise

Gehen Sie folgendermaßen vor, um benutzerdefinierte Attribute zu verwenden:

1. Öffnen Sie in der BRFplus-Workbench die Anwendung, für die Sie benutzerdefinierte Attribute definieren möchten.
2. Navigieren Sie im Abschnitt [Details](#) zur Registerkarte [Benutzerdefinierte Attribute](#).
3. Wählen Sie entweder [Vorhandene Datenobjekte hinzufügen](#) oder [Neues Datenobjekt hinzufügen](#). Das System stellt sicher, dass hierfür nur Datenobjekte des Typs Text verwendet oder angelegt werden.
4. Sichern und aktivieren Sie die Datenobjekte und die Anwendung.

Sobald Sie mindestens ein benutzerdefiniertes Attribut für eine Anwendung definiert haben, wird bei jedem Objekt in dieser Anwendung im Abschnitt [Allgemein](#) eine zusätzliche Registerkarte [Benutzerdefinierte Attribute](#) angezeigt. Hier können Sie die gewünschten Werte für Ihre benutzerdefinierten Werte eintragen.

1.2.5 Kontext und Ergebnis

Verwendung

Kontext und Ergebnis definieren die Schnittstelle für Eingabe- und Ausgabeparameter. In BRFplus werden Eingabeparameter als Kontext und Ausgabeparameter als Ergebnis bezeichnet. Kontext und Ergebnis sind Datenträger.

Funktionsumfang

Allgemein

Sowohl Kontext als auch Ergebnis bestehen aus einem Satz an Datenobjekten wie Elementen, Strukturen oder Tabellen. Ein Kontext oder ein Ergebnis des Typs *Element* kann eine Zahl, ein Boolescher Wert, eine Menge, ein Betrag oder ein Zeitpunkt sein.

Funktionen sowie Ausdrücke, Aktionen und Regeln haben einen Kontext. Eine Menge an durch den Kontext definierten Attributen wird als Eingabewertemenge verwendet, die von einer Funktion oder einem Ausdruck verarbeitet wird. Ein Kontext kann aus einem oder mehreren Elementen, Strukturen oder Tabellen bestehen, die jeweils über ihren Namen angesprochen werden.

Die Kontextdaten einer Funktion können von allen Unterausdrücken verwendet werden, die zum obersten Ausdruck oder Regelsatz der Funktion gehören. Die Eingabeparameter der Ausdrücke werden entweder mit dem Kontext der Funktion oder mit dem Ergebnis anderer Ausdrücke gefüllt.

Im Gegensatz dazu kann eine Funktion oder ein Ausdruck immer nur ein Ergebnisdatenobjekt enthalten. Wenn eine Geschäftsregel also mehr als einen Wert auf einmal zurückliefern soll, kann dies nur erreicht werden, indem als Ergebnisdatenobjekt eine Struktur oder eine Tabelle zugeordnet wird.

Wenn Sie eine Funktion mit dem Ausführungsmodus *Ereignismodus* anlegen, ordnet das System die vordefinierte Tabelle *Aktionen* als Ergebnisdatenobjekt der Funktion zu. Diese Standardzuordnung können Sie jedoch überschreiben und ein anderes, passenderes Ergebnisdatenobjekt zuordnen.

Kontextübersicht

Zu jedem Objekt, das einen Kontext hat, können Sie sich über seine verfügbaren Kontextdatenobjekte informieren. Navigieren Sie hierzu in der BRFplus-Workbench zum jeweiligen Objekt und wählen Sie *Kontextübersicht*. Das System zeigt eine Liste mit allen Datenobjekten an, die über den Kontext des Objekts verfügbar sind. Das System zeigt die Herkunft jedes Datenobjekts an (Funktionskontext oder Ergebnis). Außerdem können Sie sich in diesem Dialogfenster anzeigen lassen, ob es mögliche andere Kontextdatenobjekte gibt, die derzeit **nicht** für die Verwendung mit dem Ausdruck verfügbar sind.

Hinweis

Wenn eine Funktion im *Ereignismodus* [Seite 151] statt im Funktionsmodus ausgeführt wird, kann der Kontext geändert und zu einem Teil des Ergebnisses werden. Der Ergebnisparameter ist in diesem Fall nicht relevant.

Wenn ein Ausdruck, eine Aktion oder eine Regel mehr als einer Funktion zugeordnet sind, entspricht die Liste der verfügbaren Kontextdatenobjekte der Schnittmenge der Kontextdatenobjekte aller beteiligten

Funktionen. Dies ist notwendig, um sicherzustellen, dass der Ausdruck, die Aktion oder die Regel unabhängig von der aufrufenden Funktion richtig verarbeitet werden können.

Beispiel

Ein Entscheidungstabellenausdruck DT wird von drei verschiedenen Funktionen F1, F2 und F3 verwendet, entweder direkt als oberster Ausdruck der Funktion oder indirekt als geschachtelter Ausdruck. In der folgenden Tabelle sind die Kontextdatenobjekte für jede dieser Funktionen dargestellt:

Tabelle 8:

Funktion	Parameter
F1	NUM1, TXT1, BOOL1
F2	NUM1, TXT1
F3	NUM1, BOOL1

In diesem Szenario haben alle Funktionen lediglich das Kontextdatenobjekt **NUM1** gemeinsam. Folglich kann ein Ausdruck wie die Entscheidungstabelle, die in allen drei Funktionen verwendet wird, nur auf dieses gemeinsame Kontextdatenobjekt zugreifen. Die Verwendung eines der anderen Kontextdatenobjekte würde dazu führen, dass der Kontext des Ausdrucks für mindestens eine der drei Funktionen teilweise undefiniert wäre. Dies ist nicht zulässig.

Wenn Sie in dem oben beschriebenen Szenario DT einer vierten Funktion F4 zuordnen würden und wenn F4 im Kontext nur ein Zahl-Datenobjekt **NUM2** aufweist, das in keiner der anderen Funktionen vorhanden ist, wäre die Schnittmenge der verwendbaren Kontextdatenobjekte für DT leer.

Implizite Ergebnistypkonvertierung

Das Ergebnisdatenobjekt eines BRFplus-Objekts muss mit dem Kontextdatenobjekt kompatibel sein, an das das Ergebnis übergeben wird. Beispiel: Wenn eine Zelle in einer Entscheidungstabelle einen Ausdruck vom Typ Wertebereich enthält, müssen für die Bereichsparameter direkte Werte, Datenobjekte oder Ausdrücke desselben Typs wie die Wertebereichsparameter verwendet werden. Es gibt jedoch eine Ausnahme hierzu: In einer Funktion mit einem Ergebnisdatenobjekt vom Typ Text können Sie jeden beliebigen Ausdruck als obersten Ausdruck zuordnen, unabhängig vom Typ des Ergebnisdatenobjekts. Wenn sich der Typ des Ergebnisdatenobjekts des Ausdrucks vom Textdatenobjekt unterscheidet, das für die Funktion definiert wurde, führt BRFplus eine implizite Typkonvertierung des Ausdrucksergebnisses in einen Text durch. Die Konvertierungslogik ist dabei dieselbe, die für die Funktion `ToString` in Formelausdrücken verwendet wird. Weitere Informationen zu dieser Logik finden Sie in der Dokumentation zur Formelfunktion in der BRFplus-Workbench.

1.2.6 Objekte anlegen

Definition

In BRFplus legen Sie verschiedene Objekte an, um die Geschäftsregeln zu modellieren, die Sie bei der Entscheidungsfindung innerhalb der Anwendungsfälle Ihrer täglichen Arbeit unterstützen sollen. Sie können Objekte auf verschiedene Arten anlegen. Je nach Situation kann dies unterschiedliche Folgen haben.

Konzept

Allgemein

Alle BRFplus-Objekttypen besitzen eine Reihe von Verwaltungsdaten, die aus den gleichen Attributen zusammengesetzt sind. Die meisten dieser Attribute sind objektspezifisch. Es gibt jedoch auch einige, die vom Anwendungsobjekt abhängen, zu dem das Objekt gehört. Diese abhängigen Attribute sind:

- **Feste Abhängigkeit**
Die folgenden Anwendungsattribute werden von allen Objekten geerbt, die Sie im Kontext einer Anwendung anlegen, und können nicht auf Objektebene geändert werden:
 - **Ablagetyp**
Mit dem Ablagetyp können Sie steuern, ob ein Objekt als Systemobjekt, Customizing-Objekt oder Stammdatenobjekt behandelt wird. Sie definieren den Ablagetyp einer Anwendung und aller ihrer Objekte beim Anlegen der Anwendung. Es gibt keine Möglichkeit, diese Einstellung für die Anwendung oder deren Objekte nachträglich zu ändern.
 - **Lokale vs. transportierbare Objekte**
Sie müssen für eine Anwendung das Paket definieren, zu dem die Anwendung gehört. Das Transportverhalten der Anwendung und ihrer Objekte wird von den Transporteinstellungen des zugeordneten Pakets bestimmt. Theoretisch können Sie das Transportverhalten eines BRFplus-Objekts ändern, indem Sie der Anwendung ein anderes Paket zuordnen. Eine derartige Änderung hat jedoch meist weitreichende Folgen und wird daher nicht empfohlen.
- **Lose Abhängigkeit**
Die folgenden Anwendungsattribute werden von allen Objekten geerbt, die Sie im Kontext einer Anwendung anlegen, können aber für jedes Objekt einzeln geändert werden:
 - Einstellungen des Anwendungsprotokolls
 - Versionierungsverhalten
 - Sprach- und Versionsabhängigkeit von Objekttexten und Dokumentation

Sichern, prüfen, aktivieren

Nachdem Sie die Pflichtangaben für ein neu angelegtes Objekt gepflegt und gesichert haben, können Sie Ihre Arbeit fortsetzen. Es ist nicht notwendig, alle verfügbaren Einstellungen auf einmal zu pflegen. Bedenken Sie jedoch, dass viele Objekttypen für die produktive Verwendung detaillierte Einstellungen benötigen. Mit integrierten Prüfroutinen hilft Ihnen das System, fehlende oder unvollständige Einstellungen für jeden Objekttyp zu ermitteln.



Beispiel

Sie erstellen eine Entscheidungstabelle mit zehn Bedingungsspalten und hunderten Reihen für verschiedene Kombinationen von Bedingungswerten. Auf den ersten Blick scheint die Tabelle für den

Business Case zu genügen, den Sie sich überlegt haben. Allerdings wird die Prüfung vielleicht feststellen, dass die Datenobjekte, die Sie in den Bedingungsspalten verwendet haben, absichtlich aus dem Repository genommen wurden und nicht aus einem Funktionskontext, wie es für eine fehlerfreie Ausführung benötigt wird. Daher kann die Entscheidungstabelle zwar in sich konsistent und vollständig sein, aber trotzdem nicht aktiviert werden, solange sie nicht mit einer Funktion verbunden ist, die alle verwendeten Datenobjekte bereitstellt.

Sie können ein Objekt entweder explizit über die Schaltfläche **Prüfen** oder implizit über die Schaltfläche **Aktivieren** prüfen. Das System durchläuft dann alle Prüfroutinen, die für den jeweiligen Objekttyp definiert wurden. Immer wenn bei der Prüfung Fehler oder Inkonsistenzen festgestellt werden, erhalten Sie eine Meldung. Viele dieser Meldungen beinhalten einen zusätzlichen Langtext mit detaillierten Erklärungen des Problems und Lösungsvorschlägen.

Viele Objekttypen in BRFPplus enthalten Referenzen zu anderen Objekten. Wenn Sie versuchen, ein Objekt mit solchen Referenzen zu aktivieren, ermittelt das System, ob innerhalb der Hierarchie der verwendeten Objekte noch inaktive Objekte vorhanden sind. Ist dies der Fall, schlägt das System vor, durch rekursives Durchlaufen der Objekthierarchie alle inaktiven Objekte zu aktivieren, die es findet. Dies ist eine Voraussetzung dafür, dass das Einstiegsobjekt erfolgreich aktiviert werden kann.

Wenn eines der untergeordneten Objekte nicht aktiviert werden kann, wird der Vorgang abgebrochen und Sie erhalten eine Fehlermeldung. In diesem Fall empfehlen wir Ihnen, den Erklärungstext der Meldung aufmerksam zu lesen und herauszufinden, an welcher Stelle genau das Problem aufgetreten ist. Dies ist nicht immer offensichtlich.

Verwendungs-Konsistenzprüfung

Zusätzlich zu diesen Prüfungen von Inkonsistenzen in der Objektdefinition gibt es noch eine weitere Prüfungsart, die **Verwendungs-Konsistenzprüfung**. Mit dieser Option können Sie prüfen, ob das Objekt, das Sie gerade pflegen, von anderen Objekten verwendet wird und ob es Probleme mit der Verwendungsbeziehung oder mit den verwendenden Objekten gibt. Diese Prüfung ist nur für Objekte sinnvoll, die von anderen Objekten verwendet werden und daher für gewöhnlich nicht beim Anlegen relevant.

Öffnen Sie zur Durchführung dieser Prüfung das gewünschte Objekt im Arbeitsbereich der BRFPplus-Workbench und wählen Sie **Mehr > Verwendungs-Konsistenzprüf. >** aus.

Beispiel

Es wurde ein Element-Datenobjekt vom Typ Text definiert, dessen Text eine Maximallänge von 20 Zeichen hat. Dieses Datenobjekt wird als Ergebnisdatenobjekt für einen Ausdruck vom Typ Entscheidungstabelle verwendet, in dem bereits eine Reihe von Ergebnis-Strings definiert wurde. Wenn Sie die maximale Textlänge des Datenobjekts auf zehn Zeichen verringern und die Entscheidungstabelle Ergebniswerte enthält, die diese verringerte Maximallänge überschreiten, würde die Verwendungs-Konsistenzprüfung Sie auf dieses potenzielle Problem hinweisen.

Hinweis

Beim Durchlaufen des Datenmodells kann die Verwendungs-Konsistenzprüfung auch Objekte finden, die das aktuelle Objekt verwenden, sich jedoch in einem anderen Systemmandanten befinden. Für verwendende Objekte dieser Art kann das System keine weiteren Prüfungen durchführen. Es macht Sie jedoch darauf aufmerksam, dass betroffene Objekte in anderen Mandanten existieren und überlässt Ihnen die Entscheidung, dieselben Prüfungen in diesen Mandanten durchzuführen.

Manuelles vs. programmtechnisches Anlegen

Manuelles Anlegen von Objekten

Üblicherweise werden BRFplus-Objekte einzeln in der BRFplus-Workbench angelegt. Dies können Sie in folgenden Sichten tun:

- **Repository**
Klicken Sie in der **Repository**-Sicht mit der rechten Maustaste auf den Knoten einer beliebigen Kategorie oder eines beliebigen Objekttyps. Wählen Sie **Anlegen** aus dem Kontextmenü. Es werden Ihnen die Objekttypen angezeigt, die Sie im Kontext des Knotens, dessen Menü Sie geöffnet haben, anlegen können.

Hinweis

Im Kontextmenü eines Anwendungsobjekts bietet das **Anlegen**-Menü Unterpunkte zum Anlegen von BRFplus-Objekten aller Typen sowie ganzen Anwendungen.

- **Katalog**
Wenn Sie einen Katalog definiert haben, können Sie über das Kontextmenü des Katalogs neue Objekte anlegen. In diesem Fall werden neu angelegte Objekte gleichzeitig auch dem Katalogknoten zugeordnet, von dem aus Sie gestartet sind.
- **Favoriten** oder **Zuletzt verwendet**
In diesen beiden Sichten können Sie neue Objekte über das Kontextmenü eines Anwendungsobjekts anlegen. Wenn in diesen Sichten kein Anwendungsobjekt aufgeführt ist, können Sie hier auch keine Objekte anlegen.

Programmtechnisches Anlegen von Objekten

Sie können Objekte natürlich auch programmtechnisch über entsprechende Programmierschnittstellen (APIs) anlegen, die von BRFplus angeboten werden. Die Klasse `ECL_FDT_FACTORY` dient hierfür als Startpunkt. Dies ist vor allem hilfreich, wenn Sie eine große Anzahl von Objekten anlegen müssen, also wenn Sie beispielsweise die Systemleistung bei mehreren tausend Objekten testen möchten, bevor Sie ein Entwicklungsprojekt von solcher Dimension tatsächlich starten. In diesem Fall müssen Sie beachten, dass die Objekte, die über eine Verwendungsbeziehung verbunden sind, von unten nach oben aktiviert werden müssen. Weitere Informationen finden Sie in den Tutorien auf den BRFplus-Seiten im SAP Developers Network (SDN).

Einzelne vs. mehrere Objekte

Wie oben erwähnt, werden üblicherweise einzelne BRFplus-Objekte mit der BRFplus-Workbench angelegt und große Mengen an Objekten am besten über die Programmierschnittstellen. Es gibt allerdings eine dritte Alternative, die möglicherweise einen guten Kompromiss zwischen diesen beiden Vorgehensweisen darstellt.

Erfahrungsgemäß ist die Mehrheit der Objekte in regelbasierten Projekten vom Typ Element-Datenobjekt. Für diese Objekte bietet die BRFplus-Workbench ein Massenanlegungswerkzeug, in dem Sie in tabellarischer Struktur die grundsätzlichen Attributwerte für eine beliebige Anzahl von Objekten auf einmal eingeben können. Wählen Sie nach Eingabe dieser Daten **OK** und die Datenobjekte werden alle gleichzeitig angelegt. Diese Vorgehensweise ist wesentlich schneller als das einzelne Anlegen von Objekten. Starten Sie das Massenanlegungswerkzeug, indem Sie in der BRFplus-Workbench im Kontextmenü einer Anwendung

► **Anlegen** ► **Datenobjekt** ► **Elemente (Massenanlage)...** ► wählen.

Hinweis

Die Massenanlage ist nur für Datenobjekte des Typs Element möglich.

Wiederverwendbarkeit

Für viele Szenarien ist es wünschenswert, dieselben BRFPplus-Objekte an verschiedenen Stellen in einer Anwendung zu verwenden oder sogar in verschiedenen Anwendungen. Die Wiederverwendung von Objekten kann die Komplexität verringern und zur Verbesserung der Konsistenz einer Anwendung beitragen. Daher empfiehlt BRFPplus, wiederverwendbare Objekte anzulegen. Setzen Sie hierzu beim Anlegen ein Häkchen in das Feld *Ist wiederverwendbar*. Dieses Feld ist im Anlege-Dialog der meisten Objekte standardmäßig aktiviert. Bei Aktivierung dieser Einstellung wird der Objektname zu einem Pflichtfeld für das Anlegen, was wiederum Voraussetzung für die Wiederverwendung von Objekten ist.

Ein wiederverwendbares Objekt besitzt keinerlei Nachteile. Daher ist in der BRFPplus-Workbench das Ankreuzfeld *Ist wiederverwendbar* nicht nur standardmäßig aktiviert, sondern kann in den meisten Fällen auch nicht deaktiviert werden. In seltenen Fällen können Sie jedoch entscheiden, nicht wiederverwendbare Objekte anzulegen. Beachten Sie hierzu jedoch Folgendes:

- Nur Ausdrücke können als nicht wiederverwendbar definiert werden.
- Dies ist nur möglich, wenn Sie einen Ausdruck über den Kontext eines anderen Objekts anlegen und nicht über den Objektbaum in der *Repository*-Sicht. Beispielsweise können Sie einen unbenannten (und damit nicht wiederverwendbaren) Ausdruck im Kontext einer Funktion anlegen, indem Sie im Objektmenü des Feldes *Oberster Ausdruck* über *Anlegen...* einen neuen obersten Ausdruck anlegen.

Objektschachtelung und unbenannte Objekte

Statt Objekte über den Objektbaum im Navigationsbereich der Workbench anzulegen, können Sie in einigen Situationen ein neues Objekt direkt in ein vorhandenes Objekt einbetten. Dies ist für folgende Aufgaben möglich:

- Zuordnung eines neuen Ausdrucks zu einem übergeordneten Objekt (z.B. Zuordnung eines obersten Ausdrucks zu einer Funktion, eines Bereichsausdrucks zu der Zelle einer Entscheidungstabelle, usw.)
- Zuordnung einer neuen Regel zu einem Regelsatz

Das Anlegen von Objekten innerhalb von Objekten ist bequem, da Sie den Kontext, in dem Sie arbeiten, nicht verlassen müssen. Außerdem ist es die einzige Möglichkeit Objekte anzulegen, ohne einen Objektnamen zu definieren. Das Arbeiten mit unbenannten Objekten hat folgende Konsequenzen:

- Unbenannte Objekte können nur von dem Container-Objekt verwendet werden, in dem sie angelegt wurden.
- Unbenannte Objekte werden weder im Repository-Objektbaum noch im Objektabfragedialog noch in der Ergebnisliste der erweiterten Suche angezeigt.
- Folglich sind unbenannte Objekte nicht wiederverwendbar.
- Wenn das Container-Objekt gelöscht wird, bleiben seine enthaltenen unbenannten Objekte im System. Da diese Objekte dann unsichtbar sind, können Sie nicht darauf zugreifen und sie wie andere Objekte löschen. Um nicht verwendete unbenannte Objekte aus dem System zu löschen, wählen Sie ► *Werkzeuge* ► *Anwendungsverwaltung* ► und führen Sie die Operation *Nicht verwendete Objekte löschen* aus.

Implizites Anlegen von Objekten über Datenbindung

Sie können Datenobjekte jedes Typs (Element, Struktur, Tabelle) nicht nur über die manuelle Definition ihrer Attribute anlegen, sondern auch indem Sie eine Referenz zu einem Objekt definieren, das bereits im Data Dictionary des Backend-Systems vorhanden ist. Diese Art des Anlegens von Objekten wird Datenbindung genannt. Wenn Sie diese Option verwenden, bestimmt das System die Abhängigkeiten des referenzierten Dictionary-Objekts und versucht automatisch, das Dictionary-Objekt auf BRFPplus-Seite bestmöglich darzustellen. Je nach Situation im Dictionary können Sie mit dieser Option eine erhebliche Zeiteinsparung erreichen. Das System kann Folgendes tun:

- Element-Datenobjekte

Wenn Sie ein Datenobjekt des Typs Element an ein Datenelement im Dictionary binden, ermittelt das System, ob das Datenelement selbst an eine Domäne gebunden ist. Wenn dies der Fall ist, füllt das System das BRFplus-Datenobjekt mit allen im Dictionary definierten Domänenwerten.

- **Strukturen und Tabellen**

Wenn Sie ein Datenobjekt des Typs Struktur oder Tabelle an eine Dictionary-Struktur oder -Tabelle binden, legt das System automatisch ein Element-Datenobjekt für jedes Feld in der Struktur oder Tabelle an. Zusätzlich legt das System eine Bindungsbeziehung zum entsprechenden Dictionary-Datenelement an. Für jedes Element-Datenobjekt importiert das System auch hier die Domänenwerte, die im Dictionary für dieses Feld definiert sind.

Weitere Informationen

[Verwaltungsdaten \[Seite 39\]](#)

[Namenskonventionen für BRFplus \[Seite 36\]](#)

[Datenbindung \[Seite 66\]](#)

[Objekte löschen \[Seite 57\]](#)

1.2.7 Objekte suchen

Verwendung

Der Bestand an Objekten in BRFplus kann erfahrungsgemäß schnell wachsen. Dadurch wird es mit der Zeit immer schwieriger, ein bestimmtes Objekt zu finden. Um Ihnen die Suche nach Objekten zu erleichtern, stellt die BRFplus-Workbench daher verschiedene Methoden und Werkzeuge zur Verfügung:

- **Kataloge**

In Katalogen können Sie Gruppen von Objekten zusammenfassen, die zu einer bestimmten Geschäftsdomäne gehören. Dies ermöglicht es Ihnen, andere Objekte im Repository, die nicht zu dieser Geschäftsdomäne zählen, auszublenden. Sie können auch verschiedene Kataloge für dieselbe Domäne definieren und dort Objekte basierend auf verschiedenen Aufgaben gruppieren. Beispielsweise können Sie einen Katalog für Endbenutzer festlegen und einen mit Objekten für die Regelverwaltung. Weitere Informationen finden Sie unter [Katalog \[Seite 171\]](#).

- **Workbench-Views**

Mithilfe der Workbench-Views [Meine Anwendungen](#), [Suchergebnis](#), [Favoriten](#) und [Zuletzt verwendet](#) können Sie die Objekte gemäß verschiedener Strategien anzeigen. Weitere Informationen finden Sie unter [Die Benutzungsoberfläche \[Seite 7\]](#).

- **Objektsuche**

BRFplus stellt eine mächtige Objektsuchfunktion bereit, die Ihnen das Suchen von Objekten erleichtert. In den folgenden Abschnitten wird die Objektsuche in BRFplus im Detail beschrieben.

Funktionsumfang

Eine Suche starten

Es gibt verschiedene Möglichkeiten, die Suche zu starten. Das Dialogfenster für die Suche wird abhängig von Ihrem aktuellen Arbeitskontext jeweils leicht unterschiedlich angezeigt, um Sie bestmöglich zu unterstützen:

- Drucktaste im Navigationsbereich
Die Drucktaste "Suche" im Navigationsbereich ist der schnellste Weg, die Suche zu starten. Das Suche-Dialogfenster wird in seiner Standardansicht angezeigt. Hier können Sie Ihre Suchkriterien wie gewünscht angeben. Das Suchergebnis wird im Navigationsbereich der BRFPplus-Workbench angezeigt.
- Suche nach einer Objekt-ID
Jedes BRFPplus-Objekt, das Sie anlegen, erhält automatisch eine eindeutige ID, die als Objekteigenschaft gespeichert wird. Ein Objekt kann nur über seine ID eindeutig ermittelt werden, da ein technischer Name in anwendungsübergreifenden Szenarien häufiger als einmal vorkommen kann. Wenn Sie die ID eines Objekts kennen, können Sie es über das Workbench-Menü schnell finden, indem Sie ► [Workbench](#) ► [Objekt öffnen](#) ► wählen. Hier wird wieder das Suche-Dialogfenster angezeigt, bei dem nun das erste Suchkriterium auf [Objekt-ID ist gleich](#) voreingestellt ist. Sie müssen lediglich die ID in das Eingabefeld eintragen und können dann die Suche starten. Wenn es ein entsprechendes Objekt mit dieser ID gibt, wird es direkt in den Arbeitsbereich der Workbench geladen.
- Suche nach Objekten für die Zuordnung zu anderen Objekten während der Pflege
Beim Aufbau eines Regelmodells mit BRFPplus steht die höchstmögliche Wiederverwendbarkeit bestehender Objekte im Vordergrund. Aus diesem Grund suchen Regelmodellierer häufig nach vorhandenen Objekten, die sie dem Objekt zuordnen können, an dem sie gerade arbeiten. In diesen Situationen stellt die BRFPplus-Workbench an den Stellen ein Objektmenü bereit, an denen das Objekt zugeordnet werden soll. Wenn Sie in diesem Menü die Option [Auswählen...](#) öffnen, wird das Suche-Dialogfenster angezeigt. Es ist in einigen Suchfeldern bereits mit passenden Werten vorausgefüllt, startet die Suche und zeigt die Suchergebnisse in einer Ergebnisliste unterhalb der Suchkriterien an. Sie können dann entweder eines der Objekte aus der Ergebnisliste auswählen oder die Suchkriterien anpassen und die Suche wiederholen, bis das gewünschte Objekt in der Ergebnisliste angezeigt wird.

Suchkriterien anpassen

Die BRFPplus-Workbench ist nicht nur in Bezug auf die frei definierbaren Suchbegriffe flexibel, sondern auch auch beim Layout und den vordefinierten Einschränkungen selbst. Sie können folgende Anpassungen vornehmen:

- Anzahl an Suchkriterien
Standardmäßig wird ein Dialogfenster angezeigt, das drei Zeilen für Ihre Suchkriterien enthält. In jeder Zeile können Sie aus einer jeweiligen Liste das gewünschte Suchkriterium, den Vergleichsoperator und den Wert auswählen, nach dem Sie suchen. Wenn Sie mehr als ein Suchkriterium verwenden, hängt das Verhalten der Suche davon ab, ob Sie verschiedene Suchkriterien definiert haben oder das gleiche Suchkriterium mehrmals, jedoch mit verschiedenen Werten verwenden: Wenn Sie mindestens zwei verschiedene Kriterien definieren, werden diese intern über den logischen Operator UND verbunden. Je mehr Kriterien Sie also festlegen, desto genauer sind Ihre Suchergebnisse.
 - Verschiedene Kriterien: Wenn Sie mindestens zwei verschiedene Kriterien definieren, werden diese intern über den logischen Operator UND verbunden. Je mehr Kriterien Sie also festlegen, desto genauer sind Ihre Suchergebnisse.
 - Gleiches Kriterium mit verschiedenen Werten: Wenn Sie mindestens zwei Zeilen für das gleiche Kriterium festlegen, die jedoch unterschiedliche Werte haben, werden diese Zeilen intern über den

logischen Operator ODER verbunden. Je mehr Kriterien Sie also festlegen, desto mehr Suchergebnisse können gefunden werden.

Wenn Sie für Ihre Suche mehr als drei Zeilen benötigen, können Sie die Drucktaste "+" am Ende einer Kriterienszeile wählen. Es wird dann eine weitere Zeile hinzugefügt, bei der standardmäßig das gleiche Suchkriterium vorausgewählt ist, wie bei der Zeile, neben der Sie die Drucktaste gewählt haben. Wenn Sie stattdessen lieber weniger Suchkriterien bzw. Zeilen sehen möchten, können Sie die Drucktaste "-" wählen. Das System stellt allerdings sicher, dass mindestens ein Suchkriterium im Suchdialog angezeigt wird.

- Gegenseitige Beziehung von Suchkriterien

Im Allgemeinen sind die Suchkriterien, die Sie in einer Suche verwenden, voneinander unabhängig und können kombiniert werden. Beachten Sie jedoch, dass Sie über das Hinzufügen von Suchkriterien das Suchergebnis einschränken.

Trotz ihrer Unabhängigkeit sind einige der verfügbaren Suchkriterien miteinander verwandt. Diese verwandten Suchkriterien haben hauptsächlich den Zweck, eine Suche mit weniger Parametern zu verfeinern. Im Folgenden ist ein Beispiel einer Suche aufgeführt, die verschiedene Parametersätze verwendet:

Beispiel

Sie möchten die Datenbank nach Datenobjekten des Typs Tabelle durchsuchen, deren Name mit "ZTBL_CRM_" beginnt. Sie möchten diese Tabelle in allen Anwendungen suchen, die Sie für CRM-bezogene Anwendungsfälle angelegt haben. In einem ersten Ansatz definieren Sie die Suche über generische Kriterien, die alle BRFplus-Objektypen gemeinsam haben:

- *Anwendungsname ist gleich *CRM**
- *Objektyp ist gleich Datenobjekt*
- *Name ist gleich ZTBL_CRM_**

Dieser Satz an Parametern könnte für Ihre Suche zu Problemen führen, wenn das Namenspräfix ZTBL_CRM_ nicht nur für Tabellen, sondern auch für Strukturen oder Elemente verwendet wurde. Sie können daher Ihre Suche weiter einschränken, indem Sie eines der spezifischeren Suchkriterien verwenden:

- *Anwendungsname ist gleich *CRM**
- *Datenobjektyp ist gleich Tabelle*
- *Name ist gleich ZTBL_CRM_**

Hier wurde das generische Kriterium *Objektyp* durch das Kriterium *Datenobjektyp* ersetzt. Mit diesem Kriterium geht das System implizit davon aus, dass die Suche nur Datenobjekte durchsucht und alle anderen Objektypen nicht berücksichtigt. Es ist nicht notwendig, das Kriterium *Objektyp ist gleich Datenobjekt* in der Suche beizubehalten.

- Einschließen von Standard-BRFplus-Objekten

Wenn Sie BRFplus-Ausdrücke oder -Regeln definieren, besteht ein wichtiger Schritt darin, dem aktuell zu bearbeitenden Objekt verschiedene Objekte zuzuordnen. Jeder Ausdruck erfordert beispielsweise ein zugeordnetes Datenobjekt und fast alle Ausdrücke benötigen Kontextdatenobjekte. Üblicherweise nehmen Sie eine entsprechende Zuordnung vor, indem Sie aus dem Objektmenü neben der Position des zuzuordnenden Objekts die Option *Auswählen...* wählen. Das System zeigt ein Dialogfenster zur Objektsuche an, in dem bereits der richtige Objektyp vorausgewählt ist und die Ergebnisliste für diese Vorauswahl angezeigt wird. Zur Veranschaulichung: Wenn ein Ausdruck ein Ergebnisdatenobjekt des elementaren Datentyps Zahl benötigt, werden alle Datenobjekte des Typs Element mit der Elementart Zahl in der Ergebnisliste angezeigt, die für die Zuordnung verfügbar sind.

Sie können alternativ eines der mit BRFplus ausgelieferten vordefinierten Datenobjekte für die Zuordnung verwenden, indem Sie im Dialogfenster das Ankreuzfeld *Auch Objekte aus der Standard-BRFplus-Anwendung einschließen* markieren. Das System zeigt in diesem Fall zusätzlich zu Ihren eigenen Objekten die möglichen vordefinierten Objekte als Suchergebnis an.

- **Maximale Anzahl von Ergebnissen**
Wenn es absehbar ist, dass eine bestimmte Suche weitaus mehr Suchergebnisse enthalten wird als benötigt, können Sie die Anzahl der angezeigten Suchergebnisse begrenzen. Ist die Anzahl der Suchergebnisse größer als das aktuelle Maximum, werden Sie vom System entsprechend benachrichtigt. Der im Dialogfenster voreingestellte Wert der maximalen Anzahl von Ergebnissen kann angepasst werden. Sie können den Voreinstellungswert in den Workbench-Personalisierungseinstellungen entsprechend ändern. Der Mindestwert beträgt 10.
- **Suchkriterien ein-/ausblenden**
Wenn Sie viele Suchkriterien definiert haben, kann es vorkommen, dass der Bereich der Suchkriterien weitaus mehr Platz im Dialogfenster einnimmt als die Liste der Suchergebnisse. In diesem Fall können Sie die Höhe der Ergebnisliste erweitern, indem Sie die Suchkriterien ausblenden. Hierdurch füllt die Liste den gesamten verfügbaren Platz im Dialogfenster aus.

Besondere Auswahlkriterien

Die meisten der im Suchdialog bereitgestellten Auswahlkriterien sind selbsterklärend. Allerdings sollten für ein besseres Verständnis folgende Kriterien näher erläutert werden:


Tabelle 9:

Suchkriterium	Beschreibung
<i>Anwendungsname</i>	<p>Es wird fälschlicherweise häufig versucht, über das Suchkriterium <i>Anwendungsname</i> eine Anwendung zu finden. Allerdings werden bei einer solchen Suche alle Objekte in der Ergebnisliste angezeigt, die in dieser Anwendung vorhanden sind, einschließlich der Anwendung selbst. Der Grund hierfür ist, dass der <i>Anwendungsname</i> ein administratives Datenfeld ist, das in allen BRFplus-Objekten enthalten ist. Eine entsprechende Suche beschränkt sich demnach nicht auf Objekte des Typs Anwendung.</p> <p>Wenn Sie nach einem Anwendungsobjekt suchen möchten, gehen Sie wie folgt vor:</p> <ul style="list-style-type: none"> • <i>Objektyp ist gleich Anwendung</i> • <i>Name ist gleich <Ihr Anwendungsname></i>
<i>Aktionstyp</i> <i>Ausdrucksart</i> <i>Datenobjektyp</i>	<p>Mit diesen drei Suchkriterien können Sie Ihre Suchdefinition vereinfachen. Sie alle legen implizit bereits den gesuchten Objektyp fest, nämlich eine Aktion, ein Ausdruck oder ein Datenobjekt. Daher ist es für diese Suchkriterien nicht notwendig, ein zusätzliches Suchkriterium <i>Objektyp</i> zu definieren.</p>

Suchkriterium	Beschreibung
<i>Elementart</i>	Dies ist eine weitere Verfeinerung des oben genannten Suchkriteriums <i>Datenobjekttyp</i> . Hiermit können Sie über ein einzelnes Suchkriterium nach Datenobjekten des Typs Element suchen, die wiederum beispielsweise die Elementart Text oder Zahl aufweisen. Es ist nicht notwendig, den <i>Objekttyp</i> explizit auf Datenobjekt einzuschränken und ein weiteres Kriterium <i>Datenobjekt ist gleich Element</i> zu definieren.

Beispiel

Während eines Regelimplementierungsprojekts möchten Sie herausfinden, ob die Namen aller Formelausdrücke in einer bestimmten BRFplus-Anwendung den Namenskonventionen entsprechen, die für dieses Projekt festgelegt wurden. Sie möchten dabei vor allem Formelausdrücke berücksichtigen, die vor dem Jahr 2012 angelegt wurden. Gehen Sie hierzu folgendermaßen vor:

1. Wählen Sie in der BRFplus-Workbench .
2. Stellen Sie im Dialogfenster *Personalisierung* sicher, dass auf der Registerkarte *Allgemein* das Ankreuzfeld *Technische Namen anzeigen* markiert ist.
3. Stellen Sie sicher, dass in der Workbench im Navigationsbereich das *Repository* ausgewählt ist.
4. Wählen Sie *Suche*.
5. Definieren Sie im Dialogfenster *Suche* folgende Suchkriterien:
 - *Anwendungsname ist gleich* **<Ihr Anwendungsname>**
 - *Objekttyp ist gleich* **Ausdruck**
 - *Ausdrucksart ist gleich* **Formel**
 - *Angelegt am ist kleiner als* **01.01.2012**

In der folgenden Grafik ist ein Dialogfenster mit diesen Einstellungen dargestellt:

Search [Close]

Search Criteria

Application Name	is equal to	Z_SUPPLIER_DETE	+ -
Object Type	is equal to	Expression	+ -
Expression Type	is equal to	Formula	+ -
Created on	is less than	01.01.2012	+ -

Also include objects from default BRFplus application: ☐

Maximum Number of Results: 200

Search Clear Reset

Ok Cancel

Abbildung 3: Suchdialog mit den oben beschriebenen Suchkriterien

6. Starten Sie die Suche mit [Suchen](#).

Das System führt eine Suche gemäß Ihren Suchkriterien durch und listet die gefundenen Objekte im Navigationsbereich der BRFplus-Workbench auf. Da Sie zu Beginn festgelegt haben, dass die Objekte mit ihren technischen Namen angezeigt werden, können Sie in der ausgegebenen Liste auf einen Blick sehen, ob die Namenskonventionen eingehalten wurden oder nicht.

Weitere Informationen

[Katalog \[Seite 171\]](#)

[Workbench-Personalisierung \[Seite 12\]](#)

[Die Benutzungsoberfläche \[Seite 7\]](#)

1.2.8 Objekte löschen

Verwendung

Das Löschen von Objekten aus der BRFplus-Datenbank ist ein genau definierter Prozess, der mehrere Schritte umfasst. Mit Hilfe dieses Prozesses können Sie Ihre Arbeitsumgebung aufräumen können, ohne dabei einen unbeabsichtigten Datenverlust zu riskieren. Er umfasst verschiedene vorbereitende und endgültige Schritte des Löschens, bei denen die Verfügbarkeit und Sichtbarkeit von Objekten von nach und nach geringer wird. Ein

Objekt kann nur dann physisch und unwiderruflich gelöscht werden, wenn es die meisten (oder sogar alle) dieser Schritte durchlaufen hat. Dank dieses verzögerten und schrittweisen Ansatzes können Sie in BRFplus Ihre Objekte selbst dann noch verwalten, wenn sie auf den ersten Blick gelöscht sind. Solange ein Objekt nicht physisch gelöscht wurde, kann ein Benutzer mit Administrationsrechten es reaktivieren und zu Revisions- bzw. Prüfungszwecken oder sogar zur produktiven Verwendung verfügbar machen.

Funktionsumfang

Aus technischer Sicht sind die verschiedenen Schritte bis zum endgültigen, physischen Löschen eines Objekts nicht als eigentliches Löschen, sondern als Zuordnung interner Status umgesetzt. Dadurch kann ein Objekt in verschiedenen Stufen des Löschens vorliegen, wobei jede Stufe eine Reihe von Beschränkungen enthält, die auf das Objekt in diesem jeweiligen Status zutreffen. Lediglich der letzte Schritt in diesem Prozess (physisches Löschen) wird intern nicht mehr als Statuszuordnung, sondern als physische Löschung eines Objekts aus der Systemdatenbank ausgeführt. Nach diesem Schritt ist ein Objekt unwiderruflich gelöscht. In den folgenden Abschnitten werden die Schritte eines Objekts von der produktiven Verwendung bis zum physischen Löschen beschrieben.

Objekte als veraltet kennzeichnen

Bei der Überprüfung Ihrer Geschäftsregeln werden Sie vielleicht von Zeit zu Zeit feststellen, dass einige Objekte Ihre jeweiligen Geschäftsprozesse nicht mehr richtig widerspiegeln. Diese Objekte mögen gegenwärtig ihren Zweck noch vollkommen erfüllen. Allerdings sehen Sie vielleicht voraus, dass zukünftige Änderungen in Ihrer Geschäftsumgebung möglicherweise Änderungen am Regelobjekt erforderlich machen werden.



Beispiel

In einer BRFplus-Anwendung zur Berechnung der Einkommensteuer haben Sie eine Beziehung zwischen Einkommen und Steuer modelliert. Hierzu haben Sie eine Entscheidungstabelle verwendet, bei der jede Zeile einen bestimmten Einkommensbereich sowie die für dieses Einkommen zu zahlende Steuer enthält. Die Regierung Ihres Landes hat nun jedoch angekündigt, das Steuersystem grundsätzlich zu ändern. Es ist zukünftig eine Reihe komplizierter Berechnungsschritte nötig, um die rechtlich vorgeschriebene Steuer eines jeweiligen Einkommens zu bestimmen.

In dieser Situation ist es ersichtlich, dass der bisher ausreichende Ansatz die zukünftigen Voraussetzungen nicht mehr erfüllen wird. Angesichts dieser Tatsache kennzeichnen Sie die Entscheidungstabelle als veraltet. Dadurch erinnern Sie sich daran, dass es nichts nützt, die Entscheidungstabelle an das neue Besteuerungssystem anpassen zu wollen. Stattdessen müssten Sie sich einen anderen Ansatz mit geeigneteren Ausdrucksarten wie beispielsweise Formelausdrücken überlegen.

Wenn Sie ein Objekt als veraltet kennzeichnen, hat das folgende Auswirkungen:

- Ein veraltetes Objekt kann nicht geändert werden.
- Das Kennzeichnen eines Objekts als veraltet hat **keine** Auswirkungen auf seinen Aktivierungsstatus.
- Regeln und Ausdrücke, die ein veraltetes, aber aktives Objekt bereits verwenden, bleiben weiterhin funktionstüchtig und bereit für die produktive Verwendung.
- Inaktive Regeln und Ausdrücke, die bereits ein veraltetes Objekt verwenden, können nicht aktiviert werden.
- Ein veraltetes Objekt kann nicht erneut einem anderen Objekt zugeordnet werden. Diese Einschränkung wird im System durchgesetzt, indem veraltete Objekte in verschiedenen Auswahllisten ausgeblendet werden.

- Das Veraltet-Kennzeichen eines Objekts kann jederzeit wieder entfernt werden. Wenn das Kennzeichen entfernt wird, befindet sich das Objekt wieder in demselben Zustand, in dem es sich vor seiner Kennzeichnung als veraltet befunden hatte.

i Hinweis

Das Kennzeichnen eines Objekts als veraltet ist **keine** Voraussetzung für das Löschen des Objekts. Es gibt im System **keinen** Workflow oder Zustandsübergang, bei dem Sie ein Objekt vor dem Löschen zwingend als veraltet kennzeichnen müssten. Wenn Sie ein Objekt als veraltet ansehen, führt dies jedoch in vielen Fällen letztendlich zum Löschen des Objekts. Zusätzlich hat das Kennzeichnen von Objekten als veraltet ähnliche Folgen wie das Vormerken von Objekten zum Löschen. Deshalb wird das Kennzeichnen von Objekten als veraltet in diesem Abschnitt zusammen mit dem Vormerken von Objekten zum Löschen behandelt.

Objekte zum Löschen vormerken

Wenn Sie ein Objekt zum Löschen vormerken, machen Sie damit deutlich, dass es veraltet ist und nicht mehr verwendet werden soll. Das Vormerken eines Objekts zum Löschen hat dieselben Auswirkungen, wie oben für veraltete Objekte aufgeführt, sowie eine weitere Folge:

- Objekte, die zum Löschen vorgemerkt sind, können zentral vom System entfernt werden. Dies ist entweder über das Werkzeug zur Anwendungsverwaltung möglich (logisches Löschen, siehe unteren Abschnitt) oder über den Report `FDT_DELETE`, der entweder bei Bedarf oder nach einem benutzerdefinierten Zeitplan ausgeführt wird. Mit diesem Report können Sie Objekte löschen, die mindestens seit einer vorgegebenen Verweildauer zum Löschen vorgemerkt sind.

Daher ist es naheliegend, dass Sie ein zum Löschen vorgemerktes Objekt so schnell wie möglich überall dort ersetzen sollten, wo es verwendet wird. Dies trifft auch trotz der Tatsache zu, dass ein zum Löschen vorgemerktes Objekt in allen Kontexten, in denen es verwendet wird, weiterhin funktionstüchtig bleibt.

Objekte löschen (logisches Löschen)

Das logische Löschen von BRFplus-Objekten ist weitestgehend identisch mit Löschvorgängen, die Sie vielleicht aus anderen Anwendungen kennen. Wenn Sie ein Objekt gelöscht haben, wird es aus allen Objektlisten wie dem Repository-Baum, der Liste zuletzt verwendeter Anwendungen oder der Favoritenliste entfernt. Der Benutzer hat keine Möglichkeit, ein logisch gelöscht Objekt wiederherzustellen. Für das Löschen von Objekten gilt:

- Sie können ein Objekt nur dann löschen, wenn es nicht von einem anderen Objekt verwendet wird. Bedenken Sie, dass bei versionierten Objekten Verwendungsbeziehungen zwischen dem Objekt, das Sie löschen möchten und historischen Versionen eines anderen Objekts bestehen können. In diesem Fall müssen Sie zuerst die veraltete Verwendung entfernen, bevor Sie das referenzierte Objekt löschen können.
- Es empfiehlt sich, Objekte erst zu löschen, nachdem sie als veraltet gekennzeichnet oder zum Löschen vorgemerkt wurden. Dies ist allerdings **keine** technische Voraussetzung. Sie können Objekte bei Bedarf auch unabhängig von ihrem aktuellen Status löschen, vorausgesetzt sie werden nicht von anderen Objekten verwendet.
- Sie können Objekte auf folgende Arten logisch löschen:
 - Manuell, indem Sie zu dem entsprechenden Objekt navigieren und den Befehl [Löschen](#) für dieses Objekt verwenden.
 - Automatisch, indem Sie den Befehl [Zur Löschung vorgemerkte Objekte löschen](#) im Werkzeug zur Anwendungsverwaltung verwenden. Dieser Befehl hat lediglich auf Objekte Auswirkungen, die zum Löschen vorgemerkt wurden. Außerdem müssen die zu löschenden Objekte bereits mindestens seit der definierten Verweildauer zum Löschen vorgemerkt sein (Standard: 90 Tage).

- Sobald ein Objekt logisch gelöscht wurde, ist es in keiner Objektliste in BRFplus mehr sichtbar.
- BRFPplus bietet keinerlei Werkzeuge zur Wiederherstellung von gelöschten Objekten. Systemverwalter können Objekte jedoch trotzdem wiederherstellen, da selbst gelöschte Objekte **nicht** sofort physisch vom System entfernt werden. Weitere Informationen finden Sie im folgenden Abschnitt *Objekte Löschen (physisches Löschen)*.

Objekte löschen (physisches Löschen)

Sobald ein Objekt logisch gelöscht wurde, ist es nicht mehr in BRFPplus verfügbar. Allerdings führt das logische Löschen nicht umgehend dazu, dass das Objekt physisch gelöscht ist. Vielmehr befindet sich das gelöschte Objekt noch immer im System. Es ist dort jedoch so verborgen, dass es in BRFPplus keine Möglichkeit gibt, es wieder bereitzustellen.

Es gibt verschiedene Möglichkeiten, ein gelöschtes Objekt endgültig aus der Datenbank zu entfernen:

- Wählen Sie in der BRFPplus-Workbench **Werkzeuge** > **Anwendungsverwaltung**. Wählen Sie unter **Operation** aus der Liste der Operationen **Datenbank aufräumen**.
- Führen Sie im Backend-System den Report `FDT_DELETE` aus.
- Räumen Sie mit einem automatischen Bereinigungsverfahren die Datenbank auf. Dieser Bereinigungsverfahren erfolgt ohne manuelles Eingreifen. Er entfernt Objekte aus der Datenbank, die seit mindestens 180 Tagen logisch gelöscht sind.

Solange das physische Löschen nicht stattgefunden hat, ist das Objekt im System vorhanden und kann von einem Systemverwalter wiederhergestellt werden, wenn es aus zwingenden Gründen wieder eingesehen werden muss (beispielsweise im Zuge rechtlicher Verfahren).

Aktivitäten

Sie können Aktivitäten, die mit dem Löschen von Objekten in Zusammenhang stehen, auf zwei Arten ausführen; entweder einzeln für jedes Objekt oder als Massenaktivität.

- Einzelne Objekte
Um ein einzelnes Objekt als veraltet zu kennzeichnen, gehen Sie wie folgt vor:
 1. Öffnen Sie das zu kennzeichnende Objekt in der BRFPplus-Workbench.
 2. Wählen Sie **Mehr** > **Als veraltet kennzeichnen**.
 Um ein einzelnes Objekt zum Löschen vorzumerken, gehen Sie wie folgt vor:
 1. Öffnen Sie das vorzumerkende Objekt in der BRFPplus-Workbench.
 2. Wählen Sie **Löschen** > **Zum Löschen vormerken**.
 Um ein einzelnes Objekt zu löschen, gehen Sie wie folgt vor:
 1. Öffnen Sie das zu löschende Objekt in der BRFPplus-Workbench.
 2. Wählen Sie **Mehr** > **Verwendungsnachweis anzeigen**, um sicherzugehen, dass das Objekt von keinem anderen Objekt verwendet wird.
 3. Wählen Sie **Löschen** > **Löschen**.
- Mehrere Objekte
Um Aktivitäten, die mit dem Löschen von Objekten in Zusammenhang stehen, für mehrere Objekte gleichzeitig auszuführen, gehen Sie wie folgt vor:
 1. Wählen Sie **Werkzeuge** > **Massenänderung**.
 2. Wählen Sie im Werkzeug zur Massenänderung die Objekte aus, die Sie ändern möchten.

3. Wählen Sie *Aktionen für markierte Positionen durchführen: Als veraltet kennzeichnen, zum Löschen vormerken, Löschen*, und anschließend die gewünschte Option.

Weitere Informationen

[Massenänderung \[Seite 278\]](#)

1.2.9 Datenobjekte und Datenobjekttypen

Verwendung

Ein Datenobjekt beschreibt Daten und dient im Kontext oder Ergebnis von Ausdrücken oder Aktionen als Datenträger. Datenobjekte können über ihren Typ näher beschrieben werden. Die möglichen Datenobjekttypen sind Element, Struktur und Tabelle.

Für Datenobjekte vom Typ Element können Sie die Menge an Vergleichsoperationen definieren, die von einem bestimmten Datenobjekt unterstützt werden. Dies ist hilfreich, wenn ein Datenobjekt in einem Vergleich mit einem Bereichsausdruck verwendet wird. Wenn ein Datenobjekt die entsprechende Vergleichsoperation nicht unterstützt, kann das System diese Diskrepanz bereits bei der Modellierung feststellen und möglichen Problemen zur Laufzeit vorbeugen.

Ein Element-Datenobjekt muss einen der folgenden Untertypen aufweisen:

Tabelle 10:

Elementtyp	Text (Beschreibung)
Text	<p>Zeichenkette (maximal 255 Zeichen)</p> <div><p>i Hinweis</p><p>Die Maximallänge von Text-Datenobjekten ist auf die maximale Länge von Zeichenketten zurückzuführen, die in der Datenbank gespeichert werden können. Trotz dieser Beschränkungen für persistente Zeichenketten können Sie wesentlich längere Zeichenketten verwenden, solange diese nur für interne Berechnungen während einer BRFplus-Session eingesetzt werden. So können Sie beispielsweise in einer BRFplus-Formel mehrere Zeichenketten zu einer langen Zeichenkette verketteten, anschließend diese Zeichenkette nach einem bestimmten Muster durchsuchen und schließlich ein Formelergebnis zurückliefern (zum Beispiel wie häufig das Muster in der langen Zeichenkette vorkommt).</p></div>

Elementtyp	Text (Beschreibung)
Zahl	<p>Zahl mit Dezimalstellen</p> <p>Wenn ein Datenobjekt an ein ABAP-Dictionary-Datenelement des Typs NUMC gebunden ist, wird der Typ des Datenobjekts automatisch auf Text statt auf Zahl gesetzt. Weitere Informationen finden Sie im Abschnitt <i>Typableitung</i> in Datenbindung [Seite 66].</p>
Boolesch	Boolescher Wert
Betrag	<p>Zahl und Währung</p> <div> <p>i Hinweis</p> <p>Die Anzahl an Dezimalstellen des Zahl-Teils eines Betrags beträgt immer zehn und kann vom Benutzer nicht geändert werden. Dies liegt daran, dass diese Einstellung nur aus formalen Gründen vorhanden ist. Zur Laufzeit wird die eigentliche Anzahl der verwendeten Dezimalstellen von den zentralen Währungstabellen im Backend abgeleitet. Die Anzahl der Dezimalstellen ist somit währungsabhängig. Aus diesem Grund können Sie diese formale Einstellung in der Personalisierungsfunktion der BRFplus-Workbench ausblenden, indem Sie die Option Allgemein > Anzeigen > Technische Attribute anzeigen verwenden.</p> </div>

Elementtyp	Text (Beschreibung)
Menge	<p data-bbox="804 367 1043 394">Zahl und Mengeneinheit</p> <p data-bbox="804 418 1390 551">Zusätzlich zur obligatorischen Definition der Mengeneinheit zur Laufzeit können Sie optional während der Modellierung die Mengendimension angeben (beispielsweise Länge, Masse, Zeit). Diese Angabe hat folgende Auswirkungen:</p> <ul data-bbox="804 568 1390 842" style="list-style-type: none"> <li data-bbox="804 568 1390 629">• Zur Laufzeit prüft das System, ob die bereitgestellte Mengeneinheit für die angegebene Dimension gültig ist. <li data-bbox="804 640 1390 842">• Wenn Datenobjekte mit angegebener Mengendimension in Berechnungen eines Formelausdrucks verwendet werden, führt das System zusätzliche Formel-Konsistenzprüfungen durch, um semantische Fehler auszuschließen, die zur Laufzeit zu einem Fehler führen könnten. <div data-bbox="804 869 1390 1532"> <p data-bbox="804 875 970 909">i Hinweis</p> <p data-bbox="804 934 1382 1514">Selbst wenn es in der zentralen Dimensionstabelle eine vordefinierte Anzahl an anzuzeigenden Dezimalstellen für eine bestimmte Mengeneinheit gibt, hat die vom Benutzer angegebene Anzahl an Dezimalstellen für das Datenelement Vorrang vor diesem Systemstandard. Dadurch wird die Transparenz mengenbasierter Entscheidungen sichergestellt. Andernfalls könnten Situationen auftreten, in denen zwei Werte mit leicht unterschiedlichen Dezimalstellen nach dem Runden mit gleichen Dezimalzahlen angezeigt werden. Dies könnte die Frage aufwerfen, warum bei der Regelauswertung gerade diesen Wert verwendet. Um solche Mehrdeutigkeiten zu vermeiden, und aufgrund der Tatsache, dass BRFplus ein generisches Werkzeug ist, das nicht an die Bedürfnisse eines bestimmten Geschäftsszenarios angepasst ist, werden in BRFplus Mengen immer mit der benutzerdefinierten Anzahl an Dezimalstellen angezeigt.</p> </div>

Elementtyp	Text (Beschreibung)
Zeitpunkt	<ul style="list-style-type: none"> • Ortsdatum • Ortszeit • Ortsdatum und -zeit • Zeitstempel im UTC-Format • Ortsdatum und -zeit mit Zeitunterschied zur UTC-Zeit <div> <p>i Hinweis</p> <p>Bei den ersten drei der oben aufgelisteten Zeitpunktarten handelt es sich um abstrakte Werte, die keine expliziten Informationen zur Zeitzone enthalten. Wenn Sie genaue und Eindeutige Informationen zu Datum und Uhrzeit benötigen, verwenden Sie die beiden UTC-basierten Zeitpunktarten.</p> <p>Bei einem Zeitpunkt-Datenobjekt ist es nicht notwendig, die Zeitpunktart des Objekts zur Design-Zeit festzulegen. Sie können diesen Teil der Objektdefinition offen lassen. Dadurch kann das Objekt beliebige, gültige Daten zu Datum und Uhrzeit enthalten. Dies ist für Szenarios hilfreich, bei denen die genaue Zeitpunktart einer eingehenden Datums-/Uhrzeitangabe unbekannt ist. Wenn Sie die Zeitpunktart-Einstellung undefiniert lassen, müssen Sie jedoch die Objekte, die das Zeitpunkt-Element verwenden, besonders beachten, da für Zeitpunkte ohne angegebene Zeitpunktart keine typspezifischen Konsistenzprüfungen möglich sind.</p> </div>

Statt explizit den Typ eines Element-Datenobjekts zu definieren, können Objekte den Typ und die Objekteigenschaften auch von einem anderen Objekt erben. Dieses andere Objekt kann entweder im BRFPplus-Repository vorhanden sein oder in einem anderen Repository im Backend-System. Weitere Informationen finden Sie unter [Datenbindung \[Seite 66\]](#).

Struktur

In der folgenden Tabelle sind die strukturellen Beziehungen zwischen den verschiedenen Datenobjekttypen aufgelistet:

Tabelle 11:

Datenobjekttyp	Besteht aus
Element	Vordefinierte Datentypen (siehe Tabelle oben)

Datenobjekttyp	Besteht aus
Struktur	Element (0..n) Struktur (0..n) Tabelle (0..n)
Tabelle	Element (0..1) Struktur (0..1)

Für ein besseres Verständnis dieser Beziehungen ist es hilfreich zu wissen, wie die verschiedenen Datenobjekttypen aufeinander aufgebaut sind:

- Ein Element ist ein vordefinierter, elementarer Datentyp, der weder in kleinere Teile aufgeteilt noch in irgendeiner Form erweitert werden kann.
- Eine Struktur ist eine Kombination aus beliebig vielen Elementen und/oder Strukturen. Vor allem die Tatsache, dass eine Struktur aus anderen Strukturen bestehen kann (die wiederum ebenfalls aus Strukturen bestehen können), ermöglicht es Ihnen, Datenstrukturen von beliebiger hierarchischer Komplexität zu modellieren.

Hinweis

Auch wenn es aus architektonischer Sicht theoretisch keine Grenzen für die Komplexität von Strukturen gibt, müssen Sie folgende Beschränkungen für die praktische Verwendung beachten: In tief verschachtelten Strukturen mit mehreren Ebenen kann es beim Adressieren von bestimmten Feldern über ihren Namen dazu kommen, dass dieser vollständige Name eine lange Folge von verketteten Zeichenketten ist. Wenn der vollständige Name über 255 Zeichen lang ist, kann er nicht mehr von der Werkzeug-Infrastruktur des ABAP-Backend-Systems verarbeitet werden. Bei Überschreitung dieser Zeichenlänge ist die Codegenerierung für ein solches Objekt nicht mehr möglich.

- Die Felder in Tabellen werden definiert, indem entweder genau ein Element oder genau eine Struktur der Tabelle zugeordnet wird. Dies bedeutet, dass die Feldstruktur einer Tabelle **nicht** implizit aufgebaut wird, indem ein Feld nach dem anderen der Tabelle hinzugefügt wird. Vielmehr kann die Tabelle nur die Felder eines Struktur-Datenobjekts annehmen, das vorher angelegt wurde. Wenn Sie der Tabelle hingegen ein Element-Datenobjekt zuordnen, wird das Ergebnis eine Tabelle mit einer Spalte sein (also eine einfache Liste).

Weitere Informationen

[Datenbankobjekte anlegen \[Seite 236\]](#)

[Datenbindung \[Seite 66\]](#)

[Vergleichsoperationen \[Seite 70\]](#)

1.2.9.1 Data Binding

Use

Data objects can be bound to elements in the ABAP Data Dictionary (DDIC) or to the global data types (GDT) in Enterprise Service Repository (ESR). In addition, you can also bind an elementary data object to an already existing BRFplus data object. This enables you to implicitly define a data object's settings, as opposed to explicitly assigning a data type to a data object, manually creating a value list for it and so on.

In simple use cases where you just need a field that, for example, can hold a string for an intermediate result during rule processing, it is normally the easiest way to define the data object manually. However, when you have to handle complex data structures that are already defined in a different repository, then binding a BRFplus data object to such an already existing element is an effective way to accomplish this in a consistent and fast manner.

The advantages of using data binding are not restricted to complex structures. There are also use cases where binding a data object to a simple DDIC data element can save you a lot of work. For example, if you need to access a list of ISO country codes, you would just look up a data element in the DDIC that is used for the same purpose and has the country codes already assigned as a list of domain values. After binding the BRFplus data object to that data element, you are done: The data object has the same data type and field length as the bound data element, and all the domain values are made available to the data object.

Data binding can also help you keep your system consistent in cases where you have one or more fields that are reused at many places throughout your application. The following example shows what binding can do for you in such a scenario:

Example

The name of customer company `Akron Heating Inc.` shall be used in your BRFplus application at a number of different places. If you have to change the name from `Akron Heating Inc.` to `Akron Heating Ltd.`, this change has to be done at all these places for reasons of consistency. By binding the data object, you can avoid such repetitive and error-prone tasks.

1. Create an element data object `Company_Name` of type text with a length of 30 characters.
2. On the *Domain Values* tab, choose *Create Value*.
The system displays a dialog for defining a constant expression for the name string.
3. Define a name for the constant and enter the company name `Akron Heating Inc.` into the *Value* field.
4. Click the *Create* button.
The system navigates back to the `Company_Name` data object and displays the newly defined company name in the *Value List* on the data object's *Domain Values* tab.
5. Create another element data object named `Customer_Name`.
6. From the *Binding Type* list, choose *Bind to existing BRFplus element*.
The system presents the *Object Query* dialog.
7. In the *Object Query* dialog, look up and select the previously defined `Company_Name`.
The system navigates back to the `Customer_Name` data object that is now bound to the `Company_Name` data object.
8. Click *Create and Navigate to Object*.
The system navigates to the newly created `Customer_Name` data object. On the data object's *Domain Values* tab, you can see the company name `Akron Heating Inc.` that is inherited from the definition of the bound data object.

You can now start populating your application with the [Customer_Name](#) data object wherever this entity is needed. Once you are done, change the associated text constant's value from `Akron Heating Inc.` to `Akron Heating Ltd.`. The new name is visible at all places where you have used the [Customer_Name](#) data object.

Prerequisites

The object you want to bind to a BRFplus data object must be active. This does not only apply to the bound Data Dictionary objects but also to BW master data (see section *BRFplus data objects and SAP BW information objects* below).

Features

Supported Binding Types

Depending on the type of the BRFplus data object that you want to bind to another element, you have different possibilities. This is explained in the following table:

Table 12:

Data Object Type	Supported Binding Types	Comment
Element	DDIC object GDT BRFplus element	
Table	DDIC table type	Table type structure with all contained fields is imported automatically into BRFplus
Structure	DDIC structure GDT structure CDS View	DDIC structure with all contained fields is imported automatically into BRFplus Binding to CDS Views is available for analytical functions from SAP NetWeaver 7.51 SP00 or S/4HANA Cloud 1608. For more information, see Workflow for Analytical Functions [page 21] .

Note

- Client/MANDT is filtered out in data objects that are bound to DDIC structures and tables. Raw / binary fields are also filtered out, when not UUIDs.
- As an alternative to binding a BRFplus table data object to a DDIC table type, you can also associate a BRFplus table with a database table in the backend system. To accomplish this, leave the [Binding Type](#) setting as [No Binding](#) and choose [Create Structure from DB Table](#) instead. The system then reads the

table line type of the given database table, creates a new BRFplus structure including the structure elements, binds the new BRFplus structure to the database table structure and assigns the BRFplus structure to the current BRFplus table data object.

- Binding BRFplus objects to GDTs is currently only supported for a limited scope of SAP-internal development projects. This feature is **not** supported for external applications.
- For analytical functions, when a structure is bound to a CDS view, you can define a Value Help list using a corresponding OData service. For more information, see [Value Help Conditions](#) in the *SAP Rules Expression Language Guide*.

Type Derivation

When you bind an elementary data object to a dictionary type, the system determines the dictionary type and adjusts the subtype of the BRFplus data object such that it matches best the bound dictionary type. While this sounds trivial for basic dictionary types like `CHAR` or `INT1`, there are also some more sophisticated cases. For example, the BRFplus elementary type Boolean has no direct corresponding elementary type in the data dictionary. Here, data elements used for Boolean purposes are set up by referencing a one-character domain of type `CHAR` with two, sometimes three, fixed values. For a derivation of a BRFplus data object of type Boolean, the bound dictionary element must fulfill the following conditions:

- Elementary type: `CHAR1`
- Domain: Exactly two values defined: 'x' (True) and ' ' (False).
These two conditions are fulfilled by the standard domain `BOOLE`.
- As an alternative, BRFplus does also accept the standard data element `SYBATCH` as a valid binding reference for a Boolean data object.

Another example of a type derivation that may be surprising at first sight is the handling of the widely used built-in type `NUMC`. Although this type accepts only digits from 0 to 9, it is still technically designed as a string of number literals. Therefore, a `NUMC` field in the dictionary is always converted to an elementary data object of type text (**not** number) during data binding. This would normally lead to the consequence that within BRFplus, such fields cannot be used for mathematical operations like `SUM` or `AVERAGE` in a formula or table operation expression. However, you can still control whether such a data object shall be treated as numeric or alphanumeric. You can accomplish this on application level with the [Numeric Comparison for NUMC Fields](#) setting located on the [Default Settings](#) tab.

Domain Value Lists

Most of the business-specific data elements in the dictionary have a domain object assigned. The domain is then used to define the data type of the referencing data element and a number of technical settings. In addition, a domain can provide a predefined list of values for the referencing data elements. Value lists can be defined in different ways:

- Single Values: A list of single fixed values (for example, `1, 2, 3, 10, 20, 30, ...`).
- Intervals: A list of value ranges (for example, `'a'..'f', 'h'..'m', 'x'..'z', ...`).
- Value Table: A reference to a dictionary table where each table entry serves as a value for the referencing domain.

In addition to the domain values itself, the system lets the developer maintain a descriptive text for each value. An application can take advantage of this by using unique technical values stored in the value field while presenting the business-oriented descriptive text to the user.

When you bind a BRFplus data object to a data element in the dictionary, the system imports the domain values and presents them on the [Domain Values](#) tab in the BRFplus workbench. In addition, the system tries to

import the descriptive texts associated with the values. Here, the system behavior differs with respect to the different ways of defining the texts:

- **Single Values:**
The system imports the descriptive texts as defined in the domain definition.
- **Intervals:**
For intervals, the system does **not** import any descriptive texts. This is because in this case, descriptions can only be associated with each defined interval as a whole, but not with the individual values within each interval.
- **Value Table:**
The system analyzes the associated table in order to determine a field that is likely to be meant as the description field for the values. Among others, the following criteria are tested by the system to find out if a particular field is a candidate for descriptions:
 - Field type is **Text**.
 - Field length is 15 characters or more.
 - Field name contains **'DESC'**, **'DSCR'**, **'NAME'**, **'TEXT'**, or **'TXT'** as a substring.The first field that matches all of the above criteria is considered to be a value description field, and the system imports the field contents as value descriptions for the BRFplus data object. If none of the table fields matches the criteria, no value descriptions are associated with the data object's domain values.

Note

In the BRFplus workbench, only the first 500 domain values are displayed. This is to avoid performance problems in case of very large value lists. At runtime, however, you can of course access all the domain values that have been defined in the dictionary for the referenced data element.

Inheritance

Data objects inherit their attributes such as data type, allowed comparisons, or field length from the bound type. Value help for data objects is implemented by binding a data object to the data dictionary (DDIC) or a BRFplus data object with a defined list of domain values. Binding an elementary data object to the DDIC or to ESR enables you to reuse a list of values that has been defined in the distant repository.

Of course, a data object can only inherit attributes that are available in the bound data object. For example, a data object inherits the type of a bound DDIC object whereas the setting for allowed comparisons of the data object is not inherited and can still be defined for the inheriting data object directly. This is evident because a DDIC object does not support the concept of allowed or restricted comparison operations.

If you bind a data object to another BRFplus data object, be sure that the other data object has explicitly defined attributes rather than being bound to a third data object itself. This kind of cascading inheritance over more than one level is not supported and leads to an error message.

In contrast to this, however, it is possible to bind one BRFplus data object to another data object that is bound to a DDIC object. In this case, the attributes of the DDIC object are inherited by the involved BRFplus data objects on both levels.

BRFplus data objects and SAP BW information objects

A topic related to data binding is the association of BRFplus element data objects with data elements in the Data Dictionary that are referenced by information objects (especially characteristics) modeled in SAP Business Information Warehouse (SAP BW). In nearly all cases, these objects are accompanied by comprehensive master data tables as well as text tables. At the same time, the data element associated with a BW characteristic has normally no domain values maintained (because the already existing tables serve the

same purpose). Therefore, when you establish a binding relationship between a BRFplus element data object and a data element, the system applies the following processing steps:

1. Check if the data element is referenced by a BW characteristic. If yes, determine the related master data table and text table (otherwise, establish a standard data binding as described in this chapter above).
2. Use the related tables as a source for domain values and texts for the BRFplus data object.

With that, from your BRFplus application, you can take advantage of the data modeling that has already been done in SAP BW.

Constraints

Binding data objects to GDT structures is only supported for systems that have been defined as AP (Application Platform) systems using the Switch Framework. This kind of system is used for running SAP Business ByDesign. For more information, see the *Switch Framework* documentation in the SAP NetWeaver Library on SAP Help Portal.

More Information

[Creating Data Objects \[page 236\]](#)

1.2.10 Vergleichsoperationen

Definition

Mithilfe von Vergleichsoperationen legen Sie fest, wie das System den Vergleich zwischen einem eingehenden Wert, der als Kontextparameter an einen Ausdruck weitergegeben wurde, und einem vordefinierten Wert oder Wertebereich ausführen soll. Dieser eingehende Wert, der Testparameter, kann statt über einen Kontextparameter auch als Ergebnis eines verschachtelten Ausdrucks weitergegeben werden.

Verwendung

Vergleichsoperationen werden hauptsächlich verwendet, um innerhalb einer Geschäftsregel zwischen verschiedenen Verarbeitungszweigen zu unterscheiden. Beispielsweise dienen in einer Entscheidungstabelle die vordefinierten Werte in den Bedingungsspalten als Vergleichsparameter, mit dem ein eingehender Wert verglichen wird.

Struktur

Alle von BRFPplus unterstützten Vergleichsoperationen werden verwendet, um ein eingehendes Element-Datenobjekt (den Testparameter) mit einem vordefinierten Wert oder Wertebereich (dem Vergleichsparameter) zu vergleichen. Die formale Beschreibung einer Vergleichsoperation lautet daher:

```
<Testparameter> <Vergleichsoperator> <Vergleichsparameter>
```

Beispiel

Sie möchten bestimmen, ob ein eingehender Wert innerhalb eines vordefinierten Wertebereichs liegt.

Testparameter: `tp = 24`

Vergleichsoperator: *liegt zwischen*

Vergleichsparameter: `vp = [18..25]`

Vergleich: `tp liegt zwischen vp` (also: "24 liegt zwischen [18..25]")

Ergebnis: `wahr`

Die oben aufgeführte formale Beschreibung ist auch für Operatoren gültig, die aus Sicht des Benutzers keinen Vergleichsparameter besitzen (*ist initial* und *ist nicht initial*). Die zu dieser Gruppe gehörenden Operatoren (Gruppe 3 in der folgenden Tabelle) vergleichen den Testparameter mit einem impliziten Vergleichsparameter, den das System intern ableitet. Daher ist es nicht notwendig (und nicht möglich), selbst einen Vergleichsparameter für diese Operationsarten anzugeben.

Operatorengruppen

Ob ein bestimmter Operator verwendet werden kann oder nicht, hängt davon ab, welche Parametertypen beteiligt sind. Um die Komplexität der möglichen Kombinationen von Operatoren und Parametertypen zu verringern, können die von BRFPplus bereitgestellten Vergleichsoperatoren nach erlaubten Verwendungsmöglichkeiten gruppiert zusammengefasst werden. Für jede der unten aufgeführten Gruppen kann bestimmt werden, welche Parametertypen mit den jeweiligen Operatoren aus dieser Gruppe kombiniert werden können.

Hinweis

In diesem Abschnitt werden alle von BRFPplus unterstützten Vergleichsoperatoren beschrieben. Ob alle diese Vergleichsoperatoren verwendet werden können, hängt allerdings von der Definition des Element-Datenobjekts ab. Dieses Verhalten wird unter der Registerkarte *Eigenschaften* des Datenobjekts von der Einstellung *Erlaubte Vergleiche* gesteuert. Weitere Informationen finden Sie unter [Ein Element-Datenobjekt anlegen \[Seite 236\]](#).

Tabelle 13:

Gruppe	Beschreibung	Operatoren	Konstante	ID
1	Ordinalvergleiche	ist gleich ist nicht gleich ist kleiner als ist kleiner oder gleich ist größer als ist größer oder gleich liegt zwischen liegt nicht zwischen	GC_OPTION_EQUAL GC_OPTION_NOT_EQUAL AL GC_OPTION_LESS GC_OPTION_LESS_EQUAL GC_OPTION_GREATER GC_OPTION_GREATER_EQUAL GC_OPTION_BETWEEN GC_OPTION_NOT_BETWEEN	EQ NE LT LE GT GE BT NB
2	String-Vergleiche	enthält irgendein/e/en enthält nicht irgendein(e) enthält nur enthält nicht nur enthält Zeichenfolge enthält nicht Zeichenfolge stimmt überein mit Muster stimmt nicht überein mit Muster beginnt mit Text beginnt nicht mit endet mit Text endet nicht mit	GC_OPTION_CONTAINS_ANY GC_OPTION_CONTAINS_NOT_ANY GC_OPTION_CONTAINS_ONLY GC_OPTION_CONTAINS_NOT_ONLY GC_OPTION_CONTAINS_STRING GC_OPTION_CONTAINS_NO_STRING GC_OPTION_CONTAINS_PATTERN GC_OPTION_NOT_CONTAINS_PATTERN GC_OPTION_STARTS_WITH GC_OPTION_NOT_STARTS_WITH GC_OPTION_ENDS_WITH GC_OPTION_NOT_ENDS_WITH	CA NA CO CN CS NS CP NP SW US EW DE

Gruppe	Beschreibung	Operatoren	Konstante	ID
3	Implizite Vergleiche	ist initial	GC_OPTION_INITIAL	I1
		ist nicht initial	GC_OPTION_NOT_INITIAL	I2
		ist gültig	GC_OPTION_VALID	IV
		ist ungültig	GC_OPTION_IS_NOT_VALID	NV
4	Besondere Vergleiche	Währung ist gleich	GC_OPTION_CURRENCY_EQUALS	CE
		Währung ist unterschiedlich	GC_OPTION_CURRENCY_DIFFERS	CD
		Einheit ist gleich	GC_OPTION_UNIT_EQUALS	UE
		Einheit ist unterschiedlich	GC_OPTION_UNIT_DIFFERS	UD

Hinweis

Die impliziten Vergleichsoperatoren "ist gültig" und "ist ungültig" (Gruppe 3) sind nur für die folgenden Element-Datenobjekte definiert:

- Betrag
- Menge
- Zeitpunkt

Für den Betrag und die Menge prüft das System, ob die angegebene Währung oder Mengeneinheit in den zentralen Customizing-Tabellen des Backend-Systems verfügbar ist. Für Datenobjekte des Typs Zeitpunkt prüft das System, ob die angegebenen Datums- und Zeitwerte in ein gültiges Datum überführt werden können (beispielsweise nicht "31. April").

Die Operatoren "wird übergeben" und "wird nicht übergeben" (technische IDs S1, S2) werden nicht mehr unterstützt. Bereits vorhandene Objekte, die diese Operatoren verwenden, müssen daher angepasst werden.

Alle Operatoren werden intern durch Konstanten repräsentiert, die für die betroffenen Interfaces definiert sind (zum Beispiel IF_FDT_RANGE).

Die IDs der Operatoren aus der letzten Spalte der oberen Tabelle stellen die technischen Namen dar, die intern verwendet werden. Sie beziehen sich größtenteils auf die entsprechenden Vergleichsoperatoren, die in ABAP definiert sind. BRFplus gibt die Ausführung von Vergleichsoperationen technisch an die entsprechenden ABAP-Ausdrücke weiter. Aus diesem Grund ist das detaillierte Verarbeitungsverhalten von BRFplus-Parametern identisch mit den entsprechenden ABAP-Ausdrücken.

Datentypkombinationen für Vergleiche

Wie das System einen Vergleich durchführt, hängt vom Operator und dem Testparametertyp ab. In der folgenden Tabelle ist eine Übersicht über die verschiedenen Vergleichsansätze – zusammengefasst nach beteiligten Datentypen – aufgelistet:

Tabelle 14:

Datentyp	Anmerkungen
Text	Alphanumerischer Vergleich von Werten. Um die Komplexität zu verringern, werden Textparameter intern entsprechend der ABAP-Anweisung <code>CONVERT TEXT</code> in eine sortierbare Byte-Folge konvertiert. Mithilfe dieser Normalisierung wird Problemen vorgebeugt, die beim Vergleich von Zeichen verschiedener Codepages auftreten. Allerdings kann dieses Vorgehen zu unerwarteten Ergebnissen führen.
Zahl	Numerischer Vergleich von Werten.
Betrag, Menge	Wenn die Währung oder Einheit der Parameter unterschiedlich ist, werden die Parameter zunächst intern in eine der beteiligten Währungen oder Einheiten konvertiert. Anschließend wird ein numerischer Vergleich der Werte durchgeführt.
Zeitpunkt	Datums- und Zeitwerte werden numerisch verglichen. Wenn die zu vergleichenden Werte unterschiedliche Untertypen aufweisen, werden nur die übereinstimmenden Teile beider Werte für den Vergleich herangezogen. Wenn Sie beispielsweise einen Zeitstempel mit Datum und Uhrzeit mit einem reinen Datum (ohne Uhrzeit) vergleichen möchten, wird die Uhrzeit des Zeitstempels ignoriert und es werden lediglich die Datumsangaben miteinander verglichen.
Boolesch	Überprüfung auf Gleichheit

Im Allgemeinen kann ein Vergleich vom System nur dann durchgeführt werden, wenn beide beteiligten Parameter denselben Typ aufweisen. Als Ausnahme hierzu unterstützt das System jedoch zusätzliche Vergleichsoperationen unter folgenden Bedingungen:

- Ordinalvergleiche (Gruppe 1) können durchgeführt werden, wenn beide beteiligten Parameter in eine zulässige Zahl konvertierbar sind. In diesem Fall führt das System einen numerischen Vergleich durch. Andernfalls werden beide Parameter als Texte behandelt und es wird ein String-Vergleich durchgeführt.
- String-Vergleiche (Gruppe 2) können mit Parametern verschiedener Typen durchgeführt werden. Hierzu wird vom System eine implizite Typkonvertierung vorgenommen, sodass beide Parameter als Texte behandelt werden können.

In den folgenden Tabellen sind die verschiedenen Kombinationsmöglichkeiten aufgelistet. Es wird hierbei entsprechend der Frage zusammengefasst, ob ein Vergleich überhaupt möglich ist (mit oder ohne eine vorherige Typkonvertierung) oder nicht. Die Tabellen beschränken sich dabei auf die Operatorgruppen 1 und 2, da bei Gruppe 3 ein Typenkonflikt per Definition unmöglich ist.

In der folgenden Tabelle sind alle Vergleichsoperationen dargestellt, die das System ohne implizite Konvertierungen durchführen kann:

Tabelle 15:

Testparametertyp	Vergleichsparametertyp	Vergleichsoperator	Vergleichstyp
Zahl	Zahl	Gruppe 1	Numerisch
Text	Text	Gruppe 1	Text
Text	Text	Gruppe 2	Text
Betrag	Betrag	Gruppe 1	Betrag
Menge	Menge	Gruppe 1	Menge
Zeitpunkt	Zeitpunkt	Gruppe 1	Zeitpunkt
Boolesch	Boolesch	ist gleich, ist nicht gleich	Boolesch

In der folgenden Tabelle sind die Vergleichsoperationen aufgelistet, die das System nach der impliziten Typkonvertierung von mindestens einem der beteiligten Parameter durchführen kann:

Tabelle 16:

Testparametertyp	Vergleichsparameter- typ	Vergleichsoperator	Vergleichstyp	Kommentare
Zahl	Zahl	Gruppe 2	Text	Zahlen werden intern in Text konvertiert
Zahl	Text	Gruppe 1	Numerisch oder Text	1. Der Vergleichsparameter des Typs Text wird in eine Zahl konvertiert und ein numerischer Vergleich wird durchgeführt. 2. Ist dies nicht möglich, wird der Testparameter in einen Text konvertiert und es wird ein String-Vergleich durchgeführt.
Zahl	Text	Gruppe 2	Text	Testparameter wird in einen Text konvertiert

Testparametertyp	Vergleichsparameter- typ	Vergleichsoperator	Vergleichstyp	Kommentare
Text	Zahl	Gruppe 1	Numerisch oder Text	1. Der Testparameter des Typs Text wird in eine Zahl konvertiert und ein numerischer Vergleich wird durchgeführt. 2. Ist dies nicht möglich, wird der Vergleichsparameter in einen Text konvertiert und es wird ein String-Vergleich durchgeführt.
Text	Zahl	Gruppe 2	Text	Vergleichsparameter wird in einen Text konvertiert
Betrag	Zahl	Gruppe 2	Text	Beide Parameter werden in Texte konvertiert
Betrag	Text	Gruppe 2, Gruppe 4	Text	Testparameter wird in einen Text konvertiert
Menge	Zahl	Gruppe 2	Text	Beide Parameter werden in Texte konvertiert
Menge	Text	Gruppe 2, Gruppe 4	Text	Testparameter wird in einen Text konvertiert
Zeitpunkt	Zahl	Gruppe 2	Text	Beide Parameter werden in Texte konvertiert
Zeitpunkt	Text	Gruppe 2	Text	Testparameter wird in einen Text konvertiert

In der folgenden Tabelle sind alle Vergleichsoperationen dargestellt, die das System **nicht** durchführen kann:

Tabelle 17:

Testparametertyp	Vergleichsparametertyp	Vergleichsoperator
Zahl	Betrag	Gruppe 1, Gruppe 2

Testparametertyp	Vergleichsparametertyp	Vergleichsoperator
Zahl	Menge	Gruppe 1, Gruppe 2
Zahl	Zeitpunkt	Gruppe 1, Gruppe 2
Zahl	Boolesch	Gruppe 1, Gruppe 2
Text	Betrag	Gruppe 1, Gruppe 2
Text	Menge	Gruppe 1, Gruppe 2
Text	Zeitpunkt	Gruppe 1, Gruppe 2
Text	Boolesch	Gruppe 1, Gruppe 2
Betrag	Betrag	Gruppe 2
Betrag	Zahl	Gruppe 1
Betrag	Text	Gruppe 1
Betrag	Menge	Gruppe 1, Gruppe 2
Betrag	Zeitpunkt	Gruppe 1, Gruppe 2
Betrag	Boolesch	Gruppe 1, Gruppe 2
Menge	Menge	Gruppe 2
Menge	Zahl	Gruppe 1
Menge	Text	Gruppe 1
Menge	Betrag	Gruppe 1, Gruppe 2
Menge	Zeitpunkt	Gruppe 1, Gruppe 2
Menge	Boolesch	Gruppe 1, Gruppe 2
Zeitpunkt	Zeitpunkt	Gruppe 2
Zeitpunkt	Zahl	Gruppe 1
Zeitpunkt	Text	Gruppe 1
Zeitpunkt	Betrag	Gruppe 1, Gruppe 2
Zeitpunkt	Menge	Gruppe 1, Gruppe 2
Zeitpunkt	Boolesch	Gruppe 1, Gruppe 2
Boolesch	Zahl	Gruppe 1, Gruppe 2

Testparametertyp	Vergleichsparametertyp	Vergleichsoperator
Boolesch	Text	Gruppe 1, Gruppe 2
Boolesch	Betrag	Gruppe 1, Gruppe 2
Boolesch	Menge	Gruppe 1, Gruppe 2
Boolesch	Zeitpunkt	Gruppe 1, Gruppe 2
Boolesch	Boolesch	Gruppe 1 (<i>ist gleich</i> und <i>ist nicht gleich</i> sind jedoch erlaubt), Gruppe 2

Weitere Informationen zu ausgewählten Vergleichsoperatoren


In der folgenden Tabelle sind einige zusätzliche Informationen zur genauen Funktionsweise von Vergleichsoperatoren aufgelistet. Es werden jedoch lediglich die Operatoren erwähnt, deren Verhalten nicht selbsterklärend ist.

Zusätzlich zu den detaillierten Informationen in der Tabelle gelten folgende Regeln:

- Bei String-Vergleichen wird die Groß- und Kleinschreibung von der jeweiligen Einstellung des Wertebereich-Ausdrucks gesteuert, der als Vergleichsparameter verwendet wird.
- Im Gegensatz dazu werden Mustervergleiche unabhängig von Groß- und Kleinschreibung durchgeführt.
- Bei String-Vergleichen werden vorangehende und angehängte Leerstellen vor der Durchführung des Vergleichs vom Testparameter entfernt. Leerstellen innerhalb der Zeichenkette werden jedoch beibehalten.
- Bei Ordinalvergleichen von Zeichenketten (beispielsweise *ist kleiner als*) wird die Groß- und Kleinschreibung **nicht** berücksichtigt. So wird der Vergleich 'a' ist kleiner als 'B' als wahr bewertet, obwohl 'a' im ASCII-Code einen höheren Wert als 'B' hat.
- Im Gegensatz dazu werden Gleichheitsprüfungen (*ist gleich*) von Textparametern wie String-Vergleiche durchgeführt. Dies bedeutet, dass das System während der Auswertung die Einstellung der Groß- und Kleinschreibung berücksichtigt.

Tabelle 18:

Vergleichsoperator	Kommentar
liegt zwischen	Gibt <i>wahr</i> zurück, wenn der Testparameterwert zwischen dem Mindest- und Maximalwert eines Wertebereichs liegt, einschließlich des Mindest- und Maximalwerts (also <code>value >= low AND value <= high</code>).
enthält irgendein/e/en	Gibt <i>wahr</i> zurück, wenn der Testparameter mindestens ein Zeichen enthält, das auch im Vergleichsparameter enthalten ist. Die Folge sowie die Häufigkeit des Vorkommens der entsprechenden Zeichen kann jedoch unterschiedlich sein.

Vergleichsoperator	Kommentar
enthält nur	Gibt <code>wahr</code> zurück, wenn der Testparameter ausschließlich Zeichen enthält, die auch im Vergleichsparameter enthalten sind. Die Folge sowie die Häufigkeit des Vorkommens der entsprechenden Zeichen kann jedoch unterschiedlich sein.
enthält Zeichenfolge	Gibt <code>wahr</code> zurück, wenn die Zeichenfolge aus dem Vergleichsparameter auch im Testparameter enthalten ist. Dabei muss die Reihenfolge der Zeichen in beiden Parametern identisch sein.
ist gleich	Gibt <code>wahr</code> zurück, wenn der Wert des Testparameters identisch mit dem Wert des Vergleichsparameters ist.
stimmt überein mit Muster beginnt mit Text endet mit Text	<p>Gibt <code>wahr</code> zurück, wenn der eingegebene Testparameter mit dem Muster aus dem Vergleichsparameter übereinstimmt. Folgende Platzhalterzeichen werden unterstützt:</p> <ul style="list-style-type: none"> • <code>+</code> – ersetzt genau ein Zeichen • <code>*</code> – ersetzt beliebig viele Zeichen <div>  Beispiel <pre> Platzhalter = '*': Testparameter: tp = 'Hallo' Vergleichsparameter: vp = 'Ha*o' Vergleich: tp stimmt überein mit Muster vp Ergebnis: wahr Platzhalter = '+': Testparameter: tp = 'Held' Vergleichsparameter: vp = 'He+d' Vergleich: tp stimmt überein mit Muster vp Ergebnis: wahr </pre> </div>

Weitere Informationen

[Wertebereich-Ausdruck \[Seite 127\]](#)

1.2.11 Ausdrücke

Verwendung

Ausdrucksarten definieren die Rechenleistung von BRFplus. Jede Ausdrucksart definiert eine in sich geschlossene Recheneinheit mit klar definierter Logik. Ausdrücke verwenden einen Kontext oder andere, geschachtelte Ausdrücke, um ein Ergebnis zu berechnen, zu bestimmen oder abzuleiten. Ein Ausdruck kann demnach als Instanz einer Ausdrucksart angesehen werden, die den Regeln der Ausdrucksartlogik folgt. Ein Ausdruck kann als geschachtelter Baustein für eine Regel wiederverwendet werden, bei dem eine Funktion als oberster Ausdruck zugeordnet wird.

BRFplus enthält standardmäßig eine Menge gemeinsamer Ausdrucksarten und wird regelmäßig um neue Ausdrucksarten erweitert. Sie können jedoch auch Ihre eigenen Ausdrucksarten anlegen und sie in Regeln verwenden.

Folgende vordefinierte Ausdrucksarten sind verfügbar:

- [Boolescher Wert \(Ausdruck\) \[Seite 81\]](#)
- [GRMS-Konnektor \(Ausdruck\) \[Seite 82\]](#)
- [Fall \(Ausdruck\) \[Seite 82\]](#)
- [Konstante \(Ausdruck\) \[Seite 84\]](#)
- [Datenbankabfrage \(Ausdruck\) \[Seite 85\]](#)
- [Entscheidungstabelle \(Ausdruck\) \[Seite 89\]](#)
- [Entscheidungsbaum \(Ausdruck\) \[Seite 103\]](#)
- [Dynamischer Ausdruck \[Seite 105\]](#)
- [Formel \(Ausdruck\) \[Seite 107\]](#)
- [Funktionsaufruf \(Ausdruck\) \[Seite 111\]](#)
- [Schleife \(Ausdruck\) \[Seite 111\]](#)
- [Prozeduraufruf \(Ausdruck\) \[Seite 118\]](#)
- [Zufallszahl \(Ausdruck\) \[Seite 113\]](#)
- [Suchbaum \(Ausdruck\) \[Seite 115\]](#)
- [Tabellenoperation \(Ausdruck\) \[Seite 124\]](#)
- [Wertebereich \(Ausdruck\) \[Seite 127\]](#)

Hinweis

Folgende Ausdrucksarten, die Teil früherer BRFplus-Releases waren, sind jetzt veraltet und werden nicht länger von SAP unterstützt:

- Schrittfolge
- XSL-Transformation

Weitere Informationen finden Sie in den SAP-Hinweisen [1845412](#)  oder [1904368](#) .

1.2.11.1 Boolescher Wert (Ausdruck)

Verwendung

Der Ausdruck Boolescher Wert wird verwendet, um eine logische Bedingung gemäß den Regeln der booleschen Logik zu testen. BRFplus bietet Ihnen eine Anzahl an vordefinierten Ausdrucksvorlagen mit zwei oder drei Operanden, die viele Anwendungsfälle abdecken. Beim Definieren eines booleschen Ausdrucks ersetzen Sie die formalen Operanden (<1>, <2> usw.) durch BRFplus-Objekte, die zu einem booleschen Wert ausgewertet werden können (boolesche Kontextdatenobjekte oder Ausdrücke mit einem booleschen Ergebnisdatenobjekt).

Vorlagen

Folgende Vorlagen sind für häufig verwendete logische Bedingungen vordefiniert:

- Mindestens ein Operand ist wahr
- Alle Operanden sind wahr
- <1> und <2>
- <1> oder <2> oder <3>
- (<1> und <2>) oder <3>
- (<1> oder <2>) und <3>
- <1> und <2> und <3>

Die formale Struktur der Vorlagen ist fix. Sie können lediglich die Operatoren invertieren (d.h. ein logisches **nicht** hinzufügen bzw. entfernen).

Benutzerdefiniertes Logikschema

Wenn Sie komplexere Szenarien lösen müssen, die nicht durch die vorhandenen Vorlagen abgedeckt werden, können Sie über den Vorlagen-Menüpunkt [Benutzerdefiniert](#) Ihre eigene logische Bedingung definieren. Hier können Sie so viele Operanden festlegen, wie Sie benötigen. Unterstützte Operatoren sind **und**, **oder** und **nicht**. Den logischen Vorrang steuern Sie, indem Sie Teile des Ausdrucks in Klammern setzen.

Nachdem Sie die Struktur Ihres benutzerdefinierten booleschen Ausdrucks definiert und geprüft haben, übernehmen Sie die formale Struktur für den Ausdruck und weisen den Operanden Objekte oder Ausdrücke zu. Sie können die benutzerdefinierte Struktur Ihres Ausdrucks auch jederzeit ändern, indem Sie weitere Operanden hinzufügen oder die gewünschten Operanden entfernen. Änderungen an der Struktur von booleschen Ausdrücken werden nur für benutzerdefinierte Ausdrücke unterstützt.

Hinweis

Es ist allerdings **nicht** möglich ein Schema, das Sie für einen benutzerdefinierten booleschen Ausdruck festgelegt haben, wiederzuverwenden und weitere boolesche Ausdrücke mit demselben Schema zu instanziierten. Wenn Sie mehrere boolesche Ausdrücke mit derselben Struktur instanziierten möchten, können Sie den gesamten Booleschen Wert kopieren und ihn mit einem neuen Namen versehen.

Weitere Informationen

[Einen booleschen Ausdruck anlegen \[Seite 197\]](#)

1.2.11.2 GRMS-Anbindung (Ausdruck)

Verwendung

Die GRMS-Anbindung (GRMS: Geschäftsregel-Managementsystem) ermöglicht es Ihnen, SAP BRM oder eine beliebige andere externe Regel-Engine mit BRFplus zu verwenden.

Je nach Verbindungstyp (NetWeaver-GRMS oder extern) verwendet der Ausdruck den entsprechenden Funktionsbaustein, um die externe Regel-Engine aufzurufen. Über den Ausdruck können Sie basierend auf dem verfügbaren Vokabular die Signatur für den Aufruf angeben. Sobald die Signatur definiert ist, können Sie die mit dem Aufruf verknüpften Metadaten als XML-Schema exportieren.

Das XML-Schema beinhaltet Informationen zu den Eingabedatenobjekten, Texten und der Dokumentation, die mit den zu füllenden Datenobjekten und Rückgabeparametern verknüpft sind.

Sie können bei der Definition des Ausdrucks auch alle Parameter angeben, die spezifisch für die verwendete Regel-Engine sind. Für einen Aufruf zu SAP NetWeaver BRM sollten die beiden Parameter Projekt und Regelsatz Werte enthalten. BRFplus legt während der Ausführung des Aufrufs ein XML-Dokument an, das dem oben genannten XML-Schema entspricht und gibt dieses an die andere Regel-Engine weiter.

Weitere Informationen

[Einen GRMS-Anbindungsausdruck anlegen \[Seite 199\]](#)

1.2.11.3 Fall (Ausdruck)

Verwendung

Der Ausdruck Fall wird für die Modellierung von Anwendungsfällen verwendet, bei denen Sie einen eingehenden Wert mit einer vordefinierten Liste diskreter Werte vergleichen möchten, die Sie bereits kennen. Eine solche in sich geschlossene Liste bekannter Werte könnte beispielsweise Länderschlüssel, Regionalkennzeichen, Beförderungsmittel oder Steuerkennzeichen enthalten. Wenn ein Wert vorkommt, der nicht in Ihrer Liste enthalten ist, können Sie diesen über einen optionalen *Sonst*-Zweig abdecken. Dieser wird für alle Werte berücksichtigt, die nicht explizit vom Ausdruck behandelt werden.

Struktur

Der BRFplus-Fall-Ausdruck ähnelt der ABAP-CASE-Anweisung. Er prüft einen Eingabeparameter aus dem Kontext (Case-Parameter) gegen eine bestimmte Anzahl von Testparametern.

Für jeden dieser Testparameter wird ein entsprechender Rückgabeparameter in der *Wenn*-Tabelle angegeben. Sobald einer der Testparameter mit dem Case-Parameter übereinstimmt, wird der entsprechende

Rückgabeparameter ausgewertet und es wird ein Ergebnis zurückgeliefert. Wenn keiner der Testparameter mit dem Case-Parameter übereinstimmt, wird ein generischer Parameter ausgewertet, um das Ergebnis anzugeben.

Alle Parameter können Kontextdatenobjekte oder eingebettete Ausdrücke sein. Üblicherweise ist der Case-Parameter ein Kontextdatenobjekt, die Testparameter Konstanten und die Ergebnisparameter eingebettete Ausdrücke.

Funktionsumfang

Ergebnistyp

Sie können entscheiden, ob ein Fall-Ausdruck einen Wert zurückliefern oder eine Aktion durchführen soll:

- **Rückgabewert:** In diesem Modus müssen Sie sicherstellen, dass allen Vergleichswerten ein Rückgabewert oder ein Ausdruck zugeordnet ist, dessen Ergebnis den gleichen Typ wie das Ergebnisdatenobjekt des Fall-Ausdrucks hat.

Hinweis

Wenn Sie einen Fall-Ausdruck in der BRFplus-Workbench definieren, ordnen Sie dem Ausdruck zunächst ein Ergebnisdatenobjekt zu und definieren Sie anschließend die Ergebnisse für die Vergleichswerte. Nur unter dieser Voraussetzung haben Sie Zugriff auf passende Ausdrücke, Standardobjekte (wie den aktuellen Benutzer oder die Anmeldesprache) oder können über die direkte Werteingabe Ergebnisse eingeben. Ohne definiertes Ergebnisdatenobjekt können Sie einem Vergleichswert nur Element-Datenobjekte als Ergebnis zuordnen.

- **Aktion durchführen:** In diesem Modus stößt der Fall-Ausdruck die Aktion an, die dem Rückgabewert zugeordnet ist, der mit der Eingabe übereinstimmt. Als Ergebnisdatenobjekt wird automatisch die vordefinierte Tabelle `ACTIONS` ausgewählt, die eine Liste von IDs der auszuführenden Aktionen enthält (Ergebnisaktionen sowie alle Nachfolgeaktionen, die möglicherweise mit dieser Aktion verknüpft sind).

Nachdem Sie die Einstellungen zum Ergebnistyp vorgenommen und den Ausdruck gesichert haben, zeigt das System diese Einstellungen standardmäßig im Abschnitt [Details](#) auf der Benutzungsoberfläche an.

Hinweis

Das Ändern der Einstellungen zum Ergebnistyp führt in den meisten Fällen dazu, dass Sie alle Objekte erneut den Vergleichswerten des Ausdrucks zuordnen müssen. Dies sollten Sie bedenken, bevor Sie sich dazu entschließen, den Ergebnistyp eines bereits definierten Ausdrucks zu ändern.

Groß-/Kleinschreibung

Wenn Sie ein Kontextdatenobjekt des Typs Text als Case-Parameter des Ausdrucks zuordnen, können Sie definieren, ob die Groß-/Kleinschreibung beim Wertevergleich berücksichtigt wird.

Weitere Informationen

[Einen Fall-Ausdruck anlegen \[Seite 202\]](#)

1.2.11.4 Konstante (Ausdruck)

Verwendung

Der Ausdruck Konstante ist der einfachste Ausdruck in BRFplus. Er ermöglicht es Ihnen, einen konstanten Wert anzulegen, der verarbeitet und anschließend zurückgeliefert wird. Eine Konstante kann zu verschiedenen Zwecken verwendet werden:

- Zur Definition von Ergebniswerten für andere Ausdrücke
- Zur Definition von Testparametern für Vergleichsoperationen
- Zur Definition einer Werteliste für Element-Datenobjekte

Es sind sechs Konstantentypen verfügbar:

Tabelle 19:

Konstantentyp	Text (Beschreibung)
Text	Zeichen Eine Text-Konstante kann eine Zeichenkette von maximal 255 Zeichen enthalten.
Zahl	Zahl mit Dezimalstellen Eine Zahl-Konstante kann eine Länge von maximal 46 Ziffern ohne Exponent oder Vorzeichen aufweisen.
Boolesch	Boolescher Wert
Betrag	Zahl und Währung
Menge	Zahl und Mengeneinheit
Zeitpunkt	abstraktes Datum, abstrakte Uhrzeit, abstraktes Datum und Uhrzeit, globales Datum und Uhrzeit oder lokaler Datums- und Zeitunterschied

Hinweis

Der Typ eines Konstanten-Ausdrucks kann nur während der Anlegung definiert werden. Nachdem der Ausdruck gesichert wurde, kann der Konstantentyp nicht mehr geändert werden.

Weitere Informationen

[Einen Konstanten-Ausdruck anlegen \[Seite 204\]](#)

1.2.11.5 Datenbankabfrage (Ausdruck)

Verwendung

Mit dem Ausdruck Datenbankabfrage können Sie aus der BRFplus-Workbench heraus Datenbankabfragen ausführen. Er wird verwendet, um auf Datensätze aus Tabellen im ABAP Dictionary zuzugreifen oder Informationen über Tabellen abzurufen.

Funktionsumfang

Operationsmodi

- **Datenbeschaffungsmodus**

In diesem Modus werden die Daten aus einer Datenbanktabelle abgerufen. Der Ausdruck führt eine einfache Auswahloperation im angegebenen Feld aus. Die Auswahl wird über folgende Einstellungen definiert:

- **Auswählen-Kennzeichen**
Gibt an, ob mehrere Einträge (*Alle Einträge*) oder ein einzelner Eintrag (*Einzelerfassung*) ausgewählt werden soll.

Hinweis

Wenn Sie *Alle Einträge* für die Auswahl verwenden, muss das der Datenbankabfrage zugeordnete Ergebnisdatenobjekt vom Typ Tabelle sein.

- **Tabellenname**
Gibt die Datenbanktabelle an, von der die Daten gelesen werden. Sie können auf jede im ABAP Dictionary des Backend-Systems verfügbare Tabelle zugreifen. Es ist **nicht** notwendig, die Datenbanktabelle an ein BRFplus-Datenobjekt des Typs Tabelle zu binden.
- **Feldbedingungen**
Gibt Bedingungen für die Auswahloperation an, mit deren Hilfe Sie das Suchergebnis verfeinern können. Ausgehend von den Feldbedingungen wird die `WHERE`-Anweisung der ausgeführten Abfrage erstellt.

Welcher Wert tatsächlich aus der Tabelle gewählt wird, hängt vom Objekttyp des Ergebnisdatenobjekts ab:

- Wenn das Ergebnisdatenobjekt ein **Element** ist, dessen Name einem der Tabellenfelder entspricht, wird mit dem Ausdruck nur dieser eine Wert ausgewählt.
- Wenn das Ergebnisdatenobjekt eine **Struktur** ist, werden die Datenbankfelder gelesen, die den Strukturfeldern entsprechen.
- Wenn das Ergebnisdatenobjekt eine **Tabelle** ist, werden die Quellfelder gelesen, die den Feldern aus der Ergebnistabelle entsprechen. Anders als bei einem Datenobjekt des Typs Element oder Struktur können hier mehrere Einträge gelesen werden.

Hinweis

Das System versucht ausgehend vom Feldtyp und -namen, die Felder des Kontextdatenobjekts den Feldern des Ergebnisdatenobjekts zuzuordnen. Es kann jedoch nicht gewährleistet werden, dass diese

Zuordnung vollständig oder für den gegebenen Fall ausreichend ist. Aus diesem Grund sollten Sie die Feldzuordnungen stets prüfen und gegebenenfalls anpassen. Weitere Informationen finden Sie im unteren Abschnitt *Feldzuordnung*.

- **Aggregationsmodus**

Bei diesem Modus wird das Ergebnis einer Aggregationsfunktion zurückgeliefert, die auf eine Datenbanktabelle angewandt wurde. Folgende Aggregationsfunktionen werden unterstützt:

- Maximum
- Minimum
- Summe
- Durchschnitt
- Anzahl

Aggregationsfunktionen können auf einzelne Felder oder ganze Tabellendatensätze (Alle Felder einer Zeile) angewandt werden. Im folgenden Beispiel werden die Auswirkungen von Änderungen an dieser Einstellung verdeutlicht:

Beispiel

Sie legen einen Ausdrucks des Typs Datenbankabfrage an, der aus Tabelle T100 liest. Diese Tabelle wird für das Speichern von Nachrichtenkurztexten in einem SAP-System verwendet. Der Ausdruck soll die Anzahl der Datensätze (Aggregationsfunktion *Anzahl*) mit deutschen Kurztexten (**SPRSL = DE**) zurückliefern. In einem gegebenen System kann dieser Ausdruck je nach Art der Unterscheidung folgende Ergebnisse liefern:

Tabelle 20:

Einstellung	Ergebnis	Kommentar
Auswählen: Anzahl von Datensätze von T100 Mit Bedingung: SPRSL ist gleich DE	110650	Gesamtzahl der Tabellenzeilen, deren Sprachenfeld (SPRSL) den Wert DE aufweist.
Auswählen: Anzahl von ARBGB von T100 Mit Bedingung: SPRSL ist gleich DE	1615	Gesamtzahl der Nachrichtenklassen (ARBGB), bei denen mindestens eine Nachricht die Sprache DE aufweist.

Einstellung	Ergebnis	Kommentar
Auswählen: Anzahl von SPRSL von T100 Mit Bedingung: SPRSL ist gleich DE	1	Gesamtzahl der eindeutigen Tabellenzeilen, deren Sprachenfeld (SPRSL) den Wert DE aufweist. Obwohl über 100000 Zeilen in der Tabelle diese Bedingung erfüllen, führt die Einschränkung der Suche auf unterschiedliche Werte für das Feld SPRSL dazu, dass die Ausführung bereits nach dem ersten Treffer beendet wird. Jeder weitere Treffer würde zu Dubletten führen.

Der Typ des Element-Datenobjekts bestimmt, welche Aggregationsfunktionen unterstützt werden:

Tabelle 21:

Typ	Unterstützte Aggregationsfunktionen
Zahl, Betrag, Menge	Alle Aggregationsfunktionen
Boolesch, Text	Maximum, Minimum, Anzahl
Zeitpunkt	Anzahl

- **Existenzprüfungsmodus**

In diesem Modus wird dem Ergebnisdatenobjekt wahr oder falsch zugeordnet, je nachdem, ob mindestens ein Eintrag in der ABAP-Dictionary-Tabelle vorhanden ist, der die Bedingungen bei `Mit Bedingung` erfüllt.

Feldzuordnung

Wenn Sie eine Datenbankabfrage im Datenbeschaffungsmodus verwenden, möchten Sie eine bestimmte Teilmenge der Datensätze aus der Tabelle auswählen, die Sie als Kontextdatenobjekt verwenden.

Anschließend möchten Sie die die gewählten Datensätze in ein BRFplus-Tabellendatenobjekt speichern, das das Ergebnisdatenobjekt des Ausdrucks darstellt.

Im Allgemeinen sind diese beiden Tabellendatenobjekte nicht identisch (sonst würden Sie den ursprünglichen Inhalt der Tabelle mit dem Ergebnis der Auswahl überschreiben). Aus diesem Grund müssen Sie eine Zuordnungsbeziehung zwischen den Feldern der Quelltable und den Feldern der Zieltabelle definieren. Die Höchstzahl möglicher Beziehungen wird durch die Anzahl der verfügbaren Felder in der Zieltabelle bestimmt. Sie können folgende Funktionen verwenden, um die Feldzuordnung festzulegen:

- **Automatische Zuordnung**
Sobald Sie die Quell- und Zieltabelle definiert haben, überprüft das System die Quelltable auf Felder mit demselben Namen und Typ wie die Felder in der Zieltabelle. Jedes übereinstimmende Quellfeld wird dem entsprechenden Zielfeld zugeordnet. Wenn Sie die Programmiersprache ABAP beherrschen, kennen Sie diesen Mechanismus möglicherweise bereits als `MOVE-CORRESPONDING`-Anweisung. Die Automatische Feldzuordnung ist lediglich ein Vorschlag und kann manuell überschrieben werden.
- **Manuelle Zuordnung**

Sie können Quellfelder manuell mit Zielfeldern verknüpfen. Dies ist hilfreich bei der Zuordnung von Feldern mit gleicher Semantik, aber unterschiedlichen Namen. Bei solchen Feldern kann die automatische Zuordnung die Entsprechung nicht erkennen. Sie können auch für bereits vorhandene automatische Zuordnungen manuell aus den Quellfeldern wählen.

Für die manuelle Zuordnung bietet Ihnen das System eine Liste von Quellfeldern in einem Auswahlménü an. Dieses Ménü ist jedoch auf die ersten neun Felder der Tabelle beschränkt. Wenn die Quelltable mehr Felder enthält, können Sie über die Option *Anderes Element auswählen* darauf zugreifen.

- **Feldaggregation und Gruppierung**

Sie können festlegen, dass ein Zielfeld einen aggregierten Wert erhält, der von den Werten im entsprechenden Quellfeld abgeleitet wird. Hierzu bietet das System dieselben Aggregationsfunktionen an, die auch Operationsmodus Aggregation verfügbar sind (Maximum, Minimum, Durchschnitt, Anzahl, Summe). Eine Übersicht über die unterstützten Aggregationsfunktionen für verschiedene Datentypen finden Sie in der entsprechenden Tabelle im oberen Abschnitt *Operationsmodi*.

Wenn Sie die Feldaggregation verwenden, werden alle Zielfelder, die derzeit **keine** Aggregation verwenden, automatisch dem Kennzeichen *Gruppieren nach* zugeordnet. Dadurch wird sichergestellt, dass nur die eindeutigen Tupel dieser Felder in die Zieltabelle geschrieben werden, statt den Wert für das aggregierte Feld für jeden einzelnen Datensatz in der Auswahl zu wiederholen.

i Hinweis

Wenn ein aggregiertes Quellfeld vom Typ Betrag oder Menge ist, fügt das System automatisch eine Gruppierung nach den darin enthaltenen Währungen oder Mengeneinheiten hinzu. Diese zusätzliche Gruppierung wird **nicht** auf der Benutzungsoberfläche der Datenbankabfrage in der BRFplus-Workbench dargestellt. Sie können sie jedoch in der generierten SQL-Anweisung für die Auswahlkriterien sehen.

- **SQL-Anzeige**

Alle Einstellungen, die Sie bezüglich der Feldzuordnung, Aggregation und Gruppierung vornehmen, haben Auswirkungen auf die Art, wie das System auf die Tabelle zugreift, die als Kontext-Datenobjekt des Ausdrucks verwendet wird. Als Anwendungsentwickler könnte es hilfreich für Sie sein, die SQL-Anweisung zu sehen, die das System auf Grundlage Ihrer Einstellungen auf der Benutzungsoberfläche generiert. Hierzu bietet Ihnen das System einen Einblick in die SQL-Repräsentation Ihrer aktuellen Einstellungen. Über die Drucktaste *SQL anzeigen* können Sie die SQL-Anweisung anzeigen.

- **Standardeinstellungen wiederherstellen**

Es kann vorkommen, dass sich nach dem Modifizieren der Standardzuordnung das Verhalten des Ausdrucks unerwünscht verändert. In solchen Fällen ist es hilfreich, die vorgenommenen Änderungen rückgängig zu machen und von vorne zu beginnen. Hierzu können Sie bei der Feldzuordnung die Drucktaste *Standardzuordnung wiederherstellen* verwenden. Diese Funktion setzt alle manuellen Zuordnungen zurück und stellt die automatische Zuordnung auf Grundlage der Logik MOVE-CORRESPONDING wieder her.

Einschränkungen

Die Datenbankabfrage weist Einschränkungen in Bezug auf die Komplexität des Datenbankzugriffsbefehls (WHERE-Klausel) auf, den das System je nach gegebenen Feldern und Bedingungen ausführt. Es gelten folgende Einschränkungen:

- Maximale Anzahl von Feldern: 255

- Maximale Anzahl von Bedingungen: 255

Weitere Informationen

[Einen Datenbankabfrage-Ausdruck anlegen \[Seite 205\]](#)

[Datenbindung \[Seite 66\]](#)

[Tabellenoperation \(Ausdruck\) \[Seite 124\]](#)

1.2.11.6 Entscheidungstabelle (Ausdruck)

Definition

Eine Entscheidungstabelle wird verwendet, um eine gegebene Eingabe mit einer Liste von vordefinierten Bedingungskombinationen zu vergleichen. Jeder Kombination ist ein individuelles Ergebnis zugeordnet, das als Ausdrucksergebnis zurückgeliefert wird, wenn die eingegebenen Daten mit einer bestimmten Kombination übereinstimmen.

Struktur

Entscheidungstabellen enthalten folgende Elemente und Einstellungen:

- Mindestens eine Spalte, die die verschiedenen Bedingungen enthält, mit denen die Eingabedaten verglichen werden
- Mindestens eine Spalte, die das vom Ausdruck zurückgelieferte Ergebnis enthält, sofern alle Bedingungen in einer Zeile erfüllt sind
- Mindestens eine Zeile, die in Bezug auf ihre kombinierten Bedingungen eindeutig ist

Funktionsumfang

Allgemeines Konzept

Eine Entscheidungstabelle verarbeitet sequenziell Geschäftsregeln auf der Grundlage von bestimmten Eingaben. Zur Laufzeit werden die in die Entscheidungstabelle eingegebenen Daten der Reihe nach mit den definierten Bedingungen verglichen, beginnend bei der ersten Zeile. Alle Zellen werden ausgewertet, indem der Spaltenwert gegen eine Reihe von Bereichsvergleichen geprüft wird. Für jede Zelle wird ein boolesches Ergebnis zurückgeliefert. Wenn alle Zellen in einer Zeile als „true“ ausgewertet werden, dann wird die Evaluierung angehalten und der Ausdruck liefert das Ergebnis zurück, das mit den aktuellen Bedingungen verknüpft ist. Andernfalls wird die Verarbeitung mit der nächsten Tabellenzeile fortgesetzt, bis entweder ein passender Bedingungssatz gefunden oder das Ende der Tabelle erreicht wurde.

Hinweis

Aufgrund der sequenziellen Bearbeitung der in der Tabelle definierten Regeln (von oben nach unten), empfiehlt SAP, die Regeln so zu definieren, dass die spezifischste Bedingung in der ersten Zeile und in den Folgezeilen absteigend Regeln mit weniger spezifischen Bedingungen enthalten sind. Anderenfalls könnten Situationen auftreten, in denen eine bestimmte Eingabe, die eine unspezifische Bedingung erfüllt, zum für diese Bedingung definierten Ausdrucksergebnis führt, ohne dass das System prüft, ob eine spezifischere Bedingung vorhanden ist, die besser mit den Eingabedaten übereinstimmt.

Mit den Prüffunktionen von BRFplus können Sie diese sequenziellen Anordnungsfehler ermitteln. Das System gibt entsprechende Fehlermeldungen aus. Über die Funktion [Neu anordnen](#) korrigiert das System automatisch zumindest die offensichtlichsten Fehlanordnungen. So werden Zellen mit unspezifizierten Werten an das Ende der Tabelle verschoben.

Beispiel

Eine Entscheidungstabelle besteht aus den beiden Bedingungsspalten [Region](#) und [Land](#) sowie der Ergebnisspalte [Zahlungsziel](#). Für die Region Europa ist ein Zahlungsziel von 60 Tagen definiert. Für die Region Europa und das Land Deutschland lautet das Zahlungsziel 30 Tage. Die Eingabedaten lauten [Region= Europa](#) und [Land= Deutschland](#). Wenn die Bedingung für die Region **vor** der Kombination aus Region und Land ausgewertet wird, würde die unspezifische Prüfung der Region bereits ein Ergebnis liefern und die spezifischere Bedingung für Land und Region würde überhaupt nicht geprüft. Das bedeutet, das Zahlungsziel für die Eingabe mit [Land= Deutschland](#) würde fälschlicherweise mit 60 anstatt mit 30 Tagen ausgewertet.

Die Daten der Entscheidungstabelle bestehen aus einem Satz verschachtelter Ausdrücke für jede Tabellenzelle. Die Zellen in den Bedingungsspalten enthalten für die Ausdrücke in der Regel Wertebereiche oder Konstanten. Sie verwenden das entsprechende Spaltendatenobjekt als Testparameter. Normalerweise wird eine Bedingungsspalte definiert, indem ihr ein Datenobjekt zugeordnet wird. Bedingungsspalten können aber auch aus geschachtelten Ausdrücken abgeleitet werden. Jede Zelle in einer Zeile kann entweder einen Einzelwert, eine Enumeration eindeutiger Werte oder einen Wertebereich enthalten, auf die die Eingaben geprüft werden.

Ergebnisspalten und Ergebnisdatenobjekt

Die Datenstruktur der Ergebnisspalte wird vom Ergebnisdatenobjekt abgeleitet. Eine Entscheidungstabelle kann auch mehrere Ergebnisspalten enthalten. Dazu ordnen Sie ein Strukturdatenobjekt als Ergebnisdatenobjekt zu oder stellen einfach mehrere elementare Datenobjekte als Ergebnisspalten bereit. Im letzteren Fall können Sie ein neues Strukturdatenobjekt anlegen, das die ausgewählten Elemente enthält. Diese Struktur wird der Entscheidungstabelle dann automatisch als Ergebnisdatenobjekt zugeordnet.


In bestimmten Fällen können Sie die Ergebnisspalte auch als [Aktionsspalte](#) definieren. In diesem Fall definieren Sie eine Aktion für jede Kombination aus Bedingungen, die ausgelöst werden soll, wenn die Eingabedaten der Bedingung in einer Tabellenzeile entsprechen.

Hinweis

Sie können eine Kombination aus einer Aktion und einem Datenobjektwert in der Ergebnisspalte einer Entscheidungstabelle verwenden. Einer Ergebnisspalte kann jedoch nur eine einzige Aktion zugeordnet sein.

Technische Einstellungen ändern

In den folgenden Abschnitten werden die verschiedenen technischen Einstellungen beschrieben, mit denen Sie das Verhalten der Entscheidungstabelle Ihrem Anwendungsfall entsprechend anpassen können. Zu diesen Einstellungen gehören die Definition der Bedingung und der Ergebnisspalten sowie viele weitere Details, die fallspezifisch sind. Um Verwirrungen vorzubeugen, werden diese Einstellungen deshalb standardmäßig ausgeblendet, wenn Sie eine Entscheidungstabelle bearbeiten. Sie rufen die technische Einstellungen wie folgt auf:

- Wenn Sie eine neue Entscheidungstabelle anlegen, wird das Bild mit den technischen Einstellungen angezeigt, damit Sie die erforderlichen Funktionen wie Bedingung und Ergebnisspalten definieren können.
- Wenn Sie eine bereits vorhandene Entscheidungstabelle bearbeiten, rufen Sie die technischen Einstellungen auf, indem Sie  am rechten Rand der Symbolleiste *Details* wählen.

Hinweis

Der Zugriff auf die technischen Einstellungen einer Entscheidungstabelle unterliegt den definierten Personalisierungseinstellungen. Weitere Informationen finden Sie unter [Workbench-Personalisierung \[Seite 12\]](#) im Abschnitt *Ausdruck – Entscheidungstabelle*.

Zudem können technische Einstellungen nicht geändert werden, solange ein Filter aktiv ist, welcher den Inhalt der Entscheidungstabelle eingrenzt. Auf diese Weise wird unerwartetes und nicht eindeutiges Systemverhalten vermieden. Beispiel: Wenn ein Filter aktiv ist, sodass nur die Zeilen 5 und 12 von insgesamt 25 angezeigt werden, führt das Einfügen einer neuen Zeile zwischen den Zeilen 5 und 12 dazu, dass nicht klar ist an welcher Stelle genau die neue Zeile platziert werden soll. Deshalb deaktiviert das System Funktionen, die zu Verwirrungen bei aktiviertem Filtermodus führen könnten.

Übereinstimmungsmodi Einzelwert und Mehrfachtreffer

Eine Entscheidungstabelle kann in zwei unterschiedlichen Betriebsmodi ausgeführt werden. Der Modus ist abhängig vom Indikator *Alle übereinst. Werte zurückgeben*:

- Übereinstimmungsmodus Einzelwert (*Alle übereinst. Werte zurückgeben*= **false**)
Im Übereinstimmungsmodus Einzelwert prüft das System die Eingabedaten auf die in einer Tabellenzeile definierten Bedingungen. Wenn die Eingabe alle Bedingungen erfüllt, wird das dazugehörige Ergebnis ausgegeben und die Verarbeitung der Entscheidungstabelle stoppt. Das System gibt also das Ergebnis aus, das der ersten Zeile mit Übereinstimmung zugeordnet ist. Dieser Modus ist standardmäßig eingestellt.
- Übereinstimmungsmodus Mehrfachtreffer (*Alle übereinst. Werte zurückgeben*= **true**)
Im Übereinstimmungsmodus Mehrfachtreffer prüft das System die Eingabedaten auf die in **allen** Tabellenzeilen definierten Bedingungen. Alle übereinstimmenden Ergebnisse werden im Ergebnisdatenobjekt gesammelt und als Ergebnis der Auswertung ausgegeben.

Hinweis

Da der Übereinstimmungsmodus Mehrfachtreffer für die Auslieferung mehrerer Werte im Ergebnis des Ausdrucks konzipiert ist, muss das Ergebnisdatenobjekt für eine Entscheidungstabelle im Übereinstimmungsmodus Mehrfachtreffer immer ein Datenobjekt vom Typ „Tabelle“ sein.

Teilübereinstimmungen verarbeiten

Beide Übereinstimmungsmodi folgen dem Grundprinzip, dass eine Übereinstimmung nur vorliegt, wenn die an eine Entscheidungstabelle geleiteten Eingabedaten **allen** angegebenen Bedingungswerten in einer Zeile entsprechen. Möglicherweise gibt es aber Anwendungsfälle, für die die Eingabe mit keiner Zeile der Entscheidungstabelle übereinstimmt. In diesem Fall ist das Auslösen einer Ausnahme nicht ausreichend. In

solchen Fällen werden Teilübereinstimmungen verarbeitet. Dieser Verarbeitungsmodus wurde für folgende Szenarien entwickelt:

- Die an die Entscheidungstabelle weitergeleiteten Kontextdaten entsprechen keiner der Bedingungszeilen in der Tabelle.
- Sie möchten die Verarbeitung nicht nur auf der Tatsache basierend fortsetzen, dass keine Übereinstimmung vorliegt, sondern darauf, wie nah die Kontextdaten einer Übereinstimmung kommen (ohne eine Bedingung vollständig zu erfüllen).

Setzen Sie in diesem Fall das Kennzeichen *Bei Teilübereinstimmung Ausnahme auslösen*. Das Setzen des Kennzeichens bewirkt Folgendes: Bei der Verarbeitung der Entscheidungstabelle erfasst das System die maximale Anzahl der erfüllten Bedingungen pro Zeile. Wenn nach der Verarbeitung der letzten Tabellenzeile keine Übereinstimmung vorliegt, gibt das System die Ausnahme `CX_FDT_PARTIAL_MATCH` mit der höchsten erfassten Übereinstimmungsrate aus, die an das Attribut `CX_FDT_PARTIAL_MATCH-`

`>MV_MAX_COND_MATCHED` weitergeleitet wurde. Diese Ausnahme wird dann von der aufrufenden Anwendung verarbeitet, die basierend auf der maximalen Anzahl der erfüllten Bedingungen für die Eingabe das gewünschte Systemverhalten implementiert.

Hinweis

Das Setzen des Kennzeichens *Bei Teilübereinstimmung Ausnahme auslösen* wirkt sich negativ auf die Systemleistung aus. Grund hierfür ist einmal der erforderliche Vergleich der Anzahl an Bedingungen für jede Zeile und zum Anderen können bestimmte Optimierungsmaßnahmen während der Code-Generierung nicht angewendet werden. Deshalb empfiehlt SAP, dieses Kennzeichen nur zu setzen, wenn der Zugriff auf die maximale Anzahl an übereinstimmenden Bedingungen für Ihren Anwendungsfall unbedingt erforderlich ist.

Initialwert

Wenn keine übereinstimmende Bedingung für eine bestimmte Eingabe gefunden wird, gibt die Entscheidungstabelle eine Ausnahme aus anstatt eines Ergebnisses. Dieses Standardverhalten können Sie überschreiben, indem Sie das Kennzeichen *Initialwert zurückgeben, wenn keine Übereinstimmung gefunden wird* setzen. Wenn dieses Kennzeichen gesetzt ist, gibt die Entscheidungstabelle den Initialwert zurück, der für den Typ des Ergebnisdatenobjekts definiert ist. Auf diese Weise können Sie die nicht erfolgreiche Bearbeitung als definiertes Ergebnis berücksichtigen anstatt als Fehler und so die Verarbeitung fortsetzen.

Hinweis

Unter Umständen ist es bei dieser Option erforderlich, zusätzliche Maßnahmen zu ergreifen, damit die weiteren Verarbeitungsschritte einen Initialwert nicht mit einem wohldefinierten Ergebnis verwechseln. Beispiel: Der Initialwert für ein Ergebnisdatenobjekt des Typs „Nummer“ lautet 0. Dies könnte allerdings auch das definierte Ergebnis für eine erfüllte Bedingung sein.

Inhaltsprüfungen für Tabellen

Zusätzlich zu den Standardprüfungen auf technische Konsistenz können Sie weitere Prüfungen durchführen, um sicherzustellen, dass keine Fehler in den Daten der Entscheidungstabelle vorliegen und dass diese die erwarteten Ergebnisse aus Geschäftssicht zurückgibt. Weitere Informationen finden Sie unter [Inhaltsprüfungen für Entscheidungstabellen \[Seite 96\]](#).

Tabellenoptimierung

BRFplus stellt Werkzeuge bereit, mit denen Sie den Tabelleninhalt verdichten und die Tabellenanordnung optimieren können, um eine bessere Performance zu erzielen. Weitere Informationen finden Sie unter [Entscheidungstabellen optimieren \[Seite 97\]](#).

Optionale und erforderliche Einträge von Bedingungsdaten

Standardmäßig erfolgt die Auswertung einer Entscheidungstabelle wie folgt:

- Der Wert eines Eingabedatenfelds wird mit dem Wert in der entsprechenden Bedingungsspalte der Tabelle verglichen.
- Wenn die zwei Werte übereinstimmen, wird die Bedingungszeile als „true“ ausgewertet.
- Dieser Prozess wird für alle Bedingungen in der Tabellenzeile wiederholt, bis entweder **alle** Bedingungen erfüllt sind (Zeile ist eine Übereinstimmung), oder bis eine Bedingung **nicht** erfüllt wird. In diesem Fall wird die Verarbeitung in der nächsten Tabellenzeile fortgesetzt.

Wenn dieses Standardverhalten nicht Ihren Anforderungen entspricht, können Sie es wie folgt individuell für jede Bedingungsspalte ändern:

- **Obligatorische Eingabe**
Ist dieses Kennzeichen gesetzt, müssen Sie **zur Design-Zeit** einen Bedingungsvalue für die entsprechende Spalte in jeder Zeile, die Sie in die Tabellendefinition eingeben, bereitstellen. Eine solche Bedingung kann deshalb **nicht** zum Bereitstellen einer unspezifischen leeren Bedingung verwendet werden, die unabhängig von den Eingabedaten als `true` ausgewertet werden würde.

Beispiel

Im obigen Beispiel mit den beiden Bedingungsspalten *Region* und *Land* würde die Spalte *Region* als obligatorischer Eintrag definiert, um sicherzustellen, dass eine Eingabe mindestens mit einer der in der Entscheidungstabelle bereitgestellten Regionen übereinstimmen muss, damit eine weitere Auswertung erfolgt. Die Spalte *Land* ist dagegen **nicht** obligatorisch, sodass eine Eingabe mit übereinstimmender Region und unbekanntem Land trotzdem verarbeitet werden kann. Um dies zu erreichen, müssen Sie eine zusätzliche Tabellenzeile für jede Region, kombiniert mit einem undefinierten Land, einfügen.

Tabelleninhalt abrufen

Abhängig vom Anwendungsfall können Entscheidungstabellen eine große Anzahl an Zeilen enthalten, viele Tausend Zeilen sowie Dutzende oder Hunderte von Spalten. So kann es schwierig sein, eine bestimmte Zeile zu finden. BRFplus bietet die folgenden Tools, mit denen Sie durch den Inhalt einer Entscheidungstabelle navigieren können:

Zeilenfilter

Sie können die Filterkriterien definieren, die eine Tabellenzeile erfüllen muss, damit sie angezeigt wird. Sie können für jede Spalte einen separaten Filter definieren, der nur für diese Spalte gilt. Wenn Sie Filter für mehrere Spalten definieren, werden die Filterkriterien mit einem logischen **AND**-Operator kombiniert, bevor sie ausgewertet werden. Eine Zeile wird also nur angezeigt, wenn alle derzeit gültigen Filterkriterien gleichzeitig von den betroffenen Spalten einer Zeile erfüllt werden.

Filterbedingungen sind so lange gültig, bis Sie sie ändern oder zurücksetzen, oder bis Sie zu einem anderen Objekt navigieren. Es ist **nicht** möglich, einer Entscheidungstabelle bestimmte Filterbedingungen dauerhaft zuzuordnen.

Um einen Filter für eine Spalte zu definieren, bewegen Sie den Mauszeiger über die Überschriftszelle der gewünschten Spalte und wählen Sie das kleine Dreieck in der unteren rechten Ecke aus. Wurde für die Spalte bereits ein Filter definiert, zeigt das System ein Filtersymbol in der Zelle mit dem Spaltentitel.

Suchen

Mit diesem Tool durchsuchen Sie den Tabelleninhalt nach einer Zeichenfolge. Wenn die Suchzeichenfolge in der Tabelle vorhanden ist, wird die entsprechende Zeile hervorgehoben. Mit den Drucktasten „Nächste“ und „Vorige“ können Sie zu den weiteren Treffern der Suchzeichenfolge navigieren.

Hinweis

Für beide Tools gilt Folgendes: Der Vergleich zwischen der Filterbedingung oder Suchzeichenfolge und dem tatsächlichen Zelleninhalt erfolgt über eine implizite Inhaltsumwandlung in Textform. Das bedeutet, Sie müssen beim Definieren der Filterbedingung alles berücksichtigen, was in einer Zelle angezeigt wird. Beispiel: Wenn eine Zelle einen Wertebereich enthält, der mit **[10..20]** angezeigt wird, und Sie diese Zelle in einen Filter aufnehmen möchten, müssen Sie die eckigen Klammern explizit im Filterkriterium aufführen: **[10*]**. Wenn Sie lediglich **10*** ohne Klammern als Filterkriterium angeben, würde die Zelle ausgeschlossen, obwohl die eckigen Klammern nicht wirklich Teil des Zelleninhalts sind und nur der Indikator für den Wertebereich in der Zelle sind.

Beide Tools unterstützen Platzhalter: **+** steht für ein einzelnes Zeichen, ***** für eine beliebige Zeichenfolge. Groß-/Kleinschreibung wird bei allen Suchen und Zeichenfolgenvergleichen nicht berücksichtigt.

Wenn eine Suche mit einer gültigen Filterbedingung durchgeführt wird, wird die Suche auf die Zeilen begrenzt, die sich im Ordner befinden und daher sichtbar sind.

Tabellendaten ex- und importieren

Sie können Bedingungs- und Ergebnisdaten einer Entscheidungstabelle zwischen BRFplus und Microsoft Excel austauschen. So können Sie die Tabellendaten mit Personen teilen, die keinen Zugang zu BRFplus haben, Sie jedoch in geschäftlicher Hinsicht unterstützen können. Die erforderlichen Schritte zum Ex- und Importieren von Daten werden im Abschnitt [Daten zwischen Entscheidungstabellen und Microsoft Excel austauschen \[Seite 100\]](#) beschrieben.

Spaltenschutz

Ein großer Vorteil von BRFplus ist, dass die Kriterien für Geschäftsentscheidungen in Ihrem Unternehmen transparent und leicht zu ändern sind. Transparenz kann zwar als wichtiger Faktor gesehen werden, ist aber für manche Teile Ihres Unternehmens möglicherweise unpassend. So können Sie festlegen, dass die Entscheidungskriterien für einen bestimmten Geschäftsfall gegen nicht autorisierte Änderungen geschützt oder überhaupt nicht angezeigt werden sollen. Dazu verwenden Sie die Einstellung [Spaltenzugriffsberechtigung](#) sowohl für die Bedingungs- als auch für die Ergebnisspalten einer Entscheidungstabelle:

- **Vollzugriff (Änderungen zulässig)**
Bei dieser Einstellung kann ein Benutzer die Werte der Spaltenzellen ändern. Dies ist die Standardeinstellung.
- **Nur anzeigen (keine Änderungen möglich)**
Bei dieser Einstellung sieht der Benutzer die Werte in den Spaltenzellen, kann sie aber nicht ändern.
- **Ausgeblendet**
Bei dieser Einstellung wird die Spalte in der BRFplus-Workbench sowohl zur Design- als auch zur Laufzeit vollständig ausgeblendet. Ohne eine Prüfung der Tabelleneinstellungen erhält der Benutzer keinen sichtbaren Hinweis darauf, dass die Spalte existiert.

Beispiel

Die folgende Tabelle stellt die Entscheidungstabelle des obigen Beispiels dar, mit der das Zahlungsziel für eine Kundenrechnung ermittelt wird:

Tabelle 22:

Region*	Land	Unternehmen	Zahlungsziel (Tage)
AMERICAS	USA	Miller Inc.	90
AMERICAS	USA		75
Europa	Deutschland		30
Europa	Frankreich		45
Europa			60

Diese Tabelle enthält eine Liste mit verschiedenen Zahlungszielen entsprechend dem jeweiligen Geschäftsgebaren an verschiedenen Orten der Welt. Die Spalten *Region*, *Land* und *Unternehmen* stellen die Bedingungskriterien (Kontextdaten) dar. Die Spalte *Zahlung (Tage)* enthält die Ergebniswerte für die einzelnen Bedingungen (Ergebnisdatenobjekt). Die Tabelle berücksichtigt, dass in Deutschland Rechnungen in relativ kurzer Zeit nach dem Empfang beglichen werden, wohingegen in anderen Regionen oder Ländern ein längerer Zeitraum akzeptabel ist. Es kann auch nützlich sein, spezielle Zahlungsziele für einzelne Kunden zu definieren, abhängig von deren Kapitalstärke oder aufgrund der Erfahrungen, die Ihr Unternehmen in der Vergangenheit mit diesen Kunden gemacht hat.

Die Einträge in der Tabelle sind nach der Anzahl der verfügbaren spezifischen Informationen sortiert, beginnend mit dem spezifischen Eintrag für ein einzelnes Unternehmen in den USA und endend mit einem generischen Eintrag für alle in Europa ansässigen Kunden, die die Bedingungen der spezifischeren Zeilen nicht erfüllen.

Hinweis

Verwechseln Sie diese Sortierreihenfolge nicht mit der gängigen alphanumerischen Sortierfunktion in den meisten Table Controls. In einer Entscheidungstabelle müssen Sie die Einträge manuell sortieren, sodass sie dem jeweiligen Geschäftsfall am besten entsprechen. Es handelt sich also um eine semantische Sortierreihenfolge basierend auf Geschäftsentscheidungen und **nicht** um eine auf Namen basierende alphanumerische Reihenfolge.

Der Asterisk (*) in der Spalte *Region** gibt an, dass Sie in diese Spalte immer einen Eintrag eingeben müssen. Mit anderen Worten, die Entscheidungstabelle benötigt mindestens die Angabe der Region, um ein Zahlungsziel für den Kunden zu ermitteln.

Die Einträge der Entscheidungstabelle für die Region AMERICAS definieren ausschließlich Zahlungsziele für in den USA ansässige Kunden. Beispiel: Wenn Sie zur Laufzeit einen Kunden aus Kanada oder Brasilien eingeben, gibt die Entscheidungstabelle eine Ausnahme aus, da für diese Länder keine spezifische Bedingung vorliegt und auch keine generische Bedingung für alle Länder der Region AMERICAS (außer den USA) definiert wurde.

Weitere Informationen

[Entscheidungstabelle \(Ausdruck\) anlegen \[Seite 208\]](#)

[Inhaltsprüfungen für Entscheidungstabellen \[Seite 96\]](#)

[Entscheidungstabellen optimieren \[Seite 97\]](#)

[Daten zwischen Entscheidungstabellen und Microsoft Excel austauschen \[Seite 100\]](#)

1.2.11.6.1 Inhaltsprüfungen für Entscheidungstabellen

Definition

Zusätzlich zu den Prüfungen auf technische Korrektheit, die für alle BRFplus-Objekttypen vorhanden sind, können Sie einige inhaltsorientierte Prüfungen für eine Entscheidungstabelle durchführen. Mithilfe dieser Prüfungen stellen Sie sicher, dass die Entscheidungstabelle fehlerfrei ist und aus Geschäftssicht die erwarteten Ergebnisse liefert. Im Gegensatz zu technischen Prüfungen können Sie Inhaltsprüfungen explizit mit den entsprechenden Befehlen durchführen, die im Menü *Zusätzliche Aktionen* zusammengefasst sind. Folgende Menüeinträge sind vorhanden:

- *Vollständigkeit überprüfen*

Mit der Vollständigkeitsprüfung stellt das System sicher, dass die Werte einer bestimmten Bedingungsspalte als kontinuierliche Wertefolge ohne Lücken definiert sind. Wenn die Wertefolge Lücken aufweist, kann es vorkommen, dass ein Eingabewert nicht verarbeitet werden kann. Aus Geschäftssicht müssen jedoch alle möglichen Werte innerhalb eines gegebenen Bereichs von der Entscheidungstabelle verarbeitet werden.

Wenn das System nach Abschluss der Prüfung Wertelücken gefunden hat, schlägt es zusätzliche Tabellenzeilen vor, die die Lücke schließen würden. Auf diese Weise können Sie schnell sicherstellen, dass die Tabelle für die noch nicht abgedeckten Wertekombinationen ein definiertes Ergebnis zurückliefert. In bestimmten Fällen würden diese Zeilenvorschläge Ihre Tabelle allerdings nur formal vervollständigen. Sie müssen die Daten dann manuell anpassen, damit sie Ihre Geschäftsanforderungen erfüllen.

Beispiel

Eine Entscheidungstabelle hat unter anderem die Bedingungsspalte *Beförderungsklasse* mit einem zugeordneten Datenobjekt, das an eine Liste mit den drei Domänenwerten *Economy Class*, *Business Class* und *First Class* gebunden ist. Während der Vollständigkeitsprüfung stellt das System fest, dass bei einer bestimmten Kombination von Bedingungen nur die Werte *Business Class* und *First Class* berücksichtigt werden und keine Tabellenzeile für *Economy Class* vorhanden ist. Die Prüfung benachrichtigt Sie daher über diese potenzielle Lücke. Sie können anschließend entscheiden, ob der fehlende Wert absichtlich ausgelassen wurde oder nicht.

- *Überlappungen überprüfen*

Bei der Überlappungsprüfung untersucht das System, ob die den Bedingungsspalten zugeordneten Wertebereiche eindeutig sind und sich nicht überlappen. Bei sich überlappenden Wertebereichen könnte das System einen Eingabewert, der in diesen Bereich fällt, nicht richtig bearbeiten.

Beispiel

Eine Entscheidungstabelle hat eine Bedingungsspalte *Rechnungssumme* mit einem definierten Wertebereich 500 . . . 1000 in einer Zeile und einem weiteren Wertebereich 750 . . . 1500 in einer anderen Zeile. Für eine Eingabe von 900 als Rechnungssumme wäre es in der Entscheidungstabelle nicht möglich festzustellen, welcher der beiden Wertebereiche der zutreffende ist. Das System würde daher einfach die erste Zeile auswählen, zu der die Eingabe passt, auch wenn dies aus Geschäftssicht nicht die beabsichtigte Entscheidung wäre.

Die oben beschriebenen Prüfungen werden immer durchgeführt, wenn Sie sie explizit durch Wählen der entsprechenden Option ausführen, oder als implizite Prüfung, wenn Sie die Entscheidungstabelle aktivieren. Da diese Prüfungen sich allerdings auf Korrektheit aus Geschäftssicht beziehen, und nicht auf technische Konsistenz, können Sie selbst festlegen, wie das System sich bei einem gefundenen Problem verhalten soll. Im Dialogfenster *Tabelleneinstellungen* stehen Ihnen folgende Optionen zur Verfügung:

- *Anwendungsstandard*
Wenn ein Problem gefunden wurde, sendet das System die entsprechende Nachricht mit dem im Backend-System definierten Nachrichtentyp (in den meisten Fällen wird der Typ 'Warnung' verwendet).
- *Nachrichten als Fehler anzeigen*
Mit dieser Option werden Sie gezwungen, das Problem in der Tabellendefinition zu lösen. Andernfalls kann die Entscheidungstabelle nicht aktiviert werden.
- *Nachrichten als Warnung anzeigen*
Das System zeigt Ihnen das Problem auf, aber Sie können die Entscheidungstabelle dennoch aktivieren.
- *Keine Nachrichten anzeigen*
Auch wenn ein Problem gefunden wurde, benachrichtigt Sie das System **nicht** darüber. Sie können die Entscheidungstabelle aktivieren.

Hinweis

Inhaltsprüfungen für Tabellen sind performance-kritisch, da sie teilweise Operationen beinhalten, bei denen jede Zelle in einer Tabelle mit allen anderen Zellen verglichen werden muss. Um einer Überlastung des Systems vorzubeugen, können die Prüfungen nur ausgeführt werden, solange bestimmte Beschränkungen nicht überschritten werden. Die Vollständigkeitsprüfung kann beispielsweise nur für Tabellen mit maximal 10000 Zeilen ausgeführt werden.

1.2.11.6.2 Optimierung von Entscheidungstabellen

Definition

Die Entscheidungstabelle ist die beliebteste und am häufigsten verwendete Ausdrucksart in BRFplus. Erfahrungsgemäß erstellen Kunden häufig komplexe Tabellen mit dutzenden Spalten und tausenden (manchmal zehntausenden) Zeilen. Diese Komplexität hat jedoch negative Auswirkungen auf die Performance. Aus diesem Grund stellt BRFplus Werkzeuge bereit, mit deren Hilfe Sie den Tabelleninhalt

verdichten und die Tabellenanordnung optimieren können, um eine bessere Performance zu erzielen. Die folgenden Werkzeuge sind im Menü [Zusätzliche Aktionen](#) verfügbar:

- [Neu anordnen](#)

Mithilfe dieser Funktion können Sie Tabellenzeilen, die unbesetzte (leere) Zellen enthalten, an das untere Ende der Tabelle verschieben. Dadurch stellen Sie sicher, dass spezifischere Bedingungen in einer Tabelle zuerst verarbeitet werden.

Wenn Sie diese Funktion ausführen, werden Sie zunächst aufgefordert, eine Sortierreihenfolge für die Tabellenspalten zu definieren. Standardmäßig durchläuft das System die Spalten von links nach rechts. Es sucht für jede Spalte leere Zellen und verschiebt die entsprechenden Zeilen an das untere Ende der Tabelle. Zeilen werden höchstens ein Mal verschoben. Zeilen, die leere Zellen in weniger priorisierten Spalten enthalten, werden über bereits verschobene Zeilen, jedoch unter Zeilen ohne leere Zellen gesetzt.

Beispiel

Das folgende Beispiel veranschaulicht die Funktion [Neu anordnen](#). Die neu anzuordnende Tabelle besitzt leere Zellen an den Positionen A5 und B3.

Tabelle 23:

#	A	B
1	P100	1
2	P300	0..2
3	P100	<...>
4	P300	2
5	<...>	3
6	P200	3

Tabelle 24:

#	A	B
1	P100	1
2	P300	0..2
4	P300	2
6	P200	3
3	P100	<...>
5	<...>	3

Im ersten Schritt analysiert das System die Spalte A, findet eine leere Zelle in Zeile 5 und verschiebt Zeile 5 daher an das untere Ende der Tabelle. Nachdem Spalte A analysiert wurde, fährt das System mit Spalte B fort und findet eine leere Zelle in Zeile 3. Es verschiebt Zeile 3 nach unten, jedoch eine Zeile über die bereits verschobene Zeile 5. Alle anderen Zeilen bleiben auf ihrer ursprünglichen Position.

Tabelle 25:

#	A	B
1	P100	1
2	P300	0..2
4	P300	2
6	P200	3
5	<...>	3
3	P100	<...>

Aufgrund der geänderten Sortierreihenfolge beginnt das System mit der Analyse von Spalte B. Folglich wird zuerst Reihe 3 an das untere Ende der Tabelle verschoben und anschließend Zeile 5 auf die Position direkt über der bereits verschobenen Zeile 3 gesetzt.

- **Zeilen zusammenführen**

Bei dieser Funktion untersucht das System eine Reihe an ausgewählten Zeilen auf die Möglichkeit, in verschiedenen Zeilen definierte Bedingungen in einer einzelnen Zeile zusammenzufassen. Wenn es eine passende Lösung findet, legt es eine neue Zeile mit den zusammengefassten Bedingungen an und löscht alle Zeilen, deren Bedingungen zusammengefasst wurden. Diese Funktion setzt voraus, dass mindestens zwei Zeilen ausgewählt wurden und dass allen ausgewählten Zeilen derselbe Ergebnisdatenwert zugeordnet ist.

Beispiel

Sie möchten eine Tabelle so weit wie möglich zusammenfassen. Hierzu markieren Sie alle Zeilen der folgenden Tabelle und wählen ► **Zusätzliche Aktionen** ► **Zeilen zusammenführen** ►.

Tabelle 26:

#	A	B	Ergebnis
1	P100	1	hoch
2	P100	2	niedrig
3	P100	<...>	hoch
4	P100	3	hoch
5	P100	4	hoch
6	P200	3	hoch

Tabelle 27:

#	A	B	Ergebnis
1	P100	1; 3..4	hoch
2	P100	2	niedrig
3	P100	<...>	hoch
6	P200	3	hoch

Nach der Zusammenführung wurden die Zeilen 1, 4 und 5 in Zeile 1 zusammengefasst. Die Zeilen 4 und 5 wurden gelöscht. Die übrigen Zeilen können aus folgenden Gründen nicht zusammengeführt werden:

- Zeile 2 hat ein anderes Ergebnis als die anderen Zeilen
- Zeile 3 hat eine leere Zelle, die nicht mit spezifischen Werten kombiniert werden kann
- Zeile 6 hat einen anderen Wert in Spalte A

1.2.11.6.3 Daten zwischen Entscheidungstabellen und Microsoft Excel austauschen

Verwendung

Sie können Bedingungs- und Ergebnisdaten einer Entscheidungstabelle zwischen BRFPplus und Microsoft Excel austauschen. Diese Funktion ist für folgende Situationen hilfreich:

- Sie können Ihre Tabellendaten mit Personen teilen, die keinen Zugang zu BRFPplus haben, Sie jedoch in geschäftlicher Hinsicht unterstützen können.
- Der Anwendungsfall, den Sie mit der Entscheidungstabelle angehen möchten, wurde bereits in Ihrer Abteilung bearbeitet. Daher ist möglicherweise bereits eine Vielzahl an genau definierten Regeln vorhanden, die lediglich an BRFPplus angepasst werden müssen.
- Sie können Ihre Entscheidungstabelle in eine Excel-Tabelle exportieren und dadurch an Ihren Bedingungen weiterarbeiten, wenn Sie nicht online sind oder wenn das System nicht verfügbar ist.
- Sie können die Exportfunktion verwenden, um eine zusätzliche, leicht abrufbare Sicherung der Tabellendaten anzulegen.

➔ Empfehlung

BRFPplus versucht beim Import der Daten aus einer Excel-Tabelle, die zu importierenden Daten den Spalten der Entscheidungstabelle zuzuordnen. Um besser verstehen zu können, wie die Tabellendaten für einen erfolgreichen Import in eine Entscheidungstabelle aussehen müssen, empfehlen wir, zunächst einige Daten aus der Entscheidungstabelle in eine Excel-Tabelle zu **exportieren**. Diese Tabelle enthält die Spalten und Datenformate, die für einen fehlerfreien Import benötigt werden.

Wenn Sie die Daten einer Entscheidungstabelle in eine Excel-Datei exportieren, enthält die erzeugte Datei drei Arbeitsblätter:

- **Daten:** In diesem Arbeitsblatt finden Sie alle Bedingungs- und Ergebnisdaten, die Sie für die Entscheidungstabelle in BRFplus gepflegt haben.
- **Hilfe:** In diesem Arbeitsblatt finden Sie detaillierte Informationen zu den korrekten Datenformaten für Werte und Operatoren, die von der BRFplus-Importfunktion erwartet werden. Dieses Arbeitsblatt dient Ihrer Information und ist Teil jeder Excel-Datei, die Sie mit der BRFplus-Exportfunktion anlegen.
- **Spaltendetails:** In diesem Arbeitsblatt finden Sie technische Informationen zum jeweiligen elementaren Datenobjekt, auf das jede Tabellenspalte verweist. Mithilfe dieser Informationen können Sie entscheiden, ob ein manuell eingegebener Wert ohne Weiteres importiert werden kann. Auch dieses Arbeitsblatt ist Teil jeder Excel-Datei, die Sie mit der BRFplus-Exportfunktion anlegen.

Behandlung nicht unterstützter Datentypen

Der Datenaustausch zwischen BRFplus-Entscheidungstabellen und Excel-Dateien wird nur für diskrete Werte und Wertebereichs-Ausdrücke unterstützt. Jeden anderen Datentyp (beispielsweise einen Booleschen Ausdruck) behandelt das System – abhängig von der Übertragungsrichtung und der Einstellung des Kennzeichens *Übersteuerung von Ausdrücken mit festen Werten* (nur relevant für den Import) – entsprechend den folgenden Regeln:

Tabelle 28:

Übersteuerung von Ausdrücken	Export	Import
false (default)	Text des Ausdrucks wird aus Referenzgründen in Excel-Zelle geschrieben.	Es werden nur Einzelwerte oder Wertebereiche importiert. Alle anderen Zellendaten werden beim Import übersprungen; die entsprechende Zelle in der Entscheidungstabelle bleibt unverändert.
true		<p>Alle Zellendaten außer Einzelwerten oder Wertebereichen lösen eine Fehlermeldung aus und der Import wird gestoppt.</p> <p>Wenn die zu füllende Zelle in der Entscheidungstabelle einen Ausdruck enthält und die Excel-Zelle einen Einzelwert oder einen Wertebereich, wird der Ausdruck durch den importierten Wert ersetzt.</p>

Voraussetzungen

Der Datenaustausch mit Microsoft Excel erfolgt mithilfe des mit Microsoft Excel 2007 eingeführten XML-basierten Dateiformats (Dateiendung .XLSX). Wenn Sie eine ältere Version von Microsoft Office verwenden, können Sie im Microsoft Download Center das Microsoft Office Compatibility Pack herunterladen (<http://www.microsoft.com/en-us/download/details.aspx?id=3>). Wenn Sie das Compatibility Pack heruntergeladen haben, müssen Sie es auf jedem Computer installieren, auf dem eine ältere Version von Microsoft Office installiert ist.

Hinweis

Sie benötigen Administrationsrechte, um das Compatibility Pack zu installieren.

Aktivitäten

Daten von Entscheidungstabellen nach Microsoft Excel exportieren


1. Öffnen Sie in der BRFplus-Workbench im Bereich *Objektmanager* die Entscheidungstabelle, deren Daten Sie exportieren möchten.
2. Choose ► *Additional Actions* ► *Export to Excel* .
Sie werden gefragt, ob Sie die Excel-Datei öffnen oder speichern möchten. Je nachdem, wie Sie sich entscheiden, führt das System eine der folgenden Aktionen durch:
 - *Öffnen*: Die exportierten Daten werden in eine Excel-Datei geschrieben, die in einem lokalen Ordner für temporäre Dateien abgelegt wird. Diese Datei wird in Microsoft Excel geöffnet.
 - *Speichern*: Es öffnet sich ein Dialogfenster *Speichern unter*, in dem Sie den gewünschten Dateinamen und Speicherort für Ihre exportierte Datei angeben können. Standardmäßig wird für den Dateinamen `<Name der Entscheidungstabelle>.xlsx` vorgeschlagen.

Hinweis

Textzellen in einer Entscheidungstabelle, die nur ein Leerzeichen enthalten (also ein einzelnes ASCII-32-Zeichen), werden beim Import und Export besonders behandelt: Aus Kompatibilitätsgründen werden Leerzeichen als Folge von drei Zeichen kodiert: einem Anführungszeichen, einem Leerzeichen und einem weiteren Anführungszeichen (" " oder ASCII-34 ASCII-32 ASCII-34). Diese drei Zeichen lange Folge wird beim Export in die Excel-Datei geschrieben.

Beim Import wird die Zeichenfolge wieder in ihre ursprüngliche Form zurückgesetzt, also eine Zelle, die ein Leerzeichen (ASCII-32) enthält. Beachten Sie diese implizite Konvertierung vor allem beim Import von bereits vorhandenen Excel-Dateien. Prüfen Sie in solchen Fällen die Datei auf Zellen die ein einzelnes Leerzeichen enthalten. Ist dies der Fall, verwenden Sie **Ersetzen**-Funktion von Microsoft Excel, um alle einfachen Leerzeichen durch die oben genannte Folge aus drei Zeichen zu ersetzen. Stellen Sie sicher, dass Sie für die Durchführung das Kontrollkästchen **Gesamten Zelleninhalt vergleichen** markiert haben.

Daten von Entscheidungstabellen aus Microsoft Excel importieren

1. Öffnen Sie in der BRFplus-Workbench im Bereich *Objektmanager* die Entscheidungstabelle, in die Sie Daten aus einer Excel-Datei importieren möchten.
2. Wählen Sie ► *Zusätzliche Aktionen* ► *Aus Excel importieren* .
3. Geben Sie im Dialogfenster *Excel-Datei import* den Namen der zu importierenden Excel-Datei ein.
4. Legen Sie das Systemverhalten für Excel-Dateien fest, deren Inhalte nicht den Anforderungen für einen korrekten Import entsprechen.
Weitere Informationen hierzu finden Sie im oberen Abschnitt "Behandlung nicht unterstützter Datentypen".
5. Sichern Sie nach dem Import der Daten die Entscheidungstabelle.

i Hinweis

Beachten Sie, dass BRFplus-Datenobjekte vom Typ Text eine Maximallänge von 255 Zeichen haben. Wenn eine Excel-Zelle alphanumerische Daten enthält, die diese Länge überschreiten, wird der Inhalt beim Import nach 255 Zeichen abgeschnitten.

1.2.11.7 Entscheidungsbaum (Ausdruck)

Verwendung

Mithilfe eines Entscheidungsbaums (Ausdruck) definieren Sie einen binären Baum von Ausdrücken. Die Nicht-Blattknoten heißen Bedingungsknoten. Diese Knoten geben ein Ergebnis in Form eines Booleschen Ausdrucks zurück. Abhängig vom Ergebnis wird dann der linke oder rechte Knoten verarbeitet. Sobald ein Blattknoten erreicht wird, wird der zugewiesene Ausdruck verarbeitet und das entsprechende Ergebnis als Ergebnis des gesamten Ausdrucks ausgegeben.

Struktur

Die Struktur des Entscheidungsbaums besteht aus einer verketteten Liste von Ja-/Nein-Entscheidungen. Jeder Bedingungsknoten ist in zwei Zweige unterteilt, die die Optionen `true` oder `false` für diese Bedingung

darstellen. Jeder Zweig führt entweder zu einer weiteren Bedingung zur Verfeinerung oder zu einem Blattknoten mit einem zugewiesenen Ergebnis. Die folgende Abbildung zeigt das Grundprinzip:

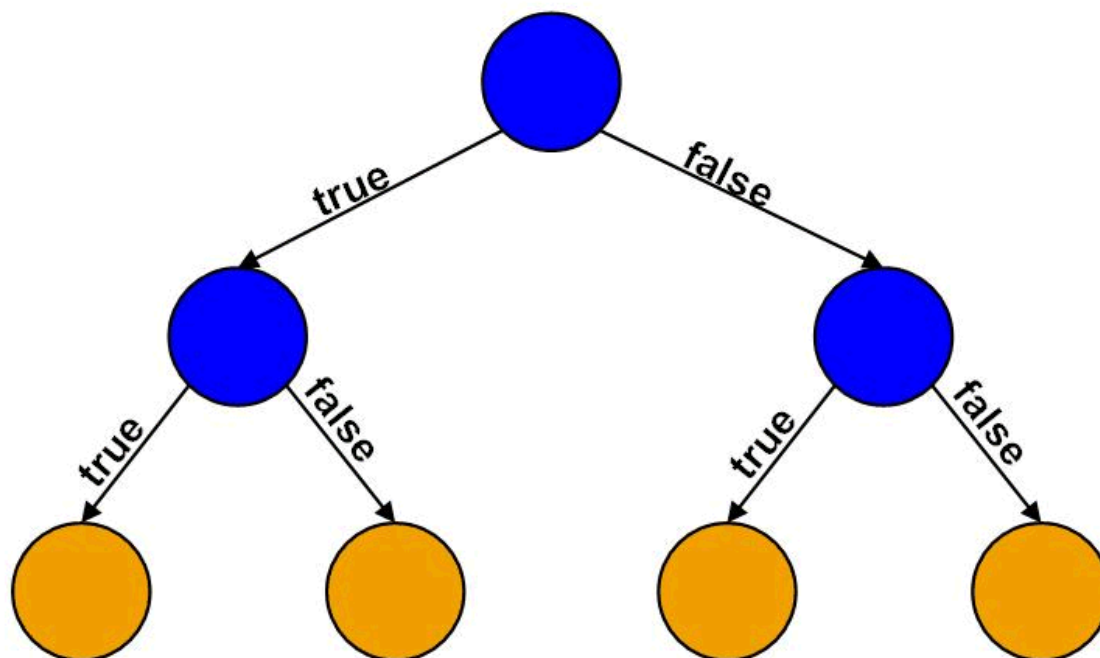


Abbildung 4: Schema des Entscheidungsbaums

Funktionen

Ergebnistypen

Sie können entscheiden, ob ein Entscheidungsbaum (Ausdruck) einen Wert zurückliefern oder eine Aktion durchführen soll:

- **Rückgabe eines Werts:** In diesem Modus müssen Sie sicherstellen, dass allen Blattknoten ein Wert oder Ausdruck zugeordnet ist, dessen Ergebnis dem Typ des Ergebnisdatenobjekts des Entscheidungsbaums (Ausdruck) entspricht.

i Hinweis

Wenn Sie einen Entscheidungsbaum in der BRFPplus-Workbench definieren, ordnen Sie dem Baumausdruck zunächst ein Ergebnisdatenobjekt zu und definieren anschließend die Ergebnisse für die verschiedenen Blattknoten. Nur unter dieser Voraussetzung gewährt Ihnen das System Zugriff auf passende Ausdrücke oder ermöglicht Ihnen die direkte Eingabe von Ergebniswerten. Ohne definiertes Ergebnisdatenobjekt können Sie einem Knoten nur Element-Datenobjekte als Ergebnis zuordnen.

- **Aktion durchführen:** In diesem Modus löst der Entscheidungsbaum die Aktion aus, die dem Blattknoten zugeordnet ist, welcher mit der Eingabe übereinstimmt. Als Ergebnisdatenobjekt wird automatisch die vordefinierte Tabelle `ACTIONS` ausgewählt, die eine Liste von IDs der auszuführenden Aktionen enthält, d. h. Ergebnisaktionen sowie alle Nachfolgeaktionen, die möglicherweise mit dieser Aktion verknüpft sind.

Nachdem Sie die Einstellungen zum Ergebnistyp vorgenommen und den Entscheidungsbaum gesichert haben, zeigt das System diese Einstellungen standardmäßig im Abschnitt *Details* auf der Benutzeroberfläche an.

i Hinweis

Das Ändern der Einstellungen zum Ergebnistyp führt in den meisten Fällen dazu, dass Sie alle Objekte erneut den Blattknoten des Entscheidungsbaums zuordnen müssen. Dies sollten Sie bedenken, bevor Sie sich dazu entschließen, den Ergebnistyp eines bereits definierten Entscheidungsbaums zu ändern.

Weitere Informationen

[Entscheidungsbaum \(Ausdruck\) anlegen \[Seite 210\]](#)

[Suchbaum \(Ausdruck\) \[Seite 115\]](#)

[Aktionstypen \[Seite 132\]](#)

1.2.11.8 Dynamischer Ausdruck (Ausdruck)

Verwendung

Der dynamische Ausdruck wird verwendet, um einen anderen Ausdruck dynamisch aufzurufen. Der aufgerufene Ausdruck wird nicht statisch zur Laufzeit definiert. Er wird vielmehr dynamisch zur Laufzeit bestimmt, basierend auf dem Auswertungsergebnis eines eingebetteten Ausdrucks, der dem dynamischen Ausdruck zur Design-Zeit zugeordnet wurde. Dies ist hilfreich, wenn Sie eine bestimmte Menge an spezialisierten Ausdrücken haben, die für die Behandlung bestimmter Anwendungsfälle optimiert wurden, welche sich im Allgemeinen ähneln, jedoch in bestimmten Details unterschiedlich sind. Mithilfe des dynamischen Ausdrucks kann der passendste Ausdruck zur Laufzeit bestimmt werden.

Funktionsumfang

Kontextzuordnung

Der dynamische Ausdruck bietet folgende Modi für die Kontextzuordnung an:

- *Vollständiger Kontext aber keine Ausdrucksvariable*
Alle verfügbaren Kontextdatenobjekte des dynamischen Ausdrucks werden an den aufgerufenen Ausdruck übermittelt.
- *Passender Kontext*
Nur diejenigen Kontextdatenobjekte werden an den aufgerufenen Ausdruck übermittelt, die mit den Kontextdatenobjekten des aufgerufenen Ausdrucks übereinstimmen.

Aufgerufener Ausdruck

Wählen Sie einen Ausdruck und ordnen Sie ihn dem dynamischen Ausdruck zu. Der zugeordnete Ausdruck wird zur Laufzeit vom dynamischen Ausdruck aufgerufen. Der aufgerufene Ausdruck muss dergestalt konfiguriert sein, dass er die vom dynamischen Ausdruck bereitgestellten Kontextdaten behandelt und ein

Text-Datenobjekt als Ergebnis zurückliefert. Das Ergebnis muss je nach Kontextdaten die ID eines bestimmten zu verarbeitenden Ausdrucks enthalten. Das Ergebnis des sekundären Ausdrucks wird anschließend an den dynamischen Ausdruck zurückgeliefert.

Ergebnisdatenobjekt

Sie können ein beliebiges Datenobjekt als Ergebnisdatenobjekt verwenden. Stellen Sie jedoch sicher, dass das Ergebnisdatenobjekt des dynamischen Ausdrucks und die Ergebnisdatenobjekte aller sekundären Ausdrücke miteinander kompatibel sind.

Einschränkungen

Die Flexibilität, die Sie mit dynamischen Ausdrücken erhalten, bringt auch einen logischen Nachteil mit sich. Aufgrund des dynamischen Wesens dieser Ausdrucksart werden eine Reihe von Prüfungen, die BRFplus für die meisten anderen Ausdrucksarten zur Design-Zeit durchführen kann, nicht für dynamische Ausdrücke unterstützt. Aus diesem Grund wird jeder eventuelle Design-Fehler möglicherweise erst zur Laufzeit sichtbar.

Anders als bei den meisten anderen Objektarten in BRFplus wird für dynamische Ausdrücke keine Codegenerierung unterstützt. Aus diesem Grund wird diese Ausdrucksart **nicht** für zeitkritische Aufgaben empfohlen sowie für Szenarien, bei denen eine hohe Datenlast entstehen kann. Aus der Tatsache, dass dynamische Ausdrücke nicht generiert werden können, ergibt sich auch die Folge, dass sie nicht für Service-Verteilungsszenarien unterstützt werden, die Sie mit SAP Decision Service Management einrichten möchten.

Beispiel

In einem Regelsystem für eine Sozialbehörde möchten Sie Bewerbungsformulare für verschiedene Leistungsarten so bearbeiten, dass sie über einen Ausdruck in das System eingegeben werden können, der einen zentralen Einstiegspunkt für den Bearbeiter darstellt. Anschließend verwenden Sie einen dynamischen Ausdruck, um die Art des zu bearbeitenden Bewerbungsformulars zu ermitteln. Im dynamischen Ausdruck verwenden Sie eine Entscheidungstabelle als eingebetteten Ausdruck. Die Entscheidungstabelle ermittelt den Ausdruck, der mit der entsprechenden Bewerbungsformularart verknüpft ist, indem sie die Informationen zur Art des Bewerbungsformulars als Kontextdaten an die Entscheidungstabelle weitergibt, und gibt anschließend das Formular zur weiteren Verarbeitung an diesen Ausdruck weiter.

Weitere Informationen

[Einen dynamischen Ausdruck anlegen \[Seite 213\]](#)

1.2.11.9 Formel (Ausdruck)

Verwendung

Mit einem Formelausdruck können Sie eine breite Palette an Berechnungen durchführen. Der Formelausdruck bietet mathematische Grundfunktionen (Addition, Subtraktion, Multiplikation, Division), erweiterte Funktionen (wie Potenzierung oder binäres UND), Vergleiche (größer, kleiner etc.) sowie eine breite Palette von Formelfunktionen für verschiedene Zwecke und Datentypen (Datums- und Zeitberechnungen, Stringverarbeitung und Stringmanipulation, Konvertierungsfunktionen, Systemfunktionen etc.). Sie können einer Formel ebenso Kommentare hinzufügen, um den Zweck eines bestimmten Berechnungsschritts zu verdeutlichen.

Funktionsumfang

Bearbeitungsmodus

Das System bietet zwei verschiedene Bearbeitungsmodi für das Pflegen eines Formelausdrucks: *Normalmodus* und *Expertenmodus*:

- *Normalmodus*
Im *Normalmodus* kann der Benutzer den Formeltext nicht direkt bearbeiten, was die Formelintegrität bewahrt. Der Benutzer kann stattdessen die verschiedenen vordefinierten Optionen nutzen, die in der BRFplus-Workbench angeboten werden, indem er auf die entsprechende Funktionstaste klickt oder eine Option aus einer Liste auswählt. Dies gilt auch für Konstantenwerte, die Sie als Operand einer Formel eingeben möchten. Hierzu müssen Sie entsprechenden Drucktasten für die Eingabe von einer *Zahl* oder eines *String*-Wertes verwenden. Ausdrücke und Datenobjekte, die als Operand einer Formel verwendet werden, werden mit den technischen Namen angezeigt.
- *Expertenmodus*
Im *Expertenmodus* kann der Benutzer den Formeltext manuell bearbeiten. Dies kann die Formelpflege deutlich beschleunigen, jedoch ist der Benutzer vollständig für die syntaktische Korrektheit des Formeltexts verantwortlich. Ausdrücke und Datenobjekte, die als Operand einer Formel verwendet werden, werden mit der ID angezeigt.

Operanden

In einem Formelausdruck können Sie elementare Datenobjekte und Ausdrücke als Operanden verwenden. Darüber hinaus können Struktur- und Tabellendatenobjekte ebenso in einer Formel verwendet werden, wenn die Formel eine Funktion enthält, die ein Datenobjekt dieser Komplexität als Parameter vorsieht (beispielsweise sehen alle Tabellenfunktionen ein Tabellendatenobjekt als Parameter vor).

Kontextdaten

Das System zeigt in der BRFplus-Workbench eine Liste von Datenobjekten an, die als Operand einer Formel verwendet werden können. Die Datenobjekte in der Liste sind vom Kontext der Funktion vererbt, zu der der Formelausdruck zugeordnet ist. Das heißt, ein eigenständiger Formelausdruck, der noch keiner Funktion zugeordnet wurde, kann keine Kontextdatenobjekte anbieten.

Hinweis

Wenn ein Formelausdruck mehr als einer Funktion zugeordnet ist, entspricht die Liste der als Operanden einer Formel verfügbaren Kontextdatenobjekte der Schnittmenge der Kontextdatenobjekte aller beteiligten

Funktionen. Dies ist notwendig, um sicherzustellen, dass die Formel unabhängig von der aufrufenden Funktion richtig berechnet werden kann.

Formelfunktionen

In einem Formelausdruck können Sie aus einer Vielzahl an Formelfunktionen wählen, die Ihnen helfen, Daten abzurufen, umzuwandeln und zu analysieren. Zur besseren Übersicht wurden die Funktionen in die folgenden Kategorien gruppiert:

Tabelle 29:

Funktionskategorie:	Beispiele
Datum und Uhrzeit	Berechnung der Differenz zwischen zwei Zeitpunkten, in verschiedenen Einheiten; Bestimmung von Schaltjahren; Extraktion von bestimmten Abschnitten zu einem bestimmten Zeitpunkt, etc.
Stringfunktionen	Verkettung und Extraktion von Strings; String-Länge; Stringähnlichkeit; etc.
Mathematische Funktionen	Logarithmen; Potenzierung; trigonometrische Funktionen; etc. i Hinweis Parameter, die an die trigonometrischen Funktionen weitergegeben werden, müssen in Radianten angegeben werden (ein vollständiger Kreis = 2 Pi).
Systemfunktionen	Abrufen von systemspezifischen Informationen, wie Fabrikkalender, Betriebssystem, Anmeldesprache, etc.
Tabellenfunktionen	Abrufen von Informationen über oder abgelegt in einem BRFplus-Tabellendatenobjekt, wie größter und kleinster Wert, Summe, Durchschnitt, Zeilenanzahl, etc.
Verschiedene Funktionen	Umrechnung von Mengen und Beträgen, Rundung von Zahlen, Überprüfung von Initialwerten, etc.

Zusätzlich zur Funktion Gruppieren, bietet das System eine weitere Möglichkeit, eine Funktion zu finden: In das Feld [Nach Text filtern](#) können Sie den Namen (oder einen Teil des Namens) einer Funktion eingeben und das System zeigt alle Funktionsnamen an, die mit dem Suchbegriff übereinstimmen. Dies ist hilfreich, wenn der Funktionsname bekannt ist. Kombiniert mit der Funktionskategorie [Alle Funktionen anzeigen](#) ist dies die schnellste Methode, um Zugriff auf die gewünschte Funktion zu erhalten.

Weitere Details zu jeder Funktionen finden Sie in der Online-Dokumentation. Dort finden Sie alle Informationen zu Parametern, Typen, Aufrufkonventionen sowie Beispiele und mehr für jede Funktion. Um die Funktionsdokumentation anzuzeigen, navigieren Sie zur betreffenden Funktion in der Liste und klicken Sie auf den Link [Anzeigen](#) in der Spalte [Dokumentation](#).

Kundeneigene Funktionen

Sie können den funktionalen Umfang der Formelausdrucksart erweitern, indem Sie zusätzliche Funktionen gemäß den Anforderungen Ihres Anwendungsfalles definieren. Hierzu definieren Sie Methoden in der Exit-Klasse, die der Anwendung zugeordnet sind und registrieren diese Methoden als Formelfunktionen. Eine detaillierte Dokumentation von kundeneigenen Formelfunktionen finden Sie im Dokument *How to Create Formula Functions* im SDN (<http://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/library/uuid/10e9c96f-0c8b-2b10-6885-f00adbe314b>).

Rundungsverhalten für Beträge

BRFplus verwendet immer höchste Präzision für alle Berechnungen, die numerische Werte enthalten, entsprechend der Anzahl der Dezimalstellen, die für die betreffenden Datenobjekte definiert wurden. Es gibt jedoch einen Sonderfall, in dem die Berechnung **nicht** mit höchster Präzision durchgeführt werden kann, nämlich wenn das Ergebnisdatenobjekt eines Formelausdrucks ein Datenobjekt des Typs *Betrag* ist. In diesem Fall verwendet BRFplus die Anzahl der Dezimalstellen, die für die Währung des Ergebnisdatenobjekts definiert wurde, und ignoriert die Anzahl der Dezimalstellen, die in BRFplus definiert wurde.

Beispiel

In einem Formelausdruck wird die folgende Berechnung verarbeitet:

25 € / 8

Obwohl das intern berechnete Ergebnis von 3,125 € präzise ist, ergibt das Ergebnisdatenobjekt des Formelausdrucks einen gerundeten Wert von 3,13 €, da für die Währung Euro nur zwei Dezimalstellen in den Customizing-Tabellen definiert sind.

Überprüfung der Mengendimension

Zusätzlich zu den Standard-Konsistenzprüfungen, die für alle Arten von BRFplus-Objekten verfügbar sind, führt das System bestimmte Prüfungen für Formelausdrücke durch, die Berechnungen mit Datenobjekten des Typs Menge enthalten, denen eine Mengendimension zugeordnet wurde. Die Herausforderung ist hierbei, dass jede Mengeneinheit zu einer bestimmten Mengendimension gehört (Zeit, Strecke, Geschwindigkeit usw.), es jedoch nur eine begrenzte Anzahl von Kombinationsmöglichkeiten von Operanden und Ergebnisdatenobjekten mit verschiedenen Dimensionen gibt, die sinnvoll sind. Hier einige Beispiele:

Tabelle 30:

Formel	Ergebnisdatenobjekt	Gültig
Strecke / Zeit	Geschwindigkeit	ja
Strecke / Masse	Anzahl	nein

Hinweis

Der Operand **Zeit** im ersten Beispiel in der Tabelle oben darf nicht mit dem BRFplus-Datenobjekttyp *Zeitpunkt* verwechselt werden. Der Operand **Zeit** ist ein Datenobjekt des Typs *Menge* mit der Dimension *Zeit*. Dieser wird zum Ausdruck von Zeiträumen verwendet, während der Objekttyp *Zeitpunkt* mit Bezug auf einen bestimmten Zeitpunkt verwendet wird.

Das System führt Überprüfungen der Mengendimension nur für Formelausdrücke mit Datenobjekten des Typs Menge durch, für die die Eigenschaft *Dimension* gepflegt wurde. Wenn keine Informationen zur

Dimension vorhanden sind, entfällt dieser Teil der Überprüfung. Im Falle eines nicht erkannten Unterschieds in der Dimension, tritt zur Laufzeit ein Fehler auf.

Beispiele

In der folgenden Formel wird die Verwendung einer integrierten Systemfunktion in einer einfachen IF-Bedingung verdeutlicht. Die Formel liefert einen numerischen Wert von 10 zurück, wenn die aktuelle Anmeldesprache Englisch ("EN") ist. Andernfalls wird als Ergebnis der Wert 0 zurückgeliefert:

```
NUMBER = IF ( SYS_INFO_LANGUAGE ( ) = 'EN' , 10 , 0 )
```

Hinweis

Bei komplexen *If-Then-Else*-Szenarien ist es empfehlenswert, den Booleschen Ausdruck zu verwenden.

Mit der folgenden Formel soll dargestellt werden, wie trigonometrische Funktionen verwendet werden können, um den Luftweg (Entfernung) zwischen zwei Orten auf der Erde zu berechnen. Um das Beispiel einfach zu halten, wird angenommen, dass die Erde eine ideale Kugel ist.

Kontext

Der Kontext der Funktion, zu der die Formel gehört, besteht aus einer Vielzahl von Datenobjekten, die teilweise vom Benutzer gefüllt werden (Startort, Zielort), teilweise aus einer Entscheidungstabelle stammen (Längengrad, Breitengrad) und teilweise von der Formel berechnet werden (Entfernung).

Tabelle 31:

Name	Beschreibung
Koordinaten	Struktur (Längengrad, Breitengrad). Werte werden mithilfe einer Entscheidungstabelle bestimmt, abhängig vom gewählten Start- und Zielort.
Startort, Zielort	Textdatenobjekte, die interaktiv vom Benutzer gefüllt werden.
Entfernung	Zahl-Datenobjekt, das als Ergebnisdatenobjekt der Formel dient.
Längengrad des Starts, Längengrad des Endes, Breitengrad des Starts, Breitengrad des Endes	Hilfsvariablen, die bei der Berechnung verwendet werden.
Pi	Konstantenwert

Formel

```
6378.137 * ARCCOS ( SIN ( Latitude Start * Pi / 180 ) * SIN ( Latitude End *  
Pi / 180 ) + COS ( Latitude Start * Pi / 180 ) * COS ( Latitude End * Pi / 180 )  
* COS ( ( Longitude End - Longitude Start ) * Pi / 180 ) )
```

Weitere Informationen

[Kontext und Ergebnis \[Seite 46\]](#)

[Anwendung \[Seite 30\]](#)

[Formel Ausdruck anlegen \[Seite 215\]](#)

1.2.11.10 Funktionsaufrufsausdruck

Verwendung

Ein Funktionsaufrufsausdruck ruft eine BRFplus-Funktion auf, ordnet den Ausdruckskontext dem Funktionskontext zu und liefert ein Ergebnis zurück. Ein Funktionsaufrufsausdruck kann je nach Eingabe verschiedene Funktionen aufrufen.

Ein Funktionsaufrufsausdruck kann beispielsweise zur Berechnung der Steuersätze in verschiedenen Ländern verwendet werden. Der Funktionsaufrufsausdruck ruft die entsprechenden Funktionen je nach Land auf und gibt die Kontextdaten weiter.

Weitere Informationen

[Funktionsaufrufsausdruck anlegen \[Seite 218\]](#)

[Prozeduraufruf \(Ausdruck\) \[Seite 118\]](#)

1.2.11.11 Schleife (Ausdruck)

Verwendung

Für einen Schleifenausdruck geben Sie an, wie oft ein Regelsatz in einem bestimmten Kontext verarbeitet wird. Wie häufig die Regeln ausgeführt werden, wird mit einer numerischen Konstante, einer Bedingung oder einer Iteration über die Zeilen einer Tabelle definiert. Der Ausdruck kann zudem Exit- und Continue-Bedingungen für die Regeln enthalten. Der Schleifenausdruck gibt entweder ein Ergebnis zurück, das für weitere Verarbeitungen verwendet wird, oder Aktionen auslösen.

Funktionsumfang

Bearbeitungsmodus

Ein Schleifenausdruck kann in einem der folgenden Verarbeitungsmodi ausgeführt werden:

- **Aktion durchführen**
In diesem Verarbeitungsmodus weist das System als Ergebnisdatenobjekt das vordefinierte Tabellenobjekt **Aktionen** zu. Nachdem die Schleifenverarbeitung abgeschlossen ist, gibt diese Tabelle eine Liste aller Aktionen aus, die über den Schleifenausdruck ausgelöst wurden.
- **Rückgabewert**
In diesem Verarbeitungsmodus können Sie ein beliebiges Datenobjekt als Ergebnisdatenobjekt zum Ausdruck zuordnen. Wenn Sie diese Einstellung wählen, müssen Sie sicherstellen, dass die zur Laufzeit verarbeiteten Regeln Änderungen am Wert des ausgewählten Datenobjekts vornehmen.

Schleifenmodus

Sie können aus den folgenden Schleifenmodi wählen, um präzise anzugeben, wie der Schleifenausdruck ausgeführt werden soll:

- **Wiederholen... Male**
Für diesen Modus geben Sie mit einer Zahl an, wie viele Iterationen die Schleife durchlaufen soll. Verwenden Sie diesen Schleifenmodus für Anwendungsfälle, bei denen Sie im Voraus die genaue Anzahl der Iterationen kennen. Soll ein so definierter Schleifenausdruck weniger Iterationen durchlaufen als definiert, muss eine optionale Exit-Bedingung erfüllt werden.

Beispiel

Sie sollen eine einjährige Prognose von Umsatzzahlen für ein bestimmtes Produkt berechnen. Diese Prognose wird in Monate unterteilt. Dazu können Sie einen Schleifenausdruck vom Typ **Wiederholen...** definieren. Für **Male** geben Sie eine feste Iterationszahl von 12 ein. So werden die Zahlen für jeden Monat des Jahres ab dem Zeitpunkt der Definition errechnet.

- **Repeat-Schleife**
Für diesen Modus müssen Sie ein Abbruchkriterium definieren, das erfüllt sein muss, damit die Verarbeitung des Schleifenausdrucks gestoppt wird. Das Abbruchkriterium wird nach jeder Iteration getestet. Das bedeutet, in einer **Repeat-Schleife** werden die für den Ausdruck definierten Regeln mindestens einmal verarbeitet.
- **While-Schleife**
Für diesen Modus müssen Sie ein Continue-Kriterium definieren, das erfüllt sein muss, damit die Verarbeitung des Schleifenausdrucks weiterhin erfolgt. Das Continue-Kriterium wird nach jeder Iteration getestet. Das bedeutet, in einer **While-Schleife** werden die für den Ausdruck definierten Regeln unter Umständen überhaupt nicht verarbeitet.
- **Für jeden Eintrag in...**
In diesem Modus ist die Anzahl der Iterationen abhängig von der Anzahl der Einträge in einer Tabelle, die als Steuerung für die Schleifeniteration verwendet werden soll. Das für die Steuerung ausgewählte Objekt muss entweder ein Tabellendatenobjekt sein oder ein Ausdruck, dem ein Tabellendatenobjekt als Ergebnisdatenobjekt zugewiesen wurde. Dies ist z. B. der Fall für mehrere Untertypen der Tabellenoperationen oder für eine Entscheidungstabelle oder einen Suchbaum (Ausdruck) im Mehrfachtreffermodus. Die Anzahl der Iterationen kann durch die Auswahl zusätzlicher Kriterien, die Sie für die Tabelle definieren können, weiter eingeschränkt werden.
Das Objekt für die Iterationssteuerung kann frei aus allen geeigneten Objekten im System gewählt werden. Es ist dabei **nicht** erforderlich, das Objekt in den Ausdruckskontext einzubeziehen.

Achtung

Für die *Repeat*- und die *While-Schleifenmodi* müssen Sie sicherstellen, dass das Abbruchs- oder Continue-Kriterium bei jeder Iteration aktualisiert und ausgewertet wird. Anderenfalls könnte das System in eine Endlosschleife geraten, die manuell abgebrochen werden muss.

Hinweis

Wenn Sie die Bedingungen für die Schleifensteuerung definiert haben und danach den Schleifenmodus ändern, werden die Referenz auf die beteiligten Objekte und die möglicherweise zusätzlich definierten Selektionskriterien sofort aus dem Schleifenausdruck gelöscht. Nachdem Sie den Ausdruck mit dem neuen Schleifenmodus gesichert haben, können Sie die früheren Einstellungen nur durch den Aufruf einer älteren Version wiederherstellen. Dies setzt natürlich voraus, dass die Versionierung für den Schleifenausdruck aktiviert ist.

Schleifenbedingungen

Für alle Schleifenmodi können Sie für jede Regel, die während einer Iteration verarbeitet wird, zusätzliche Exit- oder Continue-Bedingungen definieren. So überschreiben Sie das durch den Schleifenmodus festgelegte Schleifenverhalten. Mit Bedingungen basiert die Schleifenverarbeitung nicht nur auf der Anzahl der Iterationen, sondern auch auf allen zusätzlichen Kriterien. Voraussetzung dabei ist, dass das Testergebnis als Boolescher Wert ausgegeben wird. So können Sie in einer Bedingung z. B. testen, ob der aktuelle Benutzer berechtigt ist, eine bestimmte Regel auszuführen, wenn das aktuelle Datum und der Zeitpunkt innerhalb eines vordefinierten Gültigkeitszeitraums liegen, oder wenn die Anzahl der an den Schleifenkontext weitergegebenen Artikel die Berechnung zusätzlicher Rabatte ermöglicht, usw.

- *Exit-Bedingung*
Mit einer Exit-Bedingung können Sie zusätzliche Tests ausführen. Wenn das Testergebnis **true** lautet, wird die Verarbeitung des gesamten Schleifenausdrucks abgebrochen.
- *Continue-Bedingung*
Mit einer Continue-Bedingung können Sie zusätzliche Tests ausführen. Wenn das Testergebnis **true** lautet, wird die Verarbeitung der aktuellen Schleifeniteration abgebrochen. Die Verarbeitung des Ausdrucks wird fortgesetzt. Dabei wird die Anzahl der Iterationen um 1 erhöht und die erste Regel in der Regelreihenfolge ausgeführt.

Weitere Informationen

[Schleife \(Ausdruck\) anlegen \[Seite 221\]](#)

1.2.11.12 Zufallszahl (Ausdruck)

Verwendung

Mit der Zufallszahl (Ausdruck) integrieren Sie bewusst ein schwer einschätzbares Element in Ihre Anwendung. Dies kann z. B. bei Tests hilfreich sein, wenn Sie prüfen möchten, wie sich eine Anwendung in vielen

unterschiedlichen Versionen mit sich ständig ändernden Eingabeparametern verhält. Mit diesem Ausdruck können Sie auch einen Wert berechnen, den Sie automatisch Massenobjekten in Ihrer Anwendung zuweisen möchten – vorausgesetzt, dazu ist keine systematische Nummerierung erforderlich. Beispiele sind Kundennummer, Bestellnummer usw.

Die Zufallszahl (Ausdruck) basiert auf dem standardmäßigen ABAP-Pseudo-Zufallszahlengenerator. Immer, wenn der Ausdruck ausgewertet wird, startet dieser Zahlengenerator im Backend mit einem neuen Initialwert. So wird sichergestellt, dass die generierte Zahlenfolge unterschiedlich ist.

i Hinweis

Beachten Sie, dass der Algorithmus für das Generieren von Zufallszahlen ausschließlich für normale Geschäftsanwendungsfälle konzipiert wurde. Dies liegt daran, dass die generierten Zahlenfolgen identisch sind, so lange derselbe Initialwert verwendet wird. Deshalb ist die Zufallszahl (Ausdruck) nicht für sicherheitsrelevante Zwecke wie Kennwortverschlüsselung, Zugriffscodes oder andere Anwendungsfälle mit Verschlüsselungen geeignet und auch nicht dafür gedacht.

Funktionsumfang

Operationsmodi

Die Zufallszahl (Ausdruck) kann in zwei Operationsmodi ausgeführt werden: Der Ausdruck kann entweder eine Zufallszahl zwischen zwei Grenzwerten (Minimal- und Maximalwert) oder einen Booleschen Wert mit einer benutzerdefinierten Wahrscheinlichkeit zwischen 0 und 1 zurückgeben.

- **Zahl**
In diesem Modus gibt der Ausdruck eine Zufallszahl zwischen zwei Grenzwerten (Minimal- und Maximalwert) aus. Sie können den Bereich der möglichen Ergebnisse nicht nur durch diese Grenzwerte, sondern auch durch Festlegen der Dezimalstellen beeinflussen. In diesem Fall ist die maximale Anzahl von Dezimalstellen abhängig von der Definition des Ergebnisdatenobjekts, welches dem Ausdruck zugewiesen ist.
- **Wahrscheinlichkeit**
In diesem Modus gibt der Ausdruck einen Booleschen Wert zurück. Sie können das Verhalten des Ausdrucks beeinflussen, indem Sie die statistische Wahrscheinlichkeit für den Ausdruck zum Zurückgeben des Werts auf **TRUE** setzen. Der Wahrscheinlichkeitswert wird als Dezimalbruch zwischen 0 und 1 mit maximal drei Dezimalstellen angegeben. Bei einer Wahrscheinlichkeit von 1 ist die Ausgabe des Ausdrucks immer wahr, also **TRUE**.

i Hinweis

Wenn Sie eine Simulation für eine Zufallszahl (Ausdruck) durchführen, können Sie die Hintergrundprozesse analysieren. Hier sehen Sie, dass der Ausdruck auch im Wahrscheinlichkeitsmodus intern die Zufallszahl berechnet. Diese Zahl wird aber nicht als Ergebnis des Ausdrucks, sondern als Operand einer internen Vergleichsoperation verwendet. Der Ausdruck prüft, ob die berechnete Zahl kleiner oder gleich dem Wert ist, der als Kontext an den Ausdruck gegeben wurde. Ist dies der Fall, gibt der Ausdruck **TRUE** zurück, wenn nicht **FALSE**.

Bewertung

Wenn Sie den Operationsmodus [Zahl](#) verwenden, können Sie wie folgt Werte für den Beispielbereich definieren:

- [Direkte Eingabe](#)
Im Modus für die direkte Eingabe definieren Sie die Werte für den unteren und oberen Grenzwert des Beispielbereichs direkt. Folglich sind in diesem Modus keine Kontextdatenobjekte erforderlich. Das System kann diese statische Bewertung ganz einfach prüfen, sodass potenzielle Fehler bereits während der Design-Zeit identifiziert werden können.
- [Referenz](#)
Im Referenzmodus definieren Sie die Werte für den unteren und oberen Grenzwert des Beispielbereichs indirekt, indem Sie den Grenzwerten jeweils ein Kontextdatenobjekt oder einen Ausdruck zuweisen anstatt einen festen statischen Wert zu definieren. Durch diese dynamische Bewertung sind Sie während der Laufzeit flexibler. Andererseits steigt hier das Risiko für formale Fehler, z. B. ein Minimalwert, der höher ist als der Maximalwert, die zu einer Ausnahme führen können.

Automatische Anpassungen

BRFplus unterstützt Sie mit mehreren Prüfungen beim Definieren einer Zufallszahl (Ausdruck). Diese werden entweder im Hintergrund ausgeführt, oder Sie lösen sie über den [Check](#)-Button aus:

- Wenn Sie den Ausdruck im Zahlenmodus anlegen, weist BRFplus automatisch das integrierte Datenobjekt **NUMBER** als Ergebnisdatenobjekt zu. In den meisten Fällen ist dies gewünscht, da das Ergebnis einer Zufallszahl (Ausdruck) immer ein numerischer Wert ist. Sie können aber auch ein benutzerdefiniertes Datenobjekt vom Typ „Zahl“ auswählen.
Wenn Sie den Ausdruck im Wahrscheinlichkeitsmodus anlegen, gilt dies ebenfalls, allerdings mit dem integrierten Objekt **BOOLEAN**.
- Im Modus für die direkte Eingabe tauscht das System automatisch den Minimal- mit dem Maximalwert, wenn dieser größer als der Maximalwert ist.

Weitere Informationen

[Zufallszahl \(Ausdruck\) anlegen \[Seite 223\]](#)

1.2.11.13 Suchbaum (Ausdruck)

Verwendung

Der Suchbaum (Ausdruck) ist eine nicht-binäre Baumstruktur mit Bedingungen und Ergebnissen. Mithilfe des Baums werden unterschiedlichen Baumzweigen verschiedene geschäftsrelevante Bedingungen zugewiesen. Das System prüft die Eingabe und vergleicht sie mit den für jeden Knoten definierten Bedingungen. Suchbaumknoten verfügen immer über eine zugewiesene Bedingung und können auch ein zugeordnetes Ergebnis haben. Jedem Knoten – mit Ausnahme des Wurzelknotens – muss eine Bedingung zugewiesen sein. Ein Knoten kann mehrere untergeordnete oder Unterknoten haben. Alle Unterknoten einer Ebene, also mit demselben übergeordneten Knoten, werden als Geschwisterknoten bezeichnet. Ein Knoten ohne zugewiesene Unterknoten wird Blattknoten genannt.

Der Suchbaum führt Suchen in einem Top-down-Ansatz entsprechend den Regeln des ausgewählten Übereinstimmungsmodus durch. Wenn die zugewiesene Bedingung eines Suchbaumknotens als „true“

ausgewertet wird und der Ausdruck im Modus *Erster Treffer* oder *Mehrfachtreffer* vorliegt, gibt der Ausdruck das dem aktuellen Knoten zugewiesene Ergebnis aus.

Im Modus *Mehrfachtreffer* verzweigt die Regelauswertung auf die Unterknoten und führt dort die Suche auf ähnliche Weise fort. Wenn der Bedingungsknoten „false“ zurückgibt, wird die Suche im nächsten Geschwisterknoten (von links nach rechts) fortgesetzt. Diese Knoten werden dann samt den dazugehörigen Unterknoten durchsucht.

Hinweis

Aufgrund der sequenziellen Bearbeitungsreihenfolge des Suchbaums (Ausdruck) ist eine genaue Analyse der Anwendungsfälle, in denen Sie eine Modellierung mit dem Suchbaum durchführen möchten, erforderlich. Zudem müssen die Bedingungen entsprechend auf die Baumknoten verteilt werden. Anderenfalls kann es vorkommen, dass eine bestimmte Kombination aus Bedingungen aufgrund ihrer Position in der Baumstruktur nie getestet wird. Dies ist auch der Fall, wenn diese Kombination eine Übereinstimmung von höherer Qualität liefern würde.

Funktionen

Modus der Operation

Sie entscheiden, ob der Suchbaum ein Ergebnisdatenobjekt einen Eingabewert, der mit einer Knotenbedingung übereinstimmt ausgibt, oder ob eine Aktion ausgelöst werden soll.

Übereinstimmungsmodus

Der Suchbaum (Ausdruck) unterstützt verschiedene Operationsmodi, um die Verarbeitung eines bestimmten Eingabewerts durch den Ausdruck zu steuern. Zudem ermittelt er, ob ein bestimmter Wert mit einem der Baumknoten übereinstimmt. Folgende Übereinstimmungsmodi sind verfügbar:

- *Erster Treffer*: Im Modus *Erster Treffer* wird die Suche so lange fortgeführt, bis der erste Knoten, dessen Bedingung mit dem Eingabewert übereinstimmt, gefunden wurde. Dann wird das entsprechende Ergebnis ausgegeben, falls diesem Knoten eins zugewiesen ist. Eine weitere Verarbeitung im Baum findet nicht statt.
- *Mehrfachtreffer*: Im Modus *Mehrfachtreffer* wird die Suche für den gesamten Baum (alle Ebenen, alle Zweige) durchgeführt. Alle Ergebniswerte, die Knoten zugewiesen sind, deren Bedingungen als „true“ ausgewertet werden, werden während der Suche gesammelt und im Ergebnisdatenobjekt des Ausdrucks hinterlegt. Wenn die Bedingung eines Knotens als „false“ ausgewertet wird, wird die Verarbeitung des nächsten Geschwisterknotens des übergeordneten Knotens oder des nächsten noch nicht verarbeiteten Zweigs fortgesetzt. Dieser Vorgang wird so lange wiederholt, bis der Baum vollständig verarbeitet wurde, oder bis ein Knoten im letzten Zweig als „false“ ausgewertet wird.

Hinweis

Im Modus *Mehrfachtreffer* muss das Ergebnisdatenobjekt vom Typ „Tabelle“ sein.

- *Qualifizierter Treffer*: Im Modus *Qualifizierter Treffer* wird ein Eingabewert für eine Kette semantisch unabhängiger Bedingungen, die den verschiedenen Baumknoten zugewiesen sind, getestet. Im Gegensatz zum Modus *Erster Treffer* wird die Verarbeitung nicht nach dem Knoten mit dem ersten Treffer gestoppt, sondern wird so lange fortgesetzt, bis eine Knotenbedingung **nicht** erfüllt ist. Eine Eingabe, die die verschiedenen Bedingungen in einem bestimmten Zweig des Suchbaums am besten erfüllt, wird als

qualifizierter Treffer bezeichnet. Das dem letzten Knoten des übereinstimmenden Zweigs zugewiesene Ergebnis wird als Ergebnisdatenobjekt des Ausdrucks zurückgegeben.

Weitere Informationen zum Modus „Qualifizierter Treffer“

Im Gegensatz zu vielen anderen Baumkonstrukten bedeutet das Durchsuchen der Suchbaumzweige von oben nach unten **nicht** unbedingt, dass mit einer allgemeinen Bedingung begonnen wird, die dann im Verlauf der Suche auf jeder Ebene des Baums verfeinert wird. Beispiel: "Region = EMEA" -> "Subregion = Osteuropa" -> "Mitglied der EU = ja" -> "Land = Polen". Stattdessen kann jeder Ebene eine Bedingung zugewiesen sein, die semantisch komplett unabhängig von der Bedingung der vorigen Ebene ist. Beispiel: "Region = EMEA" -> "Branche = Bankwesen" -> "Umsatz >= 50 Milliarden Euro" -> "am NYSE notiert" usw. Dieser Pfad könnte aber ebenso in umgekehrter Reihenfolge definiert sein, also mit der Notierung am NYSE beginnen und mit EMEA enden. Dies ist abhängig vom zu modellierenden Geschäftsfall.

Der Drilldown solcher Listen mit Bedingungen endet an dem Punkt, an dem eine Bedingung nicht mehr erfüllt wird. Sobald die Bewertung diesen Punkt erreicht hat, kann man sagen, dass die aktuelle Eingabe mit allen vorigen Bedingungen bis zu diesem Punkt übereinstimmt. Der Ausdruck gibt dann den Ergebniswert (oder löst die Aktion) als qualifizierten Treffer aus, der bzw. die dem letzten übereinstimmenden Knoten zugewiesen ist.

Zusätzliche Einstellungen

Kontextdatenobjekt

Im Gegensatz zu den meisten anderen Ausdruckstypen verfügt der Suchbaum (Ausdruck) nicht über ein zentrales Kontextdatenobjekt. Jedem einzelnen Knoten des Baums kann ein anderes Datenobjekt oder ein anderer Ausdruck zugewiesen sein, das bzw. der dann als Eingabewert für den Test der Knotenbedingung verwendet wird.

Ergebnisdatenobjekt

Für das Ergebnisdatenobjekt eines Suchbaums (Ausdruck) gelten die folgenden Regeln:

- Sie müssen sicherstellen, dass alle im Baum verwendeten Ergebniswerte mit dem Typ des Ergebnisdatenobjekts kompatibel sind, damit das Knotenergebnis dem Ergebnisdatenobjekt korrekt zugeordnet werden kann.
- Im Modus **Mehrfachtreffer** muss das Ergebnisdatenobjekt ein Tabellendatenobjekt sein, da das Ergebnis in diesem Modus in der Regel aus einer Liste von Werten besteht.
- Wenn der Suchbaum (Ausdruck) so definiert ist, dass er eine Aktion auslöst und keinen Wert zurückgibt, wird das Ergebnisdatenobjekt automatisch auf die vordefinierte Tabelle **Aktionen** gesetzt.

Ergebnisse dürfen nur Blattknoten zugeordnet sein

Ist dieses Kennzeichen gesetzt, gibt der Suchbaum (Ausdruck) nur ein Ergebnis zurück, wenn alle Bedingungen in einem ganzen Baumzweig vom Wurzel- bis zu einem der Blattknoten (ohne weitere Unterknoten) erfüllt sind. Das Aktivieren dieses Kennzeichens hat folgenden Auswirkungen auf die Zuweisung von Ergebnissen zu den Baumknoten:

- Knoten mit Unterknoten darf **kein** Ergebnis zugewiesen werden.
- Blattknoten ohne weitere Unterknoten **muss** ein Ergebnis zugeordnet werden.

Initialwert zurückgeben, wenn keine Übereinstimmung gefunden wird

Ist dieses Kennzeichen gesetzt, gibt der Suchbaum (Ausdruck) den für den Typ des Ergebnisdatenobjekts definierten Initialwert zurück, wenn keine der im Baum angegebenen Bedingungen mit dem Eingabewert übereinstimmen. Anderenfalls stoppt die Regelauswertung, und eine Ausnahme wird ausgegeben.

Hinweis

Wenn Sie diese Option verwenden, unterscheiden Sie sorgfältig zwischen einem wohldefinierten Ergebniswert, der von einem Knoten mit übereinstimmender Bedingung zurückgegeben wird, und dem typenspezifischen Initialwert, der als Ergebnis einer nicht erfolgreichen Regelauswertung zurückgegeben wird. Beispiel: Wenn das Ergebnisdatenobjekt eines Suchbaums (Ausdruck) ein Datenobjekt vom Typ „Nummer“ ist, müssen Sie sicherstellen, dass keinem der Baumknoten mit Ergebnis der Wert 0 zugewiesen ist. Das System kann in einem solchen Fall nicht erkennen, ob es sich um einen gültigen numerischen Wert oder einen Indikator für eine fehlgeschlagene Regelauswertung handelt .

Weitere Informationen

[Suchbaum \(Ausdruck\) anlegen \[Seite 225\]](#)

[Entscheidungsbaum \(Ausdruck\) \[Seite 103\]](#)

1.2.11.14 Procedure Call Expression

Definition

A procedure call expression enables the execution of function modules as well as of methods of an ABAP OO class in the backend system. You can also use this expression directly to execute procedures that are stored in a database.

Note

The procedure call expression type is the renamed and extended successor expression type of the static method call expression that was offered in previous releases of BRFplus. The procedure call expression covers the full scope of the former static method call so that all existing static method call expressions from migrated legacy applications remain fully functional. However, it is not possible to create new static method call expressions.

Use

You use this expression to take advantage of the full functional power offered by the ABAP programming language when processing BRFplus business rules. You can do so in order to access routines for data validation, data conversion, or any other kind of data manipulation regardless of the degree of complexity of such routines.

Values can be passed to the parameters of the called procedure by using a list of parameter mappings that are maintained for the expression. Once the called procedure has determined a result, it is returned to the result data object of the procedure call expression.

If the procedure that you want to execute is stored in a database (stored procedure), you can directly access this kind of procedure without having to define a wrapper method manually in an ABAP OO class. This saves you programming effort and the number of steps in the execution path is reduced, thereby improving performance.

Features

Supported Procedure Types

You can assign the following types of procedures to a procedure call expression:

- Function modules
- Methods of ABAP classes
- Database procedures

Interface Methods

It is not only possible to assign a class method to a procedure call expression, but also to assign methods of a class interface that has been implemented by the selected class. To do so, proceed as follows:

1. Choose a class.
2. Select the *Use Interface Method* checkbox.
The *Interface Name* entry field is changed to edit mode.
3. To select from the interfaces that are implemented by the class, use the value help.
4. To select from the implemented interface methods suitable for the procedure call expression, use the value help in the *Method Name* field.

Automatic Filtering of Suitable Methods

When using the BRFplus workbench for modeling a procedure call expression, the system automatically determines which methods of a given class can be used for the expression at all. Only these suitable methods are then offered for selection using the value help.

Example

You can use only methods declared as public and static. However, this type of method is still not sufficiently qualified. For example, if the method signature contains a mandatory parameter whose type is determined dynamically or defined via indirection (for example, with the `TYPE REF` statement), it cannot be used for a procedure call expression because the exact type of that parameter can only be determined at runtime and is therefore unknown in the modeling environment.

However, if the same parameter is optional rather than mandatory, the method itself can be used for the procedure call expression. If this is the case, the optional parameter remains unusable and is presented in a list of *Unsupported Parameters* that you can check.

You can still enter a method that is **not** offered by the value help. In this case, using the check function for the expression makes the system trigger error messages that inform you about the reasons why the method cannot be used. It is then up to you to decide whether the method can be modified so that the system requirements are met. To see an overview of all methods of the selected class or interface that are currently **not** supported for a procedure call, choose *Unsupported Methods*.

Parameter Mapping

Once you have assigned a method, function module, or database procedure to the expression, the system presents a list of the method parameters. The list contains all of the parameters that are either mandatory or not **and** have a type that can be determined at design time assigned to them. At the same time, this means:

- Non-mandatory parameters whose type can only be determined at runtime are omitted because no mapping can be done for these at design time. These types of parameters are logged in the list of [Supported Parameters](#) below.
- Mandatory parameters, for which the type can only be determined at runtime, lead to the system qualifying the entire method as not suitable for the procedure call expression as described above in the [Automatic Filtering of Suitable Methods](#) section.

From the list of parameters, you choose the ones for which you want to provide a value and associate them with a static value assignment or an expression of the appropriate data type.

Providing Values for Different Parameter Types

Depending on the parameter type (element, structure, or table), you have to take specific steps to map the context parameters to a BRFplus data object.

- Elements
 - You can map an elementary parameter to a BRFplus element data object (or to an expression with result data object of type element). To accomplish this, assign the relevant element data object to the parameter and choose [Move Value](#) as [Move Type](#).
 - As an alternative, you can also decide to define a value at design time to be passed to the parameter as [Direct Input](#).
- Structures
 - To map a structure parameter to a BRFplus structure data object (or to an expression with result data object of type structure) as a whole, assign the desired structure data object to the parameter and choose [Move Corresponding](#) as [Move Type](#).
 - To map individual components of a structure parameter, expand the structure element tree. For each element parameter, proceed as described above for element parameters.

i Note

Mapping individual components of structure parameters is **not** supported for parameters defined as [Exporting or Returning](#).

- Tables
 - To map a table parameter to a BRFplus table data object (or to an expression with result data object of type table) as a whole, assign the desired table data object to the parameter and choose [Move Corresponding](#) as [Move Type](#).
 - You can define a [Source Table for Column Update](#). Once you have assigned a source table to the procedure call expression, you can define for each table structure element whether the source table content is mapped line by line to the table passed as context parameter ([Map Element to Source Table](#) = `True`) or whether the values found in the currently selected source table line is passed to each line of the table passed as context parameter ([Map Element to Source Table](#) = `False`).
 - To map individual components of a table parameter, expand the table structure element tree. For each element parameter, proceed as described above for element parameters.

i Note

Mapping individual components of table structure parameters is **not** supported for parameters defined as *Changing, Exporting, Or Returning*.

Automatic Parameter Mapping

If the method or function module contains any of the following parameters, BRFplus automatically maps them to the corresponding internal BRFplus objects:

Table 32:

Name	Type
IV_TIMESTAMP	IF_FDT_TYPES=>TIMESTAMP
IV_EXPRESSION_ID	IF_FDT_TYPES=>ID
IO_TRACE	REF TO IF_FDT_TRACE
IV_IN_UPDATE_TASK	ABAP_BOOL

i Note

Automatic mapping can be done only if the parameter names and types of the method or function module are exactly the same as given in the table above. In addition, the importing parameter for the trace object (IO_TRACE) must be declared as optional.

For the same reason, there is no automatic parameter mapping for an assigned database procedure. This is because the internal representation of the data types used in the database differs substantially from the data types available in BRFplus. The system can therefore only offer a technical description of the database type in the *Description* field of the parameter. Automatic mapping is not supported for procedure call expressions that have a database procedure assigned to them.

The IV_IN_UPDATE_TASK parameter lets you define whether the actions that are triggered by an associated function module shall be executed directly (*false*) or appended to the internal update task queue (*true*). Note that, depending on the parameter value, the function module owner is responsible for taking the necessary actions in the function module source code. In simulation mode, this parameter is always treated as *true*, regardless of the parameter value. This is to avoid unwanted side effects during simulation. Automatic mapping of this parameter is only supported for procedure call type *Function Module*.

The following table lists the supported parameters:

Table 33: Supported Parameters

Parameter Type	Import	Change	Export	Return	Tables (FM only)
'Simple' Type	Yes	Yes	Yes	Yes	Only tables
Reference	No	No	No	No	No
NUMERIC	Yes	Yes	Yes	Yes	N/A
ANY TABLE	Yes	Yes	Yes	Yes	Yes

Parameter Type	Import	Change	Export	Return	Tables (FM only)
INDEX TABLE	Yes	Yes	Yes	Yes	Yes
CLIKE	Yes	Yes	Yes	Yes	N/A
CSEQUENCE	Yes	Yes	Yes	Yes	N/A
ANY	Yes	Yes	No	No	No
HEX	Yes	Yes	Yes	Yes	N/A
Other Generic Type	No	No	No	No	No
RAW16	Yes	Yes	Yes	Yes	N/A
Raw Fields	No	No	No	No	No

Database Procedures

In a procedure call expression, you can directly call a stored procedure that resides in the database. This helps you saving additional programming effort, as well as improving the performance of database accesses because there are fewer execution steps.

A procedure is always assigned to one of the various database schemas that may exist in the database. Therefore, a fully qualified definition of the procedure to be used consists of the procedure's name plus the containing database schema. In the BRFplus workbench, the value help supports you in retrieving the desired procedure by offering matching procedures across the different schemas. In other words, although the schema is needed for identifying the database procedure, you do not need to delve too deeply into the database to find the object you are searching for.

The database where the procedure to be called resides can either be the database that is associated with the system in which you execute the expression, or it can be a database that runs in a different system within your landscape. In the latter case, you have to set up a secondary database connection. This can be accomplished in the scope of the application to which the procedure call expression belongs.

Note

Procedure call expressions of type *Database Procedure* are currently supported only for the SAP HANA database. In addition, there is another difference between calling a database procedure and the other types of procedure calls. While you can access the interfaces of an assigned function module or class method directly with a procedure call expression, calling a database procedure requires an additional step.

You first need to generate a proxy that acts as an interface between the ABAP layer and the database procedure to be called. Automatic proxy generation, however, requires that the types of the parameters of the database procedure are included in a well-defined set of supported field types that can be transformed into ABAP types. You can find information on the supported type mappings in the ABAP Keyword Documentation on SAP Help Portal:

[External Views for SAP HANA Views](#)

With SAP NetWeaver 740 SP05, the ABAP Managed Database Procedure (AMDP) framework has been introduced. With this framework, you do not need to set up a proxy anymore. Rather, you can now create a special type of an ABAP OO class to access the HANA database. An AMDP can be called in the same way as a normal ABAP OO method and can therefore be assigned directly to a procedure call expression of call type *Static Method*. Again, you can find information on the AMDP framework on SAP Help Portal:

[AMDP - ABAP Managed Database Procedures](#)

We recommend using ABAP Managed Database Procedures whenever you want to execute procedures directly from an ABAP program. If, in special cases, this is not possible (for example, in a stand-alone HANA database, or for procedures generated by another HANA process) you can still use call type [Database Procedure](#).

Exception Handling

It is common programming practice to define exception handlers to ensure safe and controlled system behavior if errors occur at runtime. If a procedure that you have assigned to a BRFplus procedure call expression contains one or more code lines in which an exception is raised, the system presents these exceptions in the BRFplus workbench. It is then up to you to decide what BRFplus should do if an exception actually occurs at runtime:

- [Propagated Exceptions](#)

If any of the exceptions in this list occurs at runtime, the corresponding exception instance is propagated to the application that has called the BRFplus application. The calling application is then responsible for proper exception handling. By default, the system adds all exceptions that may be raised by a called procedure to this list.

- [Ignored Exceptions](#)

If any of the exceptions in this list occur at runtime, the corresponding exception instance is **not** propagated to the application that has called the BRFplus application. In other words, the exception is suppressed and BRFplus tries to continue processing as if nothing had happened. In this scenario, it is the responsibility of the BRFplus application designer to make sure that exceptions are ignored only if this is safe with respect to the processing flow.

Caution

Whenever a called procedure raises an exception, this means that the processing is aborted immediately. There is no way to influence this elementary system behavior with the BRFplus exception handling options. From a BRFplus perspective, this means that the result value of the procedure call expression is initial, and the expression context remains unchanged.

You control the system behavior by assigning the available exceptions to either of the two lists with the help of the buttons ([Ignore Exception](#) and [Propagate Exception](#)).

More Information

[Creating a Static Method Call Expression \[page 227\]](#)

[Function Call Expression \[page 111\]](#)

1.2.11.15 Der Ausdruck „Tabellenoperation“

Definition

Der Ausdruck „Tabellenoperation“ bietet eine Vielzahl an Funktionen für den Zugriff und die Pflege von Daten, die in BRFplus-Datenobjekten vom Typ „Tabelle“ gesichert sind.

Verwendung

Mit dieser Ausdrucksart können Sie verschiedene Verwaltungsaufgaben durchführen sowie statistische Daten zu einem Datenobjekt vom Typ „Tabelle“ abrufen. Die verfügbaren Funktionen ähneln dabei der Aggregation und der Existenzprüfung des Ausdrucks „Datenbankabfrage“, allerdings haben sie einen anderen Anwendungsbereich. Die folgende Tabelle zeigt die Unterschiede zwischen diesen beiden Ausdrucksarten auf:

Tabelle 34:

Kriterium	Tabellenoperation	Datenbankabfrage
Kontextdatenobjekt	BRFplus-Tabellendatenobjekt	ABAP-Dictionary-Tabelle
Verwendung	Geschäftsorientierter Ansatz	Datenbankorientierter Ansatz
Zugriffsart	Lesen/Schreiben (es sind verschiedene Löschoperationen verfügbar)	Lesezugriff

Funktionen

Unterstützte Tabellenfunktionen

In der folgenden Tabelle sind die verschiedenen Funktionen aufgelistet, die vom Ausdruck „Tabellenoperation“ für den Zugriff auf Tabellendatenobjekte bereitgestellt werden.

Tabelle 35:

Name	Kommentar
Hat mindestens	Liefert <code>True</code> zurück, wenn die angegebene Tabelle mindestens die gegebene Anzahl an Zeilen enthält, die mit der gegebenen Bedingung übereinstimmen.
Hat genau	Liefert <code>True</code> zurück, wenn die angegebene Tabelle genau die gegebene Anzahl an Zeilen enthält, die mit der gegebenen Bedingung übereinstimmen.

Name	Kommentar
Hat nicht genau	Liefert <code>True</code> zurück, wenn die angegebene Tabelle entweder mehr oder weniger als die gegebene Anzahl an Zeilen enthält, die mit der gegebenen Bedingung übereinstimmen.
Hat weniger als	Liefert <code>True</code> zurück, wenn die angegebene Tabelle weniger als die gegebene Anzahl an Zeilen enthält, die mit der gegebenen Bedingung übereinstimmen.
Hat mehr als	Liefert <code>True</code> zurück, wenn die angegebene Tabelle mehr als die gegebene Anzahl an Zeilen enthält, die mit der gegebenen Bedingung übereinstimmen.
Hat nicht mehr als	Liefert <code>True</code> zurück, wenn die angegebene Tabelle nicht mehr als die gegebene Anzahl an Zeilen enthält, die mit der gegebenen Bedingung übereinstimmen.
Anzahl	Liefert die Anzahl an Zeilen zurück, die mit der gegebenen Bedingung in der angegebenen Tabelle übereinstimmen.
Minimum	Liefert den kleinsten Wert in der angegebenen Tabellenspalte zurück, der mit der gegebenen Bedingung übereinstimmt.
Maximum	Liefert den größten Wert in der angegebenen Tabellenspalte zurück, der mit der gegebenen Bedingung übereinstimmt.
Summe	Liefert die Summe aller Werte in der angegebenen Tabellenspalte zurück, die mit der gegebenen Bedingung übereinstimmen.
Durchschnitt	Liefert den Durchschnitt aller Werte in der angegebenen Tabellenspalte zurück, die mit der gegebenen Bedingung übereinstimmen.
Erste Zeile	Liefert die erste Zeile zurück, die mit der gegebenen Bedingung in der angegebenen Tabelle übereinstimmt.
Letzte Zeile	Liefert die letzte Zeile zurück, die mit der gegebenen Bedingung in der angegebenen Tabelle übereinstimmt.
Alle Zeilen	Liefert alle Zeilen zurück, die mit der gegebenen Bedingung in der angegebenen Tabelle übereinstimmen.

Name	Kommentar
Sortierung	<p>Sortiert die Zeilen der angegebenen Tabelle nach den Werten in den gegebenen Spalten und der Sortierreihenfolge.</p> <div> <p>i Hinweis</p> <p>Im Gegensatz zu den meisten anderen Tabellenoperationen liefert die Operation Sortieren keinen Wert zurück. Stattdessen wird das im Ausdruckskontext angegebene Tabellendatenobjekt direkt sortiert.</p> <p>Das Ergebnis des Sortierens bleibt lediglich im Speicher und wird nicht in die Datenbank gesichert.</p> <p>Sie können eine Tabelle nach höchstens fünf verschiedenen Spalten sortieren.</p> </div>
Erste Zeile löschen	Löscht die erste Zeile in der angegebenen Tabelle, die mit der gegebenen Bedingung übereinstimmt. Die geänderte Tabelle (ohne die entfernte Zeile) wird als Ergebnis zurückgeliefert.
Letzte Zeile löschen	Löscht die letzte Zeile, die mit der gegebenen Bedingung in der angegebenen Tabelle übereinstimmt. Die geänderte Tabelle (ohne die entfernte Zeile) wird als Ergebnis zurückgeliefert.
Alle Zeilen löschen	Löscht alle Zeilen, die mit der gegebenen Bedingung in der angegebenen Tabelle übereinstimmen. Die geänderte Tabelle (ohne die entfernten Zeilen) wird als Ergebnis zurückgeliefert.

i Hinweis

Es ist außerdem für alle Operationen möglich, **keine** Auswahlbedingung anzugeben. In diesem Fall wird die Operation auf alle Zeilen in der angegebenen Tabelle angewendet.

Folgende der oben aufgelisteten Operationen können nur auf numerische Tabellenfelder angewendet werden:

- Minimum
- Maximum
- Summe
- Durchschnitt

Die BRFplus-Workbench unterstützt Sie bei der Suche nach entsprechenden Spalten für eine gegebene Tabelle, indem Ihnen nur Felder angeboten werden, die den Anforderungen der jeweiligen Operation entsprechen.

Weitere Informationen

[Datenobjekt \[Seite 61\]](#)

[Der Ausdruck „Datenbankabfrage“ \[Seite 85\]](#)

1.2.11.16 Wertebereich-Ausdruck

Verwendung

Mit Wertebereich-Ausdrücken prüfen Sie, ob der Wert eines Testparameters innerhalb eines bestimmten Bereichs liegt. Das Ergebnis eines Wertebereich-Ausdrucks ist immer vom Typ „Boolesch“. Der Ausdruck gibt den Wert als „true“ oder „false“ zurück, abhängig davon ob der Wert sich innerhalb des definierten Bereichs befindet oder nicht.

Struktur

Statische und dynamische Bereichsgrenzen

Das Definieren fester Grenzwerte für einen Bereichsvergleich zur Design-Zeit ist einfach und ein guter Ausgangspunkt. Das System vergleicht die an den Ausdruckskontext gegebenen Werte zur Laufzeit mit den Werten, die für den Bereich definiert wurden, und gibt ein entsprechendes Ergebnis zurück. In bestimmten Fällen ist es jedoch erforderlich, die Bereichsgrenzen dynamisch auszuwerten, wie im folgenden Szenario.

Beispiel

Sie möchten den Tabakkonsum eines Kunden für ein Versicherungsunternehmen klassifizieren. Dazu definieren Sie die folgenden Wertebereich-Ausdrücke:

- Zigaretten pro Tag = 0
- Zigaretten pro Tag zwischen [1..10]
- Zigaretten pro Tag zwischen [11..20]
- Zigaretten pro Tag zwischen [21..50]
- Zigaretten pro Tag mehr als [50]

Die oben aufgeführten Bereiche können direkt in die Zeilen einer Entscheidungstabelle eingefügt werden. Hier weisen Sie dann jeder Tabakkonsummenge ein Klassifizierungsergebnis zu.

Es kann sich im Laufe der Zeit jedoch herausstellen, dass der angenommene Schwellenwert von >50 für die höchste Risikoklasse sich als nicht verlässlich erweist und durch weitere Faktoren, die Auswirkungen auf die körperliche Gesundheit haben könnten, angepasst werden muss. Sie können den ersten Ansatz wie folgt neu gestalten:

Ersetzen Sie für den vierten und fünften der oben definierten Bereiche den statischen Grenzwert von 50 durch einen Formelausdruck, der unter Berücksichtigung weiterer Einflussfaktoren dynamisch einen verlässlicheren Wert errechnet. Der Konsum von 50 Zigaretten pro Tag wirkt sich unter Umständen

unterschiedlich aus, abhängig vom Alter einer Person oder davon, ob die Person auf dem Land oder in einer Großstadt lebt usw.

Durch diese Änderung ändert sich der Schwellenwert zwischen der höchsten und der zweithöchsten Risikoklasse von einem statischen in einen dynamischen Wert. Dadurch werden die realen Bedingungen besser reflektiert.

Explizite und implizite Bereiche

In BRFplus können Sie Bereiche entweder explizit festlegen, indem Sie einen Wertebereich-Ausdruck definieren, der von anderen Objekten wiederverwendet werden kann, oder implizit, indem Sie die Bereichsbedingung direkt in einem übergeordneten Objekt definieren, in dem der Bereich erforderlich ist. Implizite Bereiche werden z. B. häufig zum Testen der Kontextdaten verwendet, die an eine Entscheidungstabelle (Ausdruck) geleitet wurden.

Ob Sie explizite oder implizite Bereiche verwenden, ist abhängig davon, ob Sie die Bereiche wiederverwenden möchten.

- Wenn bestimmte Wertevergleiche typisch für Ihre Anwendungsfälle und wichtig für die Ablaufsteuerung Ihrer Geschäftsprozesse sind (z. B. Clustering von Altersgruppen, Einkommen, Produktgruppen, Ländern usw.), und wenn diese Vergleiche an vielen verschiedenen Stellen in Ihrer Anwendung benötigt werden, verwenden Sie explizite und wiederverwendbare Wertebereich-Ausdrücke.
- Wenn die Vergleiche, die Sie modellieren möchten, größtenteils Ad-hoc angelegt und eher selten in Ihren Szenarien auftreten, ist es möglicherweise einfacher, implizite Bereiche zu verwenden und die Wiederverwendbarkeit außer Acht zu lassen.

Einschränkungen

Obwohl es theoretisch möglich ist, Wertebereich-Ausdrücke mit einer unbegrenzten Anzahl an Bereichsvergleichen in einem Ausdruck anzulegen, gibt es praktische Grenzen. Das System kann nur eine bestimmte Komplexität verarbeiten. Dies gilt sowohl für explizite als auch für implizite Bereiche. Es gelten folgende Einschränkungen:

- **Backend**
Im Backend kann das System Wertebereich-Ausdrücke mit mehreren Tausend Vergleichen verarbeiten. Je größer jedoch die Komplexität, desto höher die Arbeitslast für das System bei der Code-Generierung. In Extremfällen könnten die generierten Klassenmethoden das Limit der maximalen Codegröße überschreiten, die von der ABAP-Coding-Engine verarbeitet werden kann. Deshalb empfiehlt SAP, den Maximalwert der Bereichskomplexität auf ungefähr 1.000 Vergleiche pro Wertebereich-Ausdruck zu beschränken.
- **Frontend**
Das Rendern komplexer Wertebereichsdefinitionen mit der ABAP-Web-Dynpro-Technologie verursacht hohe Systemarbeitslasten, insbesondere einen hohen Speicherverbrauch. Eine Überlastung des Systems kann hier bereits auf einem Komplexitätsniveau auftreten, das erheblich geringer ist als im Backend. Deshalb ist die Anzeige von Wertebereichsdefinitionen in der Benutzeroberfläche auf maximal 99 Vergleiche pro Wertebereich-Ausdruck begrenzt.

i Hinweis

Wenn Sie Wertebereich-Ausdrücke mit mehr als 100 Vergleichen definieren müssen, können Sie dies nur programmatisch durchführen. Aufgrund der Einschränkung im Frontend ist es nicht möglich, Wertebereiche mit diesem Komplexitätsniveau interaktiv in der BRFplus-Workbench zu definieren.

Beispiel

In der unten gezeigten Preistabelle für ein Kurierunternehmen werden die Preise basierend auf der Region und dem Gewicht der Sendung berechnet. In der Tabelle kann jede Zelle in der Gewichtsspalte als Wertebereich-Ausdruck dargestellt werden. Die Zellenwerte werden zu Wertebereichsparametern und die Spalte zum Testparameter. Jede Zelle in der Gewichtsspalte gibt – abhängig von der Eingabe und der in der Tabelle beschriebenen Bedingung – einen Booleschen Wert zurück.

Tabelle 36:

Region	Gewicht	Preis
Asien-Pazifik-Raum	Unter 10 kg	40 \$
Asien-Pazifik-Raum	Zwischen 10 kg und 50 kg	55 \$
Asien-Pazifik-Raum	Über 50 kg	70 \$
Europa	Unter 10 kg	30 \$
Europa	Zwischen 10 kg und 50 kg	45 \$
Europa	Über 50 kg	60 \$

Weitere Informationen

[Wertebereich-Ausdruck anlegen \[Seite 232\]](#)

[Vergleichsoperationen \[Seite 70\]](#)

1.2.12 Ausdrucksart

Definition

Sie können benutzerdefinierte Ausdrucksarten verwenden, um den Funktionsumfang von BRFplus mit Funktionen zu erweitern, die nicht durch die vordefinierten Ausdrucksarten abgedeckt sind.

Concept

BRFplus wird mit einem Satz an vordefinierten, mächtigen Ausdrucksarten ausgeliefert. Hierzu zählen unter anderem die Entscheidungstabelle, der Suchbaum oder die Formel. Diese vordefinierten Ausdrucksarten decken eine Vielzahl an Anwendungsszenarios für Geschäftsregeln ab. Allerdings kann es vorkommen, dass sie für einen bestimmten Anwendungsfall nicht ausreichen oder umständlich zu verwenden sind. Um solche potenziellen Mängel zu umgehen, bietet BRFplus Ihnen die Möglichkeit, eigene, unternehmensspezifische Ausdrucksarten anzulegen.

Der konzeptionelle Ansatz benutzerdefinierter Ausdrucksarten ist dabei derselbe wie bei vordefinierten Ausdrucksarten. Das bedeutet, dass SAP für die vordefinierten Ausdrucksarten dieselbe technische Infrastruktur verwendet, die Ihnen für das Anlegen Ihrer eigenen Ausdrucksarten zur Verfügung steht. Daher können Sie davon ausgehen, dass das Verhalten Ihrer eigenen Ausdrucksarten in Bezug auf Performance, Zugriffskontrolle usw. dem Verhalten der vordefinierten Ausdrucksarten entsprechen wird. Benutzerdefinierte Ausdrucksarten werden wie eine horizontale Erweiterung des Anwendungsbereichs behandelt und **nicht** wie vertikale Add-Ons.

Funktionsumfang

Einstellungen und Objekte

Da das BRFplus-Element *Ausdrucksart* einen generischen Behälter für ein beliebiges Verarbeitungsverhalten darstellt, müssen Sie eine Reihe an ABAP-OO-Klassen und Interfaces angeben, die das gewünschte Systemverhalten implementieren. Folgende Einstellungen und Objekte müssen für eine benutzerdefinierte Ausdrucksart angegeben werden:

Tabelle 37:

Name	Status	Kommentar
Aktionstyp	obligatorisch	Gibt an, ob die Ausdrucksart verwendet wird, um eine Aktion durchzuführen oder ein Ergebnisdatenobjekt zurückzuliefern.
Klasse	obligatorisch	ABAP-OO-Klasse, die zur Implementierung der Geschäftslogik der Ausdrucksart verwendet wird.
Interface	obligatorisch	ABAP-OO-Interface, das für den Zugriff auf öffentliche Methoden in der Klasse verwendet wird.
Query-Klasse	optional	ABAP-OO-Klasse, die verwendet wird, um in der BRFplus-Datenbank nach Instanzen der Ausdrucksart und ihren Attributwerten zu suchen.

Name	Status	Kommentar
UI-Klasse	optional	ABAP-OO-Klasse, die für den Zugriff auf Web-Dynpro-Anwendungen verwendet wird, die mit der Ausdrucksart verknüpft sind.
Datenaustauschklasse	optional	ABAP-OO-Klasse, die verwendet wird, um eine Document Type Definition (DTD) und Datenkonvertierungsroutinen für den XML-basierten Import und Export zu definieren, der mit der Ausdrucksart zusammenhängt.

Transportobjekte

Üblicherweise müssen für jede benutzerdefinierte Ausdrucksart in einer transportierbaren Anwendung eigene Transportobjekte definiert werden, die alle benötigten Teile für den Transport des Ausdrucks von einem System in das andere zusammenfassen. Die notwendigen Informationen werden in folgenden Transportobjekten gespeichert:

- Customizing-Daten (obligatorisch)
- Systemdaten (obligatorisch)
- Verteilbare Daten (optional)

BRFplus verwendet für die vorinstallierten, von SAP ausgelieferten Ausdrucksarten die Transportobjekte **FDT0000** (Customizing), **FDT0001** (System) und **FDT0002** (Verteilung). Diese Objekte sind für die Standard-Ausdrucksarten reserviert. Sie können sie nur unter folgender Bedingung für Ihre eigenen Ausdrucksarten verwenden: Die neue Ausdrucksart muss die genaue Kopie einer Standard-Ausdrucksart sein und darf keine Erweiterungen oder Änderungen beinhalten. Eine solche duplizierte Ausdrucksart kann hilfreich sein, wenn Sie Ihr eigenes benutzerdefiniertes UI für eine Ausdrucksart erstellen oder die Ausdrucksart mit besonderen Prüfroutinen verknüpfen möchten, die nicht Teil der Standardauslieferung sind.

Hinweis

Benutzerdefinierte Ausdrucksarten, die keinerlei Datenmanipulation vornehmen, müssen keinen Transportobjekten zugeordnet werden. Dies gilt beispielsweise für Ausdrucksarten, die Eingabedaten lesen und in einem UI-Element visualisieren sollen.

Wenn Sie benutzerdefinierte Ausdrucksarten in einer lokalen Anwendung anlegen, müssen Sie keine Transportobjekteinstellungen vornehmen. Daher wird in der BRFplus-Workbench in diesem Fall keine entsprechende Registerkarte angeboten.

Weitere Informationen über Transportobjekte finden Sie in der Dokumentation *BC - Zentrale Pflege- und Transportobjekte* in der SAP NetWeaver-Bibliothek des SAP Help Portals.

1.2.13 Aktionen

Verwendung

Aktionen sind eine besondere Art von Ausdruckstypen, die zur Definition des interaktiven Teils von BRFPplus verwendet werden. In manchen Fällen können Aktionen alternativ zu Ausdrücken verwendet werden. In einer Entscheidungstabelle können Sie beispielsweise für eine Bedingungszeile definieren, dass das System eine Aktion ausführt, statt einen Ergebniswert zurückzugeben. Im Folgenden sind alle Verwendungsarten für Aktionen aufgelistet:

- Aktion, die durch eine Regel ausgelöst wird
- Aktion, die durch eine übereinstimmende Bedingung in einer Entscheidungstabelle, einem Entscheidungsbaum, einem Suchbaum oder einem Case-Ausdruck ausgelöst wird
- Aktion, die als Folgeaktion einer anderen Aktion ausgelöst wird

Aktionen haben jedoch keine Ausgabewerte. Folglich können Sie Aktionen nur als Element im letzten Schritt eines jeden Verarbeitungszweiges nutzen. In diesem Fall kann eine Aktion dazu genutzt werden, um auf die bis dahin errechneten Ergebnisse der Regel zu reagieren. Jedoch gibt es keine Möglichkeit, eine Regelberechnung als Reaktion auf eine Aktion durchzuführen, die zuvor ausgelöst wurde.

Je nach Aktionsart ist es weiterhin möglich, eine Aktion als Auslöser für komplexe Systemaktivitäten zu nutzen. Dies gilt beispielsweise für die Aktionsarten *Workflow* oder *Prozedur aufrufen*. Jedoch ist es nicht möglich, die Weiterverarbeitung als bestimmte Reaktion auf alle Änderungen, die durch die Aktion hervorgerufen wurden, zu definieren, sobald diese Aktion ausgeführt wurde.

Einige Aktionsarten entsprechen einer verwandten Ausdrucksart. Um Verwechslungen zu vermeiden, ob ein gegebenes Objekt eine Aktion oder ein Ausdruck ist, sind alle Aktionsarten mit einem angehängten (*Aktion*) gekennzeichnet. Darüber hinaus erfolgt die Benennung der Aktionsarten wenn möglich nach dem sprachlichen Muster "Objekt - Verb" (zum Beispiel im Fall von *Prozedur aufrufen (Aktion)*, im Gegensatz zur Ausdrucksart *Prozeduraufruf*).

BRFPplus unterstützt die folgenden Aktionsarten:

- [Prozedur aufrufen \(Aktion\) \[Seite 133\]](#)
- [Nachricht protokollieren \(Aktion\) \[Seite 133\]](#)
- [E-Mail senden \(Aktion\) \[Seite 135\]](#)
- [Workflow starten \(Aktion\) \[Seite 138\]](#)
- [Workflow-Ereignis \(Aktion\) \[Seite 140\]](#)

Weitere Informationen

[Aktionsarten anlegen \[Seite 252\]](#)

1.2.13.1 Prozedur aufrufen (Aktion)

Verwendung

Die Aktion Prozedur aufrufen ermöglicht das Ausführen von Funktionsbausteinen sowie Methoden einer ABAP-OO-Klasse im Backend-System. Aus funktionaler Sicht entspricht diese Aktionsart dem Ausdruck Prozeduraufruf. Lediglich folgende objekttypspezifische Unterschiede sind zu beachten:

- Wie alle Aktionsarten liefert die Aktion Prozedur aufrufen keine Ergebnisse zurück.
- Wie bei allen Aktionsarten können Sie Folgeaktivitäten definieren, die nach dem Ausführen der Aktion Prozedur aufrufen ausgelöst werden.

Weitere Informationen

[Prozeduraufruf \(Ausdruck\) \[Seite 118\]](#)

1.2.13.2 Nachricht protokollieren (Aktion)

Verwendung

Mit der Aktion Nachricht protokollieren können Sie Systemnachrichten definieren, die das System während der Regelverarbeitung unter bestimmten Umständen in das Anwendungsprotokoll schreibt. Dies ist hilfreich bei der Dokumentation des Verarbeitungsablaufs in einer Anwendung und kann zu Prüfungszwecken sogar rechtlich verpflichtend sein, abhängig von Ihrem Business Case, Ihrer Region und der dortigen Gesetzgebung.

Ob und in welchem Ausmaß Nachrichten überhaupt protokolliert werden sollen, hängt größtenteils von den jeweiligen Einstellungen für das Anwendungsobjekt ab, zu dem eine entsprechende Aktion zur Nachrichtenprotokollierung gehört. Diese Einstellungen auf Anwendungsebene werden von den Aktionen geerbt. Außerdem können Sie auf Anwendungsebene festlegen, ob die Standardeinstellungen für das Protokoll-Handling von einer Aktion individuell angepasst werden können oder nicht.

Hinweis

Mit dem Report `FDT_DEPLOYMENT_LOG` können Sie die Anwendungsprotokolle von BRFplus analysieren.

Funktionsumfang

In der folgenden Tabelle sind die Funktionen zur Protokollierung aufgelistet:

Tabelle 38:

Einstellung	Kommentar
<i>Anwendungsprotokollobjekt, Anwendungsprotokoll-Unterobjekt</i>	Legt die Anwendungsprotokollobjekte fest, die für das Aufzeichnen von Protokolleinträgen von Objekten dieser Anwendung verwendet werden (zum Beispiel Aktionen der Art <i>Nachricht protokollieren</i> . Weitere Informationen finden Sie in der SAP NetWeaver-Bibliothek des SAP Help Portals unter <i>Application Log - Bedienungshinweise für Entwickler (BC-SRV-BAL)</i> .
<i>Externer Identifikationsmodus</i>	<p>Legt fest, ob und auf welche Weise den Protokolleinträgen ein zusätzliches Tag hinzugefügt werden soll. Sie können aus folgenden Optionen wählen:</p> <ul style="list-style-type: none"> • <i>Keine</i> Es wird keine externe Identifikation in das Protokoll geschrieben. • <i>Freitexterfassung</i> Geben Sie eine Zeichenfolge ein, mit deren Hilfe Sie die relevanten Protokolleinträge identifizieren können. Die maximale Länge beträgt 100 Zeichen. • <i>Nach Referenzobjekt</i> Weisen Sie der Aktion ein Datenobjekt oder einen Ausdruck zu. Zur Laufzeit wird der Wert dieses Referenzobjekts in das Protokoll geschrieben, wenn die Aktion ausgelöst wurde.
<i>Persistieren</i>	Legt fest, ob Protokolldaten permanent in der Datenbank gespeichert werden oder nicht. Wenn die Protokolldaten nicht in der Datenbank gespeichert werden, werden sie nur zur Laufzeit im Speicher behalten und gehen nach der Sitzung verloren.

Nachrichten

Anzahl an Nachrichten

Im Bereich *Nachrichten protokollieren* legen Sie fest, welche Nachrichten in das Protokoll geschrieben werden sollen, wenn die Aktion ausgelöst wurde. Sie können jeder Aktion zur Nachrichtenprotokollierung eine beliebige Anzahl an Nachrichten zuordnen.

Freitext oder Nachrichtenwiederverwendung

Sie können entweder einen eigenen Text als Nachrichtentext eingeben oder einen im System bereits vorhandenen Nachrichtentext verwenden. Wenn Sie eine bereits vorhandene Nachricht verwenden möchten, müssen Sie die Nachrichtenklasse und Nachrichtennummer angeben. Wenn diese Nachricht Parameter enthält, werden Sie aufgefordert, Werte für diese Parameter anzugeben. Hier können Sie wiederum entweder selbst definierten Text eingeben oder ein Datenobjekt bzw. einen Ausdruck zuweisen, der zur Laufzeit ausgewertet wird.

Nachrichtentyp

Unabhängig davon, ob Sie eine Freitextnachricht oder eine bereits vorhandene Nachricht wählen, müssen Sie den Nachrichtentyp festlegen. Es gibt folgende Nachrichtentypen:

- Abbrechen
- Fehler
- Beenden
- Information
- Status
- Warnung

Hinweis

Wenn einer der oben genannten Nachrichtentypen für eine Aktion nicht verfügbar ist, liegt dies an einer Beschränkung, die auf Anwendungsebene definiert wurde.

Weitere Informationen

[Anwendung \[Seite 30\]](#)

1.2.13.3 E-Mail senden (Aktion)

Verwendung

Mit der Aktion E-Mail versenden, können Sie eine Person über eine bestimmte Situation im System benachrichtigen. In diesem Fall muss der Empfänger der E-Mail entscheiden, ob bestimmte Maßnahmen ergriffen werden müssen, um die Situation zu lösen. Die Situation kann kritisch sein oder nicht, je nachdem was für Ihr Unternehmen nützlich ist. Es ist ratsam, eine verantwortliche Person zu benachrichtigen, wenn eine kritische Situation auftritt. Jedoch gibt es Situationen oder Aktivitäten, die nicht auf ein Problem hinweisen, sondern einfach unregelmäßig auftreten. In diesem Fall können Sie mit Hilfe einer vom System automatisch versendeten Benachrichtigung, Routineprüfungen umgehen und Ihre Zeit so anderen Aufgaben widmen.

Voraussetzungen

Die SAPconnect Komponente (BC-SRV-COM) wurde für Internet-Dienste eingerichtet. Sie müssen einen entsprechenden Knoten in der SAPConnect-Administration und eine RFC-Destination vom Typ TCP/IP-Verbindung angeben, mit der Sie eine Verbindung zu einem Mail-Server herstellen. Weitere Informationen finden Sie in der SAP NetWeaver-Bibliothek des SAP Help Portals in der SAPconnect-Dokumentation (BC-SRV-COM).

Funktionsumfang

Allgemein

Nutzen Sie die Aktion E-Mail versenden, um über das System eine E-Mail mit vordefiniertem Text mit dem Standardprotokoll SMTP zu versenden. Dies ist hilfreich, wenn im Ablauf der Regelbearbeitung kritische Situationen auftreten, für die Anwendererfahrung benötigt wird. Dies gilt auch für Workflow-Szenarien denen verschiedene Sachbearbeiter zugeordnet sind. Die Verarbeitung kann dann nur fortgesetzt werden, wenn ein bestimmter Rolleninhaber die nötige Aktion ausgeführt hat.

Empfänger

Sie können entweder eine statische E-Mail-Adresse oder einen Ausdruck eingeben, um den Empfänger dynamisch zu ermitteln. Sie können ebenso beide Möglichkeiten, Empfänger zu kontaktieren, in derselben Aktion kombinieren. Dies stellt sicher, dass eine bestimmte Person in jedem Fall benachrichtigt wird (z.B. der Qualitätsmanager) sowie die für die Verarbeitung von Benachrichtigungen zuständige Person (z.B. der schichthabende verantwortliche Mitarbeiter in einem Dreischichtbetrieb).

Sie können eine beliebige Anzahl von statischen E-Mail-Adressen in das *Direkte E-Mail*-Eingabefeld durch Semikola getrennt eingeben. Beachten Sie, dass die Zahl der E-Mail-Adressen die Feldlänge des Eingabefeldes überschreiten kann.

Um die Empfänger dynamisch ermitteln zu lassen, wählen Sie einen Ausdruck, eine Konstante oder ein Kontextdatenobjekt derselben Textart. Der vom gewählten Objekt zurückgegebene String muss denselben syntaktischen Regeln folgen wie im *An*-Feld. Sie können bis zu vier verschiedene Ausdrücke für die Ermittlung von Empfängern in einer Aktion definieren.

i Hinweis

Wenn Sie verschiedene Nachrichten an eine Verteilerliste schicken müssen, können Sie diesen Anwendungsfall gestalten, indem Sie ein Tabellendatenobjekt, das zuvor mit den E-Mail-Adressen befüllt wurde, als dynamischen Empfänger zuordnen. Hierzu stellen Sie sicher, dass die Tabellenstruktur jeweils nur eine E-Mail-Adresse pro Textfeld vorsieht. Beispielsweise können Sie das Tabellendatenobjekt mit Hilfe einer Entscheidungstabelle im Mehrfachtreffermodus befüllen.

E-Mail-Text

Im *Text*-Feld können Sie nun entweder einen vordefinierten statischen Text, einen zur Laufzeit dynamisch erzeugten Text mit Hilfe von Platzhaltern in der Nachricht oder eine Kombination aus beiden eingeben. Sie können bis zu acht Platzhalter pro Aktion E-Mail versenden definieren, wovon jeder zur Laufzeit von einer Konstante, von einem Ergebnis eines Ausdrucks oder von einem Element-Datenobjekt ersetzt werden kann.

Den E-Mail-Text formatieren

Standardmäßig wird eine Klartext E-Mail vom System angelegt, die wie eine genaue Kopie davon aussieht, was Sie zur Designzeit in das *Text*-Feld eingeben. Es ist jedoch auch möglich, E-Mails im HTML-Format zu versenden, um HTML-Funktionen, wie Formatierung (Schriftfamilie, Schriftgröße, Fettschrift, Kursivschrift usw.), Tabellen, Grafiken usw. zu nutzen. Geben Sie hierzu die folgende Document Type Declaration **wie in Zeile 1** im *Text*-Feld ein:

```
<!DOCTYPE HTML>
```

Mit dieser Declaration interpretiert das System den gesamten Restinhalt der E-Mail nicht länger als Klartext, sondern als HTML-Dokument. Dies hat folgende Auswirkungen:

- Implizite Zeilenvorschübe und Absätze in E-Mails, die zuvor im Klartextformat geschrieben wurden, gehen verloren. Gleiches gilt für Einrückungen, die mit Hilfe von Leerzeichen oder Tabulatoren gesetzt wurden. Um die Zeilenvorschübe wiederherzustellen, müssen Sie explizit die dafür vorgesehenen HTML-Tags verwenden (d.h.
 oder <P>).
- Um die verschiedenen HTML-Funktionen zu nutzen, müssen Sie die entsprechenden HTML-Tags direkt in den E-Mail-Text eingeben, ähnlich wie bei der Bearbeitung einer HTML-Datei für eine Webseite.
- BRFplus kann die eingegebenen HTML-Tags nicht validieren. Sie müssen Ihre eingegebenen Tags daher selbst auf Richtigkeit und Gültigkeit prüfen.

i Hinweis

HTML-Quellcode ohne Werkzeugunterstützung zu bearbeiten, kann mühsam und fehleranfällig sein. Wenn Sie daher HTML-Unterstützung für die *E-Mail versenden* Aktion nutzen wollen, empfehlen wir, den E-Mail-Text mit Hilfe eines externen Standalone-HTML-Editors Ihrer Wahl zu erstellen, so dass Sie die Funktionen Syntaxcheck, Code-Vervollständigung, Validierung usw. nutzen können. Sobald Sie den E-Mail-Text erstellt haben, kopieren Sie den zugrunde liegenden HTML-Quellcode des externen Editors in das BRFplus.

Beispiel

In einem Regelsatz, den Sie zur Verwaltung von Kundenaufträgen für Versicherungsverträge verwenden, unterscheiden Sie zwischen eingehenden Standardaufträgen und anderen Aufträgen, die aus den folgenden Gründen speziell bearbeitet werden müssen:

- Ungewöhnlich hoher Versicherungsbetrag
- Der Auftraggeber befindet sich in einem anderen Land
- Inkonsistente Auftragsdaten

Diese Gründe werden als Ausnahme der Standard-Auftragsbearbeitung angesehen und müssen speziell bearbeitet werden. Hierzu legen Sie einen zusätzlichen Verarbeitungszweig in dem Regelsatz an, in dem die Aktion E-Mail versenden ausgelöst wird. Bei dieser Aktion geben Sie eine vordefinierte Beschreibung des Ausnahmefalls ein und ergänzen dynamische Inhalte aus dem Regelsatz Kontext, wie die Auftrags-ID oder den angefragten Versicherungsbetrag. Sie können ebenso einen Ausdruck nutzen, um einen zuständigen Mitarbeiter in Ihrem Unternehmen zu ermitteln und diesen als E-Mail-Empfänger zuzuordnen.

Ein Code-Beispiel für die Verwendung der HTML-Option:

```
<!DOCTYPE HTML><html> <head><title>[Decision needed:] Insurance amount exceeded</title> </head> <body> <p>Dear colleague:</p> <p>This is to inform you of a pending workflow issue that needs your decision:</p> <p>A customer has applied for a modification of his or her existing life insurance contract. The requested change includes an increased amount insured from previously <font:color="#000000">20.000 € to <font-color:"#ff0000"><font color="#ff0000">200.000 €</font-color:"#ff0000"></font:color="#000000"></p> <p>This would exceed our standard maximum amount of <font color="#3333ff">50.000</font> €. Please check this case and decide how to proceed.</p> <p>Best regards</p> <p>Contract management workflow</p></body> </html>
```

Die E-Mail wird dem Empfänger wie folgt angezeigt (der neue Betrag wird in rot hervorgehoben, der Standard-Höchstbetrag in blau):

Liebe Mitarbeiterin, lieber Mitarbeiter,

diese E-Mail wurde versendet, um Sie über einen ausstehenden Workflow zu informieren, der von Ihnen bearbeitet werden muss.

Ein Kunde hat einen Antrag auf Änderung seines oder ihres bestehenden Lebensversicherungsvertrages gestellt. Die beantragte Änderung beinhaltet eine Erhöhung des Versicherungsbetrages von zuvor 20.000 € auf 200.000 €.

Dies würde den Standard-Höchstbetrag von 50.000 € überschreiten. Bitte prüfen Sie diesen Fall und entscheiden, wie zu verfahren ist.

Mit freundlichen Grüßen

Vertragsmanagement Workflow

1.2.13.4 Workflow starten (Aktion)

Verwendung

Mit der Workflow starten Aktion können Sie die Workflow-Verarbeitung in BRFplus integrieren. Sie können auf alle Workflows zugreifen, die im Backend-System definiert wurden, ihnen eine Workflow starten Aktion zuordnen und das System die vordefinierten Schritte ausführen lassen, sobald die Aktion durch einen BRFplus Ausdruck ausgelöst wurde.

Zu den Unterschieden zwischen der Workflow starten Aktion und der Workflow-Ereignis Aktion, siehe [Workflow-Ereignis \(Aktion\) \[Seite 140\]](#).

Voraussetzungen

Ihr Backend-System enthält die Softwarekomponente `SAP_BASIS` mit einem niedrigeren Release als 800. In `SAP_BASIS` 800 oder höher wird die zugrunde liegende Workflow-Funktionalität dieser Aktionsart nicht unterstützt.

Sie haben im Backend-System eine Workflow-Definition mit dem [Workflow Builder](#) angelegt (Transaktion **SWDD**).

Funktionsumfang

Nach Eingabe der gewünschten Workflow-Kennung, klicken Sie [Workflow Container aktualisieren](#). Das System analysiert alle Elemente des Data Dictionary, die in der Workflow-Definition verwendet wurden. Für jedes der Elemente des Data Dictionary legt das System in der aktuellen Anwendung automatisch ein BRFplus-Datenobjekt an und stellt eine Bindungsbeziehung zu dem entsprechenden Element her.

Sie können Eingabedaten zu allen Elementen des Workflow-Containers bereitstellen. Je nach Element können Sie entweder direkt einen Wert, eine Referenz auf ein Kontext-Datenelement oder einen zur Laufzeit ausgewerteten Ausdruck eingeben.

Im Falle von komplexen Containerelementen (Struktur oder Tabelle), können Sie die Daten für diese Elemente bereitstellen, indem Sie eine Entscheidungstabelle als Eingabequelle zuordnen. Jedes der Struktur- oder Tabellenfelder muss durch ein entsprechendes Ergebnisdatenobjekt aus der Entscheidungstabelle abgedeckt werden. Mit Hilfe von zusätzlichen Kontextdatenobjekten für die Entscheidungstabelle können Sie kontrollieren, welches der vordefinierten Ergebnisse für den Workflow-Container genutzt wird.

Sie können festlegen, dass der Workflow nicht direkt startet, nachdem die Workflow starten Aktion ausgelöst wurde. Stattdessen können Sie eine Verzögerung der Ausführung von bis zu 9999 Tagen, 23 Stunden, 59 Minuten und 59 Sekunden definieren.

Hinweis

Wenn Sie eine Workflow starten Aktion prüfen oder aktivieren, überprüft das System nicht nur, ob die BRFlowplus-spezifischen Einstellungen korrekt sind. Das System erkennt zudem jede inkonsistente oder unvollständige Einstellung, die in der Definition der zugrunde liegenden Workflow-Definition korrigiert werden muss. Zum Beispiel die fehlende Zuordnung von Sachbearbeitern für die Workflows.

Beispiel

Sie führen eine Anwendung zur Verwaltung von Hausratversicherungsverträgen aus. Da der Wert des Hausrats die Höhe des Versicherungssatzes bestimmt, enthält die Anwendung einen wertbasierten Diskriminator, der sicherstellt, dass Werte, die einen gesetzten Grenzwert überschreiten, nicht direkt in das System eingegeben werden können. Das System unterscheidet mit einem initialen Case-Ausdruck zwischen erlaubten und den Grenzwert übersteigenden Werten. Die erlaubten Werte werden sofort an eine Entscheidungstabelle zur Bestimmung des Versicherungssatzes weitergegeben, wohingegen die den Grenzwert übersteigenden Werte einem Verarbeitungszweig zugeordnet werden, was die Workflow starten Aktion auslöst. Dieser Workflow wurde so definiert, dass zusätzliche Genehmigungsschritte erforderlich sind, bevor der Vertrag weiter bearbeitet werden kann. Mit diesem Szenario können Sie sicherstellen, dass die erforderlichen Risikominderungsmaßnahmen ergriffen werden, um Ihr Unternehmen davor zu schützen, Versicherungssummen zahlen zu müssen, die die Gesamtrisikoberechnung Ihres Unternehmens übersteigen.

BRFlowplus in einen bestehenden Workflow integrieren

Beide workflowbezogenen Aktionsarten, die von BRFlowplus angeboten werden, nutzen eine BRFlowplus Anwendungsregel. Ausgehend davon können Sie entscheiden, ob Sie die SAP Business Workflow Funktionalitäten nutzen wollen. Es ist jedoch ebenfalls möglich, diese Reihenfolge umzukehren. Das heißt, Sie haben bereits einen Workflow implementiert und wollen diesen um eine bestimmte BRFlowplus Funktion erweitern. Weitere Informationen zu diesem Anwendungsszenario finden Sie im Whitepaper "Integration zwischen SAP Business Workflow und BRFlowplus" auf den BRFlowplus-Seiten im SAP Community Network (siehe Abschnitt verwandte Links unten).

Weitere Informationen

[Workflow-Ereignis \(Aktion\) \[Seite 140\]](#)

[Datenbindung \[Seite 66\]](#)

[Entscheidungstabelle \(Ausdruck\) \[Seite 89\]](#)

Siehe auch *SAP Business Workflow: Referenzdokumentation* in der SAP NetWeaver-Bibliothek im SAP Help Portal.

Weitere Informationen

[Integration zwischen SAP Business Workflow und BRFplus](#) 

1.2.13.5 Workflow-Ereignis (Aktion)

Verwendung

Mit der Workflow-Ereignis Aktion können Sie die Workflow-Verarbeitung in BRFplus integrieren. Im Gegensatz zur Workflow starten Aktion müssen Sie keinen bestimmten Workflow auswählen, der ausgelöst werden soll. Stattdessen geben Sie ein Ereignis an, das für ein Objekt im Backend-System definiert wurde. Wenn das Ereignis ausgelöst wird, werden auch alle damit registrierten Workflows ausgelöst.

Voraussetzungen

Ihr Backend-System enthält die Softwarekomponente `SAP_BASIS` mit einem niedrigeren Release als 800. In `SAP_BASIS` 800 oder höher wird die zugrunde liegende Workflow-Funktionalität dieser Aktionsart nicht unterstützt.

Funktionsumfang

Allgemein


BRFplus unterstützt die folgenden Objekttypen, die Sie für die Zuordnung zu einem Ereignis verwenden können:

- ABAP-Objektklasse (**CL**)
- Business-Klasse (**BC**)
- Business-Objekt (BOR-Objekt; **BO**)

Nach Eingabe der erforderlichen Ereignis-Details, klicken Sie [Ereigniscontainer aktualisieren](#). Das System analysiert alle Elemente des Data Dictionary, die als Ereignisparameter verwendet wurden. Für jedes der Elemente des Data Dictionary legt das System in der aktuellen Anwendung automatisch ein BRFplus-Datenobjekt an und stellt eine Bindungsbeziehung zu dem entsprechenden Element des Data Dictionary her.

Sobald der Ereigniscontainer mit den Parameterfeldern befüllt wurde, können Sie die Werte definieren, die an das Ereignis weitergegeben werden sollen. Je nach Paramentertyp können Sie dies entweder durch Eingabe eines Wertes oder durch Zuordnung eines zur Laufzeit ausgewerteten Ausdrucks tun.

ABAP-Objektklasse

Für Ereignisse, die in Klassen des Typs ABAP-Objekt oder Business-Klasse definiert wurden, ist es erforderlich, dass die Klasse das **IF_WORKFLOW** Interface implementiert. Wenn Sie die Werteilfe () für die Suche nach Klassen nutzen, beschränkt das System die Suchresultate automatisch auf die Klassen, die diese Voraussetzung erfüllen.

Alle Ereignisparameter, die Sie nutzen wollen, müssen eine direkte, explizite Typdefinition haben. Die Typdefinition muss jedoch auf einem Typ entsprechen, der im Data Dictionary vorhanden ist. Wenn mindestens ein Ereignisparameter eine indirekte Typzuordnung hat (zum Beispiel, durch die Anweisungen **TYPE REF** oder **LIKE** umgesetzt)) kann das Ereignis nicht für eine Workflow-Ereignis Aktion genutzt werden.

Objektinstanz-ID

Zur Laufzeit werden Ereignisse von einer Instanz der gegebenen Klasse oder des gegebenen Businessobjekts ausgelöst. Sie können diese Instanz für eine Klasse durch die [lokale persistente Objektreferenz](#) (LPOR) identifizieren. Eine Business-Objekt (BO) Instanz kann über den Business-Objektschlüssel identifiziert werden. Für eine BRFplus Workflow-Ereignis Aktion muss der LPOR- oder BO-Schlüssel zur Laufzeit an die [Objektinstanz-ID](#) weitergegeben werden. Hierzu geben Sie den entsprechenden Wert entweder mittels eines Kontextdatenobjekts oder durch Auswertung eines Ausdrucks an die Aktion weiter. Die aufrufende Anwendung gibt die entsprechende Instanz-ID an den Kontext der aufgerufenen Funktion weiter.

Vergleich der Workflow-Ereignis Aktion und der Workflow starten Aktion

Die Hauptunterschiede zwischen den Aktionsarten Workflow-Ereignis und Workflow starten sind die folgenden:

- Der Workflow starten Aktion ist immer genau ein Workflow zugeordnet. Die Zuordnung ist Teil der Definition der Aktion. Im Gegensatz dazu kann eine Workflow-Ereignis Aktion eine beliebige Zahl (einschließlich null) von registrierten Workflows beinhalten, die auf das in der Aktion definierte Ereignis warten.
- Folglich garantiert das Auslösen einer Workflow-Ereignis Aktion **nicht**, dass ein Workflow als Reaktion auf die Aktion gestartet wird. Wenn kein Workflow-Ereignisbehandlungler vorhanden ist, der das Ereignis der Aktion überwacht, wird nichts geschehen.
- Im Gegensatz zur Workflow starten Aktion, kann mit der Workflow-Ereignis Aktion nicht ermittelt werden, welcher Workflow genau durch die Aktion ausgelöst wird. Dies wird durch die Registrierung einer bestimmten Workflow-Instanz für das betreffende Ereignis bestimmt. Der Workflow-Ereignisbehandlungler überwacht die im System ausgelösten Ereignisse dauerhaft. Sobald ein registriertes Ereignis ausgelöst wird, startet der Workflow.

Business-Klasse, Business-Objekt

Diese Konstrukte beziehen sich auf ältere Klassenkonzepte in SAP-Systemen. BRFplus unterstützt diese alten Klassentypen aus Kompatibilitätsgründen. Es wird nicht empfohlen, neue Klassen dieser Art anzulegen.

BRFplus in einen bestehenden Workflow integrieren

Beide workflowbezogenen Aktionsarten, die von BRFplus angeboten werden, nutzen eine BRFplus Anwendungsregel. Ausgehend davon können Sie entscheiden, ob Sie die SAP Business Workflow Funktionalitäten nutzen wollen. Es ist jedoch ebenfalls möglich, diese Reihenfolge umzukehren. Das heißt, Sie haben bereits einen Workflow implementiert und wollen diesen um eine bestimmte BRFplus Funktion erweitern. Weitere Informationen zu diesem Anwendungsszenario finden Sie im Whitepaper "Integration zwischen SAP Business Workflow und BRFplus" auf den BRFplus-Seiten im SAP Community Network (siehe Abschnitt verwandte Links unten).

Weitere Informationen

[Workflow starten \(Aktion\) \[Seite 138\]](#)

[Datenbindung \[Seite 66\]](#)

Siehe auch *SAP Business Workflow: Referenzdokumentation* in der SAP NetWeaver-Bibliothek im SAP Help Portal.

Weitere Informationen

[Integration zwischen SAP Business Workflow und BRFplus](#) 

1.2.14 Dynamischer Datenbank-View (DDBV)

Ein dynamischer Datenbank-View (DDBV) ist ein Objekt, das eine Verbindung zwischen bestimmten BRFplus-Ausdrücken und einer Datenquelle in SAP HANA darstellt.

Definition

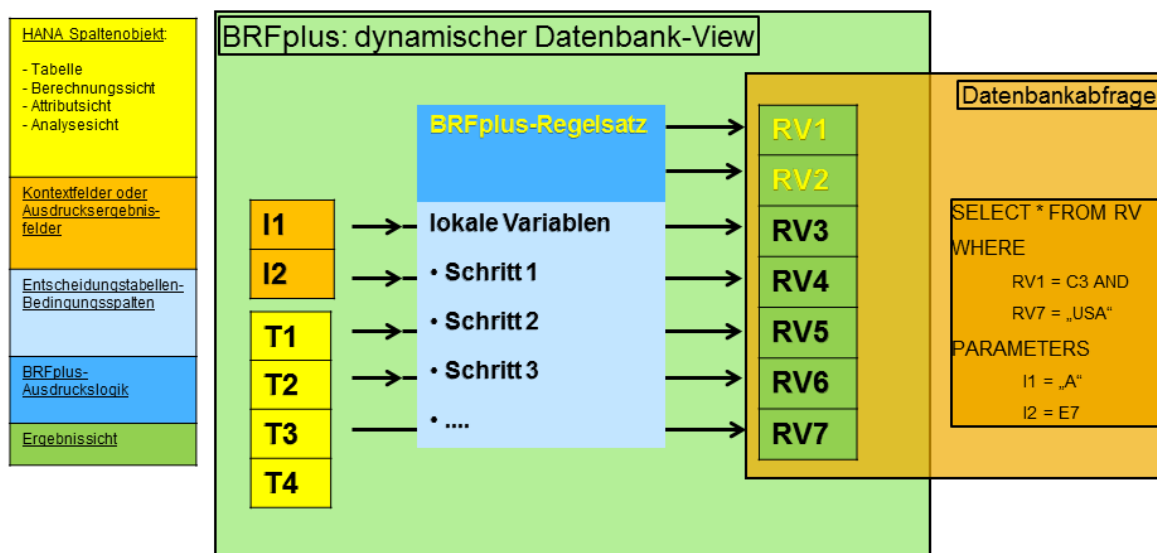
Über einen DDBV können Sie mit BRFplus auf Daten zugreifen, die in SAPs In-Memory-Datenbank SAP HANA gespeichert sind. Dies ermöglicht es Ihren Regelanwendungen, sehr große Datenmengen in sehr kurzer Zeit zu verarbeiten. Der DDBV dient als Verbindung zwischen einem bestimmten Datenbankobjekt und einem BRFplus-Ausdruck. Folgende BRFplus-Ausdrucksarten können mit einem DDBV kombiniert werden:

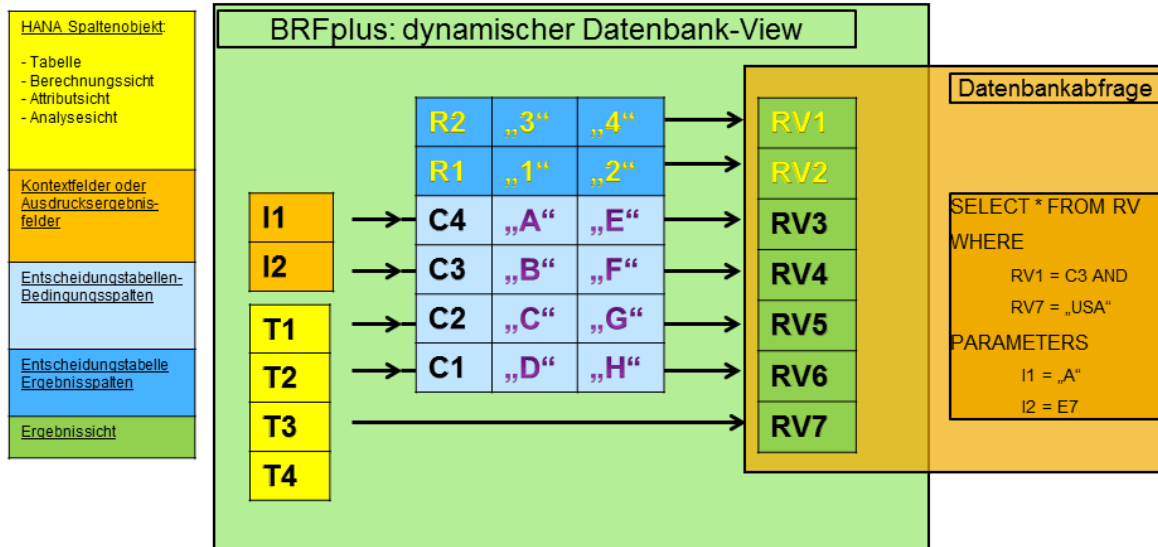
- Datenbankabfrage
- Entscheidungstabelle

Der Hauptunterschied in der Verwendung dieser Ausdrücke ist folgender: Bei einem Datenbankabfrage-Ausdruck können Sie diesem einen DDBV zuordnen und anschließend auf Daten im mit diesem DDBV verknüpften Datenbankobjekt zugreifen und sie auswerten. Mit einer Entscheidungstabelle hingegen können

Sie berechnete Ergebnisse angeben, die auf Tabellendaten und zusätzlichen Daten basieren, welche ein Benutzer möglicherweise zur Laufzeit eingibt.

Die folgenden Diagramme veranschaulichen, wie die verschiedenen beteiligten Objekttypen miteinander verbunden sind:





Voraussetzungen

Sie haben auf Ihrem Backend-System SAP NetWeaver Decision Service Management (DSM) installiert. Sie haben Zugriff auf SAP HANA-Datenbank.

Funktionsumfang

Unterstützte Datenbankobjekttypen

Ein DDBV kann mit folgenden SAP HANA-Datenbankobjekttypen verknüpft werden:

- Tabelle
- Tabellentyp
- Analysesicht
- Attributsicht
- Berechnungssicht

Nachdem Sie sich für einen Typ entschieden haben, müssen Sie nur noch den Objektnamen und das Datenbankschema des Objekts angeben. Diese Informationen finden Sie in der Entwicklungsumgebung für SAP HANA-Datenbanken, dem SAP HANA Studio.

Datenbankfelder

Nachdem Sie ein Datenbankobjekt mit dem DDBV verknüpft haben, können Sie aus der Liste der verfügbaren Datenbankfelder des gewählten Objekts diejenigen auswählen, die Sie im DDBV verwenden möchten. Wählen Sie hierzu auf der Registerkarte *Datenbankfelder* die Option *Datenbankfeld*.

Hinweis

Stellen Sie sicher, dass mindestens ein Datenbankfeld gewählt ist, denn sonst kann nicht ermittelt werden, welche Art von Daten der DDBV bereitstellen soll. Ein DDBV mit leerer Feldliste kann nicht aktiviert werden.

Eingabeparameter

Zusätzlich zu den Feldern, die vom verknüpften Datenbankobjekt bereitgestellt werden, können Sie auch weitere Kriterien definieren. Hierzu haben Sie die Möglichkeit, eine beliebige Anzahl von Eingabeparametern anzugeben. Für jeden Eingabeparameter fügt das System automatisch eine zusätzliche Spalte in die Entscheidungstabelle ein, die für berechnete Spalten verwendet wird (siehe unteren Abschnitt). Zur Laufzeit fordert das System den Benutzer auf, für jeden Eingabeparameter einen Wert anzugeben. Die Eingabewerte werden dann mit den Werten verglichen, die in den jeweiligen Spalten der Entscheidungstabelle definiert wurden und beeinflussen somit das Auswertungsergebnis während der Regelverarbeitung.

Berechnete Spalten

Auf dieser Registerkarte verknüpfen Sie den DDBV mit einer Entscheidungstabelle. Je nach Definition der Entscheidungstabelle wählen Sie eine oder mehrere Ergebnisspalten der Tabelle aus, die in den DDBV eingefügt werden sollen. Zur Laufzeit berechnet die Entscheidungstabelle das Ergebnis für die Daten, die aus dem SAP HANA-Datenbankobjekt gelesen wurden sowie für die Eingabeparameter, die an den DDBV weitergegeben wurden.

Beispiel

Sie sind Inhaber eines Reisebüros und möchten eine Rabattstrategie für bestimmte Flugangebote festlegen, die von der Fluggesellschaft sowie bestimmten Flugverbindungen abhängen. In Ihrer SAP HANA-Datenbank haben Sie bereits eine Datenbanktabelle `SFLIGHT` angelegt.

1. Wählen Sie aus der Liste der in `SFLIGHT` verfügbaren Felder `PRICE` und `CARRID` aus und fügen Sie sie den Datenbankfeldern des DDBV hinzu.
2. Fügen Sie dem DDBV auf der Registerkarte **Eingabeparameter** folgende Datenobjekte hinzu: `CONNID`, `DATE` und `NUMBER`.
3. Ordnen Sie auf der Registerkarte **Berechnete Spalten** dem DDBV eine Entscheidungstabelle zu, die die Felder `CARRID` und `CONNID` als Bedingungsspalten und ein Datenobjekt des Typs Zahl als Ergebnisspalte verwendet. Das Ergebnisdatenobjekt wird verwendet, um die Rabatte für die verschiedenen Kombinationen aus Fluggesellschaften und Flugnummern zu definieren, die Sie in der Entscheidungstabelle festgelegt haben.

1.2.15 Funktion

Verwendung

Eine Funktion stellt die Regelschnittstelle in BRFplus dar und verbindet den Anwendungscode mit dem BRFplus-Code. Sie können vom System ein ABAP-Code-Snippet generieren lassen, das Ihnen die Integration von BRFplus-Funktionen in Ihre Anwendung erleichtert.

Funktionen haben einen Kontext und ein Ergebnis. Sie importieren den Kontext aus der aufrufenden Anwendung und geben die Kontextdaten zur weiteren Verarbeitung an den zugeordneten obersten Ausdruck oder Regelsatz weiter. Nach der Verarbeitung gibt die Funktion das berechnete Ergebnis an die aufrufende Anwendung zurück.

Funktionen

Modus der Operation

Eine Funktion kann für einen der folgenden Operationsmodi vorgesehen sein:

- Funktionsmodus
Im Funktionsmodus beginnt die Ausführung der Funktion mit dem zugeordneten obersten Ausdruck. Von diesem obersten Ausdruck aus kann die Verarbeitung eine beliebige Anzahl verschachtelter Unterausdrücke durchlaufen, bis ein Ergebnis zurückgegeben wird.
- Ereignismodus
Im Ereignismodus wird die Funktion mit einer Liste von Regelsätzen verknüpft, die gemäß ihrer Ausführungspriorität und ihrer Position in der Liste ausgeführt werden.
- Funktions- und Ereignismodus
Dieser Modus stellt eine Kombination aus den beiden oben genannten Modi dar. Zur Laufzeit beginnt die Funktion mit der Verarbeitung des zugeordneten obersten Ausdrucks. Sobald die Ausdrucksauswertung abgeschlossen ist, wird die Verarbeitung der Funktionen mit den dazugehörigen Regelsätzen fortgeführt.

Weitere Informationen finden Sie unter [Operationsmodi \[Seite 151\]](#).

Signatur

Die Funktionssignatur besteht aus zwei Teilen:

- Kontext
Der Kontext ist ein Container für Datenobjekte, die Sie der Funktion als Importparameter zuweisen können. Sie wählen die Kontextdatenobjekte einer Funktion gemäß den Anforderungen der aufrufenden Anwendung aus, von der die BRFplus-Funktion aufgerufen wird. Mit den Kontextdatenobjekten einer Funktion legen Sie auch den Objektumfang fest, auf den mit den bei der Ausführung der Funktion ausgewerteten Ausdrücken zugegriffen werden kann.
- Ergebnis
Das Ergebnisdatenobjekt gibt einen Ergebniswert zurück, der anhand der Ausdrücke, aus denen die Funktion besteht, berechnet wurde.

Hinweis

Bei Funktionen im Ergebnismodus wird das Ergebnisdatenobjekt automatisch an die vordefinierte **Actions**-Tabelle gesendet, in der die von den zugeordneten Regelsätzen ausgelösten Aktionen erfasst werden.

Regelsätze

Bei einer Funktion, für die Sie auf der Registerkarte *Eigenschaften* den Operationsmodus entweder auf *Ereignismodus* oder *Funktions- und Ereignismodus* festgelegt haben, erwartet das System, dass Sie einen oder mehrere Regelsätze zuweisen, die verarbeitet werden, wenn die Funktion ausgeführt wird. Neue Regelsätze können Sie direkt im Arbeitsbereich für die Bearbeitung einer Funktion anlegen und der Funktion zuweisen. Bei bereits vorhandenen Regelsätzen dagegen müssen Sie zuerst zu dem jeweiligen Regelsatz navigieren und die gewünschte Funktion als Auslöser des Regelsatzes definieren. Sobald diese Zuordnung erfolgt ist, wird der Regelsatz auf der Registerkarte *Zugeordnete Regelsätze* der Funktion aufgelistet.

Hinweis

Wenn Sie eine Funktion ausführen, die mehr als einen Regelsatz auslöst, beginnt das System mit der Ausführung des ersten Regelsatzes und wartet, bis dieser beendet ist, bevor es mit dem nächsten Regelsatz fortfährt usw. Dies ist eine Voraussetzung dafür, dass das System Parameter, die möglicherweise von einem Regelsatz geändert wurden, von einem Regelsatz zum anderen weitergeben kann. Es findet **keine** Parallelverarbeitung der zugeordneten Regelsätze statt. Es ist auch nicht möglich, dass ein zugeordneter Regelsatz die Ausführung eines nachfolgenden Regelsatzes verhindert, sobald die Funktion gestartet wurde.

Zugeordnete Regelsätze vs. Ausführbare Regelsätze

In der BRFplus-Workbench können Sie in Hinblick auf die Regelsätze zwischen zwei verschiedenen Ansichtsmodi wählen:

- *Zugeordnete Regelsätze*

In diesem Modus zeigt das System alle Regelsätze an, die der aktuellen Funktion zugeordnet sind, unabhängig vom Aktivierungsstatus. Das bedeutet, dass möglicherweise ein oder mehrere Regelsätze angezeigt werden, die aktuell nicht zur Ausführung bereit sind und die ohne weitere Benachrichtigung übersprungen werden, wenn die Funktion ausgelöst wird. Außerdem wird bei Auswahl dieses Ansichtsmodus die Drucktaste *Regelsatz anlegen* aktiviert, mit der Sie in einem einzigen Schritt einen neuen Regelsatz anlegen und der aktuellen Funktion zuordnen können.

- *Ausführbare Regelsätze*

In diesem Modus zeigt das System nur die Regelsätze an, die der aktuellen Funktion zugeordnet wurden und entweder aktuell aktiv sind oder für die eine ältere aktive Version verfügbar ist, die für die Ausführung des Regelsatzes verwendet werden kann.

Hinweis

Das System listet alle zugeordneten Regelsätze auf, die aus technischer Sicht ausführbar sind. Das heißt, ein aktuell deaktivierter Regelsatz wird trotzdem aufgelistet, wenn eine aktive Version verfügbar ist. Im Gegensatz zum Aktivierungsstatus handelt es sich beim Aktivieren oder Deaktivieren eines Regelsatzes um eine betriebswirtschaftliche Entscheidung und **nicht** um eine technische Frage. Zur Laufzeit löst die Funktion alle ausführbaren Regelsätze aus. Ein deaktivierter Regelsatz jedoch wird

sofort abgebrochen, ohne dass Aktionen ausgeführt werden, und die Kontrolle wird an die auslösende Funktion zurückgegeben.

Ausführungspriorität von Regelsätzen

Wenn einer Funktion mehr als ein Regelsatz zugeordnet ist, werden die Regelsätze in der Reihenfolge ausgelöst, in der sie in der Liste der zugeordneten Regelsätze der Funktion angezeigt werden. Sie können die Ausführungsreihenfolge für Regelsätze beeinflussen, indem Sie für jeden beteiligten Regelsatz eine Priorität festlegen. Die Priorität ist ein numerischer Wert zwischen 0 und 99, den Sie für einen Regelsatz festlegen können. Die Standardpriorität für einen neu angelegten Regelsatz ist 0, das bedeutet, dass für diesen Regelsatz keine Priorität festgelegt ist. Die Liste der zugeordneten Regelsätze wird nach dieser Priorität in aufsteigender Reihenfolge sortiert, so dass der Regelsatz mit dem niedrigsten Prioritätswert zuerst ausgeführt wird, gefolgt vom Regelsatz mit der zweitniedrigsten Priorität usw.

Sie legen die Ausführungspriorität für die Regelsätze auf dem Bild zur Regelsatzpflege der BRFplus-Workbench fest. Geben Sie dazu den gewünschten Wert als *Priorität* im Bereich *Details* ein.

Hinweis

Abweichend von der oben beschriebenen allgemeinen Regel (niedrigere Prioritätsnummer bedeutet höhere Priorität) behandelt das System den Standardwert 0 anders. Eine Priorität von 0 bedeutet, dass der betreffende Regelsatz überhaupt keine Priorität hat. Mit anderen Worten: Ein Regelsatz mit der Priorität 0 wird erst ausgelöst, nachdem alle anderen Regelsätze derselben Funktion, die eine Priorität > 0 aufweisen, ausgeführt wurden. Das System weist Sie visuell auf diese Tatsache hin, indem es in der Prioritätsspalte für einen Regelsatz mit Prioritätsstufe 0 "undefiniert" anzeigt, während bei allen anderen Prioritätsstufen der numerische Wert angezeigt wird.

Abhängigkeiten, Kardinalität und Sichtbarkeit

Die Beziehung zwischen einer Funktion und ihren zugeordneten Regelsätzen ist auf den ersten Blick nicht unbedingt offensichtlich. Um das zugrunde liegende Konzept besser zu verstehen, bedenken Sie bitte Folgendes:

- Funktion und Regelsatz werden in einer 1:n-Beziehung modelliert. Das heißt, einer Funktion kann eine beliebige Anzahl Regelsätze zugeordnet werden, während ein Regelsatz nur von genau einer Funktion ausgelöst werden kann.
- Die Beziehung zwischen Funktion und Regelsatz wird aus der Perspektive des Regelsatzes bestimmt, **nicht** umgekehrt. Dies führt dazu, dass das System verfolgt, welche Funktion einem bestimmten Regelsatz zugeordnet ist, aber nicht andersherum. Das bedeutet, die Funktion weiß **nicht**, welchen Regelsätzen sie als Auslöser zugeordnet ist. Dies gilt, obwohl die zugeordneten Regelsätze einer Funktion in der BRFplus-Workbench für eine Funktion aufgelistet werden.
- Die einseitige Perspektive vom Regelsatz zur Funktion spiegelt sich auch in folgenden Aspekten des Systemverhaltens wieder:
 - Wenn eine Funktion einem oder mehreren Regelsätzen als Auslöser zugeordnet wurde, werden diese Regelsätze im Verwendungsnachweis der Funktion angezeigt, aber **nicht** im Bereich *Verwendungen* des Repository-Baums in der BRFplus-Workbench.
 - Wenn Sie eine Funktion aktivieren und anfordern, dass auch die untergeordneten Objekte der Funktion bei Bedarf aktiviert werden sollen, berührt diese rekursive Aktivierung **nicht** die Regelsätze, denen die Funktion möglicherweise als Auslöser zugeordnet wurde. Dasselbe gilt für alle anderen funktionsbezogenen Aktivitäten, die rekursiv ausgeführt werden können, wie zum Beispiel Kopieren oder Prüfen einer Funktion. Auch hierbei werden die Regelsätze, denen die Funktion möglicherweise zugeordnet ist, **nicht** berücksichtigt.

Code-Generierung

BRFplus bietet eine integrierte Möglichkeit zur Code-Generierung, mit der Quellcode für so viele Regelkonstrukte wie möglich in einer Funktion kompiliert wird. Mithilfe des generierten Codes können BRFplus-Regeln deutlich schneller ausgeführt werden als im Interpretationsmodus.

Simulation

Sie können die Funktionsverarbeitung simulieren, um das Verhalten der Funktion in einer Sandbox-Umgebung zu testen. Dabei erhalten Sie einen detaillierten Einblick in den Systemstatus zu jedem einzelnen Schritt der Funktionsverarbeitung.

Trace

In BRFplus können Sie anfordern, dass das System ein Verarbeitungsprotokoll angelegt, um während der Funktionsausführung alle Verarbeitungsschritte zu verfolgen. Die Trace-Informationen werden im System gespeichert und können jederzeit eingesehen werden.

Generierungsservices

Für BRFplus-Funktionen können Sie verschiedene Services nutzen, mit denen automatisch Code und andere Objekte generiert werden, die zur Überbrückung der Lücke zwischen dem ABAP Backend System, in dem eine BRFplus-Funktion gehostet wird, und anderen Systemen oder Umgebungen, die auf BRFplus zugreifen möchten, erforderlich sind. Die verschiedenen verfügbaren Services sind auf unterschiedliche Verwendungsszenarios ausgerichtet. Diese können folgendermaßen kategorisiert werden:

- **Code-Vorlagen**
Eine Code-Vorlage ist ein vordefiniertes ABAP-Code-Snippet, mit dem Sie aus Ihrer Anwendung heraus eine BRFplus-Funktion im selben System aufrufen können.
- **Funktionsbaustein**
Mit einem generierten RFC-fähigen Funktionsbaustein können Sie eine BRFplus-Funktion aufrufen, die sich in einem Remote-Backend-System befindet. Dies ist dann nützlich, wenn Sie BRFplus und die aufrufende Anwendung nicht im selben System unterbringen können – sei es aus technischen Gründen (zum Beispiel unterschiedliche Releases) oder aus organisatorischen Gründen (zum Beispiel fehlende Berechtigung für Entwicklungen im Remote System).
- **Web-Service**
Mit einem generierten Web-Service können Sie eine BRFplus-Funktion aus einer Web-Anwendung aufrufen, die nicht auf ABAP basiert oder die Sie nur über ihre exponierte Serviceschnittstelle integrieren können. Technisch gesehen werden bei der Generierung von Web-Services die BRFplus-Komponente zur Generierung von Funktionsbausteinen wiederverwendet und zusätzliche Systemobjekte generiert, die für die Serviceaktivierung benötigt werden.

Code-Vorlagen

BRFplus wird mit vordefinierten Code-Vorlagen ausgeliefert, mit denen Sie vom System ein ABAP-Code-Snippet generieren lassen können. Nachdem Sie einige kleinere Anpassungen an diesem Snippet vorgenommen haben, können Sie es kopieren und in den Code-Abschnitt der Anwendung, von der aus Sie die BRFplus-Regelausführung aufrufen möchten, einfügen. Sie können wählen, ob die Funktion mit oder ohne Trace-Informationen aufgerufen wird, und Sie können entscheiden, ob die erläuternden Inline-Kommentare im Code sichtbar sind oder nicht.

Support für Web-Services und Funktionsbausteine

Für den Funktionsaufruf unterstützt BRFplus die Generierung von Web-Services sowie von RFC-fähigen Funktionsbausteinen. Daher können Sie die leistungsstarke Regelausführung einer BRFplus-Funktion in eine

serviceorientierte Software-Umgebung oder in ein Remote-ABAP-System, das BRFplus über RFC (Remote Function Call) aufruft, integrieren.

Erweiterte Prüfungen

Zusätzlich zu den standardmäßigen Konsistenzprüfungen, die für jedes BRFplus-Objekt über die Drucktaste **Prüfen** verfügbar sind, bietet das System weitere Prüfungen für Funktionen an:

- **Zugeordnete Regelsätze**
Bei dieser Prüfung analysiert das System die Richtigkeit und Konsistenz aller Regelsätze, die einer Funktion zugeordnet sind. Diese Prüfung ist nur für Funktionen verfügbar, die im **Ereignismodus** oder im **Funktions- und Ereignismodus** ausgeführt werden.
- **Lean-Trace-Bereitschaft**
Bei dieser Prüfung führt das System einen Drilldown durch die Hierarchie aller einer Funktion zugeordneten Objekte durch und ermittelt, ob Objekte vorhanden sind, für die die Versionierung nicht eingeschaltet wurde. Wenn ein solches Objekt gefunden wird, gilt die Funktion als **nicht** für das Tracing bereit, da das Tracing einer Funktion nur dann möglich ist, wenn alle involvierten Objekte der Versionsverwaltung unterliegen.
- **Inaktive Objekte**
Bei dieser Prüfung führt das System einen Drilldown durch die Hierarchie aller einer aktiven Funktion zugeordneten Objekte durch und ermittelt, ob Objekte vorhanden sind, die aktuell inaktiv sind. Für jedes inaktive Objekt sendet das System eine entsprechende Message, mit deren Hilfe Sie das Objekt analysieren können. Aufgrund technischer Einschränkungen der Systemarchitektur kann es jedoch in einigen Fällen unmöglich sein, den vollständigen Pfad zu dem inaktiven Objekt in der Message darzustellen. In einem solchen Fall müssen Sie das entsprechende Objekt manuell nachschlagen.

i Hinweis

Aufgrund der Strategie der kaskadierenden Aktivierung von BRFplus-Objekten kann eine Situation, in der ein aktives Objekt ein inaktives Objekt verwendet, normalerweise nicht auftreten. Folgende Szenarios sind jedoch gültig und können zu einer solchen Situation führen:

- Eine aktive Funktion, die im Funktionsmodus ausgeführt wird, verwendet einen aktiven obersten Ausdruck (oder eines von dessen abhängigen Objekten), der zu einem späteren Zeitpunkt geändert und deaktiviert wird. In diesem Fall bleibt die Funktion aktiv. Wenn die Funktion ausgeführt wird, wird die letzte aktive Version des obersten Ausdrucks verwendet und **nicht** die aktuelle inaktive Version. In diesem Szenario lässt sich das Problem durch eine Reaktivierung der Funktion lösen.
- Eine aktive Funktion, die im Ereignismodus ausgeführt wird, wurde einem inaktiven Regelsatz zugeordnet. In diesem Fall kann das Problem von der BRFplus-Aktivierungsstrategie nicht direkt erkannt werden, weil die Beziehung zwischen Funktionen und Regelsätzen intern so modelliert ist, dass nur ein Regelsatz seine zugeordnete Funktion kennt, während die Funktion ihre zugeordneten Regelsätze **nicht** kennt. Daher löst eine Reaktivierung der Funktion das Problem **nicht**. Vielmehr müssen Sie alle zugeordneten Regelsätze aktivieren.

Diese Prüfung ist insbesondere dann hilfreich, wenn die Regelmodellierung als eine gemeinsame Aufgabe mehrerer beteiligter Personen mit unterschiedlichen Zuständigkeiten eingerichtet ist. In einem solchen Arbeitsszenario kann es leicht vorkommen, dass Änderungen an abhängigen Objekten vorgenommen und Sie nicht darauf aufmerksam gemacht werden. Mit der Prüfung **Inaktive Objekte** können Sie solche unerwarteten Änderungen verfolgen.

Weitere Informationen

[Operationsmodi \[Seite 151\]](#)

[Funktionen bauen \[Seite 190\]](#)

[Web-Services und Funktionsbausteine generieren \[Seite 272\]](#)

[Ausdrücke \[Seite 80\]](#)

[Regelsatz \[Seite 168\]](#)

[Tracing \[Seite 152\]](#)

[Versionierung \[Seite 176\]](#)

1.2.15.1 Operationsmodi

Verwendung

BRFplus bietet drei Operationsmodi für die Verarbeitung einer Funktion: *Funktionsmodus*, *Ereignismodus* und *Funktions- und Ereignismodus*. Die verschiedenen Optionen haben folgende Merkmale:

- *Funktionsmodus*

Der Funktionsmodus setzt die Zuordnung eines obersten Ausdrucks zu einer Funktion voraus. Während der Verarbeitung löst die Funktion den obersten Ausdruck aus und wertet diesen aus. Der oberste Ausdruck kann eine Zahl von eingebetteten Ausdrücken nutzen, um ein Ergebnis zu finden. Das Ergebnis des obersten Ausdrucks ist das Ergebnis, das für die Funktion generiert wird. Es kann beispielsweise eine Entscheidungstabelle als oberster Ausdruck einer Funktion zugeordnet werden. Die Kontextdatenobjekte werden als Input für die Spalten der Entscheidungstabelle verwendet, die für die Formulierung der Bedingungen genutzt werden. Die Entscheidungstabelle bettet daraufhin einen Formelausdruck ein. Das Ergebnis der Entscheidungstabelle wird als Funktionsergebnis zurückgegeben. Generell wird der Funktionsmodus für einfachere Anwendungsfälle mit klarem Output der Funktion verwendet.

- *Ereignismodus*

Im Ereignismodus werden Regelsätze statt eines obersten Ausdrucks verwendet, um die Verarbeitung anzustoßen. Einer Funktion können mehrere Regelsätze zugeordnet sein. Jeder Regelsatz enthält eine Zahl von Regeln. Eine Regel besteht aus einer Bedingung und einem Aktionsbereich. Diese Regeln haben zusätzliche Einschränkungen, die durch Voraussetzungen, Zeitabhängigkeit und dem Status aktiviert/deaktiviert ermittelt werden, und werden angestoßen, wenn eine Funktion verarbeitet wird. Wenn eine Regelbedingung erfüllt ist, wird die Aktion ausgeführt. Eine Aktion kann die Werte im Kontext ändern oder weitere Aktivitäten initiieren, wie z. B. einen Workflow starten oder Fehlermeldungen in ein Logfile schreiben. Alle Aktionen, die während der Verarbeitung der Funktion ausgeführt werden, werden in der vordefinierten **Aktionstabelle** gesammelt, die im Ereignismodus automatisch als Ergebnisdatenobjekt der Funktion verwendet wird.

In Regelsätzen können ebenso weitere Ausdrücke vorhanden sein, wie z.B. Entscheidungstabellen- und Formelausdrücke. Generell wird der Ereignismodus für komplexere Anwendungsfälle verwendet sowie für Anwendungsfälle, für die kein Ergebnis gebraucht wird.

- *Funktions- und Ereignismodus*

Der Funktions- und Ereignismodus ist eine Kombination aus Funktionsmodus und Ereignismodus. Ähnlich zum Funktionsmodus ist die Zuordnung eines obersten Ausdrucks und eines Ergebnisdatenobjekts

obligatorisch. Darüber hinaus können Regelsätze verarbeitet werden, so dass Aktionen ausgeführt werden können und damit eventuell der Kontext verändert wird. Zur Laufzeit beginnt die Funktion mit der Verarbeitung des zugeordneten obersten Ausdrucks. Sobald die Ausdrucksauswertung abgeschlossen ist, wird die Verarbeitung der Funktionen mit den dazugehörigen Regelsätzen fortgeführt.

Weitere Informationen

[Funktion \[Seite 146\]](#)

[Regelsatz \[Seite 168\]](#)

1.2.15.2 Tracing

Verwendung

In BRFPplus können Sie anfordern, dass das System ein Verarbeitungsprotokoll angelegt, um während der Funktionsausführung alle Verarbeitungsschritte zu verfolgen. Die Trace-Informationen werden im System gespeichert und können jederzeit eingesehen werden. Die Trace-Funktionalität folgt dem Ansatz, nur ein Minimum an Daten im System zu schreiben. Dies wird erreicht durch die Protokollierung von Objektreferenzen und Datenänderungen anstatt explizit alle verfügbaren Informationen über ein bestimmtes Objekt zu einem bestimmten Zeitpunkt zu schreiben. Dank dieser Strategie bleibt die Datenbank klein und die Performance wird gesteigert. Diese Trace-Funktionalität können Sie steuern über den Parameter **IV_TRACE_MODE** der Methode **IF_FDT_FUNCTION~PROCESS**.

Das Trace-Werkzeug wird außerdem vom Simulationswerkzeug verwendet, das für Funktionen, Ausdrücke und Aktionen zur Verfügung steht. Hier umfasst Trace alle Verarbeitungsschritte, die vom System für die vorgegebenen Parameter in einem Simulationslauf ausgeführt werden.

Voraussetzungen

Eine der folgenden Voraussetzungen muss erfüllt sein:

- Alle für Trace vorgesehenen Objekte müssen versionsgeführt sind.
- Wenn **nicht** alle für Trace vorgesehenen Objekte versionsgeführt sind: Der Zeitstempel aller Objekte in einem Trace muss dem Zeitstempel des Trace entsprechen oder älter sein.

Weitere Informationen finden Sie im unteren Abschnitt *Trace und Versionierung*.

Funktionen

Lean Trace vs. Technischer Trace

Sie können entscheiden, in welchem Umfang detaillierte Informationen im System protokolliert werden sollen. Folgende Alternativen sind verfügbar:

- **Kein Trace**
Wenn die aufrufende Anwendung keinen Trace für eine Funktion fordert, führt das System die Funktion aus und gibt das berechnete Ergebnis zurück. Es wird kein Verarbeitungsschritt protokolliert. Dies führt natürlich zu besten Performance-Ergebnissen.
- **Lean Trace**
Im Lean Trace-Modus zeichnet das System nur die Schritte des Prozesses auf, die direkt zum Prozessabschluss führen. Wenn das Programm beispielsweise eine CASE-Anweisung enthält, bei der der Eingabewert mit fünf verschiedenen Werten geprüft werden kann und nur der letzte Versuch zu einem übereinstimmenden Ergebnis führt, meldet Trace nur den fünften Versuch, jedoch **nicht** die vier fehlgeschlagenen Versuche.
- **Technischer Trace**
Im Gegensatz zum Lean Trace wird beim technischen Trace jeder einzelne Prozessschritt aufgezeichnet, unabhängig davon, ob der Schritt zum erfolgreichen Abschluss der Verarbeitung führt oder nicht. Wenn z. B. eine Routine 20 verschiedene Schritte durchläuft, bis sich herausstellt, dass der Pfad falsch war, wird dies beim technischen Trace angezeigt, während der Lean Trace diese Schritte für den Benutzer ausblendet.

Trace und Versionierung

Die Strategie, nur ein Minimum an Trace-Daten zu schreiben setzt voraus, dass alle für Trace vorgesehenen BRFplus-Objekte versionsgeführt sind. Andernfalls kann das System die betroffenen Objekte nicht ordnungsgemäß identifizieren. Dies würde dann zu einer fehlerhaften Anzeige der Trace-Daten führen. Nur unter einer Sonderkondition ist die Versionierungsanforderung relaxiert: Wenn keines der für den Trace vorgesehenen Objekte seit der letzten Trace-Aufzeichnung geändert wurde, kann das System den Trace bewerten und anzeigen. Andernfalls zeigt das System eine Fehlermeldung an, sobald Sie versuchen, auf den Trace zuzugreifen.

Wenn eine Funktion mit der Anforderung aufgerufen wird, Trace-Daten zu protokollieren, jedoch Objekte betroffen sind, die **nicht** versionsgeführt sind, ist das Systemverhalten abhängig von der Einstellung des Parameters **IV_TRACE_MODE** der Methode **IF_FDT_FUNCTION~PROCESS**:

- **L** (für **L**ean Trace)
Das System führt die Funktion im Interpretationsmodus aus. In dieser Situation ist eine Code-Generierung nicht möglich.
- **R** (für Lean Trace **R**equired)
Das System führt die Funktion **nicht** aus und löst eine Ausnahme aus. Grund dafür ist der Konflikt einer Funktion, für die **kein** Trace ausgeführt werden kann und für die gleichzeitig ein Trace ausgeführt werden **muss**. Dieser Konflikt kann nicht gelöst werden, deshalb wird die Ausführung hier abgebrochen.

Zusätzlich zu den oben genannten Parameterwerten kann der Parameter **IV_TRACE_MODE** auch folgende Werte verwenden:

- **T** (für **T**echnischen Trace)
Mit diesem Parameter schreibt das System einen Verbose an und führt während der Ausführung der Funktion eine vollständige Protokollierung durch. Der technische Trace enthält Informationen über alle Verarbeitungsschritte der Funktion, einschließlich aller Schritte, die nicht direkt zum erfolgreichen Abschluss der Verarbeitung führen. Der Trace-Modus ist insbesondere für Code-Analysen und Debugging gedacht. Eine produktive Verwendung wird aus Performance-Gründen **nicht** empfohlen.
- **""** (leer)

Wird kein Wert an den Parameter weitergegeben, zeigt dies an, dass keine Trace-Informationen geschrieben werden, solange die Funktion ausgeführt wird.

Einschränkungen

In bestimmten Verwendungsfällen werden die im Trace aufgezeichneten Informationen über Parameter dynamisch an den Trace weitergeleitet. Aus technischen Gründen kann dies dazu führen, dass der Trace für Zeilen ausgeführt wird, deren Informationen die maximale Länge überschreiten und daher abgeschnitten werden. Wenn Sie jedoch den Trace im Kontext zum Simulationswerkzeug betrachten, so sind alle Informationen verfügbar, indem Sie die Maus über die abgeschnittene Zeile bewegen. Hier werden die abgeschnittenen Daten im tooltip angezeigt.

Weitere Informationen

[Versionierung \[Seite 176\]](#)

[Simulation \[Seite 154\]](#)

1.2.15.3 Simulation

Verwendung

Das Simulieren einer Funktion, eines Ausdrucks oder einer Aktion ist hilfreich, um das Verhalten des Objekts in einer Sandbox-Umgebung zu testen. Dabei erhalten Sie einen detaillierten Einblick in den Systemstatus zu jedem einzelnen Schritt der Funktionsverarbeitung.

Hinweis

Die Simulation ist ein Werkzeug, mit dem Sie auf einfache Art das Objektverhalten testen können. Allerdings muss hierzu das Objekt, das im Simulationsmodus ausgeführt werden soll, vorher gesichert werden. Änderungen, die Sie ohne zu sichern am Objekt vorgenommen haben, werden im Simulationsmodus nicht beachtet.

Voraussetzungen

Die Funktion sowie alle ihre (direkt und indirekt) referenzierten Objekte sind aktiv oder es ist mindestens eine ältere aktive Version im System verfügbar (die zweite Voraussetzung ist nur für die Simulation im Generierungsmodus gültig).

Funktionsumfang

Simulationsmodus

Das Simulationswerkzeug bietet folgende Operationsmodi:

- **Interpretationsmodus**
In diesem Modus verwendet das System bei der Simulation die entsprechenden Objekte, indem es einen Drill Down durch ihre Design-Zeit-Definition durchführt. Für eine kurze und einmalige Simulation kann dies schneller sein, als der Generierungsmodus. Die direkte Simulation von Ausdrücken oder Aktionen (also ohne Zuordnung zu einer Funktion) wird nur im Interpretationsmodus unterstützt.
- **Generierungsmodus**
Wenn für die Funktion kein Quelltext verfügbar ist, wird in diesem Modus zunächst ein Quelltext für die an der Simulation beteiligten Objekte generiert, indem Sie auf **Ausführen** klicken. Hierbei kommt es zu einer kurzen Verzögerung bei der Ausführung, weil die Generierung der Klassen und Methoden einige Zeit in Anspruch nimmt. Der Generierungsmodus hat dennoch folgende Vorteile gegenüber dem Interpretationsmodus:
 - Das Ausführen von generiertem Quelltext ist wesentlich schneller als das Interpretieren der Objektstruktur bei jedem Simulationsdurchlauf. Der Generierungsmodus ist vor allem empfehlenswert, wenn Sie beispielsweise eine Funktion simulieren wollen, die einen Einstiegspunkt in eine sehr komplizierte Struktur von Regelsätzen und Ausdrücken darstellt. Auch wenn Sie eine Funktion in vielen Iterationen mit unterschiedlichen Testdaten simulieren möchten, ist der Generierungsmodus dem Interpretationsmodus vorzuziehen. Außerdem ist die Generierung von Quelltext nur notwendig, wenn eines der beteiligten Objekte seit der letzten Generierung geändert wurde. Wenn keine Änderungen vorgenommen wurden, wird der vorhandene Quelltext sofort geladen und ausgeführt. Dies beschleunigt die Ausführung zusätzlich.
 - Beim Generierungsmodus wird eine Funktion auf dieselbe Weise simuliert, wie sie vom Produktivsystem zur Laufzeit ausgeführt werden würde. Daher gewährt der Generierungsmodus einen genauen Einblick in das Verhalten der Funktion im Produktivsystem und lässt Rückschlüsse auf die Leistungsfähigkeit oder mögliche Probleme zu.

Hinweis

Der Generierungsmodus kann nur bei Funktionen verwendet werden, **nicht** bei Ausdrücken oder Aktionen. Es können außerdem nur Funktionen im Generierungsmodus simuliert werden, die gegenwärtig aktiv sind oder für die eine aktive Version vorhanden ist.

- **Entfernt**
Bei diesem Modus wird die Funktion über das Aufrufen des generierten Quelltexts ausgeführt, der mithilfe des an ein verwaltetes System verteilt wurde. Außerdem greift die Wertheilfe für die Kontextparameter der Funktion auf die daran gebundenen Datenelemente im verwalteten System zu.

Hinweis

Die entfernte Simulation ist nur möglich, wenn die entsprechende Anwendung, zu der die Funktion gehört, einem verwalteten System zugeordnet ist. Außerdem wird die entfernte Simulation ebenfalls nur bei Funktionen unterstützt, **nicht** bei Ausdrücken oder Aktionen.

Verwaltetes System

In diesem Abschnitt können Sie festlegen, in welchem der verfügbaren verwalteten Systeme die Funktion simuliert werden soll. Das System, das gegenwärtig der aktuellen Anwendung zugeordnet ist, wird als

Standardsystem verwendet. Sie können die Simulation jedoch auch in jedem der verwalteten Systeme ausführen, die in der Auswahlliste verfügbar sind. In der Liste sind alle verwalteten Systeme enthalten, die im aktuellen System eingerichtet wurden.

Hinweis

Dieser Abschnitt erscheint nur, wenn die aktuelle Anwendung einem verwalteten System zugeordnet ist. Die Einstellungen, die Sie hier vornehmen, sind nur in Kombination mit dem Simulationsmodus *Entfernt* wirksam.

Aktionseinstellungen

Mit einer Simulation möchten Sie das Systemverhalten testen, ohne dass dies einen Einfluss auf Ihre produktiven Geschäftsprozesse hat. Das Ausführen einer Simulation als isolierter Prozess ist relativ sicher, solange lediglich Ausdrücke und Regeln simuliert werden. Schwierig kann es hingegen werden, wenn während der Simulation Aktionen ausgeführt werden. Aktionen können Systemaktivitäten auslösen, die über die Grenzen von BRFplus hinausgehen und möglicherweise einen beträchtlichen Einfluss auf Ihre aktuellen Geschäftsprozesse haben. Beispielsweise würden Sie unter normalen Umständen bei einer Simulation keine wirklichen E-Mails verschicken oder einen System-Workflow starten wollen.

Aufgrund dieser unerwünschten Nebenwirkungen werden Aktionen bei einem Simulationsdurchlauf standardmäßig **nur teilweise** ausgeführt. Alle geschäftskritischen Schritte werden dabei ausgelassen. Wenn Sie jedoch wissen möchten, ob eine Aktion fehlerfrei ausgeführt wird, können Sie über das Kennzeichen *Aktionen ausführen* das Ausführen von Aktionen während der Simulation erzwingen.

Hinweis

Um Missverständnisse zu vermeiden, folgt eine genaue Beschreibung dessen, was intern über das Kennzeichen *Aktionen ausführen* gesteuert wird:

- Bei der Simulation werden aus technischer Sicht Aktionen, die Teil einer Funktion, eines Ausdruck oder einer übergeordneten Aktion sind, **immer** ausgeführt. Dies geschieht unabhängig davon, ob das Kennzeichen *Aktionen ausführen* gesetzt ist oder nicht. Daher werden Aktionen immer im detaillierten Simulationsprotokoll angezeigt. Wenn Sie darüber hinaus eine Simulation debuggen, werden Sie feststellen, dass das System den Quellcode verarbeitet, der zu einer Aktion gehört.
- Wenn das Kennzeichen *Aktionen ausführen* auf `false` gesetzt ist, verarbeitet das System zwar den entsprechenden Quelltext für die Aktion, allerdings werden die ausstehenden Änderungen oder Aktivitäten dieser Aktion vor dem Verlassen der entsprechenden Quelltextstelle **nicht** ausgeführt. Mithilfe einer solchen Strategie können Sie das Systemverhalten während der Simulation weitestmöglich testen, ohne dabei auf die beschriebenen Nebeneffekte achten zu müssen.

Objekttypen

Eigentlich ist die Simulation ein Werkzeug für BRFplus-Funktionen. Sie können allerdings auch direkt einen bestimmten Ausdruck oder eine bestimmte Aktion ohne Zuordnung zu einer Funktion simulieren. Dies stellt eine einfache und schnelle Möglichkeit dar, das Laufzeitverhalten Ihrer Ausdrücke zu prüfen.

Hinweis

Aus Sicht des System wird eine Simulation immer für eine Funktion ausgeführt. Dies trifft auch dann zu, wenn Sie direkt einen Ausdruck oder eine Aktion simulieren möchten. In diesem Fall wird im Hintergrund eine temporäre Funktion angelegt. Das System ordnet den Ausdruck oder die Aktion dann dieser Funktion zu und simuliert sie anschließend.

Version

Wenn das zu simulierende Objekt der Versionsverwaltung unterliegt, können Sie auswählen, welche der verfügbaren Versionen simuliert werden soll. Sie können aus folgenden Optionen wählen:

- **Letzte aktive Version**
Bei dieser Option prüft das System die Versionsliste des Objekts auf die aktuellste aktivierte Version, beginnend bei der aktuellen Version. Wenn das Objekt nie aktiviert wurde (oder die einst aktive Version mittlerweile gelöscht wurde), wird beim Starten der Simulation mit dieser Option eine Fehlermeldung ausgegeben. Diese Option ist die Versions-StandardEinstellung für Simulationsläufe.
- **Neueste Version (inaktiv)**
Bei dieser Option wird die neueste Version des Objekts zur Simulation verwendet, selbst wenn sie derzeit inaktiv ist. Damit können Sie die Auswirkungen einer Änderung im Produktivsystem prüfen, ohne dabei die korrekte Regelverarbeitung laufender Geschäftstransaktionen zu gefährden. Sie müssen selbst entscheiden, ob das simulierte Verhalten produktiv gemacht werden soll oder nicht: Wenn ja, aktivieren Sie das Objekt. Wenn nein, stellen Sie für das Objekt die vorherige aktive Version wieder her (oder verlassen Sie das geänderte Objekt einfach, ohne zu sichern). Diese Option ist nur für Objekte verfügbar, deren aktuelle Version **nicht** aktiv ist.
- **Ab Datum/Uhrzeit**
Bei dieser Option können Sie einen Zeitpunkt definieren, für den Sie simulieren möchten, wie das Laufzeitverhalten eines bestimmten Objekts zu dieser Zeit gewesen wäre. Das System vergleicht die gegebenen Datums- und Zeitangaben mit den Zeitstempeln der Objektversionen in der Datenbank. Es sucht die nächstverfügbare aktive Version des Objekts, die vor dem angegebenen Zeitpunkt aktiviert wurde. Wenn eine solche Version vorhanden ist, wird sie zur Simulation verwendet. Andernfalls gibt das System eine Fehlermeldung aus. Diese Option ist nur für Objekte verfügbar, bei denen die Versionierung aktiviert ist.

Kontextdaten

Für die Simulation von Funktionen oder Ausdrücken werden zwei verschiedene Möglichkeiten angeboten, Werte an die Kontextdatenobjekte des zu simulierenden Objekts weiterzugeben.

Manuelle Dateneingabe

Bei der manuellen Dateneingabe können Sie jedes Kontextdatenobjekt (heruntergebrochen auf seine elementaren Bestandteile) in ein entsprechendes Eingabefeld eingeben. Nachdem Sie die gewünschten Werte eingegeben haben, können Sie die Simulation ausführen und das Ergebnis prüfen.

Testdaten-Import

In diesem Modus können Sie Testdaten für die Simulation importieren, die bereits in einer externen Tabellenkalkulationsdatei zusammengestellt wurden. Dies ist aus folgenden Gründen hilfreich:

- Wenn Sie mit einem komplexen Szenario arbeiten, das eine Vielzahl verschiedener Eingabedaten benötigt, können Sie sich mithilfe einer Tabellenkalkulationsdatei die benötigten Testdaten automatisch generieren lassen. Dadurch entfällt das lästige sowie fehleranfällige manuelle Eingeben großer Mengen an Daten.
- Wenn Sie die Simulation im Batch-Modus für alle Einträge in der Import-Datei ausführen, wird eine Ergebnisdatei erstellt, in der Sie das berechnete Ergebnis für jede Kombination der Eingabedaten finden können. Diese Ergebnisdatei dient Ihnen als Simulationsprotokoll. Außerdem stellt sie ein Mittel der Prozessdokumentation dar. Sie können damit feststellen, welche Kontextdateneinträge bereits in der Simulation getestet wurden.

Bevor Sie den Testdaten-Import das erste Mal auf eine bestimmte Funktion oder einen Ausdruck anwenden, empfehlen wir Ihnen, mit der Funktion [Importfeldstruktur anlegen](#) zu beginnen. Damit legt das System eine Tabellenkalkulationsdatei an, welche die Struktur der Kontextdaten genau so widerspiegelt, wie sie von der Import-Funktion erwartet wird. Anschließend können Sie diese Exportdatei mit Testdaten füllen und die Daten zur Simulation importieren.

Nachdem Sie Ihre Testdaten für den Import in die Simulation vorbereitet haben, können Sie aus zwei Methoden der Datenverarbeitung wählen:

- **Einzeldatenverarbeitung:** Bei dieser Methode wählen Sie immer einen der importierten Testdatensätze pro Simulationslauf aus. Sie können anschließend wählen zwischen der einfachen Option [Ausführen](#) (nach der Simulation wird lediglich das Ergebnis angezeigt) und der Option [Bearbeitungsschritte ausführen und anzeigen](#) (ein detailliertes Protokoll aller Berechnungsschritte wird angezeigt). Beachten Sie dass diese beiden Optionen nur verfügbar sind, wenn nicht mehr als ein Datensatz zur Simulation ausgewählt ist.
- **Batch-Verarbeitung:** Bei dieser Methode können Sie eine beliebige Anzahl an importierten Testdatensätzen auswählen und jeden dieser Datensätze simulieren. Gehen Sie hierzu folgendermaßen vor:
 1. Nehmen Sie im Simulationswerkzeug die gewünschten Anpassungen an der Auswahl vor und klicken Sie auf [Weiter](#).
 2. Wählen Sie im Abschnitt [Simulationsdaten](#) die Option [Testdaten importieren](#), wählen Sie Tabellenkalkulationsdatei aus, die die Daten enthält, und klicken Sie auf [Importieren](#).
 3. Nachdem Sie die Daten erfolgreich importiert haben, wählen Sie die Datensätze aus, die Sie für die Simulation verwenden möchten, und wählen Sie [Ergebnis nach Excel importieren](#).

Nach Abschluss der Simulation wird eine Tabellenkalkulationsdatei erzeugt, die für jeden ausgewählten Testdatensatz eine Zeile enthält, in der die Kontextdaten und das berechnete Ergebnis für diesen Kontext angegeben sind.

Weitere Informationen

1.2.15.4 Debugging

Use

You use the debugger to analyze the detailed behavior of a function and all the expressions and rules that are called during function execution. This is helpful, for example, when you need to investigate the function behavior because the function yields unexpected results, or when you want to see how a function behaves depending on changes to the input parameters.

Prerequisites

This feature is available only in systems where SAP NetWeaver Decision Service Management (DSM) is installed.

Features

General

The BRFplus rule debugger offers several features that you may know from source code debuggers in integrated development environments (IDE). These features comprise different execution modes, the definition of breakpoints and watchpoints, a file export option for the execution trace, and access to the values of the data objects during execution. The debugger UI looks very much like the UI used in the simulation tool and offers the same features for expanding or collapsing certain rule branches, as well as a search function.

Search

In the debugger, the search function is restricted to searching for names of objects that are present in the execution trace. It does **not** operate on the message texts that make up the content of the individual trace lines. The search is **not** case-sensitive, and you can use wildcards.

Example

Suppose that a debugging trace contains the following lines:

```
If test parameter do_temperature (DO_TEMPERATURE); value 10
Include condition 1: is between 8 and 18
Operand B evaluated to true
```

In this trace snippet, the DO_TEMPERATURE in the first line refers to a data object that is evaluated while executing a boolean expression. You can find this data object name, for example, by starting a search for “do_temp*”.

As opposed to this, searching for the word “test” that is present in the same line would yield no match. This is because the string “test” is only part of a trace message but does not refer to any BRFplus object involved in the execution.

Input Data Variants

Debugging rules as well as source code is an iterative task, and a repetitive one as well. It is normal to run through the same debugging sequence times and times again to see if the modifications you have applied to the rule objects in the meantime show the desired effect or not (where the latter case calls for the next iteration). Since debugging only makes sense when you supply realistic input data, this is a point that needs attention. Supplying input data is just a routine task, but it can be tedious in case of complex input data structures, and for systematic test runs, it is even crucial to supply well-defined input data to your functions that remains stable over time.

The BRFplus debugger helps you to accomplish this important task in a convenient way with user-defined input data variants. To use data variants, proceed as follows:

1. Start the debugger for a function you want to investigate.
2. On the debugger initial screen, there are input fields for all context data objects of the function. Enter the values as desired.
3. Choose **Variant** > **Save As**.
4. Enter a name and a description for the set of input values you have just defined, and choose **OK**.
The set of values is now saved as an input data variant for the current function.

Once you have defined one or more input data variants, you can pass them to the function for debugging by choosing **Variant** > **Select**.

Validity of Input Data Variants

When you select from, or maintain, the list of already defined input data variants, you may notice that the variants are flagged as either valid or not. The system keeps track of the formal consistency of the data stored in a variant and the current signature of the function you have chosen for debugging. If the system detects any inconsistencies (for example, number of input parameters does not match, input value cannot be converted to the parameter type that is currently in effect, etc.), the variant is flagged as invalid and cannot be used for debugging.

Debugging Modes

For debugging a BRFplus function, different modes are offered. They differ in the granularity of the execution steps, but **not** in the details of the trace that is displayed. That is, even if you decide to rush through an entire function with only one mouse click, the results are still as detailed as they are in single step mode. You can choose from the following debugging modes:

- **Single Step**
In this mode, the debugger steps into each executable object (including nested objects), shows the internal rule evaluations done in that object, and waits for you to trigger the next execution step.
- **Execute**
The same as in single step mode, but without stepping into any nested objects. However, if there are any breakpoints or watchpoints defined in the scope of a nested object, the debugger would still stop at that point and wait for your commands.
- **Run**
Runs through all executable objects at once and presents the function result immediately. However, if there are any breakpoints or watchpoints defined at any step for the objects involved, the debugger would stop at that point and wait for your commands.

i Note

Even in single step mode, the granularity of debugging rules is relatively coarse compared to what you may expect from a source code debugger in a common development environment. For technical reasons, the smallest entity that the debugger can execute is one BRFplus object per step. Even if this object happens to be a formula expression containing a 30 lines formula definition, or a boolean expression that consists of 25 distinct logical operations, the debugger always executes the whole object at once. There is no way of stepping into the internal structure of an object during debugging.

However, if an object contains other executable objects in its definition (that is, rules or expressions), the debugger offers the capability to step inside such a nested object and inspect the steps to be executed within that object.

Breakpoints and Watchpoints

You can use breakpoints and watchpoints to let the debugger interrupt the execution of rules and expressions in a function. The difference between breakpoints and watchpoints is as follows:

- **Breakpoint:** With a breakpoint, you mark an expression or rule where the debugger must stop execution. Whenever the debugger reaches a step that has been marked as a breakpoint, execution is stopped.
- **Watchpoint:** A watchpoint is the definition of a value check for a context data object. These checks are defined as a simple comparison operation. If, during a debugging session, the system determines that the value of a watched data object changes and, after the change, matches the watchpoint condition, execution stops at the processing step where the new value assignment has occurred.

Using Breakpoints

To define a breakpoint, proceed as follows:

1. In the debugging screen, choose [Breakpoint](#).
The system displays a dialog box where all expressions, actions, and rules are listed that belong to the function you want to debug.
2. Decide whether you want the debugger to stop before or after executing an expression (or both).
3. Choose [Run](#) to start the debugging session.
Every time the debugger reaches an expression where a breakpoint is defined, the execution is interrupted.

Note

You can also define breakpoints by clicking the gray rectangles in the [Breakpoint](#) column of the debugging trace. This is useful when you realize during execution that you should have set a breakpoint at an earlier position. This new breakpoint will then be effective for the next debugging run in the same debugging session.

Using Watchpoints

To define a watchpoint, proceed as follows:

1. In the debugging screen, choose [Watchpoint](#).
The system displays a dialog box where all context data objects of the function you want to debug are listed.
2. In the list of data objects, select the one for which you want to define a watchpoint and choose [Create Watchpoint](#).
The data object is transferred to the [Watchpoints](#) list. In the [Details](#) section at the bottom of the dialog box, there are additional controls to define the watchpoint.
3. In the [Details](#) section, choose a comparison operation from the [Condition](#) list.
4. If you have chosen the "=" or the "<>" operator, define a value against which the data object is compared.
5. Repeat the previous steps for other data objects, or for other value checks of the same data object.
6. Once you are done, choose [Save](#) to save the watchpoints.
7. Choose [Run](#) to start the debugging session.
Every time the system determines that after a value change, any of the watchpoint conditions are fulfilled, the execution is interrupted at the expression where the value change has occurred.

More Information

[Simulation \[page 154\]](#)

1.2.15.5 Working with Test Cases

Use

Once you have modeled a function and proved that it works as desired, you have reached an important milestone. However, experience shows that under certain conditions, the same function that used to be OK suddenly starts to behave in an unexpected manner. Often, this kind of unwanted surprise depends on the data that is passed to a function via its context data objects. When you encounter such situations, the desire for systematic testing in a controlled environment arises. This is where the test case administration tool of BRFplus comes in handy.

Prerequisites

This feature is available only in systems where SAP NetWeaver Decision Service Management (DSM) is installed.

Structure

A test case is always associated with a BRFplus function, and one function can have as many test cases assigned as you desire. Each test case represents a set of data that consists of the following parts:

- Data fields for all the context data objects of a function, broken down to element level. You use these fields to set up a well-defined set of input data combinations that you want to pass to the function during testing.
- Data fields for the result data object of a function, broken down to element level. You use these fields to define the result that, in your opinion, the function should yield, based on the input values that are stored in the same test case.
- Optional data fields corresponding to the context data objects of a function. These additional fields can be used to provide expected changes to the input data during function execution.
- Metadata like test case name and documentation, date and time of creation and change, name of the user who created or changed the test case, and so on.

Features

Expected Values

While the result of a function is always a value (elementary or complex) that depends on the calculations that have been done during execution, the result of a test case run is always a boolean value indicating whether the function result is identical with the expected result or not. Because of this, it is not enough to define only the input values of a test case. Rather, you also need to define a value, which, according to your expertise and business insight, should be returned by the function for the given input values.

To define the expected value of a function in a test case, you can choose between the following *Comparison* strategies:

- **Values:** Defines a specific value as test case result. If the result data object of the function is a structure or a table, you need to define a value for each elementary data object contained in the result data object.
- **Value Ranges:** The same as for *Values*, but instead of defining a specific value you can define a value range for each element of the result data object.
- **Exception:** You may define that the given test case will trigger an exception, rather than being properly executed.
- **Custom-Defined:** If none of the options mentioned above suits your needs, you can set up an ABAP OO class that implements the `IF_FDT_TEST_TOOL_EXITS` interface. In the implementation, you can program any kind of condition that you want to compare the test case result against. For example, you may compare the result against one of the following constellations:
 - More than one value range.
 - A set of discrete, non-consecutive values.
 - The result shall depend on additional conditions (for example, other values in a result table, or the status of certain system parameters).

In addition to the expected result, you can also decide to define an expected value for each of the context data objects of a function. This is mostly useful for functions running in event mode or mixed mode, where the input values passed to the context data objects may be overwritten during execution by variables defined in a ruleset. For functions running in functional mode, defining a context comparison can also be useful. This is true for functions making use of loop expressions or procedure calls, since these expression types can also be used to modify the context data.

Variants

Once you have defined the input and output data for a test case, you store this data in the test case by saving it. In addition, you can also save the context data of a test case as a variant, and you can reimport the variant data into a given test case, thereby overwriting the current data of that test case.

Since the purpose of a test case is to store a set of predefined data that can be used to parameterize the execution of a function, and a test case variant is doing just the same, the question arises why variants are needed at all for test cases. The answer is that when you focus on the test case administration tool alone, variants in fact do bring no additional value. However, this is different as soon as you take into account that variants are designed as a BRFplus-internal reuse component that is integrated into different tools that are offered for BRFplus functions: test case administration, simulation, debugger. Once you have defined a variant for a function, you can use that variant from inside of all these tools, thus making it much easier for you to conduct systematic, consistent error-solving activities around your functions.

Execution Modes

For executing a test case, the tool offers a variety of different execution modes:

Standard execution

Running a test case in one of the standard execution modes lets the system execute the function with the context defined in the test case. The function result is logged in a test case run, together with the input data.

- *Interpretation Mode*
Executes the function in the local system in interpretation mode.
- *Generation Mode*
Generates source code for the function (if not already available) and executes the generated code in the local system.
- *Remote (Productive Version)*
Executes the function in a managed system. By default, the tool offers the system that is assigned to the application, but you can as well select any other managed system. The system executes the productive version of the function that has been deployed to the managed system as a service. If the service is not available in the selected managed system, an error message is displayed.
- *Remote (Current Version)*
Executes the function in the managed system. By default, the tool offers the system that is assigned to the application, but you can as well select any other managed system. The system uses the current version of the function in the design time system, deploys it to a sandbox area in the managed system as a service, and executes it. The sandbox area is isolated from the productive area so that the productive version of the service remains untouched.

Execution with processing step recording

For the execution modes listed above (except for interpretation mode), you can also decide that during execution, not only the input values and the result is recorded in the log, but also the detailed list of processing steps that the function has run through during execution. This gives you precise insight into the function behavior at the particular moment when the test case was executed.

Test Case Consistency

You can modify a test case as desired, but only as long as the test case has not been executed. After a test case has been executed (that is, as soon as one or more runs have been recorded for a test case), it is not possible to apply any modifications to the test case anymore. This is to ensure that the test case runs remain formally consistent with the underlying test case.

If you need to modify a test case that has already been executed, you can accomplish this by choosing one of the following approaches:

- Delete the corresponding test case runs. As soon as there is no test case run remaining, you can modify the test case.
- Create a copy of the current test case and modify the test case copy as desired.

1.2.16 Partnermodulintegration

Erläutert, wie SAP-Partner BRFplus und DSM erweitern können, um das Verteilen von Services in Nicht-ABAP-Umgebungen zu unterstützen.

Verwendung

BRFplus sowie SAP NetWeaver Decision Service Management (DSM) wurden ursprünglich als Werkzeuge auf dem ABAP-Stack von SAP NetWeaver konzipiert. Folglich werden bei der Codegenerierung von BRFplus-

Funktionen ABAP-OO-Klassen und -Methoden generiert, die entweder explizit vom Benutzer oder implizit bei der ersten Ausführung einer bestimmten BRFplus-Funktion kompiliert werden.

Auch wenn in BRFplus weiterhin standardmäßig ABAP-Code generiert wird, lässt sich diese Einschränkung in Bezug auf die Programmiersprache über einen Anwendungs-Exit-Mechanismus aufheben, den Sie für eine beliebige BRFplus-Anwendung aktivieren können. Das Verwenden von Anwendungs-Exits zu diesem Zweck ist mit einem hohen Aufwand bei der Implementierung der relevanten Interface-Methoden verbunden. Es ist daher unwahrscheinlich, dass ein Kunde diesen Aufwand auf sich nimmt. Doch es könnte durchaus eine Option sein, ein Partnerunternehmen von SAP damit zu beauftragen, die notwendigen Schritte durchzuführen. Aus diesem Grund wird diese Funktionalität als Partnermodulintegration bezeichnet und die in solchen Szenarien verwendeten Anwendungsexits werden Partner-Exits genannt.

Funktionsumfang

Nachdem Sie die relevanten Interface-Methoden implementiert und die entsprechenden Exit-Klassen einer BRFplus-Anwendung zugeordnet haben, enthält die Partnermodulintegration folgende Funktionalitäten:

- **Codegenerierung:** Wenn Sie die Codegenerierung für eine BRFplus-Funktion explizit starten, wird Ihnen entsprechend den Generierungssprachen, die Sie implementiert haben ein Auswahlménü mit zusätzlichen Zielsprachen angezeigt.
- **Simulation:** Beim Simulieren einer Funktion können Sie wählen, ob Sie die Simulation in einer Nicht-ABAP-Umgebung starten möchten.
- **Codevorlage:** Die Codevorlage, die von BRFplus bereitgestellt wird, um Ihnen das Vervollständigen des generierten Codes für eine Funktion zu vereinfachen, ist auch für die zusätzlichen Zielsprachen verfügbar, die Sie implementiert haben.
- **DSM:** Unterstützung von nicht-ABAP-basierten verwalteten Systemen. Aus Benutzersicht wird dies umgesetzt, indem ein verwaltetes System angelegt wird, dass auf eine RFC-Destination referenziert, die nicht vom Typ 3 ist (also keine ABAP-Verbindung). Alle anderen Benutzeraktivitäten im Decision Service Manager bleiben unverändert.

Vorgehensweise

Um die Partnermodulintegration einzusetzen, müssen Sie eine ABAP-OO-Klasse anlegen, die das Interface `IF_FDT_DSM_PARTNER_EXIT` implementiert. Dieses Interface enthält die Methodensignaturen, die für die im oberen Abschnitt [\[\[unresolved text-ref: Funktionsumfang\]\]](#) erwähnten Funktionen benötigt werden. Es enthält außerdem eine Liste von zugehörigen booleschen Attributen (eines für jede Methode), die angeben, ob die jeweilige Methode implementiert wurde oder nicht. Ein zusätzliches Attribut `GV_IS_PRODUCTIVE` wird verwendet, um anzugeben, ob das gesamte Modul für den Produktivbetrieb freigegeben ist oder sich noch in Entwicklung befindet.

Hinweis

Wenn ein Fabrikkalender im Design-Zeit-System implementiert ist und Sie in der Implementierung des Partner-Exit-Interfaces auf diesen Kalender angewiesen sind, müssen Sie manuell sicherstellen, dass in allen verwalteten Systemen, an die Sie einen Service verteilen möchten, ein Fabrikkalender mit denselben Customizing-Einstellungen vorhanden ist. Andernfalls besteht die Möglichkeit, dass die verteilten Services

im verwalteten System ein unerwartetes Verhalten zeigen, das sich vom Verhalten im Design-Zeit-System unterscheidet. DSM verfügt über keine integrierten Routinen, die eine solche Inkonsistenz automatisch erkennen können.

Weitere Informationen

Um die Partnermodulintegration zu ermöglichen, müssen im BRFplus-Backend-System, in der BRFplus-Workbench sowie im Decision Service Manager bestimmte Schritte durchgeführt werden. Der folgende Abschnitt enthält die entsprechenden Informationen:

Öffnen Sie nach der Implementierung des oben erwähnten Interfaces den Decision Service Manager und legen Sie ein neues verwaltetes System an. Achten Sie dabei darauf, dass Sie im geöffneten Dialogfenster [System hinzufügen](#) den Cursor auf das Feld [RFC-Destination](#) positionieren und wählen Sie anschließend [Enter](#). Dadurch wird im Dialogfenster das Eingabefeld [Partner-Exit](#) eingeblendet. Geben Sie in dieses Feld den technischen Namen der ABAP-OO-Klasse ein, den Sie als Implementierung des Interfaces `IF_FDT_DSM_PARTNER_EXIT` angelegt haben.

Empfehlungen zur Interface-Implementierung

Wir empfehlen die Verwendung des statischen Konstruktors für die Definition aller Interface-Attribute, die mit dem Vorhandensein oder Nichtvorhandensein einer bestimmten Methodenimplementierung zusammenhängen.

Es empfiehlt sich außerdem, eine private statische Methode mit dem Namen `PING` zu implementieren, auch wenn dies technisch nicht notwendig ist. Sie können diese Methode verwenden, um zu prüfen, ob die beteiligten Systeme fehlerfrei miteinander verbunden sind. Rufen Sie zuerst die Methode `PING` in jeder der anderen Methoden auf und rufen Sie erst anschließend das verwaltete System auf. Dies hilft Ihnen dabei, alle programmatischen Vorkehrungen zu treffen, falls das verwaltete System nicht erreicht werden kann. Andernfalls kann es sein, dass ein Benutzer versucht, eine bestimmte Funktionalität auszuführen und keine Antwort vom System erhält. Dies führt üblicherweise zu einer Zeitüberschreitung seitens des Systems, deren Ursache schwierig zu finden sein kann.

Nachdem Sie das Interface implementiert haben und mit dem Testen Ihrer Implementierung beginnen möchten, müssen Sie sicherstellen, dass folgende Voraussetzungen erfüllt sind:

- Der Parameter `GV_IS_PRODUCTIVE` ist auf `true` gesetzt.
- Die Methode `GET_MODULE_METADATA` wurde implementiert.

Wenn mindestens eine dieser Voraussetzungen nicht erfüllt ist, wird nicht erkannt, dass ein Partnermodul im System vorhanden ist.

Weitere Informationen

[Funktion \[Seite 146\]](#)

[Simulation \[Seite 154\]](#)

1.2.17 Regel

Verwendung

In BRFplus wird ein Regelobjekt als technische Darstellung einer einfachen Geschäftsregel verwendet, die auf einen bestimmten Geschäftsablauf angewendet wird. Die Regeln können an die steigende Komplexität von Geschäftsabläufen angepasst werden, indem der Regel zusätzliche Aktionen oder Ausdrücke hinzugefügt werden und indem eine beliebige Zahl von Regeln zu einem Regelsatz zusammengefasst werden. Tatsächlich können Regeln nur im Kontext des Regelsatzes, dem sie zugeordnet sind, ausgelöst werden. Sie können **nicht** als eigenständige Objekte ausgelöst werden.

Funktionsumfang

Eine Regel besteht aus einer umfassenden `If`-Bedingung, die einmalig durch das System getestet wird, sobald die Regel ausgelöst wird. Basierend auf dem Ergebnis der Bedingung, führt die Regel einige Operationen aus. Die Bedingung muss einen `True`-Zweig haben, wohingegen der `False`-Zweig optional ist.

Hinweis

Zusätzlich zur integrierten Bedingung jeder Regel können Sie die Ausführung einer Regel von einer Vielzahl anderer Bedingungen abhängig machen, wie z. B. Regelaktivierung, Zeitabhängigkeit usw. Sie können diese zusätzlichen Bedingungen im Kontext des Regelsatzes, dem die Regel zugeordnet ist, kontrollieren.

In einer Regel können Sie auf Kontextdaten zugreifen und diese manipulieren. Dies mag auf den ersten Blick überraschend erscheinen, da die Regel an sich keinen eigenen Kontext hat. Der Zugriff auf Kontextdaten innerhalb einer Regel bedeutet vielmehr der Zugriff auf Kontextdaten, die durch den Regelsatz, dem die Regel zugeordnet ist, verfügbar sind.

Hinweis

Eine Regel kann mehreren Regelsätzen zugeordnet sein. Daher gewährt das System nur Zugriff auf diejenigen Kontextdatenobjekte, die allen referenzierten Regelsätzen gemeinsam ist (das heißt, die Schnittmenge aller Kontextdatenobjekte aller referenzierten Regelsätze). Dies ist aus Konsistenzgründen notwendig.

Die folgenden Operationstypen werden unterstützt:

- Auslösen einer Aktion
- Verarbeitung eines Ausdrucks Das Ergebnis des Ausdrucks kann zur Aktualisierung des Kontexts genutzt werden.
- Kontextdatenobjekte initialisieren oder aktualisieren

Weitere Informationen

[Regel anlegen \[Seite 243\]](#)

1.2.18 Regelsatz

Verwendung

Ein Regelsatz ist eine Sammlung von Regeln, die in einem bestimmten Geschäftsablauf verarbeitet werden. Er dient als Einstiegspunkt für die Regelverarbeitung und kann als Alternative zu einem einer Funktion zugeordneten obersten Ausdruck verwendet werden.

Funktionsumfang

Funktionszuweisung

Damit der Regelsatz verarbeitet werden kann, muss ihm eine Funktion zugeordnet sein. Die Vorgangseinstellungen der zugeordneten Funktion müssen entweder im *Ereignismodus* oder *Funktions- und Ereignismodus* sein. Ein Regelsatz kann nur einer Funktion zugeordnet werden, wohingegen eine Funktion mehrere zugeordnete Regelsätze haben kann.

Wenn einer Funktion mehrere Regelsätze zugeordnet wurden, kann die Ausführungsreihenfolge der verschiedenen Regelsätze wie nachfolgend beschrieben beeinflusst werden. Um die Ausführungsreihenfolge zu steuern, ordnen Sie einem Regelsatz eine Priorität zu, der zur Laufzeit ausgewertet wird, wenn die zugeordnete Funktion ausgeführt wird. Um die Priorität für den Regelsatz zu setzen, öffnen Sie den Abschnitt *Details* und geben Sie den gewünschten Wert (von 00 bis 99) im Feld *Priorität* ein. Weitere Informationen finden Sie unter [Funktion \[Seite 146\]](#) im Abschnitt *Ausführungspriorität von Regelsätzen*.

Hinweis

In der BRFplus-Workbench liefert die Funktion *Verwendung* kein Ergebnis für Regelsätze. In der Kopfzeile des *Regelsatzes* können Sie jedoch einfach nachsehen, welcher Funktion ein bestimmter Regelsatz zugeordnet ist.

In früheren Releases von BRFplus war es ebenso möglich, einer Funktion eine Regel als Trigger zuzuordnen, anstatt die Funktion einem Regelsatz zuzuordnen. Dies wird nun nicht mehr unterstützt. Wenn Sie jedoch alte Anwendungen verwenden, die die individuelle Zuordnung von Regeln und Funktionen nutzen, kann das System diese Regeln weiterhin verarbeiten.

Regelzuordnung

Die in einem Regelsatz enthaltenen Regeln können entweder benannt oder unbenannt sein:

- Benannte Regeln sind Verweise auf Regelobjekte, die Sie in Ihrer BRFplus-Anwendung definiert haben. Diese Regeln sind eigenständige Objekte und können von verschiedenen Regelsätzen wiederverwendet werden.
- Unbenannte Regeln werden im Geltungsbereich eines bestimmten Regelsatzes definiert und sind für andere Objekte nicht erreichbar. Unbenannte Regeln enthalten entweder einen Verweis auf eine benannte Regel oder eine sofortige Definition der auszuführenden Aufgaben.

Hinweis

Wenn Sie eine Regel innerhalb eines Regelsatzes anlegen und Text in das Feld *Beschreibung* eingeben, gilt die Regeln weiterhin als unbenanntes Objekt. Das System kopiert den in das regelspezifische Feld *Beschreibung* des Regelsatzes eingegebenen Text in das Feld *Text* des Regelobjekts. Dabei bleibt das Feld *Name* der Regel leer.

Zusätzlich zur Trigger-Funktion definiert ein Regelsatz auch die Vorbedingungen und den Gültigkeitszeitraum jeder Regel. Die Vorbedingungen und der Gültigkeitszeitraum legen fest, ob eine Regel ausgelöst wird oder nicht. Vor diesem Hintergrund liefert das System eine Zahl von Bedingungsleveln, um festzulegen, ob die in der Regel definierten Aktivitäten ausgeführt werden sollen oder nicht:

- Die zugeordnete Funktion wird aufgerufen.
- Der Regelsatz wird aktiviert.
- Die Vorbedingung des Regelsatzes ist erfüllt.
- Die regelsatzspezifische Regelbedingung ist erfüllt.
- Die Regel ist aktiviert.
- Das aktuelle Datum und die aktuelle Uhrzeit fallen in den Gültigkeitszeitraum der Regel.
- Die Vorbedingung für eine in der Regel definierte Aktivität ist erfüllt.

Variablen

Wenn Sie zusätzliche Daten zwischen der einem Regelsatz zugeordneten Funktion und den Regeln innerhalb eines Regelsatzes austauschen möchten, können Sie die Signatur des Regelsatzes um Variablen erweitern. Sie können jedes in der Anwendung verfügbare Datenobjekt als Variable verwenden. Im Regelsatz werden alle Variablen wie Kontextelemente behandelt und daher können kontextbezogene Aktionen, wie *Wert dem Kontext zuordnen* oder *Kontext initialisieren* darauf zugreifen. Die Verbindung von Kontextdaten und Variablen wird ebenso deutlich, wenn man die Funktion *Kontextübersicht* betrachtet, die für einen Regelsatz nicht nur die Kontextdatenobjekte der zugeordneten Funktionen anzeigt, sondern auch die Regelsatzvariablen.

Hinweis

Obwohl Variablen analog zu Kontextdatenobjekten verwendet werden können ist es **nicht** möglich, eine Regelsatz-Vorbedingung basierend auf Variablen zu definieren. Eine Regelsatz-Vorbedingung muss zwingend auf ein Kontextdatenobjekt verweisen. Im Gegensatz dazu ist es jedoch möglich, eine Bedingung für Regelsatzunterbrechung basierend auf Variablen zu definieren.

Verzögerte Regelsatzverarbeitung

Wie zuvor beschrieben, können Sie die Bedingungen zur Regelsatzunterbrechung definieren, die die Regelsatzverarbeitung an einem definierten Punkt im Prozessablauf stoppen, wenn eine Bedingung erfüllt wird. Das Standardverhalten hier ist, dass alle Datenänderungen, die während der Verarbeitung der vorhergehenden Regeln angewendet wurden, im System verbleiben, während potenzielle Änderungen, die sich aus der Verarbeitung der übrigen Regeln ergeben würden, natürlich fehlen. Je nach Geschäftsablauf kann dies zu Dateninkonsistenz führen.

Um diesem potentiellen Problem vorzubeugen, können Sie festlegen, dass ein Regelsatz an dem Punkt neu gestartet wird, an dem die Verarbeitung bei der letzten Ausführung gestoppt wurde. Mit dieser Einstellung können Sie einen unterbrochenen Verarbeitungslauf fortsetzen, so dass die aus der Regelverarbeitung resultierenden Datenänderungen vervollständigt werden können. Dies ist hilfreich, wenn die für einen Regelsatz definierten Bedingungen zur Regelsatzunterbrechung in Ihrem bestimmten Geschäftsablauf erfüllt

werden können, das heißt, wenn eine erfüllte Bedingung zur Regelsatzunterbrechung eher die Regel ist, statt die Ausnahme.

Beispiel

Sie haben einen Regelsatz zur Berechnung des Stromverbrauchs von Wohnungseigentümern in einem Wohnhaus definiert. Jedoch übermitteln nicht alle Wohnungseigentümer den Stromzählerstand innerhalb der gesetzten Frist. In so einem Fall können Sie einen Regelsatz starten, um den Stromverbrauch aufgrund der bereits übermittelten Stromzählerstände zu berechnen. Wenn jedoch zusätzliche Regeln für eine statistische Analyse des Stromverbrauchs im Gebäude basierend auf den Daten der Wohnparteien im Regelsatz vorhanden sind, müssen für diese Regeln vollständige Daten eingegeben werden.

Bei einer verzögerten Regelsatzverarbeitung würde eine Bedingung zur Regelsatzunterbrechung vor den Statistik-Regeln platziert werden, um zu prüfen, ob die erforderlichen Daten vorhanden sind. Falls dies nicht getan wird, stoppt die Regelverarbeitung an diesem Punkt und kann zu einem späteren Zeitpunkt neu gestartet werden, wenn die erforderlichen Daten vorhanden sind, so dass die übrigen Regeln verarbeitet werden können.

Um die verzögerte Regelsatzverarbeitung für eine BRFplus-Anwendung zu aktivieren, gehen Sie wie folgt vor:

1. Navigieren Sie zu der Anwendung, zu der der Regelsatz gehört.
2. Auf der Registerkarte *Sonstiges* wählen Sie *Wiederaufnahme von Regelsatz aktiv*.

Sobald Sie diese vorbereitenden Schritte auf Anwendungsebene ausgeführt haben, können Sie nun die zusätzlichen Funktionen nutzen, die für die verzögerte Regelsatzverarbeitung verfügbar sind. Sie können auf diese Funktionen in der BRFplus-Workbench in Form von zusätzlichen Einstellungen für Exit-Bedingungen in einem Regelsatz zugreifen.

Hinweis

Um das Konzept der verzögerten Regelsatzverarbeitung vollständig zu verstehen, ist es wichtig, das Folgende zu berücksichtigen:

- Die verzögerte Regelverarbeitung kann auf Anwendungsebene nur für die Regelsätze gesteuert werden, die zu einer Anwendung gehören. Es ist **nicht** möglich, die verzögerte Regelverarbeitung für ausgewählte Regelsätze in einer Anwendung zu steuern.
- Die verzögerte Regelverarbeitung kann nur mittels entsprechender Programmaufrufe an das BRFplus-API genutzt werden. Die interaktive Anwendung dieser Funktion im BRFplus-Workbench-UI wird **nicht** unterstützt.
- Die verzögerte Verarbeitung wird über die folgenden API-Objekte unterstützt:

Tabelle 39:

Name	Verwendung
IF_FDT_RS_INTERRUPT_HANDLER	Stellt Methoden zur Verfügung, mit denen Sie Systemunterbrechungen abrufen können, die aufgrund einer Bedingung zur Regelsatzunterbrechung angelegt wurden.
CL_FDT_RS_INTERRUPT_HANDLER	Implementiert Interface IF_FDT_RS_INTERRUPT_HANDLER.

Name	Verwendung
IF_FDT_RS_INTERRUPT	Stellt Methoden zur Verfügung, mit denen Sie eine Unterbrechungsinstanz auf die Datenbank schreiben, den dazugehörigen Regelsatz neu starten und die gesicherten Unterbrechungsdaten von der Datenbank löschen können.

Weitere Informationen

[Regelsatz anlegen \[Seite 245\]](#)

[Funktion \[Seite 146\]](#)

[Operationsmodi \[Seite 151\]](#)

[Regel \[Seite 167\]](#)




1.2.19 Katalog

Verwendung

Ein Katalog vereinfacht den Pflegeprozess von Regeln für den Anwendungsbenutzer, da das Repository ausgeblendet und ausschließlich relevante Objekte angezeigt werden, die für die Pflege von Geschäftsregeln in einem bestimmten Geschäftsszenario erforderlich sind. Der Katalog dient als Filter für alle im Repository vorhandenen Objekte.

Für Anwendungsbenutzer zeigt ein Katalog in der Regel nur die Elemente einer BRFplus-Anwendung, die für geschäftsrelevante Anpassungen oder Erweiterungen wie z. B. Entscheidungstabellen erforderlich sind. Alle verbleibenden Elemente, wie Funktionen, Datenobjekte, Konstanten, Formeln oder Aktionen sind nicht in einem solchen geschäftsorientierten Katalog enthalten und für den Benutzer nicht sichtbar.

➔ Empfehlung

SAP empfiehlt, für Anwendungsbenutzer von BRFplus alle Sichten bis auf die Katalogsicht in der BRFplus-Workbench zu deaktivieren, um bei der Verwendung von Katalogen optimal von der Vereinfachung bei Prozessen zu profitieren. Wählen Sie dazu , danach  **Sichten** , und deaktivieren Sie alle Sichten bis auf **Katalog**.

Funktionsumfang

Objekttypen

Für jeden Katalog können Sie eine Reihe von Objekttypen definieren, die diesem Katalog hinzugefügt werden können. Mit einer begrenzten Anzahl von Objekttypen stellen Sie sicher, dass nur die Objekte zum Katalog hinzugefügt werden, die wichtig für die Zielgruppe sind. So sehen Benutzer keine hochtechnischen Objekte oder Objekte, die aus Unternehmenssicht nicht direkt zur Lösung eines bestimmten Anwendungsfalls beitragen.

Hinweis

Durch eine Eingrenzung der Liste der erlaubten Objekttypen sorgen Sie aber nicht nur dafür, dass ein Benutzer keine Objekte **hinzufügen** kann, die nicht in der Liste enthalten sind. Wenn der Katalog bereits Objekte eines Typs enthält, die Sie aus der Liste der erlaubten Objekttypen entfernen, interpretiert das System dies als Verstoß gegen die Konsistenz des Katalogs. In diesem Fall können Sie entweder den zuvor entfernten Objekttyp erneut zur Liste hinzufügen oder die Objekte, die über den entfernten Objekttyp verfügen, aus dem Katalog löschen.

Unter anderem enthält die Liste der erlaubten Objekttypen den Typ *Ausdruck*. Aus technischen Gründen bezieht sich dieser Objekttyp **nicht** ausschließlich auf Ausdrücke der BRFplus-Ausdruckstypen wie Entscheidungstabellen, Formeln usw., sondern auch auf Aktionen und Regeln.

Knotentypen

Ein Katalog kann eine beliebige Anzahl an Ordner-, Objekt- und Katalogknoten enthalten. Die einzelnen Knotentypen:

- **Ordnerknoten**
Mithilfe von Ordnerknoten legen Sie im Katalog eine Ordnerstruktur an. In den Ordnern können Sie Objekte nach verschiedenen Anwendungsfällen oder anderen für Anwendungsbenutzer hilfreichen Kriterien gruppieren.
- **Objektknoten**
Ein Objektknoten ist ein Katalogelement, welches eine Referenz auf ein BRFplus-Objekt wie einen Ausdruck oder eine Funktion enthält.
- **Katalogknoten**
Mit einem Katalogknoten richten Sie eine Referenz auf einen Knoten in einem anderen Katalog ein und können so Teile eines bereits vorhandenen Katalogs in einen anderen einbetten. Mit diesem Knotentyp können Sie bereits definierte Objektgruppen, die unter Umständen für andere Katalogszenarien relevant sind, wiederverwenden.

Objektknoten

Am einfachsten füllen Sie einen Katalog, indem Sie auf vorhandene BRFplus-Objekte referenzieren oder ein neues Objekt anlegen und gleichzeitig eine Referenz aus dem Katalog dieses Objekts einrichten. Sie können Referenzen auf Objekte aller Arten zu einem Katalog hinzufügen, solange der Objekttyp für die Verwendung in diesem Katalog zugelassen ist.

Wenn Sie einem Katalog eine Objektreferenz hinzufügen, kopiert das System den Kurztext des Objekts zu Anzeigenzwecken in den Katalog. Dieses Standardverhalten können Sie wie folgt ändern:

- **Überschreiben**
Der in den Katalog kopierte Objekttext gehört technisch gesehen zur Referenz, die vom Katalog auf das referenzierte Objekt weist. Folglich können Sie diesen Text überschreiben und einen Text verfassen, der unter Umständen besser Ihren Anforderungen für diesen Katalog entspricht. Dies hat keine Auswirkungen auf den Originaltext des referenzierten Objekts.
- **Textbindung**

Über die Option **Text binden** legen Sie fest, dass im Katalog immer der Originalkurztext des Objekts angezeigt wird. Eine Änderung des Objekttexts ist im Katalog sofort sichtbar. Mit dieser Einstellung legen Sie fest, dass das Textfeld im Katalog schreibgeschützt ist, um unerwünschte Änderungen am referenzierten Objekt zu vermeiden.

Bei aktivierter Textbindung aktiviert das System einen zusätzlichen Fallback-Mechanismus, um sicherzustellen, dass immer ein ordnungsgemäßer Text angezeigt wird: Wenn für das referenzierte Objekt nur das lange **Text**attribut gepflegt und kein **Kurztext** eingegeben wurde, wird das **Text**attribut automatisch an den Katalog gebunden. In diesem Fall wird der Objekttext aufgrund einer möglichen Längenbeschränkung für das Feld unter Umständen nicht vollständig im Katalog angezeigt.

Platzhalterknoten

Beim Anlegen von Objekt- oder Katalogknoten gehen Sie standardmäßig wie folgt vor: Sie legen einen neuen Knoten an und weisen diesem einen Objekt- oder Katalogknoten zu, indem Sie ein vorhandenes Objekt auswählen oder ein neues anlegen. Es kann jedoch vorkommen, dass Sie aus Unternehmenssicht eine klare Vorstellung der relevanten Objekte haben, die technischen Details aber noch nicht geklärt sind.

In diesen Fällen können Sie Platzhalterknoten einsetzen, um die technische Zuweisung zu einem späteren Zeitpunkt vorzunehmen. Hier definieren Sie eine Reihe von Objekten für einen Katalog aus rein semantischer Sicht und treffen keine technischen Entscheidungen. Im weiteren Verlauf des Projekts, nachdem die technischen Anforderungen geklärt wurden, rufen Sie die Platzhalterknoten in Ihrem Katalog auf und weisen Ihnen die vereinbarten Elemente zu.

Ordner- und Knotenattribute

In einem Katalog können Sie einzelne Sets von Attributen für Ordner, Objekte und Katalogknoten definieren. Sobald Sie die Attribute definiert haben, können Sie jedem Attribut für jeden Knoten oder Ordner Ihres Katalogs Werte zuweisen. Diese Attributwerte werden **nicht** von BRFplus verwendet. Es ist unter Umständen nützlich, die im Katalog hinterlegten Attributwerte aus der aufrufenden Anwendung im Backend-System aufzurufen. Sie können auch Verwaltungsinformationen auf Knotenebene hinterlegen, die dann zu Referenzzwecken in der BRFplus-Workbench aufgerufen werden können.

Beispiel

Für Ordnerknoten in einem Katalog definieren Sie zwei Attribute:

- **Anwendungsfall**
In dieses Textfeld geben Sie eine kurze Beschreibung des Anwendungsfalls ein, zu dem die Objekte in einem Ordner gehören.
- **Wiederverwendet**
In diesem Booleschen Feld geben Sie an, dass ein bestimmter Ordner von einem bestimmten Knoten eines anderen Katalogs referenziert wird. Dies unterstützt Administratoren bei der Entscheidung, ob sie einen Ordner sicher entfernen können.

Ein Attribut wählen Sie aus einem Subset der elementaren Datenobjekte aus, die Sie über Ihre Anwendung aufrufen. Dieses Subset setzt sich zusammen aus den folgenden Datentypen:

- Boolescher Wert
- Zahl
- Text

Versionierungsverhalten

Im Gegensatz zu allen anderen BRFplus-Objekttypen ist die Versionierungseinstellung für neu angelegte Katalogobjekte immer deaktiviert. Dies gilt auch, wenn die Standardeinstellung für die Versionierung in der Anwendung aktiviert ist. Auf Wunsch können Sie die Versionierung manuell für einen Katalog aktivieren.

Hinweis

Grund für dieses besondere Versionierungsverhalten bei Katalogobjekten ist, dass Änderungen an der Katalogstruktur im Grunde nie relevant für das Geschäft sind. Deshalb ist es nicht erforderlich, die verschiedenen Versionen eines Katalogs zu erfassen. Zudem werden Kataloge häufig geändert, sodass eine Versionierung zu einem hohen Speicherverbrauch führen kann.

Weitere Informationen

[Mit Katalogen arbeiten \[Seite 247\]](#)

[Workbench-Personalisierung \[Seite 12\]](#)

1.2.20 Objektfilter

Verwendung

Mit einem Objektfilter definieren Sie ein Subset von Objekttypen, die zur Design-Zeit in einer BRFplus-Anwendung verfügbar sein sollen. So stellen Sie sicher, dass für Geschäftsregeln aller Art in einer Anwendung nur die Objekttypen verwendet werden können, die dem Filter entsprechen. Das Definieren von Objektfiltern und das Zuweisen der Filter zu Benutzerprofilen ist eine typische Administratoraktivität.

Beispiel

In einer produktiven BRFplus-Anwendung haben Sie eine Reihe benutzerdefinierter Ausdruckstypen importiert, um den Funktionsumfang von BRFplus zu erweitern. Diese Ausdruckstypen sind jedoch teilweise noch Prototypen und nicht für die produktive Verwendung freigegeben. Sie können Ihre Anwendung vor unerwünschten Auswirkungen durch diese Prototyp-Ausdrücke schützen, indem Sie einen Filter definieren. So können Benutzer nur die Ausdruckstypen instanziiieren, die von SAP geliefert wurden. Alle benutzerdefinierten Ausdruckstypen werden dann für die Benutzer ausgeblendet, auf die der Filter angewendet wird.

Funktionsumfang

Allgemein

Objektfilter werden zu administrativen Zwecken zur Design-Zeit verwendet. Hauptsächlich wird mit ihnen gesteuert, welche Objekttypen für die Einrichtung einer Geschäftsregel verfügbar sind. Folglich ist die gültige

Zugriffsebene für andere Objekttypen nicht für die Verwendung von Objektfiltern relevant. Auch wenn Sie einen Objektfilter im Kontext einer bestimmten Anwendung anlegen, so ist seine Verwendung nicht auf diese Anwendung beschränkt. Objektfilter sind also anwendungsübergreifende Objekte.

Die Einstellungen von Objektfiltern wirken sich im Wesentlichen auf zwei Hauptbereiche in der BRFplus-Workbench aus:

- **Neue Objekte anlegen**
Wenn einer Anwendung ein Filter zugewiesen wurde, können Sie nur Objekte mit Typen anlegen, die in der Filterdefinition erlaubt sind. Objekttypen, die herausgefiltert wurden, sind beim Anlegen nicht verfügbar.
- **Objekttypen im Dialogfenster *Objektabfrage* auswählen**
Wenn einer Anwendung ein Filter zugewiesen wurde, können Sie nur Objekte mit Typen auswählen, die in der Filterdefinition erlaubt sind. Objekttypen, die herausgefiltert wurden, sind beim Auswählen nicht verfügbar.

Hinweis

Die Auswirkung einer Filterzuweisung beschränkt sich auf das Anlegen neuer oder das Auswählen bereits vorhandener Objekte im Dialogfenster *Objektabfrage*. Auf alle Objekte, die bereits in der Anwendung vorhanden sind, hat ein Filter **keine** Auswirkungen. Sie werden weiterhin angezeigt und können produktiv verwendet werden. Dies gilt auch für Objekte, deren Typen von einem Filter ausgeschlossen sind.

Details

Anwendungen

Auf der Registerkarte *Anwendungen* wählen Sie die Anwendungen aus, für die der Filter beim Auswählen von Objekten im Dialogfenster *Objektabfrage* gelten soll.

Objekttypen

Auf der Registerkarte *Objekttypen* finden Sie eine Liste von Objekttypen, die systemweit verfügbar sind. Standardmäßig sind alle verfügbaren Objekttypen erlaubt. Sie definieren den Filter, indem Sie bestimmte Objekttypen deaktivieren – entsprechend dem Anwendungsfall, den Sie modellieren möchten.

Hinweis

In dieser Liste sind alle Objekttypen enthalten, die im System vorhanden sind. Dazu gehören auch benutzerdefinierte Ausdrucksarten, unabhängig von der ursprünglichen Anwendung oder der Zugriffsebene. Wenn Sie also die Verwendung eines bestimmten Objekttyps erlauben, gibt es keine Garantie, dass diese Entscheidung sichtbare Auswirkungen in einer bestimmten Anwendung hat.

Einige Objekttypen sind immer erlaubt und können nicht deaktiviert werden. Dies gilt z. B. für Datenobjekte aller Typen.

Filterzuweisung

Nachdem Sie die Filtereinstellungen definiert und den Filter aktiviert haben, müssen Sie ihn einer oder mehreren Anwendungen zuweisen, damit die Einstellungen wirksam werden. Diesen Schritt führen Sie im Dialogfenster *Personalisierung* durch.


➔ Empfehlung

Nachdem Sie die Zuweisung von Filtern zu Anwendungen geändert haben, empfiehlt SAP, die Workbench-Sitzung neu zu starten. So stellen Sie sicher, dass die geänderten Einstellungen wirksam werden.

Prozess


Objektfilter bearbeiten

Gehen Sie wie folgt vor, um einen Objektfilter anzulegen:

1. Wählen Sie im Kontextmenü einer Anwendung im Repository-Baum **Anlegen** ➤ **Objektfilter** .
2. Geben Sie einen Namen und Text für den Filter an.
3. Wählen Sie **Anlegen und zu Objekt navigieren**.
Das System zeigt den neuen Filter im Arbeitsbereich an.
4. Geben Sie auf der Registerkarte **Anwendungen** an, ob der Objektauswahlfilter für alle Anwendungen oder nur für die von Ihnen angegebenen gelten soll.
5. Geben Sie auf der Registerkarte **Objekttypen** an, welche Objekttypen zum Anlegen neuer Objekte in den Anwendungen, denen der Filter zugewiesen wurde, verwendet werden können.
6. Sichern und aktivieren Sie den Filter.

Objektfilter zuweisen

Gehen Sie wie folgt vor, um einen Objektfilter in der BRFplus-Workbench zu aktivieren:

1. Wählen Sie in der oberen rechten Ecke des Workbench-Fensters .
2. Auf der Registerkarte **Filter** im Dialogfenster **Personalisierung** wählen Sie **Filter hinzufügen**, um den gewünschten Objektfilter anzugeben. Sie können aus allen im System vorhandenen Filtern auswählen, unabhängig davon, zu welcher Anwendung sie gehören.
3. Geben Sie in der Spalte **Alle Anwendungen** an, ob der Filter für alle Anwendungen im System gelten soll. Wenn Sie **Nein** wählen, müssen Sie die gewünschten Anwendungen in der Spalte **Anwendung** angeben.

Wiederholen Sie die obigen Schritte für alle weiteren Objektfilter, die Sie aktivieren möchten.

1.2.21 Versionierung

Verwendung

Mit der Versionierung können Sie Änderungen verfolgen, die an einem BRFplus-Objekt im Laufe der Zeit vorgenommen wurden. Sie basiert auf dem Zeitstempel, den das System Objekten zuordnet, wenn sie gesichert und aktiviert wurden. Es können verschiedene Versionen desselben Objekts im System verbleiben und auf dem Zeitstempel basierend verarbeitet werden.

Das System legt für jedes aktive Objekt eine neue Version an, nachdem es geändert und gesichert wurde (außer im Falle des Versionierungsmodus **Transportabhängige Versionierung** - siehe Abschnitt **Standard-**

Versionierungsverhalten). Im Gegensatz dazu wird für eine Änderung an einem inaktiven Objekt **keine** neue Version angelegt. Die Versionierung kann eingeschaltet oder ausgeschaltet werden.

Auf alte Versionen kann zugegriffen werden, indem der Zeitstempel auf die Methoden `PROCESS` oder `GET<...>` übertragen wird. Wenn Sie einen Zeitstempel übertragen und auf ein Objekt zugreifen, wird eine Version des Objekts zu dieser bestimmten Zeit zurückgegeben. Wenn kein Zeitstempel übertragen wurde, wird die zuletzt aktivierte Version zur Verarbeitung verwendet. Im Falle der Methode `GET<...>` werden die letzten Objekteigenschaften (aktiv oder inaktiv) zurückgegeben.

Beispiel

Der Ausdruck Konstante wurde am 24. Juni 2007 (Datum) um 01:56:19 (Uhrzeit) angelegt. Der Konstante wurde der Wert ABC zugeordnet. Am 14. Juli 2007 um 04:46:17 wurde der Wert zu XYZ geändert. Die Änderung wird aktiviert und gesichert. Der Wert wird erneut zu HIJ geändert, jedoch nicht aktiviert. Im Folgenden sind die Änderungen aufgeführt.

Tabelle 40:

Ereignis (Event)	Zustand	Zeitstempeländerung (Datum und Uhrzeit)	Version	Konstantenwert	Text
Ein neues Objekt wird angelegt.	Inaktiv	24.06.2007 01:56:19	1	ABC	Text ABC
Das Objekt ist eingeschaltet	Aktiv	24.06.2007 01:58:35	1	ABC	Text ABC
Der Konstantenwert wurde geändert und aktiviert.	Aktiv	14.07.2007 04:46:17	2	XYZ	Text XYZ
Der Konstantenwert wurde geändert und gesichert.	Inaktiv	12.08.2007 01:56:19	3	HIJ	

Die aktuelle Version der Konstante hat den Wert HIJ und wurde nicht aktiviert. Die BRFplus-Programmierschnittstelle (API) verarbeitet jedoch nur aktive Versionen. Die letzte aktive Version (in diesem Beispiel ist die letzte aktive Version 2) gilt, bis eine neue aktive Version angelegt wird.

Hinweis

Die Versionierung ist nur für Objekte sinnvoll, die für die Verarbeitung relevant sind. Dies gilt für fast alle BRFplus-Objekttypen, mit Ausnahme der folgenden nicht versionierten Objekttypen:

- Katalog
- Benutzerdefinierter Ausdruckstyp
Benutzerdefinierte Ausdruckstypen sind nicht versioniert, auch wenn sie für die Verarbeitung relevant sind. Der Grund dafür ist, dass BRFplus keinen Einfluss auf das Erstellen von benutzerdefinierten Ausdruckstypen hat. Es ist daher möglich, dass ein Ausdruckstyp völlig neu erstellt wird und daher im Laufe der Zeit ein unterschiedliches Verhalten zeigen.

Funktionsumfang

Standard-Versionierungsverhalten

Sie können definieren, ob ein neu angelegtes Objekt standardmäßig der Versionskontrolle unterliegt oder nicht. Diese Standardeinstellung wird auf Anwendungsebene vorgenommen und betrifft alle Objekte, die innerhalb einer Anwendung angelegt wurden. Sie können diese Einstellung für eine Anwendung in der BRFplus-Workbench unter der Registerkarte *Standardeinstellungen* anpassen.

Hinweis

Unter bestimmten Umständen kann die Verwendung von Objektversionen potenzielle „Nebenwirkungen“ haben, die zu schwer behebbaren Fehlern führen können. Aus diesem Grund hat SAP entschieden, die Standard-Versionierungsmodi *Versionierung ein* und *Versionierung erzwungen* als „nicht empfohlen“ zu definieren. Weitere Informationen finden Sie unter *Versionierung: Vor- und Nachteile [Seite 182]*.

Sie können aus folgenden Standard-Versionierungsmodi wählen:

- *Aus*
Für neu angelegte Objekte ist die Versionierung standardmäßig **ausgeschaltet**. Sie können die Versionierung zu jeder Zeit manuell für alle Objekte in der Anwendung ein- oder ausschalten. Die Versionierungseinstellung von bereits angelegten Objekten ist von diesem Standard-Versionierungsmodus **nicht** betroffen.
- *Versionierung ein (nicht empfohlen)*
Für neu angelegte Objekte ist die Versionierung standardmäßig **eingeschaltet**. Sie können die Versionierung zu jeder Zeit manuell für alle Objekte in der Anwendung ein- oder ausschalten. Die Versionierungseinstellung von bereits angelegten Objekten ist von diesem Standard-Versionierungsmodus **nicht** betroffen.
- *Versionierung erzwungen (nicht empfohlen)*
Für neu angelegte Objekte ist die Versionierung standardmäßig **eingeschaltet**. Solange dieser Standard-Versionierungsmodus aktiviert ist, wird die Versionierung aller Objekte in der Anwendung erzwungen und kann **nicht** ausgeschaltet werden.
Wenn Sie diesen Standard-Versionierungsmodus einstellen, bietet das System an, die Versionierung für alle Objekte einzuschalten, die bereits in der Anwendung vorhanden sind. Wenn Sie dies bestätigen, wird die Versionierung für alle in der Anwendung vorhandenen Objekte eingeschaltet. Wenn Sie ablehnen, bleibt die Versionierungseinstellung für alle vorhandenen Objekte unverändert. In beiden Fällen wird die Versionierung für neu angelegte Objekte eingeschaltet und kann nicht ausgeschaltet werden.
Je nach Systemstatus kann der oben erwähnte Bestätigungsdialog zusätzliche Kontrollen enthalten:
 - *Transportobjekte* Markieren Sie dieses Ankreuzfeld, um sicherzustellen, dass die geänderten Objekte in einem Transportauftrag erfasst sind. Dieses Ankreuzfeld ist nur verfügbar, wenn die Anwendung transportierbar ist und der Systemmandant dahingehend konfiguriert wurde, dass Änderungen an mandantenabhängigen Objekten **nicht** automatisch erfasst werden.
 - *Transportauftrag*: Die ID des Transportauftrags dient der Erfassung von Änderungen. Wenn die Anwendung bereits einem Transportauftrag zugeordnet wurde, wird die entsprechende Transport-ID angezeigt und kann nicht geändert werden. Anderenfalls können Sie einen passenden Transportauftrag auswählen.

Hinweis

Wenn Sie die Versionierung für alle vorhandenen Objekte einschalten, sollten Sie beachten, dass dies nur für aktive und konsistente Objekte möglich ist. Wenn es in der Anwendung inaktive Objekte gibt,

ändert das System die Versionierungs-Einstellungen für diese Objekte nicht. Sie können diese Objekte entweder einzeln oder über das Massenänderungswerkzeug manuell aktivieren.

Wenn Sie zu einem späteren Zeitpunkt entscheiden, dass die *erzwungene Versionierung* nicht länger benötigt wird, können Sie einfach einen anderen Standard-Versionierungsmodus auswählen. Wenn Sie die Versionierung für ausgewählte Objekte ausschalten möchten, ist dies eine Voraussetzung.

- **Transportabhängige Versionierung**

Für neu angelegte Objekte ist die Versionierung standardmäßig **eingeschaltet**. Dieser Modus ist eine Verfeinerung des *Versionierung erzwungen*-Modus und ist speziell darauf zugeschnitten, die Zahl der Versionen zu reduzieren und damit Speicherkapazität zu sparen. Dieser Versionierungsmodus zeigt das folgende Verhalten:

- Die transportabhängige Versionierung ist nur für Objekte in Anwendungen möglich, die transportierbar sind (Ablagetyp *System* oder *Customizing*; Entwicklungspaket richtig zugeordnet).
- Wenn Sie ein bereits aktives Objekt modifizieren, legt das System eine neue Version für dieses Objekt an. Im Gegensatz zu den anderen Versionierungsmodi bleibt diese neue Version solange gültig, bis das Objekt aus dem Design-Zeit-System in andere Systeme transportiert wurde. Das bedeutet, dass Sie in der Zeit zwischen der ersten Änderung des Objekts und dem Transport, beliebig viele Änderungen an dem Objekt vornehmen und es beliebig oft aktivieren können, ohne dabei für jede Änderung eine neue Version anzulegen.
- Die Versionierung ist für alle Objekte in einer Anwendung obligatorisch und kann daher **nicht** für einzelne Objekte ausgeschaltet werden.

Dieses Versionierungsverhalten ist in Design-Zeit-Systemen nützlich, in denen Objekte in der Entwicklung regelmäßig geändert werden. Diese Änderungen beeinträchtigen somit nicht Ihre Geschäftsprozesse, da sie in einem separaten, nichtproduktiven System vorgenommen werden. Dieser Versionierungsmodus führt ebenso zu einer automatischen Synchronisierung von Objektversionen sowohl im Entwicklungssystem als auch in den Produktivsystemen.

Wenn Sie diesen Standard-Versionierungsmodus einstellen, bietet das System an, die Versionierung für alle Objekte einzuschalten, die bereits in der Anwendung vorhanden sind. Wenn Sie dies bestätigen, wird die Versionierung für alle in der Anwendung vorhandenen Objekte eingeschaltet. Wenn Sie ablehnen, bleibt die Versionierungseinstellung für alle vorhandenen Objekte unverändert. In beiden Fällen wird die Versionierung für neu angelegte Objekte eingeschaltet und kann nicht ausgeschaltet werden.

Zeitstempel

Das System pflegt für jede Version eines Objekts einen Zeitstempel. Je nach Objektversionsstatus ändert sich die Bedeutung dieses Zeitstempels.

- Inaktiv: Solange eine Version inaktiv ist, zeigt der Zeitstempel den Zeitpunkt an, zu dem das Objekt zuletzt **gesichert** wurde.
- Aktiv: Bei einer aktiven Version, zeigt der Zeitstempel den Zeitpunkt an, zu dem das Objekt **aktiviert** wurde.

Es ist hilfreich, diese beiden Bedeutungen des Zeitstempels zu beachten, wenn Sie bestimmen müssen, welche Objektversion zur Laufzeit verwendet wird. Es kann beispielsweise die Situation entstehen, dass ein versioniertes Objekt geändert wird, für das bereits eine aktive Version existiert, die Aktivierung der geänderten Version jedoch zu einem späteren Zeitpunkt stattfindet. Das System verwendet zur Laufzeit weiterhin die ältere aktive Version, solange die geänderte aktuelle Version noch nicht aktiviert wurde. Das heißt, damit die Änderungen wirksam werden, ist **nicht** der Zeitpunkt relevant, zu dem die geänderte Version aktiviert wurde, sondern der Zeitpunkt, zu dem sie geändert wurde. Somit liefert der Zeitstempel der Version die relevantesten Daten bezüglich des Objektversionsstatus. Dieses Systemverhalten erlaubt es im Falle einer aktiven Version nicht, wann das Objekt zuletzt geändert wurde.

Versionierung von Texten und Dokumentation

Sie können auch für unter Versionsverwaltung stehende Objekte definieren, dass deren Texte und Dokumentation versionsabhängig sein sollen. Mit dieser Funktion ist es möglich, Änderungen in der Benennung aufzuzeichnen, die innerhalb verschiedener Versionen vorgenommen wurden.

Beispiel

Sie haben in einer BRFplus-Anwendung für Vergütungsmanagement eine Zahl von Datenobjekten modelliert, die für verschiedene Steuerbeträge in einem Land stehen. Nach einem Regierungswechsel können einige Steuerbeträge durch die Behörden neu benannt worden sein. Somit würde ein neuer versionsabhängiger Name für die betroffenen Datenobjekte angelegt werden, während die vorher genutzten Namen aufgrund gesetzlicher Anforderungen in einer älteren Version beibehalten werden.

Wenn sich darüber hinaus die interne Struktur eines bestimmten Objekts im Laufe der Zeit deutlich ändert, kann es empfehlenswert sein, dies in der Objektdokumentation zu erwähnen.

Beispiel

In einer BRFplus-Anwendung gibt es einen komplexen Formelausdruck, der für die Risikoprüfung als Basis für die Berechnung der Versicherungsprämie verwendet wird. Im Laufe der müssen Sie eventuell weitere Risikofaktoren hinzufügen, die zu Beginn nicht im Umfang enthalten waren. Um festzustellen, wie die überarbeitete Formel funktioniert, pflegen Sie eine versionsabhängige Dokumentation für den Ausdruck.

Hinweis

Die Versionierung von Texten und der Dokumentation führt zu einem deutlich höheren Datenbankspeicher-Kapazitätsverbrauch. Deshalb empfehlen wir Ihnen, diese Funktion nur in Fällen zu verwenden, in denen dies aus Anwendungsfallperspektive obligatorisch ist.

Versionierungsstatus

Abhängig von den verschiedenen Versionierungsmodi, die im Abschnitt *Standard-Versionierungsverhalten* beschrieben wurden, kann der Objektstatus verschiedene Werte annehmen. Diese Werte sind:

Tabelle 41:

Versionierungsstatus	Beschreibung
Nicht versioniert	Es existiert nur eine Version für ein Objekt. Alle Änderungen an einem Objekt werden in derselben Version abgelegt. Daher wird das Objekt überschrieben, ohne dass der vorherige Zustand wiederhergestellt werden kann.
Versioniert	Dieser Status führt zur Anlegung einer neuen Version zusätzlich zur Initialversion, die für jedes Objekt existiert, wenn ein zuvor aktiviertes Objekt geändert wurde. Die kumulierten Änderungen an einer Objektversion werden nach der Aktivierung in dieser Version eingefroren.

Versionierungsstatus	Beschreibung
Über Transport versioniert	Ähnlich zu <i>versioniert</i> . Eine Version wird jedoch nicht durch die Aktivierung eingefroren, sondern durch den Transport eines aktivierten Objekts in ein anderes System. Dieselbe Version kann also mehrmals aktiviert werden, solange sie nicht transportiert wird.
Nicht transportiert	Eine Version, die mit dem Versionierungsverhalten <i>Über Transport versioniert</i> angelegt wurde. Jedoch wurde das Versionierungsverhalten geändert, bevor der betreffende Transport freigegeben wurde.

In der folgenden Tabelle sehen Sie, wie eine bestimmte Änderung an einem Objekt dem Versionierungsstatus entspricht:

Tabelle 42:

Objektänderungen	Version Nr. 1	Versionierungsstatus
Neu angelegtes Objekt O, Versionierung ausgeschaltet	1	Nicht versioniert
Versionierung <i>eingeschaltet</i> für O.	1	Nicht versioniert
O aktiviert.	1	Versioniert
Versionierungsverhalten der Anwendung ist auf <i>transportabhängige Versionierung</i> gesetzt.	1	Versioniert
O geändert.	2	Versioniert
Versionierungsverhalten der Anwendung ist auf <i>Versionierung erzwungen</i> gesetzt.	2	Nicht transportiert
O aktiviert.	3	Versioniert
Transportauftrag mit dem Objekt O freigegeben.	3	Über Transport versioniert <i><Transport-ID></i>

Die Tabelle zeigt, dass bei der transportabhängigen Versionierung zwei zusätzliche Statuswerte eingeführt wurden: *Über Transport versioniert <Transport-ID>* und *Nicht transportiert*. Diese zwei Werte kennzeichnen die folgenden Situation bezüglich der Versionierung:

- *Über Transport versioniert <Transport-ID>*: Transportabhängige Versionierung ist für die Anwendung, zu der das Objekt gehört, aktiviert. Das Objekt wurde in einem Transportauftrag erfasst `<transport ID>` und der Transport wurde freigegeben.
- *Nicht transportiert*: Transportabhängige Versionierung war zunächst für die Anwendung, zu der das Objekt gehört, aktiviert. Bevor der Transport jedoch freigegeben wurde, wurde das Versionierungsverhalten auf

[Versionierung erzwungen](#) gesetzt. Folglich legt das System eine neue Version für das Objekt an. Die vorherige Version, die mit dem Transport verknüpft werden sollte, ist nun als [Nicht transportiert](#) markiert.

Weitere Informationen

[Versionierung: Vor- und Nachteile \[Seite 182\]](#)

[Anwendung \[Seite 30\]](#)

[Tracing \[Seite 152\]](#)

[Versionen festlegen \[Seite 260\]](#)

1.2.21.1 Versionierung: Vor- und Nachteile

Es folgt eine Darstellung der Vor- und Nachteile der verschiedenen Versionierungsmodi.

Obwohl die Aufzeichnung von Objektversionen eine nützliche Funktion ist und unter bestimmten Umständen sogar rechtlich verpflichtend sein kann, kann sie ebenso zu potenziellen Problemen führen: Objektversionen beizubehalten führt immer zu einer größeren Nutzung der Speicherkapazität, da alle Daten für jede Version in der Datenbank gesichert werden. Das heißt, es gibt keine Möglichkeit, nur die Unterschiede zwischen zwei Versionen zu sichern. Es gibt daneben jedoch noch weitere mögliche Nebeneffekte, die zu schwerwiegenden Fehlersituationen führen können. Aus diesem Grund hat SAP entschieden, die Standard-Versionierungsmodi [Versionierung ein](#) und [Versionierung erzwungen](#) als "nicht empfohlen" zu definieren. Diese Entscheidung wurde gefällt, um dem Benutzer von BRFplus deutlich zu machen, dass diese Versionierungsmodi mögliche Wartungsprobleme hervorrufen, die auf den ersten Blick nicht erkannt werden können, sowie eine erhöhte Speicherauslastung nach sich ziehen können, die in den meisten Fällen nicht erwünscht ist. Bezüglich der Entscheidung, welcher Versionierungsmodus verwendet werden sollte, empfehlen wir die folgende Vorgehensweise:

1. Entscheiden Sie, ob die Versionierung für die betreffende Anwendung überhaupt gebraucht wird. Wenn dies nicht der Fall ist, wählen Sie [Versionierung aus](#).
2. Wenn die Versionierung aus rechtlichen Gründen oder zur Einhaltung verbindlicher organisatorischer Regelungen gebraucht wird, wählen Sie [Über Transport ausgelöste Versionierung](#). Mit dieser Einstellung, wird eine neue Version nur dann angelegt, wenn ein Objekt in ein Test- oder Produktivsystem transportiert wird.
3. Wählen Sie die nicht empfohlenen Versionierungsmodi nur aus, wenn Sie eine lückenlose Folge aller Objektversionen aufzeichnen möchten (z. B., zu Testzwecken oder um bestimmte Anforderungen zur Qualitätssicherung zu erfüllen). Wenn Sie sich dafür entscheiden, empfehlen wir Ihnen weiterhin sicherzustellen, dass der Versionierungsmodus so schnell wie möglich auf die empfohlene Einstellung zurückgesetzt wird, um die in diesem Kapitel beschriebenen möglichen Probleme zu vermeiden.

Je mehr Objektversionen in der Datenbank existieren, desto höher ist die Zahl der Abhängigkeiten zwischen diesen Objekten sowie das Risiko möglicher verborgener Objektabhängigkeiten. Sie möchten beispielsweise ein Objekt `Object1` aus dem BRFplus-Repository löschen. Zunächst prüfen Sie, ob das betreffende Objekt noch von anderen Objekten verwendet wird. In diesem Fall, stellen Sie sicher, dass diese Verwendungsbeziehungen aufgehoben werden, so dass das Objekt `Object1` gelöscht werden kann. Es ist jedoch möglich, dass das Objekt aufgrund "historischer" Verwendungsbeziehungen, die immer noch für eine

ältere Version des Objekts `Object1` gültig sind, nicht gelöscht werden kann. Obwohl die aktuelle Objektversion nicht von einem anderen Objekt verwendet wird, kann immer noch eine Verwendungsbeziehung zwischen einer älteren Objektversion und einem anderen Objekt bestehen. In diesem Fall würde das System das Löschen des Objekts verweigern.

Um dieses Problem zu lösen, können Sie entweder das referenzierte Objekt (außer das Objekt wird weiterhin gebraucht) oder die Versionen des referenzierten Objekts `Object1` löschen. Mit dem Report `FDT_TRUNCATE_VERSIONING`, den Sie innerhalb der Transaktion `SA38` oder `SE38` starten können Sie jeweils alte Objektversionen löschen. Der Report ist ebenfalls in der Hilfsreportsammlung enthalten, die durch die Transaktion `FDT_HELPERS` aufgerufen werden kann (► [BRFplus - Löschen](#) ► [BRFplus: Discard Old Versions](#) ►). Finally, the same functionality is available in the BRFplus workbench via ► [Tools](#) ► [Application Administration](#) ►, operation type [Discard Old Object Versions](#). Beachten Sie, dass die Backend-Version des Reports im Vergleich zum BRFplus-Workbench Werkzeug zusätzliche Funktionen bietet, besonders bezüglich erweiterter Objektselektionen, die über die Grenzen einer bestimmten BRFplus-Anwendung hinausgehen. Sie können die Backend-Version des Reports ebenso in einem Batch-Job nutzen, um die alten Objektversionen regelmäßig im Hintergrund aus der Datenbank zu entfernen.

Wenn Sie den Report zur Bereinigung nutzen, beachten Sie dabei folgendes:

- Es können nur die Objektversionen durch den Report bereinigt werden, die sich in dem System befinden, in dem Sie aktuell angemeldet sind. Wenn bereits Objektversionen in Ihre Systemlandschaft transportiert wurden, sollten Sie den Report in jedem betroffenen System ausführen.
- Wenn Sie ein ganzes Objekt (das heißt, nicht nur die Objektversion) löschen und die Löschung in einem Transportauftrag aufgezeichnet wird, würde man erwarten, dass das Objekt ebenso in den System gelöscht wird, in die der Transportauftrag importiert wird. Wenn das System jedoch ermittelt, dass das zu löschende Objekt noch durch ein Objekt im Zielsystem referenziert ist, wird das Objekt lediglich zum Löschen vorgemerkt und im Zielsystem nicht tatsächlich gelöscht.

1.2.22 Berechtigungen

Verwendung

Die verschiedenen Aktivitäten, die Sie in BRFplus ausführen, sind durch das SAP-Berechtigungskonzept abgedeckt. Folglich können Sie Personen, die mit BRFplus arbeiten, verschiedene Zugriffsberechtigungen zur Modellierung und Ausführung von Regeln zuordnen.

Integration

Die im Abschnitt *Funktionsumfang* beschriebenen Berechtigungsobjekte sind in der vordefinierten Benutzerrolle [BRFplus-Administrator](#) (`SAP_BC_FDT_ADMINISTRATOR`) zusammengefasst, die mit BRFplus ausgeliefert wird.

Funktionsumfang

Es gibt folgende Berechtigungsobjekte für BRFplus:

- **FDT_ADMN**: Über dieses Berechtigungsobjekt können Sie auf verschiedene Verwaltungs- und Überwachungswerkzeuge zugreifen. Diese Werkzeuge werden nur von Entwicklern und Systemadministratoren benötigt.
- **FDT_WORKB**: Über dieses Berechtigungsobjekt steuern Sie den Benutzerzugriff auf die BRFplus-Workbench. Wenn einem Benutzer die Berechtigung zum Starten der Workbench gewährt wurde, können Sie die Zugriffsrechte des Benutzers hinsichtlich der verschiedenen Workbench-Werkzeuge, wie z.B. XML-Datentransfer oder Web-Service-Generierung, anpassen.
- **FDT_OBJECT**: Über dieses Berechtigungsobjekt steuern Sie die Berechtigung zur Anzeige sowie zum Anlegen, Ändern oder Löschen von Objekten in BRFplus (einschließlich Funktionen, Ausdrücken, Ausdruckstypen, Filtern und Anwendungen).
- **FDT_TRACE**: Über dieses Berechtigungsobjekt steuern Sie die Berechtigung zur Anzeige sowie zum Anlegen, Ändern oder Löschen von Trace-Informationen, die für die Ausführung einer Funktion in BRFplus aufgezeichnet wurden.
- **FDT_PROC**: Über dieses Berechtigungsobjekt prüfen Sie, ob ein Benutzer berechtigt ist, die Regeln, die einer BRFplus-Funktion zugeordnet sind, zu verarbeiten.

Hinweis

Zur Zeit wird die Berechtigungsprüfung nur ausgeführt, wenn eine Funktion über einen Web-Service oder einen Funktionsbaustein angestoßen wird und der Benutzer die Berechtigungsüberprüfung bereits während der Generierung des Web-Services oder des Funktionsbausteins aktiviert hat.

Aktivitäten

Legen Sie die erforderlichen Benutzerprofile an und ordnen Sie diesen dann Berechtigungsobjekte zu. Im nächsten Schritt ordnen Sie die neu angelegten Benutzerprofile den möglichen Benutzern zu.

Hinweis

Weitere Informationen bezüglich Aktivitäten in Verbindung mit den verschiedenen Berechtigungsobjekten finden Sie in der Online-Dokumentation für Berechtigungsobjekte. Diese können Sie über die Pflegeaktion *Rollenpflege* (**PFCG**) aufrufen.

Weitere Informationen

[BRFplus-Administrator \[Seite 185\]](#)

1.2.22.1 BRFplus-Administrator

Rolle `SAP_BC_FDT_ADMINISTRATOR`

Diese Rolle bietet Zugriff auf die BRFplus-Workbench, also auf die Benutzungsoberfläche, auf der Sie Regelobjekte wie Ausdrücke oder Datenobjekte anlegen und Regeln modellieren und testen. Business Rule Framework plus (BRFplus) ist ein auf ABAP basierendes Modellierungssystem für Geschäftsregeln, das von allen Anwendungen, die auf dem SAP NetWeaver ABAP-Stack aufgebaut sind, verwendet werden kann.

Ein Benutzer, dem diese Rolle im Benutzerprofil zugewiesen wurde, kann viele Aktivitäten in der BRFplus-Workbench durchführen. Dazu gehören das Anlegen, Ändern, Löschen oder Versionieren aller Objekttypen, die von BRFplus unterstützt werden. Aufgrund der vielen Berechtigungen sollte diese Rolle nur Benutzern zugewiesen werden, die mit BRFplus administrative Aufgaben durchführen. Verwenden Sie diese Rolle als Kopiervorlage für alle anderen Benutzer, um Rollen mit weniger Berechtigungen zu definieren.

Aktivität

Die Rolle enthält Berechtigungsobjekte für die folgenden Aufgaben:

- Zugriff auf einzelne Objekte
- Regelverarbeitung über Web-Service oder RFC-fähige Funktionsbausteine
- Regelverarbeitung mit Protokollierungsdaten
- Zugriff auf verschiedene Tools für Verwaltungsaufgaben in der BRFplus-Workbench

Integration

Sie müssen diese Rolle (oder eine Kopie dieser Rolle mit weniger Berechtigungen) Benutzern zuweisen, die direkt mit der BRFplus-Workbench arbeiten. Sie müssen diese Rolle außerdem Benutzern zuweisen, die mit einer Anwendung arbeiten, welche BRFplus-Funktionen verwendet und eine der im obigen Abschnitt [Aktivität](#) aufgeführten Funktionen verwendet.

1.3 Aufgaben

Verwendung

In diesem Abschnitt werden die Aufgaben beschrieben, die Sie zum Modellieren von Anwendungen in BRFplus ausführen müssen.

Die Aufgaben sind wie folgt gruppiert:

- [Anwendungen modellieren \[Seite 186\]](#)
Beschreibt, wie Anwendungen angelegt und Eigenschaften für diese festgelegt werden

- [Funktionen bauen \[Seite 190\]](#)
Beschreibt, wie Funktionen angelegt und Eigenschaften für diese festgelegt werden
- [Ausdrücke anlegen \[Seite 196\]](#)
Beschreibt, welche Aufgaben zum Anlegen verschiedener Ausdrücke erforderlich sind
- [Ausdrucksarten anlegen \[Seite 235\]](#)
Beschreibt, welche Aufgaben zum Anlegen einer Ausdrucksart erforderlich sind
- [Datenobjekte anlegen \[Seite 236\]](#)
Beschreibt, wie Sie verschiedene Typen von Datenobjekten anlegen
- [Regel anlegen \[Seite 243\]](#)
Beschreibt, welche Aufgaben zum Anlegen einer Regel erforderlich sind
- [Regelsatz anlegen \[Seite 245\]](#)
Beschreibt, welche Aufgaben zum Anlegen eines Regelsatzes erforderlich sind
- [Mit Katalogen arbeiten \[Seite 247\]](#)
Beschreibt, welche Aufgaben zum Anlegen eines Katalogs erforderlich sind
- [Aktionsarten anlegen \[Seite 252\]](#)
Beschreibt, wie Sie Aktionen in einer Anwendung anlegen und verwenden
- [Mit Tools arbeiten \[Seite 262\]](#)
Beschreibt, welche Aufgaben mit verschiedenen Tools ausgeführt werden können

1.3.1 Anwendungen modellieren

Verwendung

In diesem Abschnitt wird beschrieben, wie man in BRFplus Attribute für Anwendungen anlegt und einstellt.

In diesem Abschnitt werden folgende Themen behandelt:

- [Eine Anwendung anlegen \[Seite 186\]](#)
- [Eigenschaften festlegen \[Seite 188\]](#)
- [Objekte hinzufügen \[Seite 188\]](#)
- [Die Zugriffsebene festlegen \[Seite 189\]](#)

1.3.1.1 Eine Anwendung anlegen

Vorgehensweise

1. Wählen Sie in der Menüleiste **Workbench** > **Anwendung anlegen...**.
2. Geben Sie im daraufhin eingeblendeten Dialogfenster **Objekterzeugung** die folgenden Daten im Bereich **Allgemeine Daten** ein:

- Name

i Hinweis

Der Anwendungsname muss eindeutig sein.

Weitere Informationen finden Sie unter [Namenskonventionen \[Seite 36\]](#).

- (Optional) Kurztext
Sie können eine Kurzbeschreibung der Anwendung im Feld [Kurztext](#) eingeben.
- (Optional) Text
Sie können Zusatzinformationen über die Anwendung im Feld [Text](#) eingeben.

3. Wählen Sie im Bereich [Anwendung](#) im Dialogfenster [Objekterzeugung](#) den Speichertyp.

i Hinweis

Zum Anlegen einer Test- oder Demoanwendung aktivieren Sie das Kontrollkästchen [Lokale Anwendung anlegen](#).

Lokale Anwendungen werden nicht in andere Systeme transportiert.

Weitere Informationen finden Sie unter [Speichertyp \[Seite 35\]](#).

4. Geben Sie das Entwicklungspaket ein und wählen Sie [Anlegen](#) oder [Anlegen und zu Objekt navigieren](#).

Ergebnisse

Der Anwendungsknoten wird im Bereich [Navigation](#) angezeigt und die Anwendung wird im Bereich [Objektmanager](#) geöffnet.

Nächste Schritte

[Eigenschaften festlegen \[Seite 188\]](#)

[Zugriffsebene festlegen \[Seite 189\]](#)

[Objekte hinzufügen \[Seite 188\]](#)

1.3.1.2 Eigenschaften festlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* auf der Registerkarte *Eigenschaften* geben Sie im Feld *Anwendungskomponente* den Anwendungskontext ein.
3. Wählen Sie die Softwarekomponente, in der die Anwendung gesichert werden soll.
4. Geben Sie im Feld *Klasse für Anwendungs-Exits* die Klasse ein.

BRFplus bietet Exits für Anwendungen mit speziellen Anforderungen. Eine Klasse wird angelegt, um die speziellen Anforderungen der Anwendung zu erfüllen. Sie können die Methoden nur implementieren, wenn der Klassenname im obigen Feld registriert ist.

Nächste Schritte

[Anwendung anlegen \[Seite 186\]](#)

[Objekte hinzufügen \[Seite 188\]](#)

[Zugriffsebene festlegen \[Seite 189\]](#)

1.3.1.3 Objekte hinzufügen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
3. Wählen Sie auf der Seite *Enthaltene Objekte* den gewünschten Objekttyp aus und danach *Objekt anlegen*.

Das Dialogfeld *Objekterzeugung* wird geöffnet.

Weitere Informationen zum Anlegen von Objekten:

[Datenobjekte anlegen \[Seite 236\]](#)

[Ausdrücke anlegen \[Seite 196\]](#)

[Ausdrucksarten anlegen \[Seite 235\]](#)

[Funktion anlegen \[Seite 191\]](#)

[Regel anlegen \[Seite 243\]](#)

[Regelsatz anlegen \[Seite 245\]](#)

[Katalog anlegen \[Seite 248\]](#)

[Aktionsarten anlegen \[Seite 252\]](#)

Nächste Schritte

[Anwendung anlegen \[Seite 186\]](#)

[Eigenschaften festlegen \[Seite 188\]](#)

[Zugriffsebene festlegen \[Seite 189\]](#)

1.3.1.4 Die Zugriffsebene festlegen

Kontext

Durch das Festlegen einer Zugriffsebene für ein Objekt definieren Sie den Umfang an Objekten, die auf das aktuelle Objekt zugreifen dürfen. Zugriffsebenen werden für alle Objekttypen in BRFplus unterstützt. Standardmäßig ist jedem Objekt die restriktivste Zugriffsebene zugeordnet ([Anwendung](#)). Wenn Sie also eine andere Zugriffsebene auswählen, wird der Umfang an Objekten erweitert, die auf das aktuelle Objekt zugreifen dürfen. Es ist empfehlenswert immer die restriktivste Zugriffsebene zu verwenden und den Umfang nur nach Bedarf zu erweitern. Dies ist wichtig, weil mit erweitertem Umfang auch das Risiko ungewünschter Nebenwirkungen steigt.

Hinweis

Aus technischer Sicht ist die **Erweiterung** der Zugriffsebene auf ein Objekt unkritisch. Im Gegensatz hierzu kann die **Einschränkung** der Zugriffsebene für ein Objekt (z. B. von [Oberste Komponente](#) auf [Anwendung](#)) zu unerwarteten Ergebnissen führen, falls das Objekt bereits von einer externen Anwendung verwendet wird, die erweiterte Zugriffsrechte besitzt.

Bei Datenobjekten verhindert das System die Einschränkung der Zugriffsebene für ein Objekt, sobald das Objekt in ein anderes System übermittelt wurde. Auf diese Weise wird sichergestellt, dass Objektreferenzen, die möglicherweise in Folgesystemen angelegt wurden, ihre Gültigkeit nicht verlieren.

Die folgenden Zugriffsebenen sind verfügbar:

- **Anwendung**
Der Zugriff auf das Objekt kann über alle anderen Objekte erfolgen, die zur selben BRFplus Anwendung gehören.

- **Anwendungskomponente**
Der Zugriff auf das Objekt kann über Objekte erfolgen, die einer Anwendung angehören, die derselben Komponente zugeordnet ist wie auch die Anwendung des Objekts.
- **Übergeordnete Komponente**
Der Zugriff auf das Objekt kann über Objekte erfolgen, die einer Anwendung angehören, die einer Komponente mit derselben übergeordneten Komponente zugeordnet sind wie die Komponente, die der Anwendung des Objekts zugeordnet ist. Beispiel: Ein Objekt, das Komponente FI-GL zugeordnet ist, hat Zugriff auf ein Objekt, das FI-AA mit Zugriffsebene *Übergeordnete Komponente* zugeordnet ist, da beide Objekte dieselbe übergeordnete Komponente FI verwenden.
- **Oberste Komponente**
Der Zugriff auf die Objekte kann über alle Objekte erfolgen, die dieselbe oberste Komponente verwenden. Beispiel: Die drei Objekte A, B, und C, die den Komponenten FI-GL, FI-AR, FI-AA-AA zugeordnet sind, haben alle Zugriff auf Objekt O, das zu einer beliebigen FI-*Komponente gehört. Voraussetzung hierfür ist jedoch, dass O der Zugriffsebene *Oberste Komponente* zugeordnet ist, da alle drei Objekte dieselbe oberste Komponente für O verwenden, nämlich FI.
- **Global**
Der Zugriff auf das Objekt ist über alle anderen Objekte aller Anwendungen möglich, unabhängig von der zugeordneten Komponente.

Hinweis

Nur einer der von BRFplus unterstützten Objekttypen ist von diesen Einstellungen der Zugriffsebene **nicht** betroffen. Dies ist der Fall für Objektfilter. In diesem Fall prüft das System keine Einstellungen der Zugriffsebene, was sich wie folgt auswirkt:

- Wie alle anderen Objekttypen verwenden auch Objektfilter ein Attribut zur Definition der Zugriffsebene. Die Einstellungen der Zugriffsebene werden jedoch für Objektfilter vom System ignoriert. Dies führt dazu, dass alle im System festgelegten Objektfilter in allen Anwendungen angezeigt und verwendet werden können.
- Objekttypen, die in einem Objektfilter einschränkt werden können, werden in einer systemweiten und übergreifenden Anwendungsliste angezeigt, unabhängig von der für diesen Objekttyp definierten Zugriffsebene.

Vorgehensweise

1. Öffnen Sie das Objekt im Bereich *Objektmanager*.
2. Wählen Sie im Bereich *Allgemein* die Ebene im Feld *Zugriffsebene*.

1.3.2 Funktionen bauen

Verwendung

In diesem Abschnitt wird beschrieben, wie man in BRFplus Funktionen anlegt und mit ihnen arbeitet.

Weitere Informationen

[Eine Funktion anlegen \[Seite 191\]](#)

[Eigenschaften festlegen \[Seite 192\]](#)

[Kontext und Ergebnis zuordnen \[Seite 193\]](#)

[Den obersten Ausdruck zuordnen \[Seite 194\]](#)

1.3.2.1 Eine Funktion anlegen

Kontext

Mit der Funktion können Sie Kontext importieren und Ergebnisse zurückgeben.

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
3. Wählen Sie auf der Seite *Enthaltene Objekte* das Objekt *Funktion* im Feld *Typ* und danach *Objekt anlegen*.
4. Geben Sie im daraufhin eingeblendeten Dialogfenster *Objekterzeugung* die folgenden Daten ein:
 - Name
Weitere Informationen finden Sie unter [Namenskonventionen \[Seite 36\]](#).
 - (Optional) Kurztext
 - (Optional) Text
5. Wählen Sie *Anlegen* oder *Anlegen und zu Objekt navigieren*.

Hinweis

Wenn Sie *Anlegen* wählen, wird die neu angelegte Funktion auf der Registerkarte *Enthaltene Objekte* angezeigt.

Wenn Sie *Anlegen und zu Objekt navigieren* wählen, können Sie direkt zum Abschnitt *Eigenschaften* navigieren.

Nächste Schritte

[Eigenschaften festlegen \[Seite 192\]](#)

[Kontext und Ergebnis zuordnen \[Seite 193\]](#)

[Obersten Ausdruck zuordnen \[Seite 194\]](#)

1.3.2.1.1 **Eigenschaften festlegen**

Kontext

Die Anwendung liefert einen Kontext und verarbeitet die Funktion über einen Operationsmodus.

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* auf der Registerkarte *Eigenschaften* wählen Sie im Feld *Modus* den Operationsmodus aus.

Es sind drei Operationsmodi verfügbar:

- Funktionsmodus
- Ereignismodus
- Funktions- und Ereignismodus

i Hinweis

Wenn Sie „Funktionsmodus“ oder „Funktions- und Ereignismodus“ wählen, müssen Sie der Funktion einen obersten Ausdruck zuweisen.

Im Ereignismodus ist dies nicht erforderlich.

Weitere Informationen finden Sie unter [Operationsmodi \[Seite 151\]](#).

3. Geben Sie die Kontextklasse ein.

Nächste Schritte

[Funktion anlegen \[Seite 191\]](#)

[Kontext und Ergebnis zuordnen \[Seite 193\]](#)

[Obersten Ausdruck zuordnen \[Seite 194\]](#)

1.3.2.1.2 Kontext und Ergebnis zuordnen

Kontext

Mit dieser Funktion können Sie einen Kontext importieren und das Ergebnis nach erfolgter Verarbeitung zurückgeben. In der ereignisgetriebenen Betriebsart sind Kontext und Ergebnis optional.

Vorgehensweise

1. Öffnen Sie die Funktion im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Signatur*.

Die Registerkarte *Signatur* wird angezeigt.

3. Ordnen Sie einen Kontext zu.

Sie können ein vorhandenes oder ein neues Datenobjekt oder mehrere Elemente hinzufügen.

- **Vorhandenes Datenobjekt hinzufügen**

1. Wählen Sie *Vorhandenes Datenobjekt hinzufügen*.
2. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus und dann *Auswählen*.
Das Datenobjekt wird zum Kontext hinzugefügt.

- **Neues Datenobjekt hinzufügen**

- Wählen Sie ► *Neues Datenobjekt hinzufügen* ► *Neues Datenobjekt hinzufügen* ►
Weitere Informationen finden Sie unter *Datenobjekte anlegen* [Seite 236].




- **Mehrere Elemente hinzufügen**

1. Wählen Sie ► *Neues Datenobjekt hinzufügen* ► *Neue Elemente hinzufügen* ►
2. Wählen Sie im Dialogfenster *Neue Elemente anlegen* die Art aus der Spalte Art aus.

Type	Element Type	Reference	Name	Short Text
Build In Type	Text			
Build In Type	Text			
Build In Type	Text			
Build In Type	Text			
Build In Type	Text			
Build In Type	Text			
Build In Type	Text			
Build In Type	Text			
Build In Type	Text			
Build In Type	Text			

3. Wählen Sie die Elementart, geben Sie einen Namen ein und wählen Sie *Anlegen*.
Das neue Element wird als Kontext hinzugefügt.
4. Ordnen Sie ein Ergebnis zu.

Sie können ein Standardobjekt, ein vorhandenes Datenobjekt oder ein neues Objekt zuordnen.

- **Standardobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts.
- **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Auswählen* aus dem Kontextmenü.
 3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
Das Datenobjekt wird als Ergebnisdatenobjekt hinzugefügt.
- **Neues Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).

Nächste Schritte

[Funktion anlegen \[Seite 191\]](#)

[Obersten Ausdruck zuordnen \[Seite 194\]](#)

[Eigenschaften festlegen \[Seite 192\]](#)

1.3.2.1.3 Den obersten Ausdruck zuordnen

Kontext


Wenn Sie die funktionale und ereignisgetriebene Betriebsart wählen, müssen Sie für *Oberster Ausdruck* einen Ausdruck zuweisen.

Weitere Informationen finden Sie unter [Operationsmodi \[Seite 151\]](#).


Vorgehensweise

1. Öffnen Sie die Funktion im Bereich *Objektmanager*.
2. Weisen Sie einen vorhandenen oder neuen Ausdruck als obersten Ausdruck zu.

Einen vorhandenen Ausdruck zuweisen

1. Wählen Sie im Bereich *Details* auf der Seite *Eigenschaften* den Operationsmodus aus.
2. Wählen Sie  neben dem Feld *Oberster Ausdruck*.
3. Wählen Sie *Auswählen* aus dem Kontextmenü.
4. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus der Liste aus.

Einen Ausdruck anlegen

1. Wählen Sie  neben dem Feld *Oberster Ausdruck*.
2. Wählen Sie *Anlegen* aus dem Kontextmenü.
3. Geben Sie im daraufhin eingeblendeten Dialogfenster *Objekterzeugung* die folgenden Daten im Bereich *Allgemeine Daten* ein:
 - Name
 - Kurztext
4. Wählen Sie den Ausdrucktyp im Feld *Typ* und dann *Anlegen* oder *Anlegen und zu Objekt navigieren*.

Hinweis

Wenn Sie *Anlegen* wählen, wird der neu angelegte Ausdruck neben dem Feld *Oberster Ausdruck* angezeigt.

Nächste Schritte

[Booleschen Ausdruck anlegen \[Seite 197\]](#)

[GRMS-Anbindungsausdruck anlegen \[Seite 199\]](#)

[Fallausdruck anlegen \[Seite 202\]](#)

[Konstanten Ausdruck anlegen \[Seite 204\]](#)

[Datenbankabfrage-Ausdruck anlegen \[Seite 205\]](#)

[Entscheidungstabelle \(Ausdruck\) anlegen \[Seite 208\]](#)

[Entscheidungsbaum \(Ausdruck\) anlegen \[Seite 210\]](#)

[Dynamischen Ausdruck anlegen \[Seite 213\]](#)

[Formelausdruck anlegen \[Seite 215\]](#)

[Funktionsaufrufsausdruck anlegen \[Seite 218\]](#)

[Schleife \(Ausdruck\) anlegen \[Seite 221\]](#)

[Zufallszahl \(Ausdruck\) anlegen \[Seite 223\]](#)

[Wertebereich \(Ausdruck\) anlegen \[Seite 232\]](#)

[Suchbaum \(Ausdruck\) anlegen \[Seite 225\]](#)

[Statischen Methodenaufruf \(Ausdruck\) anlegen \[Seite 227\]](#)

[Schrittfolge \(Ausdruck\) anlegen \[Seite 229\]](#)

[XSL-Transformation \(Ausdruck\) anlegen \[Seite 233\]](#)

1.3.3 Ausdrücke anlegen

Verwendung

Dieser Abschnitt beschreibt die Vorgehensweise zum Anlegen verschiedener Ausdrucksarten. Weitere Informationen zu den einzelnen Ausdrucksarten finden Sie unter [Ausdrücke \[Seite 80\]](#). Wenn Sie eine benutzerdefinierte Ausdrucksart einrichten möchten, um Anforderungen zu bearbeiten, die nicht durch die vordefinierten Ausdrucksarten von BRFplus abgedeckt sind, finden Sie unter [Ausdrucksart \[Seite 129\]](#) weitere Informationen.

Weitere Informationen

[Booleschen Ausdruck anlegen \[Seite 197\]](#)

[GRMS-Anbindungsausdruck anlegen \[Seite 199\]](#)

[Fallausdruck anlegen \[Seite 202\]](#)

[Konstanten Ausdruck anlegen \[Seite 204\]](#)

[Datenbankabfrage-Ausdruck anlegen \[Seite 205\]](#)

[Entscheidungstabelle \(Ausdruck\) anlegen \[Seite 208\]](#)

[Entscheidungsbaum \(Ausdruck\) anlegen \[Seite 210\]](#)

[Dynamischen Ausdruck anlegen \[Seite 213\]](#)

[Formelausdruck anlegen \[Seite 215\]](#)

[Funktionsaufrufsausdruck anlegen \[Seite 218\]](#)

[Schleife \(Ausdruck\) anlegen \[Seite 221\]](#)

[Zufallszahl \(Ausdruck\) anlegen \[Seite 223\]](#)

[Suchbaum \(Ausdruck\) anlegen \[Seite 225\]](#)

[Statischen Methodenaufruf \(Ausdruck\) anlegen \[Seite 227\]](#)

[Schrittfolge \(Ausdruck\) anlegen \[Seite 229\]](#)

[Wertebereich \(Ausdruck\) anlegen \[Seite 232\]](#)

[XSL-Transformation \(Ausdruck\) anlegen \[Seite 233\]](#)

1.3.3.1 Booleschen Ausdruck anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Boolesch* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Der neu angelegte Boolesche Ausdruck wird im Bereich *Objektmanager* angezeigt.

Hinweis

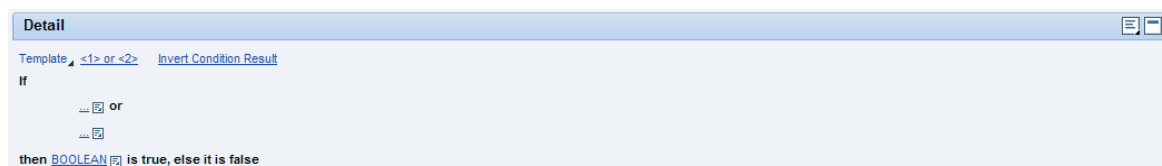
Wenn Sie *Anlegen* wählen, wird der Boolesche Ausdruck auf der Registerkarte *Enthaltene Objekte* angezeigt.

7. Im Abschnitt *Details* wählen Sie *Vorlage* und danach den Vorlagentyp aus.

Folgende Vorlagentypen stehen zur Verfügung:

- Jeder Operand ist WAHR
- Alle Operanden sind WAHR
- <1> und <2>
- <1> oder <2> oder <3>
- (<1> und <2>) oder <3>
- (<1> oder <2>) und <3>
- <1> und <2> und <3>
- Wählen Sie Ihre eigene Vorlage aus.

Der ausgewählte Vorlagentyp wird neben der Vorlage angezeigt.




8. Abhängig von der ausgewählten Vorlage können Sie dem Booleschen Ausdruck eine Bedingung hinzufügen.

Sie können das ausgewählte Datenobjekt oder den Ausdruck als Testparameter verwenden und einen Wertebereich (Ausdruck) anlegen.


- **Wert aus Standardtyp als Bedingung verwenden**

1. Wählen Sie  neben *wenn*.
2. Wählen Sie im Kontextmenü ► *Direkten Wertebereich verwenden von* ► *Standardobjekte* ►, und wählen Sie den Standardtyp aus.

- **Wert aus vorhandenem Kontextdatenobjekt als Bedingung verwenden**


1. Wählen Sie  neben *wenn*.
2. Wählen Sie im Kontextmenü ► *Direkten Wertebereich verwenden von* ► *Anderer Kontextparameter* ►.
3. Im Dialogfenster *Kontextabfrage* wählen Sie das Datenobjekt aus.

- **Wert aus vorhandenem Ausdruck als Bedingung verwenden**


1. Wählen Sie  neben *wenn*.
2. Wählen Sie im Kontextmenü ► *Direkten Wertebereich verwenden von* ► *Ausdrücke* ►.
3. Im Dialogfenster *Objektabfrage* Geben Sie den Namen des Ausdrucks ein und wählen *Suchen*. Der Ausdruck wird unterhalb der Drucktaste *Suchen* angezeigt.
4. Wählen Sie den Ausdruck aus.

Sie können einen Ausdruck oder ein Datenobjekt auch direkt als Bedingung auswählen. Der ausgewählte Ausdruck sollte als Ergebnistyp „Boolesch“ und das ausgewählte Datenobjekt sollte vom Typ „Boolesch“ sein.


- **Operand aus vorhandenem Ausdruck als Bedingung verwenden**

1. Wählen Sie  neben *wenn*.
2. Wählen Sie im Kontextmenü ► *Operand auswählen* ► *Ausdrücke auswählen* ► *Auswählen...* ►.
3. Im Dialogfenster *Objektabfrage* Geben Sie den Namen des Ausdrucks ein und wählen *Suchen*. Der Ausdruck wird unterhalb der Drucktaste *Suchen* angezeigt.
4. Wählen Sie den Ausdruck aus.

- **Operand aus vorhandenem Kontextdatenobjekt als Bedingung verwenden**

1. Wählen Sie  neben *wenn*.
2. Wählen Sie im Kontextmenü ► *Operand auswählen* ► *Datenobjekt auswählen* ► *Anderen Kontextparameter* ►.
3. Im Dialogfenster *Kontextabfrage* wählen Sie das Datenobjekt aus.

- **Neuen Ausdruck als Bedingung verwenden**

1. Wählen Sie  neben *wenn*.
2. Wählen Sie *Ausdruck anlegen* aus dem Kontextmenü.
3. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).

9. Wählen Sie *Aktivieren*.

10. Sichern Sie den Ausdruck.

1.3.3.2 Einen GRMS-Anbindungsausdruck anlegen




Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Registerkarte *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *GRMS-Konnektor* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Der neu angelegte GRMS-Konnektor wird im Bereich *Objektmanager* angezeigt.

Hinweis

Wenn Sie *Anlegen* wählen, wird der GRMS-Konnektor auf der Registerkarte *Enthaltene Objekte* angezeigt.

7. Weisen Sie ein Ergebnisdatenobjekt zu. Sie können einen Standard-Datenobjekttyp, ein vorhandenes Datenobjekt oder ein neues zuweisen.
 - **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Auswählen* aus dem Kontextmenü.
 3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
 - **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter *Datenobjekte anlegen* [Seite 236].
8. Wählen Sie den Verbindungstyp. Wählen Sie entweder *NetWeaver GRMS-Verbindung* oder *Externe GRMS-Verbindung*.
9. Wählen Sie die Destination.

Die Destination muss in den Einstellungen festgelegt werden, damit der Aufruf ordnungsgemäß weitergeleitet werden kann.

10. Geben Sie die Basiswährung ein.

Detail

Result Data Object: [<not assigned>](#) ⓘ

Connection Type:

Function Module:

Destination:

Base Currency: ⓘ

Connection Parameters | **Signature**

Context

ⓘ	Name	Text	Parameter Type	Object Type

Result

Result Parameter: [<not assigned>](#) ⓘ

11. Fügen Sie dem Ausdruck einen Kontext zu. Sie können entweder ein vorhandenes Objekt hinzufügen oder ein neues Objekt anlegen.

- **Vorhandenes Objekt hinzufügen**
 1. Wählen Sie [Vorhandenes Objekt hinzufügen](#).
 2. Im jetzt angezeigten Dialogfenster [Objektabfrage](#) wählen Sie das gewünschte Objekt aus und danach [Auswählen](#).
- **Neues Objekt hinzufügen**
 1. Wählen Sie [Neues Objekt hinzufügen](#).
 2. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).

12. Weisen Sie einen Ergebnisparameter zu. Sie können einen Standard-Datenobjekttyp oder einen vorhandenen Ergebnisparameter zuweisen oder einen neuen Ergebnisparameter anlegen.

- **Standardtyp auswählen**
 1. Wählen Sie ⓘ.
 2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Typ des Datenobjekts.
- **Vorhandenen Ergebnisparameter auswählen**

1. Wählen Sie [☐](#).
 2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objektabfrage](#) wählen Sie das Datenobjekt aus.
 - **Neuen Ergebnisparameter anlegen**
 1. Wählen Sie [☐](#).
 2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster [Objekterzeugung](#) das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).
 13. Wählen Sie [Vokabularschema generieren](#).
- Verwenden Sie die generierte XSD-Regel zum Festlegen der Vokabularregel im externen GRMS-Engine.
14. Um die Verbindungsparameter einzustellen, wählen Sie die Registerkarte [Verbindungsparameter](#).
 15. Geben Sie im Bereich [Konnektor-Aufrufparameter](#) die Werte des Projekts und die festgelegte Regel ein.

Detail

Result Data Object: [<not assigned>](#) [☐](#)

Connection Type: [NetWeaver BRMS Connection](#) ▼

Function Module: [FDT_NW_BRMS_CONNECTOR_EXECUTE](#)

Destination:

Base Currency: [☐](#)

Connection Parameters **Signature**

Connector Call Parameters

[Add Row](#) [Remove Row](#)

Name	Value
project	
ruleset	

16. Wählen Sie [Aktivieren](#).
17. Sichern Sie den Ausdruck.

1.3.3.3 Einen Fallausdruck anlegen




Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Fall* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.





Der neu angelegte Fallausdruck wird im Bereich *Objektmanager* angezeigt.

Hinweis

Wenn Sie *Anlegen* wählen, wird der Fallausdruck auf der Registerkarte *Enthaltene Objekte* angezeigt.


7. Wählen Sie den Ergebnistyp aus. Sie können den Rückgabewert auswählen oder eine Aktion durchführen.
 - **Wählen Sie Rückgabewert**
 1. Wählen Sie *Rückgabewert* im Feld *Ergebnistyp*.
 2. Weisen Sie das Ergebnisdatenobjekt zu. Sie können einen Standardtyp oder ein vorhandenes Datenobjekt auswählen oder ein neues Datenobjekt anlegen.
 - **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts. Das Element wird als Ergebnisdatenobjekt hinzugefügt.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Auswählen* aus dem Kontextmenü.
 3. Geben Sie im angezeigten Dialogfenster *Objektabfrage* den Namen des Datenobjekts in das Feld *Objektname* ein, und wählen Sie *Suchen*. Das Datenobjekt wird unterhalb der Drucktaste *Suchen* angezeigt.
 4. Wählen Sie das Datenobjekt aus. Das Datenobjekt wird als Ergebnisdatenobjekt hinzugefügt.
 - **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.



Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).

- **Wählen Sie Aktion durchführen**
 - Wählen Sie [Aktion durchführen](#) im Feld [Ergebnistyp](#).
Eine Aktion wird als Ergebnisdatenobjekt hinzugefügt.
- 8. Nehmen Sie die Definition des Fallparameters vor. Sie können einen Standardtyp oder ein vorhandenes Datenobjekt auswählen oder einen neuen Ausdruck anlegen.
 - **Standardtyp auswählen**
 1. Wählen Sie  neben [wenn](#).
 2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Standardtyp.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie  neben [wenn](#).
 2. Wählen Sie im Kontextmenü [Anderer Kontextparameter](#).
 3. Im Dialogfenster [Kontextabfrage](#) wählen Sie das Datenobjekt aus.
Das ausgewählte Datenobjekt wird als Fallparameter hinzugefügt.
 - **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie  neben [wenn](#).
 2. Wählen Sie im Kontextmenü [Anderen Parameter auswählen](#).
 3. Geben Sie im angezeigten Dialogfenster [Objektabfrage](#) den Namen des Ausdrucks in das Feld [Objektnamen](#) ein, und wählen Sie [Suchen](#).
Der Ausdruck wird unterhalb der Drucktaste [Suchen](#) angezeigt.
 4. Wählen Sie den Ausdruck aus.
Der Ausdruck wird als Fallparameter hinzugefügt.
 - **Neuen Ausdruck anlegen**
 1. Wählen Sie  neben [wenn](#).
 2. Wählen Sie aus dem Kontextmenü [Neuen Parameter anlegen....](#)
 3. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).
- 9. Geben Sie die Werte in den Feldern neben [...entspricht](#) ein.

Hinweis

Wenn keiner der Testparameter mit dem Fallparameter übereinstimmt, wird ein anderer Parameter ausgewertet, um das Ergebnis anzugeben.

- 10. Wählen Sie  neben [dann Aktion durchführen](#).
Sie können eine vorhandene Aktion zuweisen oder eine neue anlegen.
 - **Vorhandene Aktion auswählen**
 1. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
 2. Im Dialogfenster [Objektabfrage](#) wählen Sie die Aktion aus.
 - **Aktion anlegen**
 1. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 2. Im Dialogfenster [Objekterzeugung](#) legen Sie die Aktion an.
Weitere Informationen finden Sie unter [Aktionsarten anlegen \[Seite 252\]](#).
- 11. Definieren Sie einen anderen Parameter. Sie können einen vorhandenen Ausdruck zuweisen, einen neuen Ausdruck anlegen oder einen Wert einfügen.
 - **Vorhandenen Ausdruck auswählen**

1. Wählen Sie  neben *Andernfalls Aktion durchführen*.
2. Wählen Sie *Auswählen* aus dem Kontextmenü.
3. Geben Sie im angezeigten Dialogfenster *Objektabfrage* den Namen des Ausdrucks in das Feld *Objektname* ein, und wählen Sie *Suchen*.
Der Ausdruck wird unterhalb der Drucktaste *Suchen* angezeigt.
4. Wählen Sie den Ausdruck aus der Liste aus.
- **Neuen Ausdruck anlegen**
 1. Wählen Sie  neben *Andernfalls Aktion durchführen*.
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
 3. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter *Ausdrücke anlegen* [Seite 196].
12. Wählen Sie *Aktivieren*.
13. Sichern Sie den Ausdruck.

1.3.3.4 Einen Konstanten-Ausdruck anlegen

Kontext

Ein konstanter Ausdruck dient zum Definieren von Ergebniswerten für andere Ausdrücke.

Hinweis

Ein konstanter Ausdruck kann nur Element-Datentypen verwenden.

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Registerkarte *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Konstante* im Feld *Typ*.
Der Bereich *Konstante* wird angezeigt.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text

6. Wählen Sie im Bereich *Konstante* die Art der Konstanten im Feld *Typ* aus.

Abhängig von der Art der gewählten Konstanten werden unterschiedliche Felder für die Werteingabe angezeigt.

Weitere Informationen finden Sie unter *Konstanter Ausdruck* [Seite 84].

7. Wählen Sie *Anlegen und zu Objekt navigieren*.

Die Ergebnisdaten werden im Bereich *Details* angezeigt.

Standardmäßig wird die Art der ausgewählten Konstanten als Ergebnisdatenobjekt angezeigt.

Hinweis

Wenn Sie *Anlegen* wählen, wird der konstante Ausdruck auf der Registerkarte *Enthaltene Objekte* angezeigt. Wählen Sie den Namen des angelegten Objekts aus.

8. Wenn Sie das Ergebnisdatenobjekt ändern möchten, wählen Sie .

Sie können ein Standardelement oder ein vorhandenes Datenobjekt auswählen oder ein neues Ergebnisdatenobjekt anlegen.

- **Standardtyp auswählen**

- Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts. Das Element wird als Ergebnisdatenobjekt hinzugefügt.

- **Vorhandenes Datenobjekt auswählen**

- 1. Wählen Sie im Kontextmenü *Auswählen...*
- 2. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus. Das Datenobjekt wird als Ergebnisdatenobjekt hinzugefügt.

- **Neues Datenobjekt anlegen**

- 1. Wählen Sie *Anlegen* aus dem Kontextmenü.
- 2. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an. Weitere Informationen finden Sie unter *Datenobjekte anlegen* [Seite 236].

9. Wählen Sie *Aktivieren*.

10. Sichern Sie den Ausdruck.

1.3.3.5 Datenbankabfrage-Ausdruck anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.

4. Im Dialogfenster *Objekterzeugung* wählen Sie *DB-Abfrage* im Feld *Typ*.

5. Geben Sie folgende Daten ein:

- Name
- (Optional) Kurztext
- (Optional) Text

6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Der neu angelegte Datenbankabfrage-Ausdruck wird im Bereich *Objektmanager* angezeigt.

Hinweis

Wenn Sie *Anlegen* wählen, wird der Datenbankabfrage-Ausdruck auf der Registerkarte *Enthaltene Objekte* angezeigt.

7. Wählen Sie den Modus aus.


Sie können aus den Modi „Datenabruf“, „Aggregation“ oder „Existenzprüfung“ auswählen.

- **Datenabrufmodus auswählen**


In diesem Modus werden die Daten aus der Datenbanktabelle abgerufen.

1. Wählen Sie *Alle Einträge* oder *Einzelerfassung*.
2. Geben Sie den Namen der Tabelle in das Textfeld ein, und drücken Sie die Eingabetaste.
3. Weisen Sie mithilfe der Felder der Datenbanktabelle eine Bedingung zu.
Sie können ein vorhandenes Datenobjekt auswählen, ein neues anlegen oder Ergebnisse aus der Tabelle erstellen.


- **Vorhandenes Datenobjekt auswählen**

1. Wählen Sie .
2. Wählen Sie *Auswählen* aus dem Kontextmenü.
3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.

- **Neues Datenobjekt anlegen**

1. Wählen Sie .
2. Wählen Sie *Anlegen* aus dem Kontextmenü.
3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).



- **Ergebnisse aus Tabelle erstellen**

1. Wählen Sie .
2. Wählen Sie im Kontextmenü *Ergebnisse aus Tabelle erstellen*.
3. Ein Dialogfenster mit dem Feld aus der Datenbanktabelle wird angezeigt. Wählen Sie die erforderlichen Felder aus.
4. Geben Sie einen Namen für die Struktur, Tabelle oder das Element ein.





- **Aggregationsmodus auswählen**

In diesem Modus wird das Ergebnis der Aggregation aus der Datenbanktabelle zurückgegeben.

1. Wählen Sie *Aggregation* im Feld *Auswahlmodus*.
2. Geben Sie den Namen der Tabelle in das Textfeld ein, und drücken Sie die Eingabetaste.
3. Wählen Sie die Aggregationsart aus. Sie können wählen aus „Maximum“, „Minimum“, „Durchschnitt“, „Anzahl“ oder „Summe“.
4. Wählen Sie *Eindeutig* oder *Alle*.

-
5. Weisen Sie mithilfe der Felder der Datenbanktabelle eine Bedingung zu. Sie können ein vorhandenes Datenobjekt auswählen oder ein neues anlegen.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Auswählen* aus dem Kontextmenü.
 3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
 - **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).
 - **Existenzprüfungsmodus auswählen**

In diesem Modus wird dem Ergebnisobjekt „true“ oder „false“ zugewiesen. Ausschlaggebend ist, ob mindestens ein Eintrag die WHERE-Bedingung der DDIC-Tabelle erfüllt.

 1. Wählen Sie *Existenzprüfung* im Feld *Auswahlmodus*.
 2. Geben Sie den Namen der Tabelle in das Textfeld ein, und drücken Sie die .
 3. Weisen Sie mithilfe der Felder der Datenbanktabelle eine Bedingung zu.
Sie können ein vorhandenes Datenobjekt auswählen, ein neues anlegen oder einen Booleschen Ausdruck auswählen.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Auswählen* aus dem Kontextmenü.
 3. Im Dialogfenster *Elementabfrage* geben Sie den Namen des Datenobjekts ein und wählen *Suchen*.
Das Datenobjekt wird unterhalb der Drucktaste *Suchen* angezeigt.
 4. Wählen Sie das Datenobjekt aus.
 - **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).
 - **Booleschen Ausdruck auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Boolesch* aus dem Kontextmenü.
 - 8. Wählen Sie *Aktivieren*.
 - 9. Sichern Sie den Ausdruck.

1.3.3.6 Entscheidungstabelle (Ausdruck) anlegen




Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Registerkarte *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Entscheidungstabelle* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Der neu angelegte Entscheidungstabelle-Ausdruck wird im Bereich *Objektmanager* angezeigt.

Hinweis

Wenn Sie *Anlegen* wählen, wird der Entscheidungstabelle-Ausdruck auf der Registerkarte *Enthaltene Objekte* angezeigt.

7. Weisen Sie das Ergebnisdatenobjekt zu. Sie können ein Standarddatenobjekt, ein vorhandenes Datenobjekt oder ein neues zuweisen.
 - **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Auswählen* aus dem Kontextmenü.
 3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
 - **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
Die Ergebnisspalte für die Entscheidungstabelle wird automatisch hinzugefügt, sobald die *Spalte als Ergebnis* ausgewählt wird.
 3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter *Datenobjekte anlegen [Seite 236]*.

Hinweis

Wenn Sie einen Entscheidungstabelle-Ausdruck als obersten Ausdruck einer Funktion anlegen, muss das zugewiesene Ergebnisdatenobjekt der Tabelle denselben Typ verwenden, der auch der

Funktion zugewiesen wurde oder er muss dem Objekt entsprechen, das für die Funktion verwendet wird.

8. Fügen Sie Spalten zur Entscheidungstabelle hinzu. Abhängig davon, wo Sie die Spalte hinzufügen möchten, wählen Sie [Spalte anhängen](#) oder [Spalte einfügen](#).

Sie können einen Funktionskontext, ein Datenobjekt, einen Ausdruck oder eine Aktion als Spalte hinzufügen.

Hinweis

Um eine Spalte am Ende der Tabelle hinzufügen, wählen Sie [Spalte anhängen](#).

Um eine Spalte an der ausgewählten Stelle hinzuzufügen, wählen Sie [Spalte einfügen](#).

○ Funktionskontext hinzufügen

1. Wählen Sie ► [Spalte anhängen](#) ► [Von Kontextdatenobjekten...](#) ►.
2. Im jetzt angezeigten Dialogfenster [Kontextabfrage](#) wählen Sie den Kontext aus und wählen Sie dann [Auswählen](#).

○ Einen sonstigen Kontext hinzufügen

1. Wählen Sie ► [Spalte anhängen](#) ► [Von sonstigen Datenobjekten...](#) ►.
2. Im Dialogfenster [Objektabfrage](#) wählen Sie das Datenobjekt aus und wählen [Auswählen](#).

○ Vorhandenen Ausdruck hinzufügen

1. Wählen Sie ► [Spalte anhängen](#) ► [Ausdruck auswählen...](#) ►.
2. Im Dialogfenster [Objektabfrage](#) wählen Sie den Ausdruck aus und danach [Auswählen](#).

○ Neuen Ausdruck hinzufügen

1. Wählen Sie ► [Spalte anhängen](#) ► [Ausdruck anlegen...](#) ►.
2. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).

○ Aktion hinzufügen

- Wählen Sie ► [Spalte anhängen](#) ► [Aktionsspalte anlegen...](#) ►
Die Aktionsspalte wird in der Entscheidungstabelle angezeigt.

9. Definieren Sie die Entscheidungstabelle, indem Sie Zeilen hinzufügen. Um Zeilen hinzuzufügen, wählen Sie [Zeile anhängen](#) im Bereich [Tabellendaten](#).

10. Fügen Sie Werte zur Entscheidungstabelle hinzu. Sie können Werte direkt hinzufügen oder einen Ausdruck als Wert festlegen.

Hinweis



Außerdem können Sie Daten aus einer Excel-Datei importieren.

Weitere Informationen finden Sie unter [Daten zwischen Entscheidungstabellen und Microsoft Excel austauschen \[Seite 100\]](#)

○ Wert hinzufügen

1. Wählen Sie ..., das neben  angezeigt wird.
2. Geben Sie die entsprechenden Werte ein und wählen Sie [Fertig](#).

○ Einen bestehenden Ausdruck als Wert hinzufügen

1. Wählen Sie .
2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
3. Wählen Sie im Dialogfenster [Objektabfrage](#) den Ausdruck aus.
- **Einen neuen Ausdruck als Wert hinzufügen**
 1. Wählen Sie .
 2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).
11. Wählen Sie [Aktivieren](#).
12. Speichern Sie die Entscheidungstabelle.

1.3.3.7 Entscheidungsbaum (Ausdruck) anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich [Objektmanager](#).
2. Im Abschnitt [Details](#) wählen Sie die Registerkarte [Enthaltene Objekte](#).
Die Registerkarte [Enthaltene Objekte](#) wird angezeigt.
3. Wählen Sie [Ausdruck](#) im Feld [Typ](#) und danach [Objekt anlegen](#).
4. Im Dialogfenster [Objekterzeugung](#) wählen Sie [Entscheidungsbaum](#) im Feld [Typ](#).
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie [Anlegen und zu Objekt navigieren](#).

Der neu angelegte Entscheidungsbaum wird im Bereich [Objektmanager](#) angezeigt.

Hinweis

Wenn Sie [Anlegen](#) wählen, wird der Entscheidungsbaum auf der Registerkarte [Enthaltene Objekte](#) angezeigt.

7. Wählen Sie den Ergebnistyp aus. Sie können den Rückgabewert auswählen oder eine Aktion durchführen.


- **Wählen Sie Rückgabewert**

1. Wählen Sie *Rückgabewert* im Feld *Ergebnistyp*.
2. Weisen Sie das Ergebnisdatenobjekt zu.
Sie können ein Standarddatenobjekt, ein vorhandenes Datenobjekt oder ein neues zuweisen.


- **Standardtyp auswählen**

1. Wählen Sie .
2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts.

- **Vorhandenes Datenobjekt auswählen**

1. Wählen Sie .
2. Wählen Sie *Auswählen* aus dem Kontextmenü.
3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus und wählen *Auswählen*.

- **Neues Datenobjekt anlegen**





1. Wählen Sie .
2. Wählen Sie *Anlegen* aus dem Kontextmenü.
3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).

Hinweis

Das Ergebnisdatenobjekt kann ein Element, eine Struktur oder ein Tabellendatenobjekt sein.

8. Weisen Sie dem Entscheidungsbaum einen Wurzelknoten zu. Sie können die direkte Eingabe oder einen vorhandenen Ausdruck auswählen oder einen neuen Ausdruck anlegen.

- **Direkte Eingabe wählen**

1. Wählen Sie im Kontextmenü des Knotens  *Bedingung eingeben...*  *Direkte Eingabe* .
2. Wählen Sie , um die eingehenden Parameter zu definieren.
Sie können einen Standardtyp, ein vorhandenes Datenobjekt oder einen Ausdruck auswählen.

- **Standardtyp auswählen**

- Wählen Sie im Kontextmenü *Standardobjekte* und danach den Standardtyp.

- **Vorhandenes Datenobjekt auswählen**

1. Wählen Sie im Kontextmenü *Anderer Kontextparameter*.
2. Im Dialogfenster *Kontextabfrage* wählen Sie das Datenobjekt aus.

- **Vorhandenen Ausdruck auswählen**

1. Wählen Sie *Ausdrücke* aus dem Kontextmenü.

2. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.
 3. Wählen Sie den Vergleichsoperatoren und geben Sie die Werte ein.
 4. Wählen Sie *Fertig*.
- **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie im Kontextmenü des Knotens ► *Bedingung eingeben...* ► *Auswählen* ►.
 2. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.
 - **Neuen Ausdruck anlegen**
 1. Wählen Sie im Kontextmenü des Knotens ► *Bedingung eingeben...* ► *Anlegen* ►.
 2. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter *Ausdrücke anlegen* [Seite 196].

i Hinweis

Der Wurzelknoten kann nur eine Bedingung sein.

Der ausgewählte Ausdruck wird als Wurzelknoten festgelegt und darunter werden zwei untergeordnete Knoten angezeigt.



9. Fügen Sie dem untergeordneten Knoten eine Bedingung oder ein Ergebnis hinzu. Sie können dem untergeordneten Knoten einen vorhandenen oder neuen Ausdruck hinzufügen, abhängig vom Ergebnis der Bedingung im Wurzel.
 - **Vorhandenen Ausdruck als Ergebnis zuweisen**
 1. Wählen Sie im Kontextmenü des positiven untergeordneten Knotens ► *Ergebnis eingeben* ► *Auswählen* ►.
 2. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.
 - **Neuen Ausdruck als Ergebnis zuweisen**
 1. Wählen Sie im Kontextmenü des positiven untergeordneten Knotens ► *Ergebnis eingeben* ► *Neu...* ►.
 2. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter *Ausdrücke anlegen* [Seite 196].
 - **Wählen Sie „Direkte Eingabe“.**
 1. Wählen Sie im Kontextmenü des Knotens ► *Bedingung eingeben...* ► *Direkte Eingabe* ►.
 2. Wählen Sie *Ein*.
Sie können einen Standardtyp, ein vorhandenes Datenobjekt oder einen Ausdruck auswählen.
 - **Standardtyp auswählen**
 - Wählen Sie im Kontextmenü *Standardobjekte* und danach den Standardtyp.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie im Kontextmenü *Anderer Kontextparameter*.

2. Im Dialogfenster *Kontextabfrage* wählen Sie das Datenobjekt aus.
- **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie *Ausdrücke* aus dem Kontextmenü.
 2. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.
3. Wählen Sie den Vergleichsoperatoren und geben Sie die Werte ein.
4. Wählen Sie *Fertig*.
- **Vorhandenen Ausdruck als Bedingung zuweisen**
 1. Wählen Sie im Kontextmenü des negativen untergeordneten Knotens ► *Bedingung eingeben* ► *Auswählen* ►.
 2. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus und danach *Auswählen*.
- **Neuen Ausdruck als Bedingung zuweisen**
 1. Wählen Sie im Kontextmenü des negativen untergeordneten Knotens ► *Bedingung eingeben* ► *Anlegen* ►.
 2. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter *Ausdrücke anlegen* [Seite 196].

i Hinweis

- Sie können beliebig viele Knotenebenen im Entscheidungsbaum anlegen.
- Alle Blätterknoten müssen Ergebnisknoten sein.

10. Aktivieren Sie den Ausdruck.

11. Sichern Sie den Ausdruck.




1.3.3.8 Dynamischen Ausdruck anlegen



Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Dynamischer Ausdruck* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.
Der neu angelegte dynamische Ausdruck wird im Bereich *Objektmanager* angezeigt.

Hinweis

Wenn Sie [Anlegen](#) wählen, wird der dynamische Ausdruck auf der Registerkarte [Enthaltene Objekte](#) angezeigt.

7. Weisen Sie ein Ergebnisdatenobjekt zu. Sie können einen Standardtyp oder ein vorhandenes Datenobjekt auswählen oder ein neues Datenobjekt anlegen.
 - **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Typ des Datenobjekts.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objektabfrage](#) wählen Sie das Datenobjekt aus.
Das Datenobjekt wird als Ergebnisdatenobjekt hinzugefügt.
 - **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster [Objekterzeugung](#) das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).
8. Wählen Sie den Modus aus. Sie können [Vollständiger Kontext aber keine Ausdrucksvariable](#) oder [Passender Kontext](#) wählen.

Der Modus [Vollständiger Kontext aber keine Ausdrucksvariable](#) leitet den verfügbaren Kontext an den dynamisch ermittelten Ausdruck weiter. Der Modus [Passender Kontext](#) leitet lediglich die übereinstimmenden Kontextobjekte weiter.
9. Wählen Sie den aufgerufenen Ausdruck aus. Sie können einen vorhandenen Ausdruck auswählen oder einen neuen anlegen.
 - **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie .
 2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objektabfrage](#) wählen Sie den Ausdruck aus.
 - **Neuen Ausdruck anlegen**
 1. Wählen Sie .
 2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).
10. Aktivieren Sie den Ausdruck.
11. Sichern Sie den Ausdruck.




1.3.3.9 Einen Formelausdruck anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Registerkarte *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Formel* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Hinweis

Wenn Sie *Anlegen* wählen, wird die angelegte Formel auf der Registerkarte *Enthaltene Objekte* angezeigt.

7. Fügen Sie das Ergebnisdatenobjekt hinzu. Sie können einen Standardtyp auswählen, ein neues Datenobjekt anlegen oder ein vorhandenes Datenobjekt hinzufügen.
 - **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Auswählen* aus dem Kontextmenü.
 3. Im Dialogfenster *Kontextabfrage* wählen Sie das Datenobjekt aus.
 - **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).
8. Legen Sie die Formel mithilfe von Ausdrücken, Datenobjekten, mathematischen und logischen Operatoren an.

General

Name: Short Text:

Application: [Z_TAX_CALCULATION](#) Access Level:

[Show More](#)

Detail

[Switch to Expert Mode](#) [Delete Formula Element](#)

Result Data Object: [PAYABLE TAX AMOUNT](#)

Formula:
$$= (25000\text{EUR} * 0.1) + (50000\text{EUR} * 0.2) + (\text{GROSS_SALARY} - 100000\text{EUR} * 0.3)]$$

Selected Token: 0.3

Use Right Click to add more objects in the Formula [Hide Quick Help](#)

Context	
Name	Description
GROSS_SALARY	GROSS_SALARY

Formula Functions	
Filter by Category: Show All Functions	
Name	Description
ARCCOS	Arc Cosinus
ARCSIN	Arc Sinus
ARCTAN	Arc Tangent
CONCATENATE	Concatenate
CONDENSE	Trims off leading and trailing spaces
COS	Cosine
COSH	Hyperbola Cosinus
DIV	Quotient
DT_DURATION_DIFF	Calculates the duration between two TPs (returns Decimals)
DT_DURATION_DIFF_INT	Calculates duration between two TPs (returns Integer only)
DT_GET_CURRENT_DATE	Gets the current date

Beispiel

Wenn Sie eine Formel anlegen möchten, z. B. $3*1$, dann führen Sie im Formeleditor folgende Schritte aus:

1. Wählen Sie **Zahl**.
2. Geben Sie im Dialogfenster **Einfügen** und dann **3** ein und wählen Sie **OK**.
Die Zahl 3 wird im Feld **Formel** angezeigt
3. Wählen Sie *****.
4. Wählen Sie **Zahl**.
Geben Sie im Dialogfenster **Einfügen** und dann **1** ein und wählen Sie **OK**.
9. Sie können die Formel auch im Expertenmodus anlegen.
 1. Wählen Sie **In den Expertenmodus wechseln**.
Neben dem Feld **Formel** wird ein Textfeld angezeigt.

Detail

[Switch to Normal Mode](#)

Result Data Object: [PAYABLE TAX AMOUNT](#)

Formula =

Use Right Click to add more objects in the Formula

Context

Name	Description
GROSS_SALARY	GROSS_SALARY

Formula Functions

Filter by Category: [Show All Functions](#)

Name	Description
ARCCOS	Arc Cosinus
ARCSIN	Arc Sinus
ARCTAN	Arc Tangent
CONCATENATE	Concatenate
CONDENSE	Trims off leading and trailing spaces
COS	Cosine
COSH	Hyperbola Cosinus
DIV	Quotient
DT_DURATION_DIFF	Calculates the duration between two TPs (returns Decimals)
DT_DURATION_DIFF_INT	Calculates duration between two TPs (returns Integer only)
DT_GET_CURRENT_DATE	Gets the current date

2. In der Formel können Sie Ausdrücke, Datenobjekte, Strings, Zahlen, Betrag, Menge und Kommentare einfügen.

Im Kontextmenü des Textfelds:

- Zum Anlegen eines Ausdrucks.
 1. Wählen Sie ► [Ausdruck einfügen](#) ► [Anlegen](#) ►.
 2. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).
- Zum Auswählen eines vorhandenen Ausdrucks.
 1. Wählen Sie ► [Ausdruck einfügen](#) ► [Auswählen](#) ►.
 2. Im Dialogfenster [Objektabfrage](#) wählen Sie den Ausdruck aus.
- Zum Anlegen eines Datenobjekts.
 1. Wählen Sie ► [Datenobjekt einfügen](#) ► [Anlegen](#) ►.
 2. Legen Sie im jetzt angezeigten Dialogfenster [Objekterzeugung](#) das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).
- Zum Auswählen eines vorhandenen Datenobjekts.
 1. Wählen Sie ► [Datenobjekt einfügen](#) ► [Auswählen](#) ►.
 2. Im Dialogfenster [Kontextabfrage](#) wählen Sie das Datenobjekt aus.

Detail

[Switch to Normal Mode](#)

Result Data Object: [PAYABLE TAX AMOUNT](#) [g]

Formula = $(25.000,00\text{EUR} * 0,1) + (50.000,00\text{EUR} * 0,2) + (001CC411E33E1DDE84D8DB80E5F8C0E8 - 100.000,00\text{EUR} * 0,3)$

Use Right Click to add more objects in the Formula [Hide Q...](#)

Context	
Name	Description
GROSS_SALARY	GROSS_SALARY

/ * - +
 ^ & ()
 = < > <= >=
 < <= > >=
 AND OR NOT IF
 String
 Number
 Amount
 Quantity
 Timepoint
 Comment
 Expression

Formula Functions
 Filter by Category: [Show All Functions](#)

Name	Description
ARCCOS	Arc Cosinus
ARCSIN	Arc Sinus
ARCTAN	Arc Tangent
CONCATENATE	Concatenate
CONDENSE	Trims off leading and trailing spaces
COS	Cosine
COSH	Hyperbola Cosinus
DIV	Quotient
DT_DURATION_DIFF	Calculates the duration between two TPs (returns Decimals)
DT_DURATION_DIFF_INT	Calculates duration between two TPs (returns Integer only)
DT_GET_CURRENT_DATE	Gets the current date

Hinweis

Sie können auch komplexe Formeln mithilfe von verschachtelten Operanden anlegen.

10. Wählen Sie [Aktivieren](#).
11. Sichern Sie den Ausdruck.

1.3.3.10 Funktionsaufrufsausdruck anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich [Objektmanager](#).
2. Im Abschnitt [Details](#) wählen Sie die Registerkarte [Enthaltene Objekte](#).
 Sie Seite [Enthaltene Objekte](#) wird angezeigt.
3. Wählen Sie [Ausdruck](#) im Feld [Typ](#) und danach [Objekt anlegen](#).
4. Im Dialogfenster [Objekterzeugung](#) wählen Sie [Funktionsaufruf](#) im Feld [Typ](#).
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie [Anlegen und zu Objekt navigieren](#).

Hinweis

Wenn Sie [Anlegen](#) wählen, wird der Funktionsaufrufsausdruck auf der Registerkarte [Enthaltene Objekte](#) angezeigt.




7. Weisen Sie ein Ergebnisdatenobjekt zu.

Sie können ein Standarddatenobjekt zuweisen, ein vorhandenes Datenobjekt auswählen oder ein neues anlegen.


○ **Standardtyp auswählen**

1. Wählen Sie .
2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Typ des Datenobjekts.

○ **Vorhandenes Datenobjekt auswählen**

1. Wählen Sie .
2. Wählen Sie im Kontextmenü  [Auswählen...](#) .
3. Im Dialogfenster [Objektabfrage](#) wählen Sie das Datenobjekt aus.

○ **Neues Datenobjekt anlegen**

1. Wählen Sie .
2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
3. Legen Sie im jetzt angezeigten Dialogfenster [Objekterzeugung](#) das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).


8. Weisen Sie dem Ausdruck eine Zielfunktion zu.

Als Zielfunktion können Sie einen Standardtyp auswählen, eine neue Funktion anlegen oder eine vorhandene verwenden.


○ **Standardtyp auswählen**

1. Wählen Sie .
2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Standardtyp.

○ **Neue Funktion anlegen**

1. Wählen Sie .
2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
3. Im Dialogfenster [Objekterzeugung](#) legen Sie die Funktion an.
Weitere Informationen finden Sie unter [Funktion anlegen \[Seite 191\]](#).
4. Weisen Sie der Funktion einen obersten Ausdruck zu.
Weitere Informationen erhalten Sie unter [Obersten Ausdruck zuordnen \[Seite 194\]](#).
5. Weisen Sie der Funktion Kontext und Ergebnis zu.
Weitere Informationen erhalten Sie unter [Kontext und Ergebnis zuordnen \[Seite 193\]](#).

○ **Vorhandene Funktion auswählen**

1. Wählen Sie .
2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
3. Im Dialogfenster [Objektabfrage](#) wählen Sie die Funktion aus.
Das Kontextdatenobjekt der Zielfunktion wird unter [Zielfunktionskontext-Datenobjekt](#) im Abschnitt [Mapping](#) angezeigt.

Detail

[Show Result Data Object](#)
 Target Function: [TAX_CALCULATOR](#)

Mapping

Source Context Data Object/Expression	Target Function Context Data Object
Not Mapped	GROSS SALARY

9. Ordnen Sie die Kontextobjekte dem Quellausdruck und der zugehörigen Zielfunktion zu.

Sie können ein Datenobjekt oder einen Ausdruck auswählen.

- **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Standardtyp.
- **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie unter der Spalte [Quell-Kontextdatenobjekt/Ausdruck](#).
 2. Wählen Sie im Kontextmenü [Anderer Kontextparameter](#).
 3. Im Dialogfenster [Kontextabfrage](#) wählen Sie das Datenobjekt aus.
Das Kontextdatenobjekt wird unter [Zielfunktionskontext-Datenobjekt](#) im Abschnitt [Mapping](#) angezeigt.
- **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie .
 2. Wählen Sie [Ausdrücke](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objektabfrage](#) wählen Sie den Ausdruck aus.
Der Kontextausdruck wird unter [Zielfunktionskontext-Datenobjekt](#) im Abschnitt [Mapping](#) angezeigt.
- **Neuen Ausdruck anlegen**
 1. Wählen Sie .
 2. Wählen Sie [Ausdruck anlegen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).

10. Aktivieren Sie den Ausdruck.

11. Sichern Sie den Ausdruck.

1.3.3.11 Schleife (Ausdruck) anlegen




Vorgehensweise






1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Schleife* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.



Die neu angelegte Schleife (Ausdruck) wird im Bereich *Objektmanager* angezeigt.

Hinweis

Wenn Sie *Anlegen* wählen, wird die Schleife (Ausdruck) auf der Registerkarte *Enthaltene Objekte* angezeigt.

7. Wählen Sie den Ergebnistyp aus. Sie können den Rückgabewert auswählen oder eine Aktion durchführen.
 - **Wählen Sie Rückgabewert**
 1. Wählen Sie *Rückgabewert* im Feld *Ergebnistyp*.
 2. Weisen Sie das Ergebnisdatenobjekt zu. Sie können einen Standardtyp oder ein vorhandenes Datenobjekt auswählen oder ein neues Datenobjekt anlegen.
 - **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Auswählen* aus dem Kontextmenü.
 3. Geben Sie im angezeigten Dialogfenster *Objektabfrage* den Namen des Datenobjekts in das Feld *Objektname* ein, und wählen Sie *Suchen*.
Das Datenobjekt wird unterhalb der Drucktaste *Suchen* angezeigt.
 4. Wählen Sie das Datenobjekt aus.
 - **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter *Datenobjekte anlegen* [Seite 236].

- **Wählen Sie Aktion durchführen**
 - Wählen Sie [Aktion durchführen](#) im Feld [Ergebnistyp](#).
Eine Aktion wird als Ergebnisdatenobjekt hinzugefügt.
- 8. Wählen Sie den Schleifenmodus aus.
 - **Wählen Sie Wiederholen... Male... Modus**
 1. Wählen Sie [Wiederholen...Male...](#) im Feld [Schleifenmodus](#).
 2. Geben Sie einen Wert im Feld [Folgende Operationen ausführen](#) ein.
 3. Weisen Sie der Operation eine Regel zu.
 - **Wählen Sie Ausführen... Bis... Modus**
 1. Wählen Sie [Ausführen...Bis...](#) im Feld [Schleifenmodus](#).
 2. Weisen Sie der Operation eine Regel zu.
 3. Weisen Sie die Exit-Bedingung zu. Sie können einen vorhandenen Ausdruck auswählen oder einen neuen anlegen.
 - **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie  neben [Bis](#) im Bereich [Details](#).
 2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
 3. Geben Sie im angezeigten Dialogfenster [Objektabfrage](#) den Namen des Ausdrucks in das Feld [Objektnamen](#) ein, und wählen Sie [Suchen](#).
Der Ausdruck wird unterhalb der Drucktaste [Suchen](#) angezeigt.
 4. Wählen Sie den Ausdruck aus.
 - **Neuen Ausdruck anlegen**
 1. Wählen Sie  neben [Bis](#) im Bereich [Details](#).
 2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).
 - **Wählen Sie While... Do... Modus**
 1. Wählen Sie [While...Do...](#) im Feld [Schleifenmodus](#).
 2. Weisen Sie die Exit-Bedingung zu. Sie können einen vorhandenen Ausdruck auswählen oder einen neuen anlegen.
 - **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie .
 2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
 3. Geben Sie im angezeigten Dialogfenster [Objektabfrage](#) den Namen des Ausdrucks in das Feld [Objektnamen](#) ein, und wählen Sie [Suchen](#).
Der Ausdruck wird unterhalb der Drucktaste [Suchen](#) angezeigt.
 4. Wählen Sie den Ausdruck aus.
 - **Neuen Ausdruck anlegen**
 1. Wählen Sie .
 2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).
 - **Wählen Sie Für jeden Eintrag in... Modus**
 1. Wählen Sie [Für jeden Eintrag in.....](#) Im Feld [Schleifenmodus](#).
 2. Weisen Sie die Exit-Bedingung zu. Wählen Sie  neben [Für jeden Eintrag in](#) im Bereich [Details](#).

3. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
4. Im Dialogfenster [Objektabfrage](#) wählen Sie das Objekt aus.
9. Weisen Sie eine Regel zu.
 - **Vorhandene Regel auswählen**
 1. Wählen Sie  neben den Bereich [Operationen](#).
 2. Wählen Sie im Kontextmenü ► [Regel an erster Position hinzufügen](#) ► [Vorhandene Regel auswählen...](#) ►.
 3. Im Dialogfenster [Objektabfrage](#) geben Sie den Namen der Regel ein und wählen [Suchen](#). Die Regel wird unterhalb der Drucktaste [Suchen](#) angezeigt.
 4. Wählen Sie die Regel.
 - **Neue Regel anlegen**
 1. Wählen Sie  neben den Bereich [Operationen](#).
 2. Wählen Sie im Kontextmenü ► [Regel an erster Position hinzufügen](#) ► [Regel anlegen...](#) ►. Eine neue Regel wird im Bereich [Operationen](#) angezeigt. Weitere Informationen finden Sie unter [Regel anlegen \[Seite 243\]](#).
10. Aktivieren Sie den Ausdruck.
11. Sichern Sie den Ausdruck.

1.3.3.12 Zufallszahl (Ausdruck) anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich [Objektmanager](#).
2. Im Abschnitt [Details](#) wählen Sie die Registerkarte [Enthaltene Objekte](#).

Die Registerkarte [Enthaltene Objekte](#) wird angezeigt.
3. Wählen Sie [Ausdruck](#) im Feld [Typ](#) und danach [Objekt anlegen](#).
4. Im Dialogfenster [Objekterzeugung](#) wählen Sie [Zufallszahl](#) im Feld [Typ](#).
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie im Bereich [Zufallszahl](#) den Modus.

Wählen Sie zwischen den Modi [Zahl](#) und [Wahrscheinlichkeit](#).

Im Zahlenmodus wird eine Zufallszahl innerhalb eines festgelegten Bereichs zurückgegeben.

Im Wahrscheinlichkeitsmodus gibt der Ausdruck einen Booleschen Wert zurück, sofern die festgelegte Wahrscheinlichkeit zwischen 0 und 1 erfüllt wird.

- **Zahlenmodus auswählen**

1. Wählen Sie [Zahl](#) im Feld [Modus](#).
2. Geben Sie Werte in den Feldern [Minimum](#) und [Maximal](#) ein.

- **Wahrscheinlichkeitsmodus auswählen**

1. Wählen Sie [Wahrscheinlichkeit](#) im Feld [Modus](#).
2. Geben Sie einen Wert zwischen 0 und 1 im Feld [Wahrscheinlichkeit](#) ein.

7. Wählen Sie [Anlegen und zu Objekt navigieren](#).

Die Zufallszahl (Ausdruck) wird im Bereich [Objektmanager](#) angezeigt.

i Hinweis

Wenn Sie [Anlegen](#) wählen, wird die Zufallszahl (Ausdruck) auf der Registerkarte [Enthaltene Objekte](#) angezeigt.

8. Wählen Sie [Ergebnisdatenobjekt einblenden](#).

Abhängig vom ausgewählten Modus wird automatisch ein Standarddatenobjekt eingestellt.

Im Zahlenmodus wird ein Datenobjekt mit Typ Zahl eingestellt, während im Wahrscheinlichkeitsmodus ein Datenobjekt mit Typ Boolesch eingestellt wird.

Als Ergebnisdatenobjekt können Sie auch einen Standardtyp oder ein vorhandenes Datenobjekt auswählen oder ein neues anlegen.


- **Standardtyp auswählen**

1. Wählen Sie .
2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Typ des Datenobjekts.


i Hinweis

Abhängig vom gewählten Modus sind zwei Standardtypen, Zahl und Boolesch, verfügbar.



- **Vorhandenes Datenobjekt auswählen**



1. Wählen Sie .
2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
3. Im Dialogfenster [Objektabfrage](#) wählen Sie das Datenobjekt aus.

- **Neues Datenobjekt anlegen**

1. Wählen Sie .
2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
3. Legen Sie im jetzt angezeigten Dialogfenster [Objekterzeugung](#) das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).

9. (Optional) Sie können auch Ausdrücke als Werte für die Modi verwenden.

1. Wählen Sie die Drucktaste [In den Referenzmodus wechseln](#).
Wenn Sie in den Referenzmodus wechseln, werden die Werte in den Wertfeldern verworfen.
Das Wertfeld wird durch das Symbol  ersetzt.
2. Wählen Sie einen vorhandenen Ausdruck oder legen Sie einen neuen an.
 - **Andere Kontextparameter auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü [Andere Kontextparameter](#).
 3. Im Dialogfenster [Kontextabfrage](#) wählen Sie das Datenobjekt aus.

- **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Ausdrücke* aus dem Kontextmenü.
 3. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.
 - **Einen Ausdruck anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Ausdruck anlegen* aus dem Kontextmenü.
 3. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter *Ausdrücke anlegen* [Seite 196].
10. Aktivieren Sie den Ausdruck.
11. Sichern Sie den Ausdruck.

1.3.3.13 Suchbaum (Ausdruck) anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Registerkarte *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Suchbaum* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Der Suchbaum wird im Bereich *Objektmanager* geöffnet.

Hinweis


Wenn Sie *Anlegen* wählen, wird das angelegte Regelobjekt auf der Registerkarte *Enthaltene Objekte* angezeigt.

7. Wählen Sie den Rückgabebetyp, um zwischen Rückgabewert oder Aktion durchführen auszuwählen.
 - **Wählen Sie Rückgabewert**
 1. Wählen Sie *Rückgabewert* im Feld *Ergebnistyp*.
 2. Weisen Sie das Ergebnisdatenobjekt zu.
Als Ergebnisdatenobjekt können Sie einen Standardtyp oder ein vorhandenes Datenobjekt auswählen oder ein neues anlegen.

- **Standardtyp auswählen**

1. Wählen Sie .
2. Wählen Sie im Kontextmenü *Standardobjekte* und danach das Standarddatenobjekt.

- **Vorhandenes Datenobjekt auswählen**


1. Wählen Sie .
2. Wählen Sie *Auswählen* aus dem Kontextmenü.
3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus und wählen *Auswählen*.

Hinweis

Um mehrere übereinstimmende Ergebnisse zurückzugeben, aktivieren Sie das Kontrollkästchen *Alle gefundenen Übereinstimmungen zurückgeben*.

Bei zugeordneten Ergebnisdatenobjekt muss es sich um ein Tabellendatenobjekt handeln.

- **Neues Ergebnisdatenobjekt anlegen**

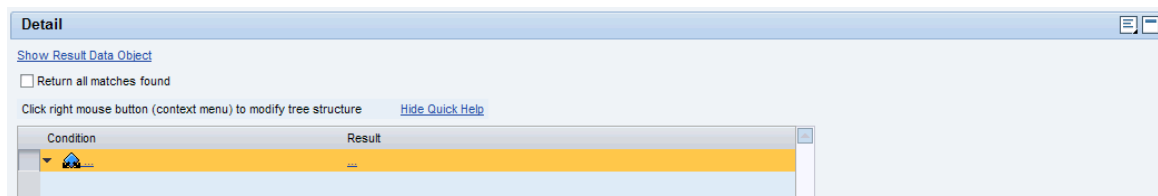
1. Wählen Sie .
2. Wählen Sie *Anlegen* aus dem Kontextmenü.
3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* ein Datenobjekt an.
Weitere Informationen finden Sie unter *Datenobjekte anlegen [Seite 236]*.

- **Wählen Sie Aktion durchführen**





- Wählen Sie *Aktion durchführen* im Feld *Ergebnistyp*.
Eine Aktion wird als Ergebnisdatenobjekt hinzugefügt.

8. Dem Wurzelknoten eine Bedingung zuordnen

In der Spalte Bedingung können Sie die direkte Eingabe oder einen vorhandenen Ausdruck wählen oder einen neuen Ausdruck anlegen.



- **Direkte Eingabe wählen**

1. Wählen Sie im Kontextmenü des Wurzelknotens  *Bedingung festlegen...*  *Direkte Eingabe* .
2. Wählen Sie , um die eingehenden Parameter zu definieren.
Sie können einen Standardtyp, ein vorhandenes Datenobjekt oder einen Ausdruck auswählen.

- **Standardtyp auswählen**

- Wählen Sie im Kontextmenü *Standardobjekte* und danach den Standardtyp.

- **Vorhandenes Datenobjekt auswählen**

1. Wählen Sie im Kontextmenü *Anderer Kontextparameter*.
2. Im Dialogfenster *Kontextabfrage* wählen Sie das Datenobjekt aus.

- **Vorhandenen Ausdruck auswählen**

1. Wählen Sie *Ausdrücke* aus dem Kontextmenü.
2. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.
3. Wählen Sie den Vergleichsoperatoren und geben Sie die Werte ein.
4. Wählen Sie *Fertig*.

- **Vorhandenen Ausdruck auswählen**

1. Wählen Sie im Kontextmenü des Wurzelknotens ► *Bedingung festlegen...* ► *Auswählen* ⌵.
 2. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.
 - **Neuen Ausdruck anlegen**
 1. Wählen Sie im Kontextmenü des Wurzelknotens ► *Bedingung festlegen...* ► *Anlegen* ⌵.
 2. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* einen neuen Ausdruck an.
Weitere Informationen finden Sie unter *Ausdrücke anlegen* [Seite 196].
 9. Ordnen Sie dem Wurzelknoten ein Ergebnis zu.
- In der Spalte Ergebnis können Sie einen vorhandenen Ausdruck wählen oder einen neuen Ausdruck anlegen.
- **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie im Kontextmenü des Wurzelknotens ► *Ergebnis festlegen* ► *Auswählen* ⌵.
 2. Im Dialogfenster *Objektabfrage* wählen Sie einen Ausdruck aus.
 - **Neuen Ausdruck anlegen**
 1. Wählen Sie im Kontextmenü des Wurzelknotens ► *Ergebnis festlegen* ► *Anlegen* ⌵.
 2. Im Dialogfenster *Objekterzeugung* legen Sie einen Ausdruck an.
Weitere Informationen finden Sie unter *Ausdrücke anlegen* [Seite 196].
 10. Legen Sie einen Kindknoten an.
 - Wählen Sie im Kontextmenü des Wurzelknotens ► *Knotenoperationen* ► *Kindknoten anlegen* ⌵.

i Hinweis

Sie müssen zunächst alle erforderlichen Kindknoten anlegen, bevor Sie dem zuerst angelegten Kindknoten eine Bedingung und ein Ergebnis zuordnen können.

11. Weisen Sie dem Kindknoten Bedingungen und Ergebnisse zu.
12. Aktivieren Sie den Ausdruck.
13. Sichern Sie den Ausdruck.

1.3.3.14 Statischer Methodenaufruf (Ausdruck) anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Statischer Methodenaufruf* im Feld *Typ*.

5. Geben Sie folgende Daten ein:

- Name
- (Optional) Kurztext
- (Optional) Text




6. Wählen Sie [Anlegen und zu Objekt navigieren](#).

Der neu angelegte statische Methodenaufruf (Ausdruck) wird im Bereich [Objektmanager](#) angezeigt.

i Hinweis

Wenn Sie [Anlegen](#) wählen, wird der statischen Methodenaufruf (Ausdruck) auf der Registerkarte [Enthaltene Objekte](#) angezeigt.

7. Weisen Sie ein Ergebnisdatenobjekt zu. Sie können einen Standardtyp oder ein vorhandenes Datenobjekt auswählen oder ein neues Datenobjekt anlegen.

- **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Typ des Datenobjekts.
- **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objektabfrage](#) wählen Sie das Datenobjekt aus.
Das Datenobjekt wird als Ergebnisdatenobjekt hinzugefügt.
- **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster [Objekterzeugung](#) das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).

8. Geben Sie den Namen der Klasse ein oder wählen Sie  im Feld [Klassenname](#).

9. Wählen Sie die relevante Klasse in der angezeigten Liste aus und wählen Sie [OK](#).

Der ausgewählte Klasse wird hinzugefügt.

10. Fügen Sie dem Ausdruck einen Teilausdruck hinzu. Sie können einen vorhandenen Ausdruck auswählen oder einen neuen anlegen.

- **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie [Ausdruck auswählen](#).
 2. Im Dialogfenster [Objektabfrage](#) wählen Sie den Ausdruck aus.
- **Neuen Ausdruck anlegen**
 1. Wählen Sie [Ausdruck anlegen](#).
 2. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).

11. Aktivieren Sie den Ausdruck.

12. Sichern Sie den Ausdruck.

1.3.3.15 Schrittfolge (Ausdruck) anlegen

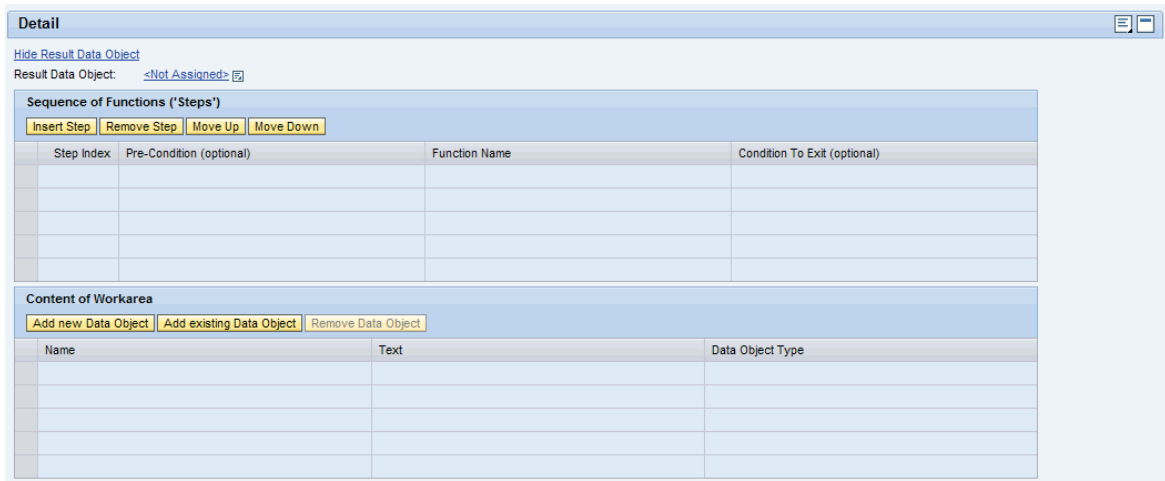
Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterstellung* wählen Sie *Schrittfolge* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Die Schrittfolge (Ausdruck) wird im Bereich *Objektmanager* angezeigt.


Hinweis

Wenn Sie *Anlegen* wählen, wird die neu angelegte Schrittfolge (Ausdruck) auf der Registerkarte *Enthaltene Objekte* angezeigt.



Detail

[Hide Result Data Object](#)

Result Data Object: [<Not Assigned>](#) 

Sequence of Functions ('Steps')

[Insert Step](#) [Remove Step](#) [Move Up](#) [Move Down](#)

Step Index	Pre-Condition (optional)	Function Name	Condition To Exit (optional)

Content of Workarea

[Add new Data Object](#) [Add existing Data Object](#) [Remove Data Object](#)

Name	Text	Data Object Type

7. Weisen Sie ein Ergebnisdatenobjekt zu.

Als Ergebnisdatenobjekt können Sie einen Standardtyp zuweisen, ein vorhandenes Datenobjekt auswählen oder ein neues anlegen.

- **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts.

- **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie [☒](#).
 2. Wählen Sie im Kontextmenü [Auswählen...](#)
 3. Im Dialogfenster [Objektabfrage](#) wählen Sie das Datenobjekt aus.
 - **Neues Ergebnisdatenobjekt anlegen**
 1. Wählen Sie [☒](#).
 2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster [Objekterzeugung](#) ein neues Objekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).
8. Weisen Sie Funktionsfolgen zu. Wählen Sie [Schritt einfügen](#).

In der Tabelle wird ein neuer Schritt angezeigt.

Sequence of Functions ('Steps')			
Insert Step Remove Step Move Up Move Down			
Step Index	Pre-Condition (optional)	Function Name	Condition To Exit (optional)
1	<Not Assigned> ☒	<Not Assigned> ☒	<Not Assigned> ☒

i Hinweis


Vor der Verarbeitung des Schritts bewertet das System die dazugehörige Vorbedingung. Die Vorbedingung ist ein Ausdruck, welcher einen Booleschen Wert zurückgeben muss. Wenn der Wert der Vorbedingung als „true“ bewertet wird, überspringt das System den entsprechenden Schritt.

9. (Optional) Fügen Sie eine Vorbedingung hinzu.

Als Vorbedingung können Sie einen Standardtyp, einen vorhandenen oder einen neuen Ausdruck hinzufügen

- **Standardtyp auswählen**
 1. Wählen Sie [☒](#).
 2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Standardtyp.
 - **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie [☒](#).
 2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objektabfrage](#) wählen Sie den Ausdruck aus.
 - **Neuen Ausdruck anlegen**
 1. Wählen Sie [☒](#).
 2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).
10. Fügen Sie eine Funktion hinzu.
- **Standardtyp auswählen**
 1. Wählen Sie [☒](#).
 2. Wählen Sie im Kontextmenü [Standardobjekte](#) und danach den Standardtyp.
 - **Vorhandene Funktion auswählen**
 1. Wählen Sie [☒](#).
 2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
 3. Im Dialogfenster [Objektabfrage](#) wählen Sie die Funktion aus.

- **Neue Funktion anlegen**

1. Wählen Sie .
2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
3. Im Dialogfenster *Objekterzeugung* legen Sie die Funktion an.
Weitere Informationen finden Sie unter [Funktion anlegen \[Seite 191\]](#).

11. (Optional) Fügen Sie eine Exit-Bedingung hinzu.

Sie können einen neuen Ausdruck anlegen oder einen vorhandenen auswählen.


i Hinweis

Exit-Bedingungen müssen Ausdrücke mit einem Booleschen Ergebnisdatenobjekt sein.


- **Standardtyp auswählen**

1. Wählen Sie .
2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Standardtyp.

- **Vorhandenen Ausdruck auswählen**

1. Wählen Sie .
2. Wählen Sie [Auswählen](#) aus dem Kontextmenü.
3. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.

- **Neuen Ausdruck anlegen**

1. Wählen Sie .
2. Wählen Sie [Anlegen](#) aus dem Kontextmenü.
3. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).

12. Weisen Sie im Feld *Inhalt des Arbeitsbereichs* Inhalte hinzu.

Sie können ein vorhandenes oder ein neues Datenobjekt hinzufügen.

- **Vorhandenes Datenobjekt hinzufügen**

1. Wählen Sie [Vorhandenes Datenobjekt hinzufügen](#).
2. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus und wählen [Auswählen](#).

- **Neues Datenobjekt anlegen**

1. Wählen Sie [Neues Datenobjekt hinzufügen](#).
2. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter [Datenobjekte anlegen \[Seite 236\]](#).

13. Aktivieren Sie den Ausdruck.

14. Sichern Sie den Ausdruck.

1.3.3.16 Wertebereich (Ausdruck) anlegen




Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
3. Wählen Sie auf der Seite *Enthaltene Objekte* das Objekt *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Wertebereich* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Der Wertebereich (Ausdruck) wird im Bereich *Objektmanager* angezeigt.

Hinweis

Wenn Sie *Anlegen* wählen, wird der Wertebereich (Ausdruck) auf der Registerkarte *Enthaltene Objekte* angezeigt.

7. Nehmen Sie die Definition des eingehenden Parameters vor. Sie können einen Standardtyp, ein vorhandenes Datenobjekt oder einen Ausdruck als eingehenden Datenparameter auswählen.
 - **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Standardtyp.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Andere Kontextparameter*.
 3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
 - **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Ausdrücke* aus dem Kontextmenü.
 3. Geben Sie im angezeigten Dialogfenster *Objektabfrage* den Namen des Ausdrucks in das Feld *Objektname* ein, und wählen Sie *Suchen*.

Der Ausdruck wird unterhalb der Drucktaste *Suchen* angezeigt.
 4. Wählen Sie den Ausdruck aus.

Detail

If GROSS SALARY (Type : Amount)

is equal to 0,00

Change

and unless ...

then ... is true, otherwise it is false.

8. Wählen Sie im Bereich *Details* den Vergleichsoperator aus und geben Sie den Wert für den eingehenden Parameter ein.
9. Aktivieren Sie den Ausdruck.
10. Sichern Sie den Ausdruck.

1.3.3.17 XSL-Transformation (Ausdruck) anlegen

Vorgehensweise




1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Registerkarte *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Ausdruck* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *XSL-Transformation* im Feld *Typ*.
5. Geben Sie folgende Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Die XSL-Transformation (Ausdruck) wird im Bereich *Objektmanager* angezeigt.

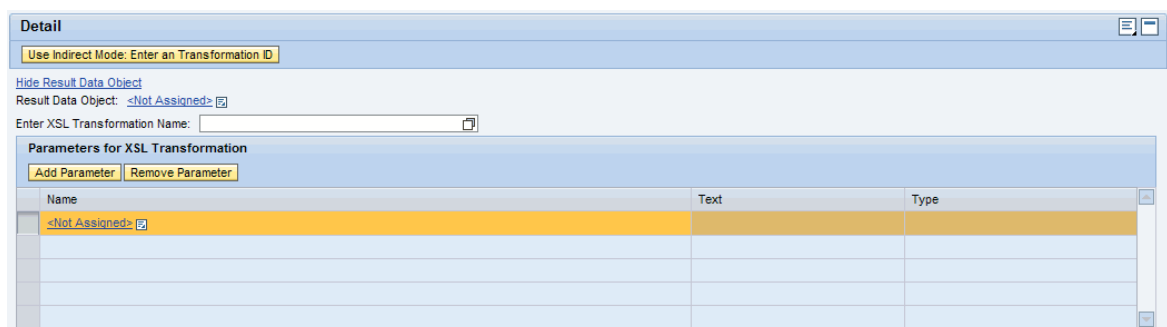
Hinweis

Wenn Sie *Anlegen* wählen, wird die XSL-Transformation (Ausdruck) auf der Registerkarte *Enthaltene Objekte* angezeigt.

7. Wählen Sie *Ergebnisdatenobjekt einblenden*. Als Ergebnisdatenobjekt können Sie einen Standardtyp zuweisen, ein vorhandenes Datenobjekt auswählen oder ein neues anlegen.
 - **Standardtyp auswählen**

1. Wählen Sie .
2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Typ des Datenobjekts.
- **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Auswählen* aus dem Kontextmenü.
 3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
Das ausgewählte Datenobjekt wird als Ergebnis für die XSL-Transformation (Ausdruck) hinzugefügt.
- **Neues Datenobjekt anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Anlegen* aus dem Kontextmenü.
 3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
Weitere Informationen finden Sie unter *Datenobjekte anlegen* [Seite 236].
8. Wählen Sie *Parameter hinzufügen*.

Der Tabelle *Parameter für XSL-Transformation* wird eine neue Zeile hinzugefügt. Als Parameter können Sie einen Standardtyp, ein vorhandenes Datenobjekt oder einen vorhandenen oder neuen Ausdruck hinzufügen.



Detail

Use Indirect Mode: Enter an Transformation ID

Hide Result Data Object





Result Data Object: <Not Assigned>

Enter XSL Transformation Name:

Parameters for XSL Transformation

Add Parameter Remove Parameter

Name	Text	Type
<Not Assigned>		

- **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Standardtyp.
- **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Anderer Kontextparameter*.
 3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
Das ausgewählte Datenobjekt wird als Parameter für die XSL-Transformation (Ausdruck) hinzugefügt.
- **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Ausdruck auswählen...* aus dem Kontextmenü.
 3. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus und danach *Auswählen*.
Der ausgewählte Ausdruck wird als Parameter für die XSL-Transformation (Ausdruck) hinzugefügt.
- **Neuen Ausdruck anlegen**
 1. Wählen Sie .
 2. Wählen Sie *Ausdrücke* aus dem Kontextmenü.
 3. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.

Weitere Informationen finden Sie unter [Ausdrücke anlegen](#) [Seite 196].

9. Aktivieren Sie den Ausdruck.
10. Sichern Sie den Ausdruck.

1.3.4 Ausdrucksarten anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich [Objektmanager](#).
2. Im Abschnitt [Details](#) wählen Sie die Registerkarte [Enthaltene Objekte](#).
3. Wählen Sie auf der Seite [Enthaltene Objekte](#) das Objekt [Ausdrucksart](#) im Feld [Typ](#) und danach [Objekt anlegen](#).
4. Geben Sie im daraufhin eingeblendeten Dialogfenster [Objekterzeugung](#) die folgenden Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
5. Wählen Sie [Anlegen und zu Objekt navigieren](#).

Die neu angelegte Ausdrucksart wird im Bereich [Objektmanager](#) angezeigt.

Hinweis

Wenn Sie [Anlegen](#) wählen, wird die neu angelegte Ausdrucksart auf der Registerkarte [Enthaltene Objekte](#) angezeigt.

6. Geben Sie im Bereich [Details](#) folgende Daten ein:
 - [Klasse](#)
Geben Sie den Namen der Klasse ein, über die die Schnittstelle implementiert wird.
 - [Schnittstelle](#)
Geben Sie den Namen der Schnittstelle ein, die die spezifischen Attribute der Ausdrucksart enthält.
 - [Rechercheklasse](#)
Geben Sie die Rechercheklasse ein. Die Rechercheklasse bietet eine Schnittstelle zum Suchen nach Objekten und gibt Details des Objekts zurück, die den Suchkriterien entsprechen.
 - [UI-Klasse](#)
Die UI-Klasse bietet ein entsprechendes Ausdrucksartobjekt, das zum Anlegen der Ausdrucksart benötigt wird.
 - [Datenaustauschklasse](#)
7. Aktivieren Sie die Ausdrucksart.
8. Sichern Sie die Ausdrucksart.

1.3.5 Datenobjekte anlegen

Verwendung

Mithilfe von verschiedenen Datenobjekten können Sie einen Kontext oder ein Ergebnis erstellen. Datenobjekte stellen in BRFplus Felder, Strukturen und Tabellen dar.

Es gibt drei Arten von Datenobjekten:

- Element-Datenobjekte
- Struktur-Datenobjekte
- Tabellen-Datenobjekte

Weitere Informationen

[Ein Element-Datenobjekt anlegen \[Seite 236\]](#)

[Ein Struktur-Datenobjekt anlegen \[Seite 241\]](#)

[Ein Tabellen-Datenobjekt anlegen \[Seite 242\]](#)

1.3.5.1 Creating an Element Data Object

Procedure

1. Open the application in the *Object Manager* panel.
2. Under the *Detail* section, choose the *Contained Objects* tab.
The *Contained Objects* tab page appears.
3. Choose *Data Object* from the *Type* field and choose *Create Object*.
4. In the *Object Creation* dialog box that appears, choose *Element* from the *Type* field.
The *Element* section appears.
5. Choose the type of element you want to create from the *Element Type* field, under the *Define Element Properties* section.
For more information, see [Data Object \[page 61\]](#)
6. Under the *General Data* section, enter a name for the element and choose *Create and navigate to object*.
The data object appears in the *Object Manager* panel.

i Note

If you choose [Create](#) button, the element data object appears under the [Contained Objects](#) tab. Click the name of the created object to define the element properties.

7. Add values to the element data object that you have created. You can create values, or refer to existing values, or bind the data object.




- **Create New Values**

1. In the [Domain Values](#) tab page, choose [Create Values](#).
2. In the [Object Creation](#) dialog box that appears, enter name and value for the data object.
3. Choose [Create](#).
4. Depending upon the element you create, the related value fields appear in the [Object Creation](#) dialog box. Enter the related values in the fields that appear.

The following table helps you to create new values and set element attributes:

Table 43:

Element Type	Values	Element Attributes(Optional)	Description
(all types)	Name and Text fields appear for all the element types. You can enter name and text appropriately for all the element types	(Optional) If you want to set additional attributes of an element, choose Element Attributes(Optional) tab.	
Text	Enter text in the Value field.	Length	You can enter an attribute with a maximum length of 255 in the Value field.
Number	Enter a number in the Value field	Length	You can enter an attribute with a maximum length of 31. The length represents the number of digits without exponent or sign.
		Decimals	You can enter up to 10 maximum number of decimal places.
		Positive values only	Select the checkbox to enter only positive values.
Boolean	You may choose True or False from the Value field.		

Element Type	Values	Element Attributes(Optional)	Description
Amount	Enter amount and currency in the <i>Value</i> and <i>Currency</i> fields.		
Quantity	Enter quantity and unit in the <i>Value</i> and <i>Unit</i> fields.		
Timepoint	<p>You can choose the following different timepoint options:</p> <ul style="list-style-type: none"> ○ Date In the <i>Date</i> field, choose . Choose the date from the calendar. ○ Time In the <i>Time</i> field that appears, enter time in hh:mm:ss format. ○ Date Time In the <i>Date</i> and <i>Time</i> fields that appear, choose  and select the date from the calendar. In the <i>Time</i> field enter time in hh:mm:ss format. ○ Universal Time In the <i>Date</i> field, choose  and select the date from the calendar. In the <i>Time</i> field enter time in hh:mm:ss format. 		

Note

1. You can set element attributes only for *Number*, *Timepoint* and *Text* element types.
2. If the value of a constant does not fit to the element attributes (for example, exceeds the length) then you will get a warning message. You can still activate the object and such calculations are carried on with decfloat precession.
3. The element attributes only affect the design time and not the runtime of the BRFplus.

○ Select from Existing Values

1. In the [Domain Values](#) tab, choose [Select Values](#).
2. In the [Object Query](#) dialog box that appears, select the value and choose [Select](#).

- **Use an Existing OData-based Value List**

For analytical workflows, when an element belongs to a structure that is bound to a CDS view, you can refer to a value list that was provided by an OData service. (The OData service can be modeled in the same CDS view as the structure is bound to, or in a different CDS view.)

1. Choose the [Value Help](#) tab.
2. Enter the following settings for the OData service that provides the value help:
 - **Service URL:** The URL of the OData service with the format `/sap/opu/odata/sap/<EXTERNAL SERVICE NAME>`; for example, `/sap/opu/odata/sap/FLIGHT_CDS/`.
 - **Property Path:** The `Target` attribute of the value list `Annotations` element (from the OData service `$metadata`). In the following code sample, the property path is `FLIGHT_CDS.FLIGHTType/carrid`.

Sample Code

```
<Annotations xmlns="http://docs.oasis-open.org/odata/ns/edm"
  Target="FLIGHT_CDS.FLIGHTType/carrid">
  <Annotation Term="Common.ValueList">
  .....
  ...
  </Annotation>
</Annotations>
```

Note

The value help must conform to the following:

- The value help value (key) must be a string value.
- The value help can only have a single-value (key) field.

For more information, see:

- [Providing Value Help](#) in the *SAP - ABAP Programming Model for SAP Fiori* guide
 - [Value Help Conditions](#) in the *Rule Expression Language for Analytical Workflow* guide
 - **Bind the Element Data Object**
You can bind the data object if you want to have a value help from where you can select values. Under the [Element Properties](#) tab, choose one of the following options from the [Select Binding Type](#) dropdown list:
 - **Bind to an Existing Element**
 1. Choose [Bind an existing BRFplus element](#) option to bind the object to an existing element.
 2. In the [Object Query](#) dialog box that appears, select the data object.
 - **Bind to a Data Dictionary object**
 1. Choose the [Bind a data dictionary \(DDIC\) element](#) to bind the object to an element object in the data dictionary.
 2. In the [Bind DDIC](#) dialog box that appears, select the DDIC element and choose [Select](#).
- For more information, see [Data Binding \[page 66\]](#)

8. If you want to restrict the number of allowed comparison types for an element, choose the relevant allowed comparison from the [Allowed Comparisons](#) field on the [Element Properties](#) tab.

Example

You can set restrictions on comparisons of an element, for example, in decision tables or decision trees. If you select *Equals* comparison for an element that is used in a decision table, then you can have only one or no value and you cannot use less or greater comparisons.

The following table lists the allowed comparison types for an element:

Table 44:

Allowed Comparisons for an element	Description
No Restriction	Data object can be used in all types of comparison operations which are supported by the system for the particular type of the data object. This is the default for elementary data objects of type <i>Text</i> .
Equals	Data object can only be tested for equality against one single value.
Single Value	Data object can only be tested for equality or inequality against one single value. This is the default for elementary data objects of type <i>Boolean</i> .
Value List	Data object can only be tested for equality or inequality against a list of values.
Non-textual	Data object can be used in all types of comparison except for string comparisons (like contains string or matches pattern). This is the default for elementary data objects of the following types: <ul style="list-style-type: none">○ Amount○ Number○ Quantity○ Timepoint
User-Defined	Data object can be used in those types of comparison operations that are explicitly mentioned in a list of allowed operations. This list has to be defined individually for that particular data object. For example, you can define that a data object may be used in <i>greater than</i> and <i>greater than or equal to</i> comparisons, but not in <i>less than</i> and <i>less than or equal to</i> comparisons.

9. Activate the data object.
10. Save the data object.

Next Steps

[Comparison Operations \[page 70\]](#)

1.3.5.2 Struktur-Datenobjekt anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Registerkarte *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Datenobjekt* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Struktur* im Feld *Typ*.
5. Geben Sie einen Namen für das Struktur-Datenobjekt ein, und wählen Sie *Anlegen und zu Objekt navigieren*.

Sie werden direkt zum Struktur-Datenobjekt navigiert, das sie angelegt haben.

Hinweis

Wenn Sie *Anlegen* wählen, wird das angelegte Struktur-Datenobjekt auf der Registerkarte *Enthaltene Objekte* angezeigt. Wählen Sie den Namen des angelegten Strukturobjekts aus.

6. Definieren Sie die Datenbindung.
Sie können das Datenobjekt an eine Data-Dictionary- (DDIC) oder eine GDT-Struktur binden. In der Dropdown-Liste *Bindungstyp auswählen* stehen folgende Optionen zur Verfügung:
 - Wählen Sie *DDIC-Struktur anbinden* und im daraufhin angezeigten Dialogfenster *DDIC-Struktur anbinden* die gewünschte DDIC-Struktur und danach *Auswählen*.
 - Wählen Sie *GDT-Struktur anbinden* und im daraufhin angezeigten Dialogfenster *GDT-Struktur anbinden*, und geben Sie die erforderlichen Details ein.Weitere Informationen finden Sie unter [Datenbindung \[Seite 66\]](#).
7. Definieren Sie die Struktur. Sie können der Struktur ein vorhandenes oder ein neues Datenobjekt hinzufügen.
 - **Neues Datenobjekt hinzufügen**
 1. Wählen Sie *Neues Datenobjekt hinzufügen* und im Dropdown-Menü den gewünschten Datenobjekttyp.
 2. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* das Datenobjekt an.
 - **Vorhandenes Datenobjekt hinzufügen**
 1. Wählen Sie *Vorhandenes Datenobjekt hinzufügen*.

2. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus und wählen *Auswählen*.
8. Aktivieren Sie das Datenobjekt.
9. Sichern Sie die Änderungen.

1.3.5.3 Ein Tabellen-Datenobjekt anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Registerkarte *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Datenobjekt* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Tabelle* im Feld *Typ*.
5. Geben Sie einen Namen für die Tabelle ein, und wählen Sie *Anlegen und zu Objekt navigieren*.

Sie werden direkt zum Tabellendatenobjekt navigiert.

Hinweis

Wenn Sie *Anlegen* wählen, wird das angelegte Tabellendatenobjekt auf der Registerkarte *Enthaltene Objekte* angezeigt. Wählen Sie den Namen des angelegten Datenobjekts aus.

6. Wenn Sie zum Auswählen der Werte die Werthilfe nutzen möchten, binden Sie das Datenobjekt an.
 1. Wählen Sie *Aktuelles Objekt sollte gebunden sein* im Feld *DDIC-Binding*.
 2. Wählen Sie *DDIC-Tabellentyp anbinden*.
 3. Wählen Sie im Dialogfenster *DDIC anbinden* einen DDIC-Tabellentyp und wählen Sie dann *Auswählen*.

Weitere Informationen finden Sie unter [Datenbindung \[Seite 66\]](#).


7. Weisen Sie einen Tabellenzeilentyp zu.

Sie können auf ein vorhandenes Datenobjekt verweisen oder ein neues anlegen.


Hinweis

Einen Tabellenzeilentyp können Sie nur zuweisen, wenn Sie *Aktuelles Objekt sollte nicht gebunden sein* im Feld *DDIC-Binding* auswählen.

- **Vorhandenes Datenobjekt auswählen**

1. Wählen Sie  neben *Tabellenzeilentyp*.
2. Wählen Sie im Kontextmenü *Auf vorhandenen BRFplus Typ verweisen...*
3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.

- **Neues Datenobjekt anlegen**

1. Wählen Sie  neben *Tabellenzeilentyp*.
2. Wählen Sie im Kontextmenü *Neuen BRFplus Typ anlegen...*
3. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* ein Element-Datenobjekt an. Weitere Informationen finden Sie unter *Ein Element-Datenobjekt anlegen [Seite 236]*.
4. Definieren Sie im Bereich *Element* die Bindung.
 - **Objekt anbinden**
 1. Wählen Sie *Aktuelles Objekt sollte gebunden sein* im Feld *Datenflussdefinition*.
 2. Wählen Sie den Bindungstyp.
Sie können auswählen zwischen *DDIC-Element anbinden* oder *GDT-Element anbinden* oder *Vorhandenes FDT-Element anbinden*.
5. Wählen Sie die Elementart.

Hinweis

Eine Elementart können Sie nur definieren, wenn Sie *Aktuelles Objekt sollte nicht gebunden sein* im Feld *Datenflussdefinition* auswählen.

8. Aktivieren Sie das Datenobjekt.
9. Sichern Sie die Änderungen.

1.3.6 Regel anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
3. Wählen Sie auf der Seite *Enthaltene Objekte* das Objekt *Regel* im Feld *Typ* und danach *Objekt anlegen*.
4. Geben Sie im daraufhin eingeblendeten Dialogfenster *Objekterzeugung* die folgenden Daten ein:
 - Name
 - (Optional) Kurztext
 - (Optional) Text
5. Wählen Sie *Anlegen und zu Objekt navigieren*.

Die neu angelegte Regel wird im Bereich *Objektmanager* angezeigt.

Hinweis

Wenn Sie *Anlegen* wählen, wird das angelegte Regelobjekt auf der Registerkarte *Enthaltene Objekte* angezeigt.

6. Weisen Sie der Regel eine Bedingung zu.

Als Bedingung können Sie einen Standardtyp, ein vorhandenes Kontextdatenobjekt oder einen vorhandenen oder neuen Ausdruck hinzufügen.

1. „Wertebereich“ als Bedingung verwenden
 - **Standardtyp auswählen**
 1. Wählen Sie  neben *Kein Ausdruck zugeordnet*.
 2. Wählen Sie im Kontextmenü ► *Wertebereich verwenden von* ► *Standardobjekte* ►, und wählen Sie den Standardtyp aus.
 - **Vorhandenes Kontextdatenobjekt auswählen**
 1. Wählen Sie  neben *Kein Ausdruck zugeordnet*.
 2. Wählen Sie im Kontextmenü ► *Wertebereich verwenden von* ► *Anderer Kontextparameter* ►.
 3. Im Dialogfenster *Kontextabfrage* wählen Sie das Datenobjekt aus.
2. „Ausdruck“ und „Datenobjekt“ als Bedingung verwenden
 - **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie  neben *Kein Ausdruck zugeordnet*.
 2. Wählen Sie im Kontextmenü ► *Bedingung auswählen* ► *Ausdrücke auswählen* ► *Auswählen...* ►.
 3. Geben Sie im angezeigten Dialogfenster *Objektabfrage* den Namen des Ausdrucks in das Feld *Objektnamen* ein, und wählen Sie *Suchen*.
Der Ausdruck wird unterhalb der Drucktaste *Suchen* angezeigt.
 4. Wählen Sie den Ausdruck aus.
 - **Standarddatenobjekt auswählen**
 1. Wählen Sie  neben *Kein Ausdruck zugeordnet*.
 2. Wählen Sie im Kontextmenü ► *Bedingung auswählen* ► *Datenobjekt auswählen* ► *Standardobjekte* ►, und wählen Sie den Typ des Datenobjekts aus.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie  neben *Kein Ausdruck zugeordnet*.
 2. Wählen Sie im Kontextmenü ► *Bedingung auswählen* ► *Datenobjekt auswählen* ► *Anderen Kontextparameter* ►.
 3. Im Dialogfenster *Kontextabfrage* wählen Sie das Datenobjekt aus.
3. Neuen Ausdruck verwenden
 1. Wählen Sie  neben *Kein Ausdruck zugeordnet*.
 2. Wählen Sie *Ausdruck anlegen* aus dem Kontextmenü.
 3. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).

i Hinweis

Die Bedingung muss ein Kontextdatenobjekt oder ein verschachtelter Ausdruck sein, dessen Wert oder Ergebnis in einen Booleschen Wert umgewandelt werden kann.

7. Weisen Sie der Regel eine Aktion zu.
 - **Aktion zuordnen**

Sie können eine vorhandene Aktion wählen oder eine neue anlegen.
 - **Vorhandene Aktion auswählen**
 1. Wählen Sie *Hinzufügen* neben *Dann*.

2. Wählen Sie im Kontextmenü ► [Aktion durchführen](#) ► [Auswählen](#) ►.
3. Im Dialogfenster [Objektabfrage](#) wählen Sie die Aktion aus.
- **Neue Aktion anlegen**
 1. Wählen Sie [Hinzufügen](#) neben [Dann](#).
 2. Wählen Sie im Kontextmenü ► [Aktion durchführen](#) ► [Anlegen](#) ►.
 3. Im Dialogfenster [Objekterzeugung](#) legen Sie die Aktion an.
Weitere Informationen finden Sie unter [Aktionsarten anlegen](#) [Seite 252].
- **Wert dem Kontext zuordnen**
 - **Vorhandenen Kontextparameter auswählen**
 1. Wählen Sie [Hinzufügen](#) neben [Dann](#).
 2. Wählen Sie im Kontextmenü ► [Wert dem Kontext zuordnen](#) ► [Anderer Kontextparameter](#) ►.
 3. Im Dialogfenster [Kontextabfrage](#) wählen Sie das Datenobjekt aus.
 4. Geben Sie Werte für das Datenobjekt ein.
 - **Ausdruck zuordnen**
 - **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie [Hinzufügen](#) neben [Dann](#).
 2. Wählen Sie im Kontextmenü ► [Ausdruck verarbeiten](#) ► [Auswählen](#) ►.
 3. Im Dialogfenster [Objektabfrage](#) wählen Sie den Ausdruck aus.
 - **Neuen Ausdruck anlegen**
 1. Wählen Sie [Hinzufügen](#) neben [Dann](#).
 2. Wählen Sie im Kontextmenü ► [Ausdruck verarbeiten](#) ► [Anlegen](#) ►.
 3. Im Dialogfenster [Objekterzeugung](#) legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen](#) [Seite 196].
8. Fügen Sie auf dieselbe Weise Aktionen zum Abschnitt [Sonst](#) der Regel hinzu.
9. Aktivieren Sie die Regel.
10. Sichern Sie die Regel.

1.3.7 Regelsatz anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich [Objektmanager](#).
2. Im Abschnitt [Details](#) wählen Sie die Registerkarte [Enthaltene Objekte](#).
Die Registerkarte [Enthaltene Objekte](#) wird angezeigt.
3. Wählen Sie [Regelsatz](#) im Feld [Typ](#) und danach [Objekt anlegen](#).
4. Geben Sie im daraufhin eingeblendeten Dialogfenster [Objekterzeugung](#) die folgenden Daten ein:
 - Name

- (Optional) Kurztext
- (Optional) Text

5. Wählen Sie *Anlegen und zu Objekt navigieren*.

Der Regelsatz wird im Bereich *Objektmanager* geöffnet.

6. Fügen Sie dem Regelsatz eine Vorbedingung hinzu.

1. Wählen Sie *Vorbedingung zuordnen*.
2. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.
Der Ausdruck wird unterhalb des Abschnitts *Details* angezeigt.

Hinweis

Der ausgewählte Ausdruck muss ein Boolesches Ergebnis zurückgeben.

7. Weisen Sie der Regel eine Funktion zu.

1. Wählen Sie *Funktion zuordnen*.
2. Im Dialogfenster *Objektabfrage* wählen Sie die Funktion aus.
Die Funktion wird unterhalb des Abschnitts *Details* angezeigt.

8. Weisen Sie dem Regelsatz Regeln zu. Sie können eine vorhandene Regel hinzufügen oder eine neue anlegen.

Neue Regel anlegen

1. Wählen Sie *Regel einfügen* *Anlegen* .
2. Geben Sie die Beschreibung für die Regel ein.
3. Weisen Sie der Regel eine Bedingung und eine Aktion zu.
Weitere Informationen finden Sie unter [Regel anlegen \[Seite 243\]](#).

Vorhandene Regel auswählen

1. Wählen Sie *Regel einfügen* *Anlegen* .
 2. Im Dialogfenster *Objektabfrage* wählen Sie die Regel aus.
Die Regel wird unterhalb des Abschnitts *Regeln* angezeigt.
9. Fügen Sie eine Exit-Bedingung hinzu.

Sie können eine Bedingung anlegen oder auswählen.

- **Bedingung anlegen**

1. Wählen Sie *Exit-Bedingung einfügen* *Anlegen* .

- Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck als eine Bedingung an.
 - **Bedingung auswählen**
 1. Wählen Sie ► *Exit-Bedingung einfügen* ► *Auswählen* ►.
 2. Im Dialogfenster *Objekterzeugung* wählen Sie den Ausdruck als eine Bedingung aus.
10. Fügen Sie eine weitere Regel hinzu. Sie können eine vorhandene Regel hinzufügen oder eine neue anlegen.
- **Regel anlegen**
 - Wählen Sie ► *Regeloperationen* ► *Nächste Regel einführen durch...* ► *Anlegen* ►.
Die Funktion wird unterhalb des Abschnitts *Regeln* angezeigt.
Weitere Informationen finden Sie unter *Regel anlegen* [Seite 243].
 - **Vorhandene Regel hinzufügen**
 1. Wählen Sie ► *Regeloperationen* ► *Nächste Regel einführen durch...* ► *Auswählen* ►.
 2. Im Dialogfenster *Objektabfrage* wählen Sie die Regel aus.
11. Aktivieren Sie den Regelsatz.
12. Sichern Sie den Regelsatz.

Nächste Schritte

[Regel anlegen](#) [Seite 243]

[Aktionsarten anlegen](#) [Seite 252]

1.3.8 Mit Katalogen arbeiten

Verwendung

In diesem Abschnitt werden die verschiedenen Aufgaben beschrieben, die Sie für einen Katalog in BRFPplus ausführen müssen.

Weitere Informationen

[Katalog anlegen](#) [Seite 248]

[Objekttypen bearbeiten](#) [Seite 251]

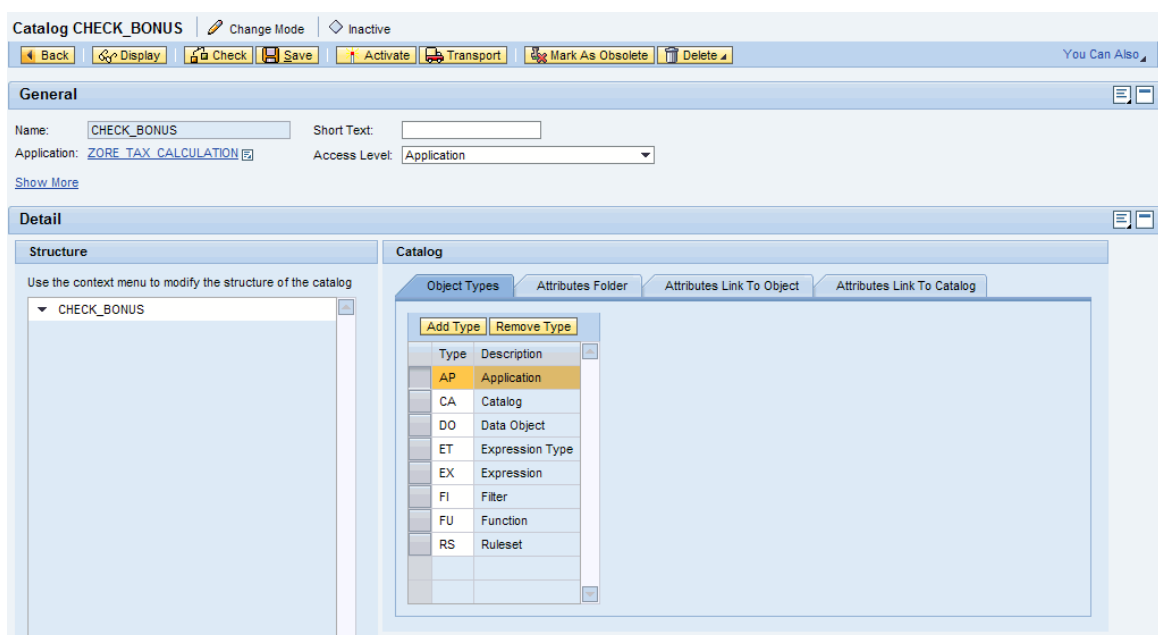
[Attribute hinzufügen](#) [Seite 251]

1.3.8.1 Katalog anlegen

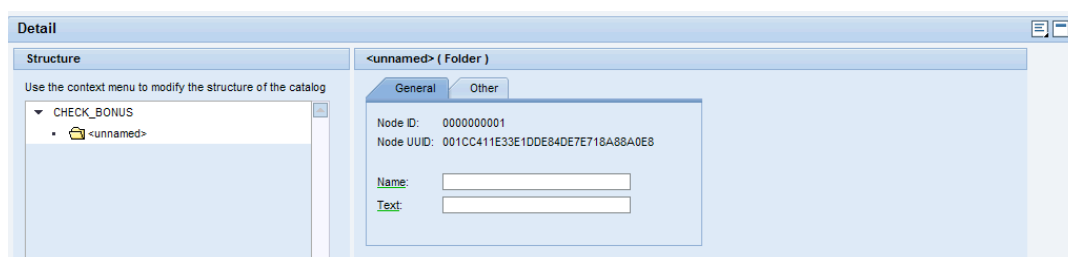
Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Sie Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie im Feld *Typ* die Option *Katalog* und dann *Objekt anlegen*.
4. Geben Sie im Feld *Objekterzeugung*, das nun angezeigt wird, den Namen des Katalogs ein, und wählen Sie *Anlegen und zu Objekt navigieren*.

Der neu angelegte Katalog wird im Bereich *Objektmanager* angezeigt.



5. Sie können dem Wurzelknoten des Katalogs Ordner-, Objekt- und Verknüpfungsknoten hinzufügen.
 - **Ordnerknoten hinzufügen**
 1. Geben Sie im Unterabschnitt *Struktur* im Kontextmenü des Katalogs *Ordnerknoten anlegen...*.
Der Ordnerknoten wird unterhalb des Wurzelknotens angezeigt.



2. Geben Sie auf der Registerkarte *Allgemein* folgende Daten ein:

- Name
Der Name wird in der Baumhierarchie der Katalogpflegesicht angezeigt. Der Name wird als Backup verwendet, wenn der Text nicht verfügbar ist.
- Text
Der Text wird im Katalog-Browser-Modus angezeigt.

Hinweis

Sie können dem Ordner auch Unterknoten hinzufügen.

- **Objektknoten hinzufügen**


Der Objektknoten verweist auf BRFplus-Objekte. Sie können einen Einstiegsknoten hinzufügen, ein vorhandenes Objekt auswählen oder ein neues Objekt hinzufügen.

Wenn Sie einen Knoten ohne Bindung anlegen möchten, verwenden Sie die Option "Einstiegsknoten anlegen".


- **Einstiegsknoten anlegen**

1. Wählen Sie im Unterabschnitt *Struktur* im Kontextmenü des Katalogs ► *Objektknoten anlegen* ► *Einstiegsknoten anlegen...* ►.
Der Objektknoten wird angezeigt.
2. Geben Sie auf der Registerkarte *Allgemein* den Namen des Blattknotens ein.
3. Ordnen Sie dem Objektknoten ein Objekt zu. Sie können dem Objektknoten entweder ein vorhandenes oder ein neues Objekt zuordnen.

- **Vorhandenes Objekt hinzufügen**

1. Wählen Sie auf der Registerkarte *Allgemein* die Option .
2. Wählen Sie im Kontextmenü *Objekt auswählen*.
3. Geben Sie im angezeigten Dialogfenster *Objektabfrage* den Namen des Ausdrucks in das Feld *Objektnamen* ein, und wählen Sie den Objekttyp.
4. Wählen Sie *Suchen*.
5. Markieren Sie das Objekt.

- **Neues Objekt hinzufügen**


1. Wählen Sie .
2. Wählen Sie im Kontextmenü *Objekt anlegen*.
3. Geben Sie im Abschnitt *Allgemeine Daten* den Namen des Objekts ein, und wählen Sie im Feld *Typ* den Typ aus.
4. Wählen Sie *Anlegen und zu Objekt navigieren*.
Das Objekt wird im Bereich *Objektmanager* geöffnet.

Hinweis

Wenn Sie *Anlegen* wählen, wird das Objekt angelegt und dem Objektknoten zugeordnet.

- **Vorhandenes Objekt auswählen**

1. Wählen Sie im Unterabschnitt *Struktur* im Kontextmenü des Katalogs ► *Objektknoten anlegen* ► *Vorhandenes Objekt auswählen...* ►.
2. Geben Sie im angezeigten Dialogfenster *Objektabfrage* den Namen des Ausdrucks in das Feld *Objektnamen* ein, und wählen Sie den Objekttyp.

3. Wählen Sie [Suchen](#).
Das Objekt wird unterhalb der Drucktaste [Suchen](#) angezeigt.
4. Markieren Sie das Objekt.
- **Neues Objekt anlegen**
 - Wählen Sie im Unterabschnitt [Struktur](#) im Kontextmenü des Katalogs [Objektknoten anlegen](#), und legen Sie das Objekt an.
- **Verknüpfung zum Katalog hinzufügen**
 - **Einstiegs-knoten anlegen**
 1. Wählen Sie im Unterabschnitt [Struktur](#) im Kontextmenü ► [Katalogknoten anlegen](#) ► [Einstiegs-knoten anlegen...](#) ►.
 2. Geben Sie den Namen des Verknüpfungsknotens ein.
 3. Wählen Sie .
 4. Wählen Sie im Kontextmenü [Katalogknoten auswählen](#).
 5. Wählen Sie im Dialogfeld [Knoten auswählen](#), das nun angezeigt wird, auf der Registerkarte [Katalog auswählen](#) den Katalog aus, und wählen Sie [Auswählen](#).
 6. Wählen Sie auf der Registerkarte [Knoten auswählen](#), die nun geöffnet wird, den Knoten aus, und wählen Sie [OK](#).
Der Katalog wird der Verknüpfung zugeordnet.

i Hinweis

Der Wurzelknoten kann ebenfalls verknüpft werden.

- **Vorhandenen Katalogknoten auswählen**
 1. Wählen Sie im Unterabschnitt [Struktur](#) im Kontextmenü ► [Katalogknoten anlegen](#) ► [Katalogknoten auswählen...](#) ►.
 2. Wählen Sie im Dialogfeld [Knoten auswählen](#), das nun angezeigt wird, auf der Registerkarte [Katalog auswählen](#) den Katalog aus, und wählen Sie [Auswählen](#).
 3. Wählen Sie auf der Registerkarte [Knoten auswählen](#), die nun geöffnet wird, den Knoten aus, und wählen Sie [OK](#).
Der Katalog wird der Verknüpfung zugeordnet.

i Hinweis

Der Wurzelknoten kann ebenfalls verknüpft werden.

6. Aktivieren Sie den Katalog.
7. Sichern Sie den Katalog.


Nächste Schritte

[Attribute hinzufügen \[Seite 251\]](#)

[Objekttypen bearbeiten \[Seite 251\]](#)

1.3.8.2 Objekttypen bearbeiten

Vorgehensweise

1. Öffnen Sie den Katalog im Bereich *Objektmanager*.
2. Wählen Sie in der Registerkarte *Objekttypen Typ hinzufügen*.
Es wird eine neue Zeile in die Tabelle eingefügt.
3. Geben Sie in der neuen Zeile den Objekttyp ein.
Sie können den Objekttyp auch aus einer Liste auswählen.
 1. Wählen Sie .
 2. Wählen Sie im Dialogfenster den Typ und dann *OK*.

Nächste Schritte

[Katalog anlegen \[Seite 248\]](#)

[Attribute hinzufügen \[Seite 251\]](#)

1.3.8.3 Attribute hinzufügen

Vorgehensweise

Der Wurzelknoten des Katalogs enthält die Eigenschaften des Katalogs.

- **Attribute zum Strukturknoten hinzufügen**
 1. Öffnen Sie den Katalog im Bereich *Objektmanager*.
 2. Wählen Sie unter *Katalog* die Registerkarte *Attributordner*.
Die Registerkarte *Attributordner* wird angezeigt.
 3. Wählen Sie *Attribut hinzufügen*.
 4. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
- **Attribute zum Objektknoten hinzufügen**
 1. Öffnen Sie den Katalog im Bereich *Objektmanager*.
 2. Wählen Sie unter *Katalog* die Registerkarte *Attribute mit Objekt verknüpfen*.
Die Registerkarte *Attribute mit Objekt verknüpfen* wird angezeigt.
 3. Wählen Sie *Attribut hinzufügen*.
 4. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
Das Datenobjekt wird zum Objektknoten hinzugefügt.

- **Attribute zum Verknüpfungsknoten hinzufügen**

1. Öffnen Sie den Katalog im Bereich *Objektmanager*.
2. Wählen Sie unter *Katalog* die Registerkarte *Attribute mit Katalog verknüpfen*.
Die Registerkarte *Attribute mit Katalog verknüpfen* wird angezeigt.
3. Wählen Sie *Attribut hinzufügen*.
4. Im Dialogfenster *Objektanfrage* wählen Sie das Datenobjekt aus.
Das Datenobjekt wird zum Verknüpfungsknoten hinzugefügt.

Hinweis

Für jeden Knotentyp können Sie sechs verschiedene Attribute definieren.

Weitere Informationen

[Katalog anlegen \[Seite 248\]](#)

[Objekttypen bearbeiten \[Seite 251\]](#)

1.3.9 Aktionsarten anlegen

Verwendung

Dieser Abschnitt beschreibt die Schritte, die durchgeführt werden müssen, um verschiedene Aktionsarten anzulegen.

Weitere Informationen

[Nachrichtenprotokollaktion anlegen \[Seite 252\]](#)

[E-Mail versenden Aktion anlegen \[Seite 255\]](#)

[Statischer Methodenaufwurf Aktion anlegen \[Seite 256\]](#)

1.3.9.1 Meldungsprotokollaktion anlegen

Verwendung

Die Meldungsprotokollaktion wird verwendet, um Meldungen in ein Anwendungsprotokoll zu schreiben. Meldungen werden anhand der Meldungsklassendefinition im Backend erstellt oder durch das Eingeben von

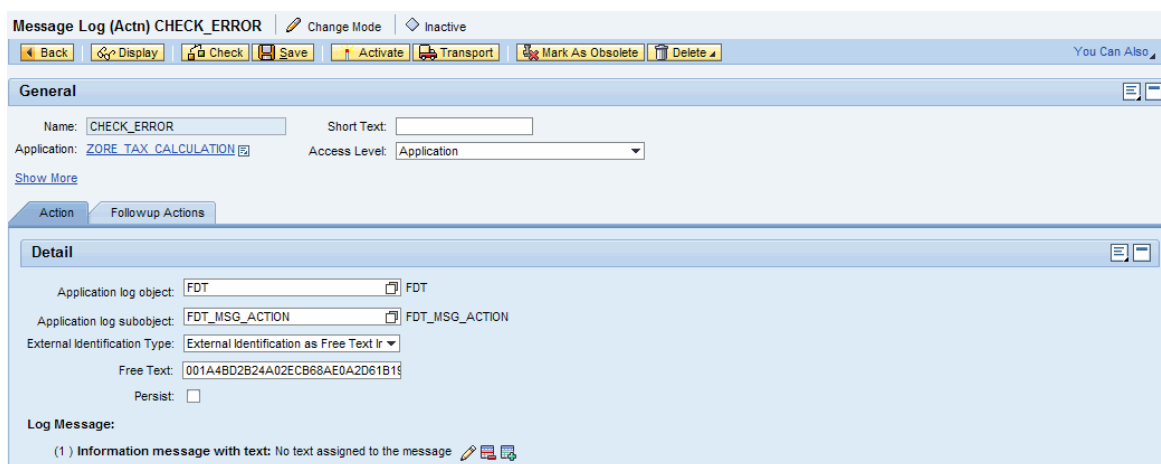
Freitext mit Parametern. Die Parameter der Meldungen werden direkt gefüllt oder von den Kontextdatenobjekten oder einem eingebetteten Ausdruck zur Laufzeit abgeleitet.

Vorgehensweise





1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Die Registerkarte *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Aktion* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterstellung* wählen Sie *Meldungsprtkll (Akt)* im Feld *Typ*.
5. Geben Sie den Namen der Aktion ein.
6. Wählen Sie *Anlegen und zu Objekt navigieren*.
Sie werden direkt zum Aktionsobjekt navigiert, das sie angelegt haben.

Hinweis

Wenn Sie *Anlegen* wählen, erscheint das Aktionsobjekt unter der Registerkarte *Enthaltene Objekte*.



7. Geben Sie im Bereich *Detail* das Anwendungsprotokollobjekt und -Teilobjekt ein.
Die Felder *Anwendungsprotokollobjekt* und *Anwendungsprotokoll-Teilobjekt* geben den Ort an, an dem sich Objekt und Teilobjekt befinden.
8. Wählen Sie die externe Identifikationsart. Sie können eine der folgenden Optionen wählen:
 - **Keine externe Identifikation im Protokoll**
 - Wählen Sie *Keine externe Identifikation im Protokoll*
 - **Externe Identifikation als Freitexterfassung**
 1. Wählen Sie *Externe Identifikation als Freitexterfassung*.
Das Freitextfeld erscheint.
 2. Geben Sie den Text in das Feld *Freitext* ein.
 - **Externe Identifikation als Referenzobjekt**
 1. Wählen Sie *Externe Identifikation als Referenzobjekt*.
Das Referenzobjektfeld erscheint.
 2. Ordnen Sie das Referenzobjekt zu. Sie können einen Standardtyp, ein vorhandenes Datenobjekt oder einen vorhandenen Ausdruck auswählen.

- **Standardtyp auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Standardtyp.
 - **Vorhandenes Datenobjekt auswählen**
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Andere Kontextparameter*.
 3. Im Dialogfenster *Kontextabfrage* wählen Sie das Datenobjekt aus.
 - **Vorhandenen Ausdruck auswählen**
 1. Wählen Sie .
 2. Wählen Sie *Ausdrücke...* aus dem Kontextmenü.
 3. Geben Sie im angezeigten Dialogfenster *Objektabfrage* den Namen der Anwendung in das Feld *Objektnamen* ein, und wählen Sie *Suchen*.
Der Ausdruck wird unterhalb der Drucktaste *Suchen* angezeigt.
 4. Wählen Sie den Ausdruck aus.
9. Wählen Sie zum Sichern der Meldungen in der Datenbank das Ankreuzfeld *Persistieren*.
10. Wählen Sie .
- Eine Meldungsvorlage erscheint.
Sie können entweder eine vorhandene Meldung auswählen oder eine neue anlegen.

Log Message:

(1) Information message with text: No text assigned to the message

Use Predefined Message Text: Type: Information

- **Neue Nachricht anlegen**
 1. Geben Sie den Text der Meldung ein.
 2. Wählen Sie die Art der Aktionsmeldung im Feld *Art* aus.
Die unterschiedlichen Arten von Aktionsmeldungen sind:
 - Information
 - Abbruch
 - Fehler
 - Status
 - Warnung
 - Exit
 - **Vorhandene Meldung verwenden**
 1. Wählen Sie die Drucktaste *Vordefinierte Nachricht verw.*.
 2. Geben Sie die Meldungsklasse, -nummer und -art ein.
Enthält die Meldung dynamische Platzhalter, geben Sie den Wert entweder direkt ein oder verwenden Sie die Referenz.
11. Aktivieren Sie die Aktion.
12. Sichern Sie die Aktion.

Hinweis

Wenn Sie eine Meldung mit Parametern einer Meldungsprotokollaktion zuordnen möchten, stellen Sie sicher, dass die im Meldungskurztext verwendeten Parameter nummeriert sind (&1, &2, &3, &4). Auch

wenn es syntaktisch zulässig ist, die Nummern auszulassen, erkennt BRFplus solche unqualifizierten Parameter nicht als Parameter und zeigt Sie als „&“ an, statt sie mit einem Wert zu ersetzen.

Weitere Informationen

[Nachfolgeaktionen hinzufügen \[Seite 258\]](#)

1.3.9.2 E-Mail versenden Aktion anlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Sie Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Aktion* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterstellung* wählen Sie *E-Mail versenden (Akt)* im Feld *Typ*.
5. Geben Sie den Namen der Aktion ein.
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Sie werden zum angelegten Aktionsobjekt weitergeleitet.





i Hinweis

Wenn Sie *Anlegen* wählen, erscheint das Aktionsobjekt unter der Registerkarte *Enthaltene Objekte*.

7. Geben Sie im Bereich *Details* folgende Daten ein:
 - Die E-Mail-Adresse des Empfängers im Feld *An*.
Sie können auch die E-Mail-Adressen mehrerer Empfänger eingeben, indem Sie diese mit einem Semikolon (;) trennen.
 - Geben Sie Betreff und Text der E-Mail ein.

i Hinweis

Für Betreff und Text der E-Mail können Sie dynamische Platzhalter eingeben und diese mit einem Datenobjekt oder einem Ausdruck verbinden. Ein Platzhalter besteht aus einem Ampersand und einem numerischen Wert, z. B. &1, &2... Sobald der Versand der E-Mail ausgelöst wird, bewertet das System die Datenobjekte oder Ausdrücke. Die Platzhalter werden dann in der E-Mail durch diesen Wert ersetzt.

1. Geben Sie Platzhalter in den Feldern *Betreff* und *Körper* ein.
 2. Wählen Sie die Drucktaste *Platzhalter aktualisieren*.
Der Bereich *Platzhalter in Nachricht* wird angezeigt.
 3. Sie können die Platzhalter mit einem Standardtyp, einem vorhandenen Datenobjekt oder einem Ausdruck verbinden.
 - 1. *Mit Standardtyp verbinden*
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Standardobjekte* und danach den Standardtyp.
 - 2. *Mit vorhandenem Datenobjekt verbinden*
 1. Wählen Sie .
 2. Wählen Sie im Kontextmenü *Anderer Kontextparameter*.
 3. Im Dialogfenster *Objektabfrage* wählen Sie das Datenobjekt aus.
 - 3. *Mit vorhandenem Ausdruck verbinden*
 1. Wählen Sie .
 2. Wählen Sie *Ausdrücke* aus dem Kontextmenü.
 3. Im Dialogfenster *Objektabfrage* wählen Sie den Ausdruck aus.
 - 4. *Mit neuem Ausdruck verbinden*
 1. Wählen Sie .
 2. Wählen Sie *Ausdruck anlegen* aus dem Kontextmenü.
 3. Im Dialogfenster *Objekterzeugung* legen Sie den Ausdruck an.
Weitere Informationen finden Sie unter [Ausdrücke anlegen \[Seite 196\]](#).
 4. Führen Sie die oben für Platzhalter beschriebenen Schritte auch für den Textkörper der E-Mail aus.
8. Aktivieren und sichern Sie die Aktion.

Nächste Schritte

[Nachfolgeaktionen hinzufügen \[Seite 258\]](#)

[Aktionsarten anlegen \[Seite 252\]](#)

1.3.9.3 Statischer Methodenaufruf Aktion anlegen

Kontext

Eine statische Methodenaufruf Aktion dient zum Aufrufen einer statischen Methode in einer Klasse, die eine spezielle Schnittstelle implementiert. Hierfür benötigen Sie eine Klasse, die die Schnittstelle implementiert und dann die statische Methode dieser Klasse aufruft.

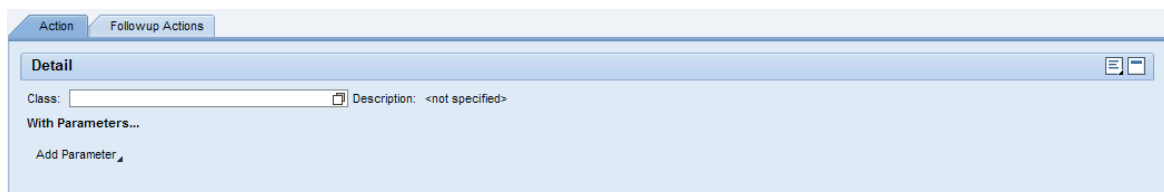
Vorgehensweise




1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Im Abschnitt *Details* wählen Sie die Registerkarte *Enthaltene Objekte*.
Sie Seite *Enthaltene Objekte* wird angezeigt.
3. Wählen Sie *Aktion* im Feld *Typ* und danach *Objekt anlegen*.
4. Im Dialogfenster *Objekterzeugung* wählen Sie *Statische Methode (Akt.)* im Feld *Typ*.
5. Geben Sie den Namen der Aktion ein.
6. Wählen Sie *Anlegen und zu Objekt navigieren*.

Sie werden direkt zum Aktionsobjekt navigiert, das Sie angelegt haben.

Hinweis

Wenn Sie *Anlegen* wählen, wird das angelegte Aktionsobjekt auf der Registerkarte *Enthaltene Objekte* angezeigt. Klicken Sie auf den Namen des angelegten Objekts, um die Aktionsmeldungen zu definieren.



7. Geben Sie den Namen der Klasse ein oder wählen Sie  im Feld *Klasse*.
8. Wählen Sie die relevante Klasse in der angezeigten Liste aus und wählen Sie *OK*.
Der ausgewählte Klasse wird hinzugefügt.
9. Fügen Sie Parameter zur Aktion hinzu.
Sie können einen vorhandenen Ausdruck hinzufügen oder einen neuen anlegen.
 - **Vorhandenen Ausdruck hinzufügen**
 1. Wählen Sie im Bereich *Details* ► *Parameter hinzufügen* ► *Auswählen...* .
 2. Geben Sie im angezeigten Dialogfenster *Objektabfrage* den Namen des Ausdrucks in das Feld *Objektname* ein, und wählen Sie *Suchen*.
Der Ausdruck wird unterhalb der Drucktaste *Suchen* angezeigt.
 3. Wählen Sie den Ausdruck aus.
 - **Neuen Ausdruck hinzufügen**
 1. Wählen Sie im Bereich *Details* ► *Parameter hinzufügen* ► *Anlegen...* .
 2. Im Dialogfenster *Objekterzeugung* geben Sie den Namen des Ausdrucks ein, wählen die Ausdrucksart und dann *Anlegen*.
Weitere Informationen finden Sie unter *Ausdrücke anlegen* [Seite 196].
10. Aktivieren Sie die Aktion.
11. Sichern Sie die Aktion.

Nächste Schritte

[Nachfolgeaktionen hinzufügen \[Seite 258\]](#)

1.3.9.4 Nachfolgeaktionen hinzufügen

Vorgehensweise

Nachfolgeaktionen können zu Meldungsprotokollaktionen und Aktionen für den statischen Methodenaufruf hinzugefügt werden.

1. Öffnen Sie die Aktion im Bereich *Objektmanager*.
2. Im Abschnitt *Allgemein* wählen Sie die Registerkarte *Nachfolgeaktionen*.
Die Seite *Nachfolgeaktionen* wird geöffnet.
3. Sie können eine neue Aktion hinzufügen oder eine vorhandene Aktion zur Meldungsprotokollaktion hinzufügen.
 - **Neue Aktion hinzufügen**
 1. Wählen Sie *Neue Aktion hinzufügen*.
 2. Legen Sie im jetzt angezeigten Dialogfenster *Objekterzeugung* eine neue Aktion an.
 - **Vorhandene Aktion hinzufügen**
 1. Wählen Sie *Vorhandene Aktion hinzufügen*.
 2. Im jetzt angezeigten Dialogfenster *Objektabfrage* wählen Sie die gewünschte Aktion aus und danach *Auswählen*.
Die ausgewählte Aktion wird als Nachfolgeaktion hinzugefügt.
4. Aktivieren Sie die Aktion.
5. Sichern Sie die Aktion.

Weitere Informationen

[Meldungsprotokollaktion anlegen \[Seite 252\]](#)

[Statischer Methodenaufruf \(Aktion\) anlegen \[Seite 256\]](#)

[E-Mail versenden \(Aktion\) anlegen \[Seite 255\]](#)

1.3.10 Konfiguration der Benutzereinstellungen

Verwendung

Sie können die Benutzungsoberfläche gemäß Ihren Anforderungen und persönlichen Präferenzen konfigurieren.

Vorgehensweise

1. Wählen Sie in der Menüleiste ► [Workbench](#) ► [Personalisieren](#) ►.
2. Nehmen Sie im angezeigten Dialogfenster [Personalisierung](#) die gewünschten Änderungen in den verschiedenen Registerkarten vor.
3. Wählen Sie [Sichern](#), damit die aktuellen Einstellungen wirksam werden.

Hinweis

Sie können alle im Dialogfenster [Personalisierung](#) angebotenen Einstellungen auf die Systemeinstellungen zurücksetzen. Klicken Sie hierzu auf [Auf Standardeinstellungen zurücksetzen](#).

Einige der Personalisierungseinstellungen werden erst wirksam, nachdem die BRFplus Workbench neu gestartet wurde.

1.3.11 Objekte transportieren

Kontext


Mit BRFplus können Sie über einen Transportmechanismus System- und Customizing Objekte von einem System in ein anderes übertragen.

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich [Objektmanager](#).
2. Wählen Sie [Transport](#).
3. Im Dialogfenster [Bestätigung des Transports](#) geben Sie im Feld [Transportauftrag](#) die Transportanforderung ein.
4. Wählen Sie [OK](#).

1.3.12 Versionen festlegen

Vorgehensweise

1. Öffnen Sie die Anwendung im Bereich *Objektmanager*.
2. Wählen Sie  neben den Bereich *Allgemein*.
3. Wählen Sie *Versionierung einschalten* aus dem Kontextmenü.
4. Wählen Sie im Dialogfenster *Bestätigung der Änderung des Versionierungsmodus*, aktivieren Sie das Kontrollkästchen *Referenzierte Objekte einbinden* und wählen Sie *OK*.

Die Liste der betroffenen Objekte wird angezeigt.

5. Wählen Sie *OK*.

Nächste Schritte

[Versionierung \[Seite 176\]](#)

1.3.13 Aktuelle Objekte zu Favoriten hinzufügen

Vorgehensweise

1. Wählen Sie im Bereich *Navigation* das Objekt aus, welches zu den *Favoriten* hinzugefügt werden soll.
2. Wählen Sie in der Menüleiste ► *Workbench* ► *Aktuelles Objekt zu Favoriten hinzufügen...* ►

Das Objekt wird jetzt im Bereich *Navigation* im Abschnitt *Favoriten* angezeigt.

1.3.14 Verwendung von Objekten überprüfen

Kontext

Über den Verwendungsnachweis können Sie Beziehungswissen festlegen und abrufen. Die Überprüfung der Verwendung von Objekten ist erforderlich, wenn Sie das Trace- und Simulationstool nutzen. Der Verwendungsnachweis verhindert Inkonsistenzen, wenn Benutzer Objekte löschen möchten, die noch von anderen Objekten verwendet werden.

Vorgehensweise

1. Wählen Sie im Bereich *Navigation* die Anwendung aus, für die Sie die Objektverwendung überprüfen möchten.

Die Anwendung wird im Bereich *Objektmanager* geöffnet.

2. Wählen Sie die Registerkarte *Enthaltene Objekte*.

Die Seite *Enthaltene Objekte* wird angezeigt.

3. Wählen Sie im Feld *Typ* den Objekttyp aus.

4. Wählen Sie das Objekt aus der Liste, und wählen Sie *Alternativ können Sie auch > den Verwendungsnachweis anzeigen >*.

Die Verwendungen für das ausgewählte Objekt werden im Dialogfenster *Verwendungsnachweis* angezeigt.

1.3.15 Objekte in einer Anwendung löschen

Vorgehensweise

1. Wählen Sie die Anwendung, in der sich das Objekt befindet, im Bereich *Repository* oder *Zuletzt verwendet* aus.
2. Wählen Sie im Bereich *Details* unter *Enthaltene Objekte* den Objekttyp aus, den Sie im Feld *Typ* löschen möchten.

Eine Liste der ausgewählten Objekttypen wird angezeigt.

Beispiel

Wenn Sie ein Datenobjekt löschen möchten, wählen Sie *Datenobjekt* im Feld *Typ*.

3. Aus dieser Liste können Sie das zu löschende Objekt auswählen.

Hinweis

Bevor Sie das Objekt löschen, überprüfen Sie mithilfe des Verwendungsnachweises die Objektverwendung.

Weitere Informationen finden Sie unter [Verwendung von Objekten überprüfen \[Seite 261\]](#).

4. Sie können das Objekt entweder direkt löschen oder zum Löschen vormerken.
- Löschen eines Objekts, das in keiner Version mehr verwendet wird
 1. Wählen Sie ► [Löschen](#) ► [Löschen](#) ►.
 2. Wählen Sie im Dialogfenster [Bestätigung der Löschung OK](#).
 - Löschen eines Objekts, das zum Löschen vorgemerkt ist
 1. Wählen Sie ► [Löschen](#) ► [Nur Objekte löschen, die zum Löschen vorgemerkt sind](#) ►.
 2. Wählen Sie im Dialogfenster [Bestätigung der Löschung OK](#).
 - Vormerken eines Objekts, das in keiner Version mehr verwendet wird
 1. Wählen Sie ► [Löschen](#) ► [Zum Löschen vormerken](#) ►.
 2. Wählen Sie im Dialogfenster [Bestätigung des Vormerkens zur Löschung OK](#).

Oberhalb der Menüleiste wird eine Bestätigungsmeldung angezeigt.

Ergebnisse


Das Objekt wird gelöscht.

1.3.16 Mit Tools arbeiten

Verwendung

In diesem Abschnitt werden die verschiedenen in BRFplus verfügbaren Toolkomponenten beschrieben.

Hinweis

Bis auf das Simulationstool sind alle Tools nur im Expertenmodus verfügbar. In den Expertenmodus wechseln Sie, indem Sie in der oberen rechten Ecke des Workbench-Fensters  und danach [Benutzermodus](#) wählen.

- [Anwendungsverwendung prüfen \[Seite 266\]](#)
- [Nicht verwendete Objekte löschen \[Seite 269\]](#)
- [Objekte im Transportauftrag analysieren \[Seite 269\]](#)
- [Funktion simulieren \[Seite 270\]](#)
- [Während der Simulation Werte aus einer Excel-Datei importieren \[Seite 271\]](#)

- [Während der Simulation Inhalte in eine Excel-Datei exportieren \[Seite 272\]](#)
- [Web-Services generieren \[Seite 272\]](#)
- [XML-Daten im- und exportieren \[Seite 274\]](#)

1.3.16.1 Anwendungsverwaltung

Verwendung

Mit dem Tool zur Anwendungsverwaltung führen Sie verschiedene Vorgänge aus, um die BRFplus-Datenbank zu bereinigen und veraltete, nicht mehr verwendete Objekte oder Objektversionen zu entfernen. So enthält die Datenbank nur produktive Objekte, was wiederum zu einer Reduzierung potenzieller Probleme in Bezug auf übergreifende Abhängigkeiten und beim Transport führt.

Alle verfügbaren Vorgänge des Tools zur Anwendungsverwaltung basieren auf Reports, die im Backend über die Transaktion **FDT_HELPERS** aufgerufen werden können. Die Backend-Versionen der Reports verfügen häufig über noch mehr Funktionen als das Tool zur Anwendungsverwaltung. Deshalb empfiehlt SAP Systemadministratoren, die Backend-Version zu verwenden. Das Tool zur Anwendungsverwaltung richtet sich an Gelegenheitsbenutzer. Andererseits bietet das Tool zur Anwendungsverwaltung in der BRFplus-Workbench Vorteile, da es dem Benutzer einen ausführlichen Überblick darüber gibt, was ein Vorgang genau ausführt. Nachdem Sie eine Auswahl vorgenommen haben, sammelt das System alle betroffenen Objekte und zeigt diese in einer Liste an. Dieser Überblick auf der Ebene von Einzelobjekten ist nur im Tool zur Anwendungsverwaltung verfügbar, nicht im Backend.

Funktionsumfang

Das Tool zur Anwendungsverwaltung bietet folgende Vorgänge:

Auswahl

Alle im Tool zur Anwendungsverwaltung verfügbaren Vorgänge werden für den minimalen Objektumfang einer Anwendung ausgeführt. Einige der Vorgänge können für mehrere Anwendungen und zusätzliche Selektionskriterien wie Transportverhalten, Speichertyp, Anwendungskomponente, Softwarekomponente usw. ausgeführt werden.

Nachdem Sie den Objektumfang definiert und **Übernehmen** gewählt haben, ermittelt das System die Anwendungen, die den Kriterien entsprechen **und** die Objekte enthalten, die durch den auszuführenden Vorgang geändert werden. Diese Anwendungen werden in der Benutzungsoberfläche für Sie aufgelistet. So können Sie prüfen, ob der Vorgang im gewünschten Umfang ausgeführt wird. Ist dies nicht der Fall, können Sie die Selektionskriterien einfach ändern.

Über die Liste der ausgewählten Objekte können Sie das System veranlassen, den entsprechenden Vorgang nur für einzelne Objekte auszuführen anstatt für die gesamte Auswahl. Beispiel: In einer Liste mit nicht verwendeten Objekten, die gelöscht werden sollen, können Sie nur einige Objekte löschen und die Massenoperation abbrechen.

Hinweis

Wenn Sie Selektionskriterien für einen Vorgang definiert haben, der mehrere Kriterien gleichzeitig unterstützt, und dann zu einem anderen Vorgangstyp wechseln, der nur die Auswahl einer Anwendung unterstützt, gehen alle zusätzlichen Selektionskriterien verloren. In einem solchen Fall gibt das System aber eine Meldung aus, damit Sie Ihre Auswahl sichern können.

Operationen

Nicht verwendete Objekte löschen

Mit diesem Vorgang löschen Sie alle Objekte in einer Anwendung, die nicht von anderen Objekten referenziert werden. Objekte, die nicht mehr verwendet werden:

- Objekte, die nur zu Testzwecken angelegt wurden
- Objekte, die früher verwendet aber durch ein anderes Objekt ersetzt wurden
- Objekte, die als eingebettetes Objekt eines Container-Objekts angelegt wurden, und deren Container in der Zwischenzeit gelöscht wurde

Nicht verwendete Objekte sind in zwei Kategorien unterteilt:

- Benannte Objekte
Nicht verwendete benannte Objekte werden meistens durch die ersten beiden oben aufgeführten Gruppen abgedeckt.
- Nicht benannte Objekte
Nicht verwendete und nicht benannte Objekte werden automatisch als Reaktion einer bestimmten Benutzeraktion im Hintergrund vom System angelegt. Beispiel: Wenn Sie in früheren Releases von BRFplus einen Bereich in die Bedingungszone einer Entscheidungstabelle eingegeben haben, legte das System automatisch einen nicht benannten Bereichsausdruck in derselben Anwendung an und richtete eine Referenzbeziehung zwischen der Entscheidungstabelle und dem Bereichsausdruck ein. Diese Art von Objekten ist besonders schwer zu finden, da sie weder im Repository-Baum noch in anderen Objektlisten aufgeführt sind.

Nachdem Sie Ihre Auswahl vorgenommen haben, wählen Sie [Übernehmen](#). Das System sammelt die betroffenen Objekte. Wenn nicht verwendete benannte Objekte gefunden werden, sind diese in einer Liste aufgeführt und die Drucktaste [Löschen](#) ist aktiviert. Werden nicht verwendete und nicht benannte Objekte gefunden, wird die Drucktaste [Unbenannte nicht verwendete Obj. bereinigen](#) aktiviert.

Hinweis

Im Gegensatz zu den meisten anderen Vorgängen kann der Vorgang [Nicht verwendete Objekte löschen](#) nicht im Testmodus ausgeführt werden. Wenn Sie [Löschen](#) oder [Bereinigung](#) wählen, führt das System den Vorgang aus und entfernt die betroffenen Objekte aus der Datenbank. Der Testmodus wird in diesem Fall durch die Liste der betroffenen Objekte ersetzt, die vom System gefüllt wird, wenn Sie die Selektionsdaten auf das Tool anwenden. Außerdem werden die bei diesem Vorgang gesammelten Objekte sowieso nicht verwendet, es besteht also keine Gefahr, dass Daten verloren gehen.

Wenn Sie den Vorgang erfolgreich durchgeführt haben, empfiehlt SAP, dass Sie dieselbe Auswahl noch einmal anwenden. So ermitteln Sie, ob nach dem ersten Löschlauf neue nicht verwendete Objekte in der Anwendung vorliegen. Dies ist der Fall, wenn Objekte im System zuvor von nicht verwendeten Objekten genutzt wurden, die im ersten Löschlauf entfernt wurden. Nach dem Lauf wird das von einem gelöschten nicht verwendeten Objekt genutzte Objekt selbst zu einem nicht verwendeten Objekt und kann in einem zweiten Lauf gelöscht werden.

Zum Löschen vorgemerkte Objekte löschen

Mit diesem Vorgang löschen Sie logisch alle Objekte aus einer definierten Reihe von Anwendungen, die zum Löschen vorgemerkt sind und nicht von anderen Objekten referenziert werden. Nachdem Sie Ihre Auswahl vorgenommen haben, wählen Sie [Übernehmen](#). Das System sammelt die Anwendungen mit den betroffenen Objekten.

Bevor ein Objekt durch diesen Vorgang tatsächlich gelöscht wird, muss neben der Löschvormerkung und der Tatsache, dass **keine** Referenz durch andere Objekte vorliegt, noch ein drittes Kriterium erfüllt sein: Basierend auf dem aktuellen Systemdatum zieht das System den so genannten Datenerhaltungszeitraum, einen vordefinierten Zeitraum, vom aktuellen Datum ab. Das so ermittelte Datum dient als Schwellenwert: Nur Objekte, die vor diesem Schwelldatum zur Löschung vorgemerkt wurden, werden tatsächlich gelöscht.

Der Standardwert für den Datenerhaltungszeitraum beträgt 90 Tage. Sie können das vom System ermittelte Schwelldatum im Tool zur [Anwendungsverwaltung](#) aber manuell anpassen.

Hinweis

Dieser Vorgang kann nur für ganze Anwendungen durchgeführt werden. Einzelne Objekte können von der Verarbeitung weder ausgeschlossen noch in sie einbezogen werden.

Alte Objektversionen verwerfen

Mit diesem Vorgang löschen Sie veraltete Objektversionen in einer definierten Reihe von Anwendungen logisch. Das System entfernt alle Objektversionen, die älter als das angegebene Schwelldatum sind. Die Version, die am Schwelldatum verwendet wurde, wird so geändert, dass das zugewiesene Schwelldatum das Startdatum der Version ist. Das System sorgt dafür, dass alle Abhängigkeiten zwischen den betroffenen Objekten bestehen bleiben, sodass es weiterhin möglich ist, das System auf den Status am Tag des Schwelldatums zurückzusetzen. Verworfenen Objektversionen verbleiben nach der Verarbeitung physisch in der Datenbank, können aber nicht mehr aufgerufen werden. Nach einem Aufbewahrungszeitraum entfernt das System sie mithilfe eines zusätzlichen Bereinigungslaufs.

Achtung

Beachten Sie, dass eine Objektversionierung meistens aus wichtigen Gründen erfolgt, z. B. um gesetzliche Vorschriften einzuhalten oder aus Auditing-Gründen. Gehen Sie deshalb sorgfältig vor, wenn Sie diesen Vorgang verwenden. Sie sollten absolut sicher sein, dass die zu verwerfenden Objektversionen auch wirklich nicht mehr benötigt werden.

Datenbank aufräumen

Mit diesem Vorgang bereinigen Sie die Datenbank explizit von den deaktivierten Objekten und Objektversionen, die als Ergebnis der oben beschriebenen Vorgänge [Zum Löschen vorgemerkte Objekte löschen](#) und [Alte Objektversionen verwerfen](#) in der Datenbank vorliegen. Wenn Sie diesen Vorgang ausführen, werden sowohl versteckte Objektversionen als auch logisch gelöschte Objekte physisch aus der Datenbank gelöscht und können nicht wieder abgerufen werden.

Objekte neu organisieren

Mit diesem Vorgang organisieren Sie die interne Struktur bestimmter Ausdrücke, z. B. von Entscheidungstabellen, Schleifen, Booleschen Ausdrücken, in einer BRFPplus-Anwendung, die aus einem früheren Release von BRFPplus übernommen wurden. So wird einmal die Performance verbessert und die Wartung solcher Legacy-Objekte vereinfacht.

Weitere Informationen

[Objekte löschen \[Seite 57\]](#)

[Versionierung \[Seite 176\]](#)

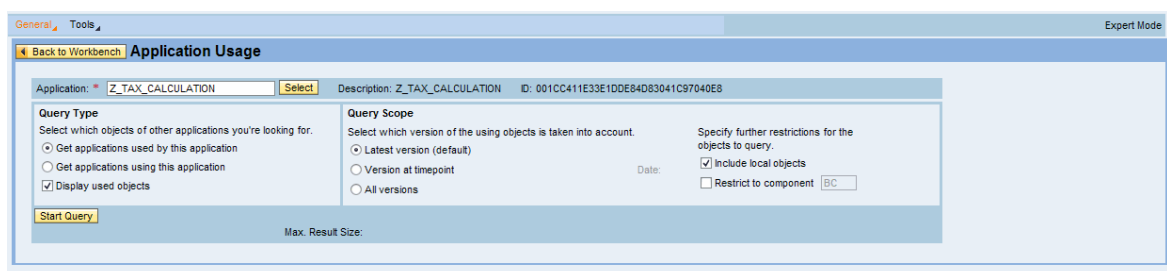
1.3.16.2 Anwendungsverwendung prüfen

Voraussetzungen

Sie arbeiten im Expertenmodus.

Vorgehensweise

1. Wählen Sie **Werkzeuge** > **Anwendungsverwendung**.
Die Seite Anwendungsverwendung wird geöffnet.
2. Geben Sie den Namen der Anwendung im Feld **Anwendung** ein und wählen Sie **Auswählen**.
3. Im Dialogfenster **Objektanfrage** wählen Sie die Anwendung aus der Tabellenliste aus.
Die ausgewählte Anwendung wird auf der Seite Anwendungsverwendung angezeigt.



4. Wählen Sie den Recherchetyp.
Sie können Anwendungen auswählen, die von der ausgewählten Anwendung genutzt werden oder auch Anwendungen, die die ausgewählte Anwendung nutzen.
5. Wählen Sie den Rechercheumfang.
Sie können die neueste Version oder die Version zu einem bestimmten Zeitpunkt oder alle verfügbaren Versionen auswählen.
6. Wählen Sie **Query starten**.
Das Query-Ergebnis wird unterhalb der Taste **Query starten** angezeigt.

1.3.16.3 Dictionary-Verwendungen prüfen

Verwendung

Eine der leistungsstarken Funktionen in BRFplus ist die Möglichkeit, ein Datenmodell basierend auf den ursprünglichen Datenobjekten zu erstellen, wie sie im Data Dictionary des Backend-Systems für die aufrufende Anwendung definiert sind. Mit diesem Tool können Sie die Verwendung aller Dictionary-Objekte in einem festgelegten Umfang analysieren.

Voraussetzungen

Sie führen die BRFplus-Workbench im Expertenmodus aus.

Funktionsumfang

Auswahl

Es gibt verschiedene Ansätze, sowohl den Umfang an Objekten, die das Tool nach potenziellen Verwendungen (Suchdomäne) durchsucht, als auch den Umfang an Objekten zu steuern, die tatsächlich an Dictionary-Elemente gebunden sind.

Suchdomäne

Wenn Sie das Tool über das Menü [Extras](#) starten, sind alle Eingabefelder zur Reduzierung des Umfangs leer. Eine Suche mit dieser Einstellung würde dazu führen, dass das Tool das gesamte System durchsucht. Dies wäre sehr zeitaufwändig. Deshalb müssen Sie einen Dictionary-Typ oder eine BRFplus-Anwendung (oder beides) als Suchdomäne angeben.

Die Suchdomäne können Sie weiter auf eine bestimmte Anwendungskomponente eingrenzen. Wenn Sie dies angeben, gibt das Tool nur die Dictionary-Verwendungen aus, die in BRFplus-Anwendungen gefunden wurden, welche einer bestimmten Anwendungskomponente zugewiesen sind.

Hinweis

Die Einträge in den Feldern [Dictionary-Typ](#) und [Anwendung](#) müssen eindeutige Namen enthalten. Platzhalter werden hier nicht unterstützt.

Objektumfang

Dictionary-Elemente können nur an BRFplus-Datenobjekte gebunden werden. Sie können jedoch definieren, dass das Tool eine Liste der Verwendungen für verschiedene Datenobjekte ausgeben soll oder die Ergebnisliste so eingrenzen, dass ausschließlich elementare Datenobjekte, Strukturen, Tabellen oder eine Kombination dieser Objekte aufgeführt werden.

Ergebnis

Nachdem Sie die Selektionskriterien definiert und [Suche starten](#) gewählt haben, ermittelt das System alle Datenobjekte, die den Kriterien entsprechen, und führt diese in der Ergebnisliste auf. Für jeden Eintrag in der

Liste zeigt das System den Namen des BRFplus-Datenobjekts, seinen Typ (elementar, Struktur oder Tabelle), den Dictionary-Typ, an den es gebunden ist und die BRFplus-Anwendung, zu der es gehört. Sowohl der Name des Datenobjekts als auch der Name der Anwendung werden als Links angezeigt, sodass Sie durch Klicken auf den Link zum jeweiligen Objekt navigieren können.

Datenbindung aktualisieren

In der Spalte [Bindung](#) wird mithilfe eines Ampelsymbols angegeben, ob die Bindung zwischen Datenobjekt und Dictionary-Element noch aktuell ist oder nicht. Ist das Ampelsymbol rot, müssen Sie das Datenobjekt prüfen und die Bindung aktualisieren. Gehen Sie dazu wie folgt vor:

1. Wählen Sie den Namen des Datenobjekts.
Das System zeigt das Detailbild für das Datenobjekt.
2. Wählen Sie [Bearbeiten](#).
3. Wählen Sie [Bindung aktualisieren](#).
Das System aktualisiert das Datenobjekt entsprechend dem aktuellen Status des Dictionary-Elements, an das es gebunden ist.
4. Wählen Sie [Aktivieren](#), um das aktualisierte Datenobjekt zu sichern und zu aktivieren.
5. Wählen Sie [Schließen](#), um zum Tool für Dictionary-Verwendungen zurückzukehren.

Hinweis

Die oben aufgeführten Schritte sind unter Umständen nicht ausreichend für die Aktualisierung, abhängig von der Ursache, warum eine Bindungsbeziehung als veraltet angezeigt wird. Wenn das gebundene Dictionary-Element z. B. gelöscht wurde, nachdem die Bindung eingerichtet wurde, ist es natürlich **nicht** möglich, das Datenobjekt durch Wählen von [Bindung aktualisieren](#) zu aktualisieren. In einem solchen Fall müssen Sie untersuchen, was mit dem Dictionary-Element passiert ist und entweder ein neues Element mit derselben Struktur einrichten oder eine Bindungsbeziehung zu einem anderen Dictionary-Element herstellen, welches das ursprüngliche Element möglicherweise ersetzt hat.




Auch wenn die Aktualisierung der Datenbindung erfolgreich ist und Sie das Datenobjekt aktiviert haben, gibt es keine Garantie dafür, dass die Regeln, die dieses Datenobjekt verwenden, wie vor der Aktualisierung funktionieren. BRFplus kann die Änderungen an den Eigenschaften und der Struktur gebundener Dictionary-Elemente nicht erfassen. Eine Datenbindung kann z. B. veraltet sein, weil der Kurztext eines Datenelements geändert wurde, oder weil 10 Felder aus einer Dictionary-Struktur gelöscht wurden. Sie müssen den Grund und die Folgen von vorgenommenen Änderungen analysieren.

Weitere Informationen

[Datenbindung \[Seite 66\]](#)

1.3.16.4 Nicht verwendete Objekte löschen


Vorgehensweise

1. Wählen Sie  [Werkzeuge](#)  [Anwendungsverwaltung](#) .
- Das Tool zur Anwendungsverwaltung wird angezeigt.
2. In der Dropdown-Liste [Operation](#) wählen Sie [Nicht verwendete Objekte löschen](#).
3. Geben Sie im Abschnitt [Objektumfang](#) den Namen der gesuchten Anwendung ein, deren nicht verwendete Objekte Sie löschen möchten, und wählen Sie [Übernehmen](#).

Hinweis

In dieser Funktion können Sie nur den Namen einer Anwendung eingeben. Platzhalter werden nicht unterstützt.

Das System führt in der Liste [Benannte nicht verwendete Objekte](#) die nicht verwendeten Objekte auf, die zurzeit in der ausgewählten Anwendung vorliegen. Die Objekte sind nach Objekttyp sortiert.

4. Sie haben jetzt folgende Möglichkeiten:
 - Wenn Sie alle nicht verwendeten Objekte in der gewählten Anwendung löschen möchten, wählen Sie [Löschen](#).
 - Wenn Sie nur einzelne Objekte löschen möchten, navigieren Sie zum gewünschten Objekt und wählen  in der Spalte [Löschen](#).

Im Abschnitt [Nachrichten](#) werden die Anzahl und weitere Details zu den gelöschten Objekten angezeigt.

1.3.16.5 Objekte in einem Transportauftrag analysieren

Voraussetzungen

Sie arbeiten im Expertenmodus.

Kontext

Mit diesem Werkzeug können Sie BRFplus-Objekte anzeigen, analysieren und prüfen, die in einem bestimmten Transportauftrag erfasst wurden. Abhängig davon, ob es im Quell- oder Zielsystem ausgeführt wird, werden verschiedenen Aktivitäten ausgeführt.

- Im **Quellsystem** werden nicht freigegebene Transporte mit Objekten angezeigt, die Fehler oder Inkonsistenzen aufweisen. Das Werkzeug bietet auch eine Option, um zu den betroffenen Objekten zu navigieren und Fehler oder Inkonsistenzen zu korrigieren, damit der Transport fehlerfrei nach der Verarbeitungsprüfung vor Export freigegeben werden kann.
- Im **Zielsystem** werden Transporte angezeigt, bei denen während der Verarbeitungsprüfung nach Import Fehler aufgetreten sind, wodurch importierte Objekte vorhanden sind, die nicht im Zielsystem aktiviert wurden. Sobald die Problemursache behoben wurde, kann die Verarbeitungsprüfung nach Import erneut gestartet werden.

Beispiel

Datenobjekte, die an ein DDIC-Objekt gebunden sind, das nicht im Zielsystem vorhanden ist, können ggf. Probleme verursachen. Auch kundeneigene Ausdruckstypen können zu dieser Fehlerart führen, sofern Instanzen dieses Ausdruckstyps in zugrundeliegende Klassen und Schnittstellen übertragen wurden, in denen der Ausdrucktyp noch nicht im Zielsystem vorhanden ist.

Vorgehensweise

- Wählen Sie ► [Werkzeuge](#) ► [Transportanalyse](#) ►.

Die Seite Transportanalyse wird angezeigt.

1.3.16.6 Funktion simulieren

Verwendung

Das Simulationswerkzeug arbeitet ähnlich wie das Testen von Funktionsbausteinen in der ABAP Workbench. Mit diesem Werkzeug können Sie beliebige Arten von Datenobjekten eingeben, die Eingabe verarbeiten und ein Ergebnis zurückgeben.

Voraussetzungen

Die Funktion muss aktiviert sein.

Vorgehensweise

1. Öffnen Sie die Funktion im Bereich [Objektmanager](#).
2. Wählen Sie [Simulation starten](#).
Die Drucktaste [Simulation starten](#) ist nur bei aktivierter Funktion verfügbar.

3. Geben Sie im Dialogfenster *Simulation* im Bereich *Werte von Kontextdatenobjekten pflegen* die Werte neben dem Kontextfeld ein.
Außerdem können Sie Werte aus einer Excel-Datei importieren.
Weitere Informationen erhalten Sie unter [Während der Simulation Werte aus einer Excel-Datei importieren \[Seite 271\]](#)
4. Im Bereich *Simulationsmodus* können Sie *Nur Ergebnis anzeigen* oder *Ergebnisse der Zwischenschritte ebenfalls anzeigen* auswählen.
5. Wählen Sie *Simulation ausführen*.
Das Ergebnis wird auf der Simulationsseite angezeigt.

Hinweis

Beachten Sie, dass nicht nur die Funktion aktiviert sein muss, sondern auch die Ausdrücke und sonstigen Objekte, die zu verarbeiten sind. Meldet die Simulation unerwartete Ergebnisse oder Fehlermeldungen, liegt dies ggf. daran, dass Sie eines der betroffenen Objekte geändert haben, ohne das Objekt anschließend zu aktivieren.

1.3.16.7 Während der Simulation Werte aus einer Excel-Datei importieren

Vorgehensweise

1. Öffnen Sie die Funktion im Bereich *Objektmanager*.
2. Wählen Sie *Simulation starten*.
Die Drucktaste *Simulation starten* ist nur bei aktivierter Funktion verfügbar.
3. Wählen Sie auf der Simulationsseite *Durchsuchen...* neben dem Feld *Excel-Datei auswählen*.
4. Wählen Sie im Dialogfenster *Datei auswählen* die Excel Datei.
Der Pfad der ausgewählten Datei wird neben dem Feld *Excel-Datei auswählen* angezeigt.
5. Wählen Sie *Testdaten importieren*.

1.3.16.8 Während der Simulation Kontext in eine Excel-Datei exportieren

Vorgehensweise

1. Öffnen Sie die Funktion im Bereich *Objektmanager*.
2. Wählen Sie *Simulation starten*.
Die Drucktaste *Simulation starten* ist nur bei aktivierter Funktion verfügbar.
3. Wählen Sie auf der Simulationsseite *Kontext in Excel exportieren*.
4. Das Dialogfenster *Datei-Download* wird angezeigt. Wählen Sie *Öffnen* oder *Sichern*.

1.3.16.9 Web-Services und Funktionsbausteine generieren

Verwendung




Das Werkzeug zum Generieren von Web-Services und RFC-fähigen Funktionsbausteinen unterstützt Anwendungsentwickler sowie Fremdarbeiter beim Verarbeiten von Regeln, die in BRFplus definiert wurden. Mit diesem Werkzeug können Sie Web-Services und Funktionsbausteine automatisch und mit minimaler Interaktion anlegen.

Beispiel

Sie haben eine BRFplus Funktion für eine regelbasierte Berechnung von Frachtkosten angelegt. Diese Funktion wird über Anwendung A aufgerufen. Möglicherweise ist die Kostenberechnungslogik so generisch, dass dieselben Regeln auch zur Frachtkostenberechnung in Anwendung B verwendet werden können. Bei B handelt es sich jedoch um eine Java-basierte Webanwendung, die die BRFplus Funktion nicht direkt im ABAP Backend System aufrufen kann. Um dieses Problem zu umgehen, müssen Sie für die fragliche BRFplus Funktion einen Web-Service generieren. Durch diese Vorgehensweise kann Webanwendung B die BRFplus Funktion über den generierten Web-Service aufrufen.

Vorgehensweise

Web-Service generieren

1. Wählen Sie  *Werkzeuge*  *Web-Service-Generierung* .
2. Wählen Sie auf der Seite Web-Service-Generierung *Funktion auswählen*.
3. Im Dialogfenster *Objektabfrage* geben Sie den Namen der Funktion und Anwendung ein und wählen *Suchen*.

4. Wählen Sie eine Funktion aus der Liste aus.
Die ausgewählte Funktion wird im Bereich *Funktion* angezeigt.

Hinweis

Für eine andere Funktion wählen Sie *Andere Funktion auswählen*.

Sie können das Werkzeug zum Generieren auch über das Kontextmenü einer Funktion im Navigationsbereich der BRFplus Workbench starten. Oder Sie verwenden die Taste *Generieren* im Arbeitsbereich einer Funktion, die Sie aktuell bearbeiten. Bei beiden Vorgehensweisen wird die zu generierende Funktion dann bereits vom System voreingestellt.

5. Konfigurieren Sie den Web-Service. Geben Sie im Abschnitt *Web-Services* folgende Daten ein:
 1. Zum Anlegen einer neuen Funktionsgruppe aktivieren Sie das Kontrollkästchen *Funktionsgruppe anlegen*. Anderenfalls wird der Funktionsbaustein in der vorhandenen Funktionsgruppe generiert.
 2. Um die vorhandene Funktionsgruppe zu verwenden, geben Sie den Namen der Funktionsgruppe im Feld *Funktionsgruppe* ein.
 3. Geben Sie den Namen des Funktionsbausteins für den generierten Remote Function Call (RFC)-Funktionsbaustein im Feld *Funktionsbaustein* ein.
 4. Geben Sie den Namen und Kurztext für den Web-Service in den Feldern *Web-Service-Name* und *Web-Service-Kurztext* ein.
 5. Zum Festlegen der Sicherheitsebene öffnen Sie die Worthilfe im Feld *Web-Service-Sicherheitsprofil*. Wählen Sie das gewünschte Sicherheitsprofil aus.
6. Nehmen Sie im Bereich *Transport* die gewünschten Eingaben vor.
 1. Das Entwicklungspaket, in dem Sie den Web-Service generieren möchten.
Wenn der Web-Service nicht in andere Systeme transportiert werden soll, aktivieren Sie das Kontrollkästchen *Lokales Paket*. Zu diesem Zweck können Sie auch *\$TMP* im Feld *Entwicklungspaket* eingeben.

Hinweis







Wenn Sie den Web-Service in einem nicht lokalen Entwicklungspaket generieren und es transportiert werden kann, müssen Sie die Objekte transportieren.

Weitere Informationen finden Sie unter *Transport von Objekten [Seite 259]*.

2. Geben Sie einen Workbench-Auftrag ein.
7. Wählen Sie *Generieren*.
Der Web-Service wird automatisch mit allen erforderlichen Objekten generiert.

Den generierten Web-Service aktivieren

Sobald alle oben genannten Schritte abgeschlossen sind, generiert das System die erforderlichen Objekte. Der neu generierte Web-Service kann jedoch nicht unmittelbar verwendet werden. Dies liegt an anwendbaren Sicherheitsstandards, die eine explizit Aktivierung durch einen autorisierten Benutzer vorschreiben. Aus diesem Grund müssen Sie folgende Schritte manuell ausführen:

1. Starten Sie im Backend-System die Transaktion **SICF**.
2. Geben Sie im Feld *Service-Name* den Namen des generierten Service ein und wählen Sie  *Programm* 
Ausführen .
- Das System erweitert den Service-Baum bis die gewünschten Services angezeigt werden.
3. Wählen Sie  *Service/Host*  *Anzeigen/Ändern* .

Das System zeigt Ihnen alle Einstellungen an, die für den generierten Service angelegt wurden.

4. Überprüfen Sie die Einstellungen und ändern Sie diese nach Bedarf.
5. Sichern Sie die Änderungen und verlassen Sie die Transaktion.
6. Starten Sie die Transaktion **SOAMANAGER**.
In einem neuen Browser-Fenster wird das Werkzeug *SOA-Management* gestartet.
7. Wählen Sie ► *Service-Administration* ► *Konfiguration einzelner Services* ►.
8. Geben Sie den Servicenamen (oder einen Teil des Namens) im Feld *Suchmuster* ein.
9. Wählen Sie in der Tabelle *Suchergebnisse* die Zeile des gewünschten Service aus und klicken Sie auf *Auswahl übernehmen*.
10. Klicken Sie auf der Registerkarte *Konfigurationen* auf *Endpunkte anlegen*. Überprüfen Sie die vom System vorgeschlagenen Standardeinstellungen für *Servicename*, *Beschreibung*, und *Neuer Binding-Name*.
11. Sobald der neu definierte Endpunkt generiert wurde, können Sie die Konfigurationsdetails pflegen (*Provider-Sicherheit*, *Transporteinstellungen* usw.).

Sobald Sie alle oben genannten Schritte abgeschlossen und die Einstellungen gesichert haben, können Sie den Web-Service verwenden.

Funktionsbaustein generieren

Das Generieren eines RFC-fähigen Funktionsbausteins für eine BRFplus Funktion ist ein Subset zum Generieren eines Web-Services. Aus diesem Grund sind die oben genannten Schritte für Funktionsgruppen und Funktionsbausteine auch zum Generieren eines Funktionsbausteins anwendbar.

1.3.16.10 Export und Import von XML-Daten

Verwendung

Sie können BRFplus-Objekte in eine XML-Repräsentation konvertieren. Dies ist hilfreich, wenn Sie Objekte zwischen Systemen, die sich nicht in derselben Systemlandschaft befinden, übertragen möchten, oder wenn kein Transportanschluss durch das SAP Transportsystem bereitgestellt wird.

Voraussetzungen

Sie führen die BRFplus-Workbench im *Expertenmodus* aus.

Funktionsumfang

Export

Die XML-Exportfunktion arbeitet objektbasiert, das heißt, Sie können die Funktion starten, indem Sie ein BRFplus-Objekt auswählen, das Sie exportieren möchten. Es ist jedoch möglich, den Umfang eines XML-Exports auf folgende Arten zu erweitern:

- **Referenzierte Objekte einbinden:** Mit dieser Einstellung wird das ausgewählte Objekt gemeinsam mit allen Objekten exportiert, die durch das ausgewählte Objekt direkt oder indirekt referenziert sind.
- **Anwendung einschließen:** Mit dieser Einstellung wird das ausgewählte Objekt gemeinsam mit der BRFplus-Anwendung exportiert, zu der es gehört.

Zusätzlich zur oben beschriebenen Auswahl des Umfangs, können Sie den Export weiter verfeinern, indem Sie eine bestimmte **XML-Version** und ein bestimmtes **XML-Schema** für den Export auswählen. Je nach Auswahl, passt das System das Format und die Syntax der exportierten Daten an. Dies ist in Situationen hilfreich, in denen Sie BRFplus-Objekte in ein System mit einem niedrigeren Release als das Quellsystem transportieren müssen. Es ist natürlich immer empfehlenswert, die höchste XML-Version, die vom Zielsystem unterstützt wird, für den Export zu verwenden, so dass das System den größtmöglichen Funktionsumfang zur Verfügung hat.

Die XML-Schemaauswahl ermöglicht es Ihnen, aus verschiedenen Formen der XML-basierten Objektrepräsentation zu wählen:

- **Intern**
Wie der Name bereits vermuten lässt, rendert das **interne** Schema die Schema-Elementnamen auf dieselbe Weise wie intern in BRFplus. Mit diesem Schema muss vom System daher keine Umwandlung während des XML-Imports oder XML-Exports durchgeführt werden.
- **Extern**
Das **externe** Schema entspricht dem **internen** Schema funktionell, rendert jedoch alle Schema-Elementnamen in Upper-Camel-Case-Notation statt in Großbuchstaben. Dies verbessert die Lesbarkeit der XML-Daten, was eine manuelle Analyse der XML-Daten erleichtert.

i Hinweis

Starten Sie die Importfunktion im Zielsystem um herauszufinden, welche die höchste vom Zielsystem unterstützte XML-Version ist. Die höchste unterstützte Version wird auf dem Bildschirm angezeigt.

Die XML-Schemaauswahl ist nur verfügbar, wenn Sie die entsprechende Einstellung im Dialogfenster **Personalisierung** ausgewählt haben.

Import

Die XML-Importfunktion ist komplementär zur Exportfunktion. Sie wird verwendet, um XML-Daten, die zuvor aus BRFplus in ein anderes System exportiert wurden, erneut zu importieren.

Bevor Sie die Daten importieren, ist es empfehlenswert, den Import zunächst als **Testlauf** zu starten. Das System prüft die Importdaten, die hochgeladen werden sollen, auf Inkonsistenzen und andere mögliche Probleme, ohne dabei die Daten in das Zielsystem zu schreiben. Sie können die Prüfergebnisse zur Bereinigung und Korrektur der Importdaten nutzen, bevor Sie den tatsächlichen Datenimport starten.

Wenn das Zielsystem, in das Sie die Daten importieren möchten, **nicht** am Ende der Transportkette in Ihrer Systemlandschaft stehen, müssen die importierten Daten in nachgelagerte Systeme transportiert werden. Hierzu müssen Sie die ID eines geeigneten Transportauftrags eingeben, den Sie zuvor im Zielsystem für den Import angelegt haben.

Importtypen

Zum Import von XML-Daten in das System bietet BRFPplus die folgenden Importtypen an:

- **Standard**
Mit dieser Einstellung können Sie Änderungen für im Zielsystem änderbare Objekte importieren.
- **Reparieren**
Mit dieser Einstellung können Sie Inhalte reparieren, die im System normalerweise nicht änderbar sind, da sie mittels einem normalen Transport in das System importiert wurden. Die Reparaturfunktion sollte nur genutzt werden, um die Änderungen am Inhalt, die durch einen SAP-Korrekturhinweis bereitgestellt wurden, zu importieren. Sie können diese Einstellung verwenden, um eine inhaltliche Korrektur in einem System vorzunehmen, bevor das entsprechende Support Package mit dem korrigierten Inhalt ins System importiert wird.
- **Lokale Kopie**
Mit dieser Einstellung können Sie eine lokale Kopie einer nicht-lokalen Anwendung in der Quelldatei anlegen.

i Hinweis

Sie können diese Einstellung nur verwenden, wenn die Anwendung in der Quelldatei **nicht** lokal ist und noch **nicht** mit einem normalen Transport ins Zielsystem importiert wurde. Diese Prüfung stellt sicher, dass lokale Kopien die transportierten Daten nicht überschreiben. Dies ist wichtig, da mithilfe dieser Einstellung eine exakte Kopie der importierten Objekte entsteht, einschließlich der ursprünglichen ID der importierten Objekte. Damit ist die ID der betreffenden Objekte nicht länger global eindeutig, sodass weitere Transporte zu unvorhersehbaren Ergebnissen führen können.

Beachten Sie, dass die Auswahl eines bestimmten Importtyps nur möglich ist, wenn die entsprechende Personalisierungseinstellung aktiviert wurde. Andernfalls wird die Auswahl des Importtyps nicht angezeigt und das System verwendet den **Standard**-Importtyp.

Verarbeitung von Importkonflikten

Wenn das System ein Objekt in den Importdaten ermittelt, das dieselbe ID hat wie ein bereits vorhandenes Objekt im Zielsystem, hängt die weitere Verarbeitung von der Versionierungseinstellung dieses Objekts ab.

- Versionierung ist aktiv
Das in den Importdaten enthaltene Objekt wird als neue Version des Objekts im Zielsystem importiert.
- Versionierung ist **nicht** aktiv
Das in den Importdaten enthaltene Objekt überschreibt das vorhandene Objekt im Zielsystem ohne vorherige Ankündigung. Der Benutzer kann dieses Verhalten nicht beeinflussen.

Export und Import über Anwendungen

Sowohl der Export als auch der Import von XML-Daten kann nur für Objekte durchgeführt werden, die zu derselben BRFPplus-Anwendung gehören. Wenn Sie Objekte von verschiedenen Anwendungen übertragen möchten, müssen Sie den Export- und Importprozess dahingehend aufteilen, dass Sie für jede betroffene Anwendung eine eigene Datendatei verwenden. Wenn Sie ein Projekt, an dem mehrere Anwendungen beteiligt sind, in das Zielsystem importieren, beachten Sie, dass die Datendateien basierend auf der Verwendungsbeziehung der verschiedenen Anwendungen verarbeitet werden: Beginnen Sie mit dem Import der Objekte, die von Objekten einer anderen Anwendung verwendet werden, bevor Sie sie verwendeten Objekte importieren. Andernfalls schlägt der Import aufgrund von Objektreferenzen fehl, die das System nicht aufheben kann.

Aktivitäten

Export

Gehen Sie wie folgt vor, um ein BRFplus-Objekt aus dem aktuellen System zu exportieren:

1. Wählen Sie **Werkzeuge** > **XML-Export**.
Das System zeigt das Bild **XML-Export** an.
2. Klicken Sie auf **Ein Objekt auswählen**, um alle BRFplus-Objekte auszuwählen, die im aktuellen System verfügbar sind und die Sie exportieren möchten.
3. Wählen Sie **Referenzierte Objekte einbeziehen**, wenn Sie alle Objekte in den Export einbeziehen möchten, die durch das ausgewählte Objekt referenziert sind.
4. Wählen Sie **Anwendung einschließen**, wenn Sie das Anwendungsobjekt in den Export einbeziehen möchten, zu dem das ausgewählte Objekt gehört.
5. Wählen Sie die höchste Versionsnummer aus der Dropdown-Liste **XML-Version**, die vom Zielsystem unterstützt wird.
6. Wählen Sie das XML-Schema aus der Dropdown-Liste **XML-Schema**, das für Ihre Zwecke am besten geeignet ist.
7. Klicken Sie auf **XML-Datei generieren**, um die XML-Konvertierung zu starten.

Hinweis

XML-Datei generieren ist nur aktiv, wenn Sie ein Objekt zum Export ausgewählt haben.

8. Klicken Sie auf **XML-Datei herunterladen**, um einen Ordner in dem Dateisystem zu definieren, in dem das System die im vorigen Schritt generierte XML-Datei ablegen soll.

Import

Gehen Sie wie folgt vor, um eine Datei, die BRFplus-Objektdaten enthält, in das aktuelle System zu importieren:

1. Wählen Sie **Werkzeuge** > **XML-Import**.
Das System zeigt das Bild **XML-Import** an.
2. Stellen Sie sicher, dass die XML-Version der zu importierenden Daten niedriger als oder gleich der **Höchsten unterstützen XML-Version** ist, wie im Bild **XML-Import** beschrieben.
3. Geben Sie den Namen der zu importierenden **XML-Datei** ein. Sie können die Datei ebenfalls auswählen, indem Sie die Drucktaste **Suchen** wählen und durch das Dateisystem navigieren.
4. Wählen Sie den geeigneten **Importtyp** aus.
5. Entscheiden Sie, ob die zu importierenden Daten in einen Transportauftrag geschrieben werden sollen:
 - Für BRFplus-Objekte, die nur lokal im Zielsystem verwendet werden, ist kein Transport notwendig. Gleiches gilt, wenn keine nachgelagerten Systeme in Ihrer Systemlandschaft Transporte vom Zielsystem für den Import erhalten.
 - Geben Sie eine **Customizing-Transportauftrags-ID** für BRFplus-Objekte ein, die als Customizing-Objekte definiert sind. Dies ist für kundeneigene Objekte notwendig.
 - Geben Sie eine **Workbench-Transportauftrags-ID** für BRFplus-Objekte ein, die als Systemobjekte definiert sind. Dies ist für von SAP ausgelieferte Objekte notwendig.
6. Entscheiden Sie, ob Sie den XML-Import als **Testlauf** ausführen möchten, oder nicht.
Der **Testlauf**-Modus ist ein Simulationsmodus, in dem das System alle Importschritte durchführt, ohne Daten ins Zielsystem zu schreiben. Sie können diesen Modus verwenden, um zu prüfen, ob die

Importdaten konsistent und vollständig sind, bevor physische Änderungen am Zielsystem vorgenommen werden.

7. Klicken Sie auf [Datei hochladen](#), um entweder den Testlauf oder den XML-Datenimport zu starten.

Weitere Informationen

[Ablagetypen \[Seite 35\]](#)

[Versionierung \[Seite 176\]](#)

1.3.16.11 Massenänderung

Verwendung

Die BRFplus-Workbench bietet eine benutzerfreundliche Oberfläche zum Anlegen und Bearbeiten der verschiedenen Objekte, die in einer Anwendung für Geschäftsregeln enthalten sind. Wenn jedoch dieselben Änderungen an vielen Objekten vorgenommen werden müssen, kann das Navigieren zwischen den Objekten kompliziert und aufwändig sein. Deshalb verfügt BRFplus über ein Tool zur Massenänderung, mit dem Sie Objektänderungen schnell und problemlos ausführen können. Die Anzahl der betroffenen Objekte, die geändert werden müssen, ist dabei völlig egal.

Das Tool unterstützt die folgenden Massenänderungen:

- Prüfen
- Aktivieren
- Als veraltet markieren
- Kennzeichen "Veraltet" zurücksetzen
- Zum Löschen vormerken
- Löschvormerkung aufheben
- Löschen
- Zugriffsebene auswählen
- Versionierungsstatus einstellen
- Umbenennen
- Kopieren
- In XML exportieren

Mit dem Tool zur Massenänderung können Sie mehrere Objekte auch mit leicht unterschiedlichen Änderungen modifizieren, also nicht genau dieselbe Änderung auf alle ausgewählten Objekte anwenden.



Beispiel

Sie haben eine neue Anwendung für Geschäftsregeln erstellt. Sie haben eine Reihe von Datenobjekten angelegt und dazu das erste Datenobjekt ohne weitere Änderungen einfach kopiert. Sie können alle diese Datenobjekte nun in das Tool zur Massenänderung laden und die Namen und Texte für alle Objekte individuell bearbeiten. Wenn Sie fertig sind, wählen Sie [Sichern](#). Dann schreibt das System alle diese Änderungen gleichzeitig in die Datenbank.

Funktionen

Umfang

Der Umfang der Objekte, die in das Tool zur Massenänderung geladen werden, ist abhängig davon, wie Sie das Tool starten. Sie haben folgende Optionen:

- Hauptmenü
Wählen Sie ► [Werkzeuge](#) ► [Massenänderung](#) ►. Das Systemverhalten ist abhängig vom aktuellen Status der Workbench:
 - Kein Objekt ausgewählt
Das System lädt alle Anwendungen in das Tool, die vom aktuell angemeldeten Benutzer angelegt wurden.
 - Objekt ausgewählt
Das System lädt die Anwendung, zu der das ausgewählte Objekt gehört, in das Tool und markiert es.
- Kontextmenü im Navigationsbereich
Wählen Sie im Kontextmenü eines im Navigationsbereich angezeigten Objekts ► [Tools](#) ► [Massenänderung](#) ►. Das System lädt die Anwendung, zu der das ausgewählte Objekt gehört, in das Tool und markiert es.
- Massenänderung durchsuchen
Wählen Sie im geöffneten Tool zur Massenänderung [Suchen](#), und geben Sie die Suchkriterien ein. Alle Objekte, die den Suchkriterien entsprechen, werden in das Tool geladen. Dies ist die einzige Option, bei der das Wurzelement in jeder Reihe nicht unbedingt eine Anwendung ist.

Hinweis

Im Tool zur Massenänderung können Sie konfigurieren, dass das System die Inhalte aller Anwendungen anzeigt, die der aktuell angemeldete Benutzer angelegt hat. Wählen Sie dazu [Meine Anwendungen anzeigen](#). Als Ergebnis wird derselbe Objektumfang aufgerufen wie beim Starten des Tools zur Massenänderung ohne ausgewählte Objekte (siehe oben).

Nachdem das System die Objektliste gefüllt hat, können Sie die aufgeführten Objekte einzeln rekursiv erweitern, um auf alle Objekte zuzugreifen, die davon verwendet werden.

Objektauswahl

Sie markieren Objekte und stellen so eine Objektauswahl zusammen, auf die Sie eine bestimmte Änderung anwenden möchten. Änderungen können an allen Objekten vorgenommen werden, die aktuell im Tool geladen sind. Dies gilt auch für Objekte, die zu anderen Anwendungen gehören.

Einzelne Objekte auswählen

Sie markieren die zu ändernden Objekte, indem Sie mit der Drucktaste am Zeilenanfang die gewünschte Zeile auswählen. Wenn Sie mehrere Objekte auswählen möchten, klicken Sie auf die jeweiligen Tabellenzeilen und halten dabei die `Umschalt-` oder die `Strg-` Taste gedrückt. Wenn Sie alle Objekte in das Tool laden möchten, wählen Sie [Alle Positionen auswählen](#).

Objekthierarchien auswählen

Häufig muss eine Änderung nicht nur auf ein bestimmtes Objekt angewendet werden, sondern auch auf alle Objekte, die vom ersten Objekt verwendet werden. Um diesen Vorgang durchzuführen, setzen Sie für das erste Objekt das Kennzeichen [Verwendungen](#). Dann nimmt das System alle untergeordneten Objekte in die Auswahl auf.

Hinweis

Das Tool kann allerdings keine anwendungsübergreifenden Objektbeziehungen auflösen. Solche anwendungsübergreifenden Konstellationen können für Objekte auftreten, deren Zugriffsebene als Systemstandard auf eine weniger begrenzte Ebene als *Anwendung* festgelegt wurde. Hier stoppt der Objekt-Drilldown, wenn das Tool ein abhängiges Objekt ermittelt, das aus einer anderen Anwendung stammt. Alle Aktivitäten, die Sie zu diesem Zeitpunkt für die ausgewählten Objekte initiieren, werden gestoppt, und die Objekte in der anderen Anwendung bleiben unverändert.

Abhängig vom definierten Umfang der Objekte, die in das Tool zur Massenänderung geladen werden, (siehe Abschnitt *Umfang*) kann das Auswählen von Objekthierarchien dazu führen, dass dasselbe Objekt mehrmals in der Objektliste aufgeführt wird. Wenn Sie so ein Objekt auswählen, markiert das System es an allen Stellen, an denen es in der Liste aufgeführt ist. Grund für die mehrfache Auflistung ist, dass ein Objekt als geschlossene Einheit gesehen werden kann und gleichzeitig als Teil anderer Objekte.

Kennzeichen „Verwendungen“: Zusatzinformationen

Die Semantik des Kennzeichens *Verwendungen* für die im Tool zur Massenänderung angezeigten Objekte ist auf den ersten Blick nicht offensichtlich. Beschreibung der verschiedenen Status für dieses Kennzeichen:

Tabelle 45:

Status	Beschreibung
<i>selektiert</i>	<p>Gibt an, dass alle Unterobjekte des übergeordneten Objekts (sowohl direkte als auch indirekte), markiert werden. Dabei werden nur Objekte derselben Anwendung als übergeordnete Objekte berücksichtigt.</p> <p>Dieser Status erfordert, dass alle benötigten Informationen im System verfügbar sind. Dies ist z. B. nicht der Fall, wenn der Objektbaum nach dem Start des Tools zur Massenänderung nicht aufgeklappt wurde. Dies kann sogar zu Situationen führen, in denen dasselbe Objekt mehrmals im Baum angezeigt wird und an einer Stelle das Kennzeichen <i>Verwendungen</i> gesetzt ist und an einer anderen nicht. Dies ist abhängig vom Aufklappstatus des Baums.</p> <div> <p>i Hinweis</p> <p>Aufgrund der BRFplus-Objektarchitektur gibt es leider keine einfache Möglichkeit, dem Tool alle nötigen Informationen für eine umfassende Markierung gleichzeitig bereitzustellen. Da die Objekte intern in einem Netzwerk und nicht in einer Hierarchie strukturiert sind, sind unter Umständen unzählige bidirektionale Beziehungen zwischen den Objekten vorhanden. Deshalb gibt es keinen technischen Indikator für das Tool, an welcher Stelle es beim Nachvollziehen dieser Beziehungen stoppen soll. Deshalb schränkt das System die Analyse der markierten Objekte auf die Objekte ein, die durch das Aufklappen bestimmter Baumzweige dem Tool bereits bekannt sind.</p> </div>
<i>nicht selektiert</i>	<p>Gibt eine der folgenden Möglichkeiten an:</p> <ul style="list-style-type: none"> • Das System verfügt nur über unzureichende Informationen zu den Unterobjekten des aktuellen Objekts. • Keines der Unterobjekte ist markiert.
<i>nicht spezifiziert</i>	<p>Gibt an, dass einige Unterobjekte markiert sind, aber mindestens eines nicht.</p>

i Hinweis

Objekte, die keine Unterobjekte haben können, werden immer mit dem *Verwendungstatus nicht selektiert* angezeigt. Dies gilt für Element-Datenobjekte.

Sie können festlegen, ob die Spalte *Verwendungen* im System angezeigt oder ausgeblendet wird. Wählen Sie dazu ► *Andere Aktionen* ► *Spalte „Verwendungen“ ein-/ausblenden* ►. Dies kann nützlich sein, wenn Ihr Bildschirm eine niedrige Bildschirmauflösung hat und Sie Platz für andere Spalten schaffen möchten.

Objektdaten

Für jedes geladene Objekt im Tool zur Massenänderung zeigt das System die wichtigsten Verwaltungsdaten in der Liste an:

- Name
- Text
- Typ
- Versionierungsstatus
- Zugriffsebene

Wenn Sie weitere Informationen zu einem bestimmten Objekt benötigen, um festzulegen, ob dieses geändert werden soll oder nicht, können Sie die gesamten Objektdaten durch Auswählen des Namens in der Spalte *Objekt* aufrufen. Das System zeigt dann dieselben Informationen zum Objekt an, die Sie auch im Arbeitsbereich der Workbench beim Bearbeiten eines Objekts sehen.

Änderungspuffer

Funktionen „Änderungspuffer“ und „Rückgängig“

Mit Ausnahme der Prüffunktion verfügen Sie für alle Aktivitäten im Tool zur Massenänderung über Schreibzugriff auf die markierten Objekte. Um zu vermeiden, dass Änderungen versehentlich in die Datenbank geschrieben werden, schreibt das System die Änderungen für jede Aktivität zunächst in einen Pufferbereich. Danach sind alle Drucktasten für die Aktivität deaktiviert, außer *Sichern* und *Rückgängig*. So können Sie die geänderten Objekte in der Liste noch einmal überprüfen. Im Anschluss haben Sie folgende Möglichkeiten:

- Wenn alles wie gewünscht geändert wurde, wählen Sie *Sichern*.
- Ist dies nicht der Fall, wählen Sie *Rückgängig*. Dann setzt das System alle markierten Objekte auf ihren vorigen Status zurück.

Nachdem Sie *Sichern* oder *Rückgängig* gewählt haben, aktiviert das System die Drucktasten für die Aktivität wieder. Die zuvor markierten Objekte bleiben markiert, um weitere Änderungen vornehmen zu können.

Hinweis

Die Funktion *Rückgängig* ist nicht nur für globale Änderungen verfügbar, die Sie mithilfe der verschiedenen Drucktasten für eine Aktivität vorgenommen haben. Sie können damit auch direkte Änderungen vornehmen, die Sie manuell auf einzelne Objekte im Tool zur Massenänderung angewendet haben, z. B. geänderte Texte. Die Funktion *Rückgängig* kann jedoch nur auf die Positionen angewendet werden, die aktuell in der Liste ausgewählt sind.

Benachrichtigungskonzept

Das System prüft für jedes zu ändernde ausgewählte Objekt, ob die gewünschte Änderung angewendet werden kann oder nicht. Ermittelt das System Probleme, die die Änderung des Objekts verhindern, gibt es eine entsprechende Meldung aus. Zudem finden Sie in der Spalte *Nachrichten* ein Symbol, das angibt, ob die Änderung erfolgreich war oder nicht.

Hinweis

Wenn Sie eine Änderung auf viele Objekte anwenden und diese Änderung für einen Großteil der Objekte fehlschlägt, führt dies zu einer langen Liste von Meldungen im Nachrichtenbereich. In diesem Fall ist es unter Umständen schwierig, die einzelnen Meldungen den entsprechenden Objekten zuzuordnen. Um zwischen einem Objekt und der dazugehörigen Systemmeldung zu navigieren, wählen Sie in der Spalte **Nachricht** des betreffenden Objekts das Symbol. Dann zeigt das System im Nachrichtenbereich nur die Meldung für dieses Objekt an. Wenn Sie das Meldungssymbol eines anderen Objekts wählen, wird der Nachrichtenbereich entsprechend aktualisiert.

Aktivitäten für Massenänderungen

Prüfen

Mit dieser Funktion führt das System eine Konsistenzprüfung für alle markierten Positionen durch. Nach Abschluss der Prüfung zeigt das System alle Prüfmeldungen in sequenzieller Reihenfolge oben im Bild an. Zudem wird für Positionen mit Problemen in kleines Symbol in der Spalte **Nachrichten** angezeigt, das angibt, welche Art von Problem für diese Position vorliegt. Durch diese doppelte Meldungsangabe sehen Sie sowohl wiederkehrende Probleme, die auf eine größere Anzahl von Objekten zutreffen als auch objektspezifische Probleme.

Hinweis

Das Prüfen von Objekten ist die einzige Aktivität im Tool zur Massenänderung, für die kein Schreibzugriff auf die markierten Positionen verfügbar ist. Folglich sind die Drucktasten **Sichern** und **Rückgängig** nach einer Prüfung nicht erforderlich.

Aktivieren

Mit dieser Funktion aktivieren Sie alle markierten Positionen. Dies ist hilfreich, da Sie in der BRFplus-Workbench Objekte nicht zwingend aktivieren müssen. Es kann also vorkommen, dass Sie eine Reihe von inaktiven Objekten angelegt haben, die Sie anderenfalls aufwändig manuell aktivieren müssten. Die Funktion zur Massenaktivierung ist in diesem Fall sehr praktisch.

Hinweis

Im Gegensatz zur Aktivierung einzelner Objekte in der BRFplus-Workbench unterstützt das Tool zur Massenänderung die automatische Aktivierung von Unterobjekten nicht. Sie können diesen Vorgang jedoch trotzdem durchführen, indem Sie das Kennzeichen **Verwendungen** setzen.

Als veraltet markieren/Kennzeichen „Veraltet“ zurücksetzen

Mit dieser Funktion kennzeichnen Sie alle markierten Positionen als veraltet. Objekte, die als veraltet markiert sind, verbleiben in Kontexten funktional, in denen sie bereits verwendet werden, können aber nicht in Kombination mit neuen Regelaufstellungen eingesetzt werden. Mit der Complementary-Funktion können Sie das Veraltet-Kennzeichen für alle markierten Objekte wieder entfernen. Dann stehen diese Objekte wieder für die produktive Verwendung bereit.

Zum Löschen vormerken/Löschvormerkung aufheben

Mit dieser Funktion merken Sie alle markierten Positionen zum Löschen vor. Zum Löschen vorgemerkte Objekte verbleiben in Kontexten funktional, in denen sie bereits verwendet werden. Es ist jedoch nicht möglich, Änderungen auf die referenzierenden Objekte vorzunehmen, ohne die Referenz zuvor zu ersetzen. Mit der

Complementary-Funktion können Sie das Kennzeichen „Löschvormerkung aufheben“ für alle markierten Objekte wieder entfernen. Dann stehen diese Objekte wieder für die produktive Verwendung bereit.

Löschen

Mit dieser Funktion löschen Sie alle markierten Positionen. Die betroffenen Objekte stehen in BRFPplus dann nicht mehr zur Verfügung.

Achtung

Setzen Sie die Aktionen *Als „veraltet“ markieren*, *Zum Löschen vormerken* und *Löschen* nur nach sorgfältiger Überlegung ein. Ausführliche Informationen zu den Auswirkungen dieser Aktionen finden Sie unter [Objekte löschen \[Seite 57\]](#).

Zugriffsebene auswählen

Mit dieser Funktion ändern Sie die Zugriffsebene aller markierten Positionen. Dies ist sehr nützlich, wenn Sie die Zugriffsebene für eine ganze BRFPplus-Anwendung ändern müssen, beispielsweise nach einer Änderung der Sicherheitsanforderungen oder nach Änderungen in der Systemlandschaft.

Versionierungsstatus einstellen

Mit dieser Funktion schalten Sie den Versionierungsstatus aller markierten Positionen ein oder aus. Dies ist z. B. hilfreich, wenn Sie zu Testzwecken die Versionierung für eine Anwendung aktiviert hatten. Wenn Sie sicher sind, dass die Anwendung wie gewünscht funktioniert, können Sie die Versionierung abschalten, da keine Versionskontrolle mehr erforderlich ist. Im umgekehrten Fall müssen Sie z. B. aufgrund geänderter Vorschriften ab einem bestimmten Zeitpunkt eine spezifische Anwendung mit Versionskontrolle ausführen.

Umbenennen

Mit dieser Funktion führen Sie eine einfache Umbenennungsaktion für die markierten Positionen durch. Das System fordert Sie auf, ein Präfix, Suffix oder beides einzugeben, die dann dem Namen der markierten Positionen hinzugefügt werden. Mit dieser einfachen Aktion sparen Sie viel Zeit, da Änderungen am technischen Namen eines Objekts nicht alltäglich sind. Die Aktion wird deshalb durch zusätzliche Navigationsschritte in der Workbench abgesichert. Im Tool zur Massenänderung können Sie diese Sicherheitsschritte umgehen.

Hinweis

BRFPplus identifiziert seine Objekte über die ID, nicht über den Namen. Innerhalb des Geltungsbereichs von BRFPplus ist das Ändern von technischen Namen daher unkritisch. Allerdings ist es möglich, dass eine Objekt von einem ABAP-Backend-Programm, das BRFPplus aufruft, direkt über seinen Namen referenziert wird. Sie müssen daher sicherstellen, dass alle externen, auf Namen basierten Verweise auf BRFPplus-Objekte mit den Änderungen innerhalb von BRFPplus übereinstimmen.

Die Umbenennung betrifft lediglich den technischen Namen eines Objekts, nicht den Kurztext oder Text. Den Text der betroffenen Objekte können Sie manuell im Tool zur Massenänderung ändern. Den Kurztext können Sie jedoch nur in der Workbench ändern.

Aufgrund der Logik des Umbenennungsmechanismus wird der aktuelle Name der betroffenen Objekte immer als Teil des neuen Namens beibehalten. Ursprüngliche Namen können mit dieser Funktion nicht ersetzt werden.

Kopieren

Mit dieser Funktion erstellen Sie eine Kopie der markierten Positionen, entweder in der aktuellen oder in einer anderen Anwendung. Diese Funktion liefert verschiedene Ergebnisse, abhängig von den gewählten Optionen. Weitere Informationen finden Sie unter [Objekte mit dem Tool zur Massenänderung kopieren \[Seite 285\]](#).

In XML exportieren

Mit dieser Funktion legen Sie eine XML-Version der markierten Positionen an und exportieren diese in eine externe Datei. Die Funktion basiert auf der standardmäßigen XML-Exportfunktion von BRFplus. Sie haben hier aber zusätzlich die Option, bestimmte Objekte, deren XML-Darstellung in derselben Exportdatei enthalten ist, zusammenzuführen. Dies ist z. B. dann nützlich, wenn eine Reihe von Objekten in einem Remote-System beschädigt wurden, und Sie diese durch einen XML-Import der ursprünglichen Versionen ersetzen möchten – ohne andere Objekte im Zielsystem zu bearbeiten.

Weitere Informationen

[Verwaltungsdaten \[Seite 39\]](#)

[XML-Daten im- und exportieren \[Seite 274\]](#)

[Versionierung \[Seite 176\]](#)

[Objekte löschen \[Seite 57\]](#)

1.3.16.11.1 Objekte mit dem Tool zur Massenänderung kopieren

Verwendung

Mit dem Tool zur Massenänderung kopieren Sie BRFplus-Objekte in folgenden Situationen:

- Sie benötigen eine Kopie einer Reihe bereits vorhandener Objekte für ein neues Szenario einer Geschäftsregel. Diese weist eine identische Struktur einer bereits vorhandenen Anwendung auf, an der lediglich einige Änderungen vorgenommen werden müssen.
- Sie benötigen eine Kopie eines sehr komplexen Objekts und möchten sicherstellen, dass alle abhängigen Objekte dieses Objekts ebenfalls kopiert werden.

Funktionsumfang

Das Tool zur Massenänderung bietet verschiedene Optionen, mit denen Sie genau steuern können, wie Objekte kopiert werden und wie diese kopierten Objekte in Bezug auf ihre abhängigen Objekte aussehen. Diese Optionen sind über verschiedene Menübefehle verfügbar, kombiniert mit dem Status des Kennzeichens [Verwendungen](#) der zu kopierenden Objekte:

Szenario 1: Markierte Positionen kopieren, Referenzen auf abhängige Objekte bleiben unverändert

Gehen Sie für dieses Szenario wie folgt vor:

- Wählen Sie ► **Kopieren** ► **Markierte Positionen kopieren** ►.
- Kennzeichen **Verwendungen**: **Nicht** gesetzt

In diesem Szenario kopiert das System alle markierten Positionen. Die neu angelegten Objekte enthalten Referenzen auf genau dieselben abhängigen Objekten wie das ursprüngliche Objekt.

Beispiel

Sie möchten eine Kopie der Funktion **F** anlegen. Für Funktion **F** wurde die Entscheidungstabelle **DT** als oberster Ausdruck zugewiesen. **DT** verwendet die drei Datenobjekte **DO1**, **DO2** und **DO3**. Wenn Sie Funktion **F** mit den obigen Einstellungen kopieren, sieht das Ergebnis wie folgt aus:

Tabelle 46:

Vorher	Nachher	Kommentar
Funktion F	Funktion F	Referenz auf DT
	Funktion F'	Neue Funktion F' mit Referenz auf DT wie Funktion F , d. h. auf dasselbe Objekt
Entscheidungstabelle DT	Entscheidungstabelle DT	Wird von F und F' verwendet
Datenobjekt DO1	Datenobjekt DO1	Wird von DT verwendet
Datenobjekt DO2	Datenobjekt DO2	Wird von DT verwendet
Datenobjekt DO3	Datenobjekt DO3	Wird von DT verwendet

Szenario 2: Markierte Positionen und abhängige Objekte kopieren, Referenzen auf abhängige Objekte bleiben unverändert

Gehen Sie für dieses Szenario wie folgt vor:

- Wählen Sie ► **Kopieren** ► **Markierte Positionen kopieren** ►.
- Kennzeichen **Verwendungen**: **Gesetzt**

In diesem Szenario kopiert das System alle markierten Positionen. Da das Kennzeichen **Verwendungen** gesetzt ist, werden die abhängigen Objekte der übergeordneten Objekte ebenfalls kopiert. Die neu angelegten übergeordneten Objekte referenzieren jedoch weiterhin auf dieselben abhängigen Objekten wie die ursprünglichen Objekte.

Durch das Setzen des Kennzeichens **Verwendungen** für Funktion **F** werden alle Objekte dieses Beispielszenarios ausgewählt:

- Funktion **F**
- Entscheidungstabelle **DT**, wird von **F** verwendet
- Datenobjekte **DO1**, **DO2**, **DO3**, werden von **DT** verwendet

Beispiel

Sie möchten eine Kopie der Funktion **F** anlegen. Für Funktion **F** wurde die Entscheidungstabelle **DT** als oberster Ausdruck zugewiesen. **DT** verwendet die drei Datenobjekte **DO1**, **DO2** und **DO3**. Wenn Sie die markierten Objekte mit den obigen Einstellungen kopieren, sieht das Ergebnis wie folgt aus:

Tabelle 47:

Vorher	Nachher	Kommentar
Funktion F	Funktion F	Referenz auf DT
	Funktion F'	Neue Funktion F' mit Referenz auf DT wie Funktion F , d. h. auf dasselbe Objekt
Entscheidungstabelle DT	Entscheidungstabelle DT	Wird von F und F' verwendet
	Entscheidungstabelle DT'	Neue Entscheidungstabelle DT' wurde als Kopie von DT angelegt, wird aber zurzeit nicht verwendet
Datenobjekt DO1	Datenobjekt DO1	Wird von DT und DT' verwendet
Datenobjekt DO2	Datenobjekt DO2	Wird von DT und DT' verwendet
Datenobjekt DO3	Datenobjekt DO3	Wird von DT und DT' verwendet
	Datenobjekt DO1'	Neues Datenobjekt, zurzeit nicht verwendet
	Datenobjekt DO2'	Neues Datenobjekt, zurzeit nicht verwendet
	Datenobjekt DO3'	Neues Datenobjekt, zurzeit nicht verwendet

Szenario 3: Markierte Positionen und abhängige Objekte kopieren, Referenzen auf Kopien abhängiger Objekte ändern

Gehen Sie für dieses Szenario wie folgt vor:

- Wählen Sie **Kopieren** > **Markierte Positionen rekursiv kopieren**.
- Kennzeichen **Verwendungen**: **Nicht** gesetzt

In diesem Szenario kopiert das System alle markierten Positionen. Das System kopiert für jede Position auch rekursiv die Gesamthierarchie der abhängigen Objekte. Die kopierten Objekte entsprechen denen im oben beschriebenen *Szenario 2*, obwohl nur Funktion **F** markiert wurde. Nachdem alle Objekte kopiert wurden, legt das System die ursprüngliche Objekthierarchie mithilfe der neu angelegten Objekte erneut an. Anders ausgedrückt: Die ursprünglichen Objekte werden durch die neu angelegten Objekte vollständig gespiegelt.

Beispiel

Sie möchten eine Kopie der Funktion **F** anlegen. Für Funktion **F** wurde die Entscheidungstabelle **DT** als oberster Ausdruck zugewiesen. **DT** verwendet die drei Datenobjekte **DO1**, **DO2** und **DO3**. Wenn Sie Funktion **F** mit den obigen Einstellungen kopieren, sieht das Ergebnis wie folgt aus:

Tabelle 48:

Vorher	Nachher	Kommentar
Funktion F	Funktion F	Referenz auf DT
	Funktion F'	Neue Funktion F' mit Referenz auf DT'
Entscheidungstabelle DT	Entscheidungstabelle DT	Wird von F verwendet
	Entscheidungstabelle DT'	Neue Entscheidungstabelle DT' , wird von F' verwendet
Datenobjekt DO1	Datenobjekt DO1	Wird von DT verwendet
Datenobjekt DO2	Datenobjekt DO2	Wird von DT verwendet
Datenobjekt DO3	Datenobjekt DO3	Wird von DT verwendet
	Datenobjekt DO1'	Wird von DT' verwendet
	Datenobjekt DO2'	Wird von DT' verwendet
	Datenobjekt DO3'	Wird von DT' verwendet

Weitere Hinweise

Mit dem im letzten beschriebenen Szenario verwendeten Befehl *Markierte Positionen rekursiv kopieren* führt das System implizit einen Drilldown von den markierten Objekten der obersten Ebene bis zu allen dazugehörigen Komponenten und Unterkomponenten durch. Deshalb kann der Befehl *Markierte Positionen rekursiv kopieren* nicht in Kombination mit einem gesetzten Kennzeichen *Verwendungen* verwendet werden.

Auch wenn Sie mit dem Tool zur Massenänderung Objekte von einer Anwendung in eine andere kopieren können, so kann das Tool keine anwendungsübergreifenden Objektbeziehungen auflösen. Solche anwendungsübergreifenden Konstellationen können für Objekte auftreten, deren Zugriffsebene als Systemstandard auf eine weniger begrenzte Ebene als *Anwendung* festgelegt wurde. Hier stoppt der Objekt-Drilldown, wenn das Tool ein abhängiges Objekt ermittelt, das aus einer anderen Anwendung stammt. Damit die Kopie funktioniert, müssen Sie die fehlenden Objekte manuell hinzufügen.

Ausschlussklauseln und rechtliche Aspekte

Coding-Beispiele

Bei dem in der vorliegenden Dokumentation enthaltenen Quell- und/oder Objektcode für Software („Code“) handelt es sich ausschließlich um eine beispielhafte Darstellung. Dieser Code ist in keinem Fall für die Nutzung in einem produktiven System geeignet. Der Code dient ausschließlich dem Zweck, beispielhaft aufzuzeigen, wie Quelltext erstellt und gestaltet werden kann. SAP übernimmt keine Gewährleistung für die Funktionsfähigkeit, Richtigkeit und Vollständigkeit des hier abgebildeten Codes, und SAP übernimmt keine Haftung für Schäden, die durch die Nutzung des Codes entstehen, sofern solche Schäden nicht durch vorsätzliches oder grob fahrlässiges Verhalten der SAP verursacht wurden.

Barrierefreiheit

Die in der Dokumentation der SAP-Bibliothek enthaltenen Informationen stellen Kriterien der Barrierefreiheit aus Sicht von SAP zum Zeitpunkt der Veröffentlichung dar und sollen keineswegs obligatorische Richtlinien sein, wie die Barrierefreiheit von Softwareprodukten zu gewährleisten ist. SAP lehnt insbesondere jede Haftung in Bezug auf dieses Dokument ab, (die nicht aus dem vorsätzlichen oder grob fahrlässigen Handeln der SAP resultieren), aus dem weder direkt noch indirekt irgendwelche vertraglichen Verpflichtungen entstehen.

Geschlechtsneutrale Sprache

Die SAP-Dokumentation ist, sofern sprachlich möglich, geschlechtsneutral formuliert. Je nach Kontext wird die direkte Anrede mit „Sie“ oder ein geschlechtsneutrales Substantiv (wie z.B. „Fachkraft“ oder „Personentage“) verwendet. Wenn, um auf Personen beiderlei Geschlechts Bezug zu nehmen, die dritte Person Singular nicht vermieden werden kann oder es kein geschlechtsneutrales Substantiv gibt, wird aus Gründen der besseren Lesbarkeit durchgängig die männliche Form des Substantivs und des Pronomens verwendet. Hierdurch wird sichergestellt, dass die Dokumentation verständlich bleibt.

Internet-Hyperlinks

Die SAP-Dokumentation kann Hyperlinks auf das Internet enthalten. Diese Hyperlinks dienen lediglich als Hinweis auf ergänzende und weiterführende Dokumentation. SAP übernimmt keine Gewährleistung für die Verfügbarkeit oder Richtigkeit dieser ergänzenden Information oder deren Nutzbarkeit für einen bestimmten Zweck. SAP übernimmt keine Haftung für Schäden, die durch die Nutzung solcher Informationen verursacht werden, es sei denn, dass diese Schäden von SAP grob fahrlässig oder vorsätzlich verursacht wurden. Informationen zur Klassifizierung von Links finden Sie unter: <http://help.sap.com/disclaimer>.



**go.sap.com/registration/
contact.html**

© 2017 SAP SE oder ein SAP-Konzernunternehmen Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP SE oder ein SAP-Konzernunternehmen nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP SE oder deren Vertriebsfirmen angebotenen Softwareprodukte können Softwarekomponenten auch anderer Softwarehersteller enthalten. Produkte können länderspezifische Unterschiede aufweisen.

Die vorliegenden Unterlagen werden von der SAP SE oder einem SAP-Konzernunternehmen bereitgestellt und dienen ausschließlich zu Informationszwecken. Die SAP SE oder ihre Konzernunternehmen übernehmen keinerlei Haftung oder Gewährleistung für Fehler oder Unvollständigkeiten in dieser Publikation. Die SAP SE oder ein SAP-Konzernunternehmen steht lediglich für Produkte und Dienstleistungen nach der Maßgabe ein, die in der Vereinbarung über die jeweiligen Produkte und Dienstleistungen ausdrücklich geregelt ist. Keine der hierin enthaltenen Informationen ist als zusätzliche Garantie zu interpretieren.

SAP und andere in diesem Dokument erwähnte Produkte und Dienstleistungen von SAP sowie die dazugehörigen Logos sind Marken oder eingetragene Marken der SAP SE (oder von einem SAP-Konzernunternehmen) in Deutschland und verschiedenen anderen Ländern weltweit. Alle anderen Namen von Produkten und Dienstleistungen sind Marken der jeweiligen Firmen.

Zusätzliche Informationen zur Marke und Vermerke finden Sie auf der Seite <http://www.sap.com/corporate-de/legal/copyright/index.epx>.