

HP Service Manager

Für unterstützte Windows®- und UNIX®-Betriebssysteme

Softwareversion: 9.30

Handbuch zum Dokumentenmodul

Datum der Dokumentveröffentlichung: Juli 2011

Datum des Software-Release: Juli 2011



Rechtliche Hinweise

Garantie

Die Garantiebedingungen für Produkte und Services von HP sind in der Garantieerklärung festgelegt, die diesen Produkten und Services beiliegt. Keine der folgenden Aussagen kann als zusätzliche Garantie interpretiert werden. HP haftet nicht für technische oder redaktionelle Fehler oder Auslassungen.

Die hierin enthaltenen Informationen können ohne vorherige Ankündigung geändert werden.

Eingeschränkte Rechte

Vertrauliche Computersoftware. Gültige Lizenz von HP für den Besitz, Gebrauch oder die Anfertigung von Kopien erforderlich. Entspricht FAR 12.211 und 12.212; kommerzielle Computersoftware, Computersoftwaredokumentation und technische Daten für kommerzielle Komponenten werden an die US-Regierung per Standardlizenz lizenziert.

Copyright-Hinweis

© Copyright 1994-2011 Hewlett-Packard Development Company, L.P.

Marken

Adobe™ ist eine Marke von Adobe Systems Incorporated. Java™ und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken von Sun Microsystems, Inc. in den USA und anderen Ländern.

Java™ ist eine in den USA eingetragene Marke von Sun Microsystems, Inc.

Microsoft® und Windows® sind in den USA eingetragene Marken der Microsoft Corporation.

Oracle® ist eine in den USA eingetragene Marke der Oracle Corporation, Redwood City, California.

UNIX® ist eine eingetragene Marke von The Open Group.

Aktualisierte Dokumentation

Auf der Titelseite dieses Dokuments befinden sich die folgenden bezeichnenden Informationen:

- Software-Versionsnummer zur Angabe der Version der Software
- Datum der Dokumentveröffentlichung, das bei jeder Änderung des Dokuments ebenfalls aktualisiert wird
- Datum des Software-Release, das angibt, wann diese Version der Software veröffentlicht wurde

Unter der unten angegebenen Internetadresse können Sie überprüfen, ob neue Updates verfügbar sind, und sicherstellen, dass Sie mit der neuesten Version eines Dokuments arbeiten:

<http://h20230.www2.hp.com/selfsolve/manuals>.

Für die Anmeldung an dieser Website benötigen Sie einen HP Passport. Hier können Sie sich für eine HP Passport-ID registrieren:

<http://h20229.www2.hp.com/passport-registration.html>.

Alternativ können Sie auf den Link **New user registration** (Neue Benutzer registrieren) auf der HP Passport-Anmeldeseite klicken.

Wenn Sie sich beim Support-Service eines bestimmten Produkts registrieren, erhalten Sie ebenfalls aktualisierte Softwareversionen und überarbeitete Ausgaben der zugehörigen Dokumente. Weitere Informationen erhalten Sie bei Ihrem HP-Kundenbetreuer.

Support

Besuchen Sie die HP Software Support Online-Website von HP unter:

<http://www.hp.com/go/hpsoftwaresupport>

Auf dieser Website finden Sie Kontaktinformationen und Details zu Produkten, Services und Supportleistungen von HP Software.

Der Online-Support von HP Software bietet Kunden mit Hilfe interaktiver technischer Support-Werkzeuge die Möglichkeit, ihre Probleme intern zu lösen. Als Valued Support Customer können Sie die Support-Website für folgende Aufgaben nutzen:

- Suchen nach interessanten Wissensdokumenten
- Absenden und Verfolgen von Support-Fällen und Erweiterungsanforderungen
- Herunterladen von Software-Patches
- Verwalten von Support-Verträgen
- Nachschlagen von HP-Supportkontakten
- Einsehen von Informationen über verfügbare Services
- Führen von Diskussionen mit anderen Softwarekunden
- Suchen und Registrieren für Softwareschulungen

Für die meisten Support-Bereiche müssen Sie sich als Benutzer mit einem HP Passport registrieren und anmelden. In vielen Fällen ist zudem ein Support-Vertrag erforderlich. Hier können Sie sich für eine HP Passport-ID registrieren:

<http://h20229.www2.hp.com/passport-registration.html>.

Weitere Informationen zu Zugriffsebenen finden Sie unter:

http://h20230.www2.hp.com/new_access_levels.jsp

Inhalt

Handbuch zum Dokumentenmodul.....	1
Inhalt.....	5
Übersicht über das Dokumentenmodul.....	9
Was ist das Dokumentenmodul?.....	9
Vorteile des modularen Ansatzes.....	9
Konsistenz.....	9
Kürzere Entwicklungszeiten.....	10
Flexibilität.....	10
Zugriff auf das Dokumentenmodul.....	10
Objekte.....	11
Erstellen und Aktualisieren von Objekten.....	12
Formular "Objektdefinition" und Felder.....	12
Feldbeschreibungen des Registers "Objektinformationen".....	13
Feldbeschreibungen des Registers "Sperrern".....	15
Feldbeschreibungen des Registers "Revisionen".....	16
Feldbeschreibungen des Registers "Variablen/Globale Listen".....	16
Feldbeschreibungen des Registers "Aktivitäten".....	17
Feldbeschreibungen des Registers "Alerts".....	18
Feldbeschreibungen des Registers "Genehmigungen".....	19
Feldbeschreibungen des Registers "Warteschlangen verwalten".....	21
Feldbeschreibungen des Registers "Ansichten/Vorlagen".....	23
Feldbeschreibungen des Registers "Benachrichtigungen".....	23
Feldbeschreibungen des Registers "Suchkonfiguration".....	24
Register "Definierte Abfragen".....	24
Register "Bereiche".....	25
Zustände.....	27
Suche.....	27
Datensatzlisten.....	27

Anzeigen eines einzelnen Datensatzes.....	27
Durchsuchen eines Datensatzes.....	27
Integrationstipps für die Display-Anwendung.....	28
Erstellen und Aktualisieren von Zuständen.....	28
Formular "Zustandsdefinition" und Felder.....	28
Prozesse.....	31
Erstellen und Ändern eines Prozesses.....	31
Formular "Prozessdefinition" und Felder.....	31
Register "Anfangsausdrücke".....	32
Register "Anfangs-Javascript".....	33
Register "RAD".....	33
Register "Endausdrücke".....	33
Register "End-Javascript".....	34
Register "Nächster Prozess".....	34
Dokumentenmodulressourcen.....	35
DEFAULT-Objekt.....	35
RAD-Anwendungen.....	35
se.search.engine.....	36
se.list.engine.....	36
se.view.engine.....	36
RAD-Anwendungsfluss.....	36
Integrationstipps für die Display-Anwendung.....	37
Basisfunktionen in "se.search.engine".....	38
Basisfunktionen in "se.view.engine".....	39
Basisfunktionen in "se.list.engine".....	42
Lokale Variablen.....	42
Übersicht über die Fehlerbehebung.....	45
Untersuchen des Anwendungspfads durch das Dokumentenmodul.....	45
Auffinden des verwendeten DBDicts oder Objekts.....	45
Auffinden des Zustands für die Datensätze.....	45
Auffinden des Namens des Prozesses.....	46
Untersuchen von Anwendungsfehlern.....	46

Druckwerte von Variablen oder Ergebnisse von Ausdrücken	47
Übersicht über das Arbeitsauftragsbeispiel.....	49
Erstellen einer Tabelle.....	50
Hinzufügen von Schlüsselfeldern zu einer Tabelle	51
Erstellen eines Formulars.....	52
Erstellen einer Kopie eines Formulars.....	53
Erstellen eines Links für das Arbeitsauftragsformular.....	54
Erstellen einer Datei für Sequenznummern.....	55
Erstellen einer Objektdefinition.....	55
Erstellen einer Initialisierungsprozessdefinition.....	57
Erstellen von Bildschirmdefinitionen für die Display-Anwendung.....	57
Erstellen von Optionsdefinitionen für die Display-Anwendung.....	59
Erstellen einer Zustandsdefinition.....	64
Hinzufügen einer Schaltfläche zum Schließen von Arbeitsaufträgen.....	66
Erstellen eines Assistenten für den Arbeitsauftrag.....	68
Hinzufügen eines Prozessdefinitions-Datensatzes.....	71
Erstellen eines Eingabeformulars für den Assistenten.....	72
Ändern der Formulare zum Schließen und zum Aktualisieren von Incidents.....	74
Erstellen eines Alias zum Verknüpfen in der Tabelle "probsummary".....	76
Ändern der Definition des Prozesses "im.set.close".....	77
Testen des Arbeitsauftragsbeispiels.....	79

Kapitel 1

Übersicht über das Dokumentenmodul

Beim Dokumentenmodul handelt es sich um ein Anpassungswerkzeug, das Möglichkeiten zum Anpassen des Systems bietet, ohne dass dafür RAD-Änderungen erforderlich sind. Es bietet eine zentralisierte Methode zum Festlegen von Berechtigungen und Verhalten für Standardaktionen wie das Listen, Anzeigen und Suchen. Es verbessert die Konsistenz modulübergreifend. Die drei Hauptkomponenten des Dokumentenmoduls sind Objekte, Zustände und Prozesse. Prozesse können für eine modularere Programmierung wiederverwendet werden. Ihre Integration erfolgt nahtlos, was zu einer Verkürzung der Entwicklungszeiten beiträgt.

Was ist das Dokumentenmodul?

Das Dokumentenmodul umfasst eine Reihe von Werkzeugen und Methoden zum Entwickeln und Ändern von Service Manager-Workflows. Das Dokumentenmodul wurde ursprünglich dafür konzipiert, zugrundeliegende Basisfunktionen zu entwickeln, die verschiedene Module innerhalb von Service Manager unterstützen, die Konsistenz der Benutzerschnittstelle zwischen Modulen verbessern und die Menge an erforderlichem Code für neue Module reduzieren.

Das Dokumentenmodul erweitert die Funktionalität der Display-Anwendung mit vereinfachten und umfangreicheren Aktionen, darunter insbesondere Aktionen mit mehreren Anwendungsaufrufen. Zusätzlich unterstützt das Dokumentenmodul die Verwendung von über einen Join verknüpften Tabellen und Basis-Format Control-Aufrufen. Es wurde so entwickelt, dass es die Bedürfnisse der meisten Kunden vordefiniert erfüllt und dennoch flexibel ist. Die Beziehung zwischen Objekten, Zuständen und Prozessen ist hierarchisch.

Das Dokumentenmodul steuert das Verhalten von Objekten. Wann immer ein Formular geöffnet wird, wird auf ein Objekt verwiesen. Das Objekt bestimmt das Verhalten für den Zustand des Formulars (**Geöffnet**, **Liste**, **Suche** usw.). Objekte definieren das Tabellenverhalten insgesamt. Innerhalb des Objekts beschreibt ein Zustand die Position eines Datensatzes in seinem Lebenszyklus (**Geöffnet**, **Liste**, **Suche** usw.). Innerhalb des Zustands werden verschiedene Prozesse ausgeführt, abhängig von den Aktionen, die von einem Benutzer für den Datensatz initiiert werden. Zustände definieren auch, wie das System einen Datensatz anzeigt und welche Optionen (Aktionen) zu bestimmten Zeiten oder unter bestimmten Umständen zur Verfügung stehen. Je nach Zugriffsberechtigung des Benutzers können Zustände beispielsweise Aktionen wie **Speichern** bestimmen.

Prozesse werden aus Zuständen basierend auf der Aktion des Benutzers aufgerufen. Der Prozess verwendet RAD-Ausdrücke, Javascript und ruft vorhandene RAD-Anwendungen zum Ausführen von Aktionen für den aktuellen Datensatz auf.

Vorteile des modularen Ansatzes

Zu den Vorteilen des modularen Designs zählen Konsistenz in der Entwicklung, kürzere Entwicklungszeiten und Flexibilität.

Konsistenz

Das Modul ermöglicht für sämtliche Anwendungen die Ausführung unter Verwendung derselben grundlegenden RAD-Anwendungen und trägt dadurch zur Konsistenz der Service Manager-

Anwendungssuite bei. Basisfunktionen wie Sperren, Alerts, Genehmigungen und die Verwendung einer Datenbankliste stehen für alle Module in gleicher Weise zur Verfügung, da sie alle dieselbe Codebasis verwenden.

Kürzere Entwicklungszeiten

Die Modularität des Dokumentenmoduls ermöglicht das Wiederverwenden vorhandenen Codes und vorhandener Prozesse.

Flexibilität

Das Dokumentenmodul verwendet Prozessdatensätze als Mechanismus zum Ändern des Verhaltens von Modulen in Service Manager-Anwendungen. Sie können einen neuen Prozess mit einem anderen Verhalten als dem das Basissystems erstellen. Hierzu müssen Sie den ursprünglichen Prozess weder ändern noch vom System entfernen. Darüber hinaus können Sie die Basisprozesse des Systems mit Ihren eigenen Prozessen überschreiben. Dies bietet dem Systementwickler mehr Flexibilität beim Anpassen von Service Manager an die Anforderungen des Unternehmens.

Zugriff auf das Dokumentenmodul

So greifen Sie auf das Dokumentenmodul zu:

1. Starten Sie einen Service Manager-Client und melden Sie sich als Administrator an.
2. Klicken Sie im Service Manager-Systemnavigator auf **Anpassung**.
3. Klicken Sie auf **Dokumentenmodul**. Von hier aus können Sie auf die drei Hauptbereiche zugreifen, um Objekte, Prozesse oder Zustände zu definieren. Sie können auch Alerts und Genehmigungen einrichten, die von Objekten aufgerufen werden sollen, oder Sie können Suchkonfigurationsdatensätze erstellen, die von Objekten verwendet werden.

Kapitel 2

Objekte

Objekte sind eine grundlegende Gruppe von Definitionen, die das Verhalten von Datensätzen bestimmen und die Definitionen und entscheidenden Regeln festlegen. Objekte entsprechen Database Dictionary-Datensätzen (DBDict-Datensätzen) in Service Manager. Wenn eine Tabelle nicht über einen dedizierten Objektdatensatz verfügt, wendet das Dokumentenmodul die im DEFAULT-Objekt festgelegten Einstellungen an. Alle Objekte sollten über definierte Listen- und Standardzustände verfügen. Wenn nicht anders angegeben, lauten die Standardzustände **db.browse**, **db.list**, **db.search** und **db.view**.

HINWEIS: Ändern oder löschen Sie den DEFAULT-Objektdatensatz **nicht**, da dies zu unvorhersehbaren Ergebnissen führt.

Der Objektdatensatz legt die Definitionen und entscheidenden Regeln für das Verhalten der Tabelle innerhalb des Dokumentenmoduls fest. Beispielsweise kann dazu Folgendes zählen:

- Die Anwendung, die zum Erstellen des Benutzerprofils innerhalb dieses Objekts verwendet wird und bestimmt, welche Aktionen diesem Benutzer für die Datensätze der Tabelle zur Verfügung stehen
- Die unter bestimmten Umständen verwendeten Zustandsdatensätze (weitere Details finden Sie im Abschnitt zu den Zuständen)
- Die Kategorie, die Phase und Paging-Datenamen für das Objekt
- Der Name des Nummerdatensatzes, der für dieses Objekt verwendet werden soll
- Die Verwendung der Sperroption für dieses Objekt
- Das Einrichten von Revisionen für Datensätze in der Tabelle
- Welche Variablen für Prozesse zur Verfügung stehen sollen, die für dieses Objekt ausgeführt werden
- Welche globalen Listen immer zur Verfügung stehen sollen, wenn dieses Objekt verwendet wird
- Die Verwendung von Aktivitätsdatensätzen
- Wie Alerts für dieses Objekt verarbeitet werden
- Wie Genehmigungen für dieses Objekt verarbeitet werden
- Die Einstellungen für die Arbeitswarteschlangen
- Die Möglichkeiten zum Einrichten von persönlichen und globalen Ansichten und Standardvorlagen
- Die Benachrichtigungen beim Hinzufügen, Aktualisieren oder Löschen eines Datensatzes in diesem Objekt
- Die Möglichkeit zum Konfigurieren zusätzlicher Suchoptionen für dieses Objekt

Klicken Sie auf dem Formular **Objektdefinition** auf **Suchen**, um eine Liste der vordefinierten Objekte anzuzeigen.

Informationen zum Anzeigen einer Liste der Felder und Feldbeschreibungen für die Objektdefinitionen finden Sie unter [Formular "Objektdefinition" und Felder auf Seite 12](#).

Erstellen und Aktualisieren von Objekten

So erstellen Sie ein Objekt:

1. Wechseln Sie zum Dokumentenmodul. Informationen zu den dazu erforderlichen Schritten finden Sie unter [Zugriff auf das Dokumentenmodul auf Seite 10](#).
2. Doppelklicken Sie auf **Objekte**. Das Formular **Objekte** wird geöffnet.
3. Verwenden Sie die Register im Formular **Objekte**, um die Felder auszufüllen, die erforderlich sind, um ein Objekt zu erstellen, das die von Ihnen gewünschten Funktionen ausführt. Zusätzliche Informationen finden Sie in den Feldbeschreibungen.

So aktualisieren Sie eine Objektdefinition:

1. Greifen Sie auf das Dokumentenmodul zu. Informationen zu den dazu erforderlichen Schritten finden Sie unter [Zugriff auf das Dokumentenmodul auf Seite 10](#).
2. Geben Sie im Feld **Dateiname** den Namen des Objekts ein, das Sie aktualisieren möchten, oder klicken Sie auf **Suchen**, um nach dem Objekt zu suchen.

Formular "Objektdefinition" und Felder

Die Feldbeschreibungen für das Formular **Objektdefinition** lauten folgendermaßen:

Feldname	Beschreibung
Dateiname <i>file.name</i>	Geben Sie den DBDict-Namen für das Objekt ein, bzw. verwenden Sie den DBDict-Namen, der dem Namen dieses Objekts entspricht. (Erforderlich)
Allgemeiner Name <i>message</i>	Das System füllt diese Informationen auf der Grundlage der Daten unter Datenrichtlinien aus. Dies ist ein allgemeiner Name für das Objekt. Der allgemeine Name kann ein einfacher Name sein, beispielsweise "Arbeitsauftrag".
Eindeutiger Schlüssel <i>unique.field</i>	Das System füllt diese Informationen auf der Grundlage der DBDict-Daten aus. Dies ist der eindeutige Schlüssel für das Objekt.

Dieses Formular weist darüber hinaus die folgenden Register auf. Die Felder in diesen Registern werden in den Tabellen zu den Feldbeschreibungen für das jeweilige Register beschrieben.

- [Feldbeschreibungen des Registers "Objektinformationen" auf Seite 13](#) – Gibt die allgemeinen Eigenschaften und das Verhalten des Objekts an.

- [Feldbeschreibungen des Registers "Sperrungen" auf Seite 15](#) – Bestimmt das Sperrverhalten für das Objekt.
- [Feldbeschreibungen des Registers "Revisionen" auf Seite 16](#) – Verfolgt Revisionen für das Objekt.
- [Feldbeschreibungen des Registers "Variablen/Globale Listen" auf Seite 16](#) – Beschreibt lokale und globale Variablen, die vom Objekt verwendet werden.
- [Feldbeschreibungen des Registers "Aktivitäten" auf Seite 17](#) – Verfolgt Revisionen für das Objekt.
- [Feldbeschreibungen des Registers "Alerts" auf Seite 18](#) – Definiert, wo die Alerts und die Bedingungen zum Erstellen von Alerts festgelegt werden sollen.
- [Feldbeschreibungen des Registers "Genehmigungen" auf Seite 19](#) – Legt Genehmigungsoptionen für das Objekt fest.
- [Feldbeschreibungen des Registers "Warteschlangen verwalten" auf Seite 21](#) – Steuert, wie Warteschlangen angezeigt werden, sowie die Thread-Verarbeitung und wer Ansichten (Inboxes) erstellen kann.
- [Feldbeschreibungen des Registers "Ansichten/Vorlagen" auf Seite 23](#) – Definiert, ob ein Benutzer globale und persönliche Ansichten und Vorlagen für das Objekt erstellen kann.
- [Feldbeschreibungen des Registers "Benachrichtigungen" auf Seite 23](#) – Gibt Benachrichtigungen an, die bei Hinzufügungs-, Löschungs- oder Aktualisierungsaktivität eines Datensatzes im Objekt automatisch gesendet werden.
- [Feldbeschreibungen des Registers "Suchkonfiguration" auf Seite 24](#) – Steuert die verfügbaren Optionen im Menü **Weitere Aktionen** des Suchbildschirms.

Feldbeschreibungen des Registers "Objektinformationen"

Dieses Register gibt die allgemeinen Eigenschaften und das Verhalten des Objekts an.

Die Feldbeschreibungen für das Register **Objektinformationen** lauten folgendermaßen:

Feldname	Beschreibung
Beschreibungsfeld <i>desc.field</i>	Gibt eine Kurzbeschreibung des Objekts an.
Profilanwendung <i>profile.appl</i>	Gibt die RAD-Anwendung für die Erstellung des Profils an, das bestimmt, ob ein Benutzer bestimmte Funktionen (z. B. Hinzufügen und Löschen) ausführen kann. Beispiel: db.environment . (Erforderlich)
Profilvariable <i>profile.variable</i>	Gibt eine Variable an, auf die bei jedem Aufruf dieses Objekts zugegriffen werden kann, ohne dass dafür ein Zugriff auf den Umgebungsdatensatz erfolgt. Beispiel: \$L.env . (Erforderlich)
Nummerdatensatzname <i>number.record</i>	Definiert eine Nummernklasse für das Objekt, auf die entweder von einem Aufruf der getnumb-Anwendung oder über einen Prozess-, Format Control- oder RAD-Aufruf aus zugegriffen werden kann. Kann

Feldname	Beschreibung
	zum Abrufen einer Sequenznummer als eindeutiger Schlüssel für den Datensatz verwendet werden. Beispiel: EXWorkOrder .
Kategorietabellenname <i>category.file.name</i>	Gibt den Tabellennamen an, der mit der Kategorietabelle verbunden ist, die dem Kategoriewert des Objekts zugeordnet ist. Wenn Sie einen Datensatz dieses Typs anzeigen und ein Kategoriefeld vorhanden ist, sucht das Objekt den Kategorietabellennamen sowie nach dem Datensatz mit dem entsprechenden Namen. Wenn die Suche erfolgreich verläuft, speichert das System den Kategoriedateinamen als eine Variable: \$L.category .
Phasentabellenname <i>phase.file.name</i>	Gibt den Tabellennamen an, der mit der Phasentabelle verbunden ist, die diesem Objekt zugeordnet ist, falls anwendbar. Wenn das System einen Datensatz dieses Typs anzeigt und ein Phasenfeld vorhanden ist, sucht das Objekt den Phasentabellennamen und wählt einen Datensatz mit dem entsprechenden Namen aus. Wenn die Suche erfolgreich verläuft, speichert das System den Phasentabellennamen als eine Variable: \$L.phase .
Paging-Tabellenname <i>paging.file</i>	Gibt den Name der Tabelle an, die zum Speichern von Seiten verwendet wird. Seiten werden als vollständige Kopien des aktuellen Datensatzes vor Anwendung der neuesten Aktualisierungen gespeichert. Dies geschieht jedes Mal, wenn ein Datensatz aktualisiert wird. Dadurch wird ein detailliertes Prüfprotokoll erstellt.
Basis-Format Control <i>master.fc</i>	Gibt den Namen des Basis-Format Control-Datensatzes an, sofern ein solcher für das Datensatzobjekt vorhanden ist. Basis-Format Control ermöglicht Ihnen das Definieren eines Datensatzes in Form von Format Control-Anweisungen, die auf sämtliche Phasen in diesem Bereich angewendet werden, beispielsweise auf Change Management-Anforderungsphasen. Normalerweise entspricht der Name der Basis-Format Control dem Namen der DBDict oder aller Kategorien in Incident Management, Problem Management oder Service Desk.
Joindef <i>joindef</i>	Gibt den Namen des joindef-Datensatzes an, der zum Verbinden mehrerer Tabellen verwendet wird, beispielsweise joincomputer , oder einen Ausdruck, der als ein gültiger joindef-Name ausgewertet wird, beispielsweise joindef in \$L.category .
Statusfeld <i>statusField</i>	Gibt den Namen des Felds an, das die Statusinformationen des Datensatzes enthält.
Zugewiesen an-Felder <i>assignedToFields</i>	Gibt den Namen des Felds an, das das Feld mit dem Namen des Sachbearbeiters für dieses Objekt enthält. Auf dieses Feld wird Bezug genommen, wenn von Ordnerzugang bestätigt wird, dass der Datensatz dem angemeldeten Bearbeiter zugewiesen ist.

Feldname	Beschreibung
Arbeitsgruppenfelder <i>workgroupFields</i>	Gibt den Namen des Felds an, das das Feld mit der Zuweisungsgruppe für dieses Objekt enthält. Auf dieses Feld wird Bezug genommen, wenn von Ordnerzugang bestätigt wird, dass der Datensatz einer beliebigen Arbeitsgruppe des angemeldeten Bearbeiter zugewiesen ist.
Öffnen-Zustand <i>open.state</i>	Gibt den Zustandsdefinitionsdatensatz an, der beim Öffnen eines neuen Datensatzes verwendet werden soll.
Schließen-Zustand <i>close.state</i>	Gibt den Zustandsdefinitionsdatensatz an, der beim Schließen eines vorhandenen Datensatzes verwendet werden soll.
Listenzustand <i>list.state</i>	Gibt den Zustandsdefinitionsdatensatz an, der beim Anzeigen einer Liste mit Datensätzen verwendet wird.
Standardzustand <i>default.state</i>	Gibt den Namen des Zustands an, der standardmäßig für das Objekt verwendet werden soll. Der Standardzustand wird zum Bearbeiten eines Datensatzes in diesem Objekt verwendet.
Suchen-Zustand <i>search.state</i>	Gibt den Zustandsdefinitionsdatensatz an, der für die Suche verwendet werden soll.
Durchsuchen-Zustand <i>browse.state</i>	Definiert den Zustandsdefinitionsdatensatz, der beim Sperren von Datensätzen verwendet werden soll. Dieses Feld definiert einen schreibgeschützten Zustand, bei dem ein Datensatz aktuell von einem anderen Benutzer gesperrt ist.
Manuelle Zustände <i>manual.states</i>	Gibt die Reihe der Zustände an, die mit diesem Objekt verwendet werden können, mit Ausnahme der Lebenszykluszustände Öffnen , Schließen , Listen , Anzeigen , Suchen oder Durchsuchen .

Feldbeschreibungen des Registers "Sperren"

Das Register **Sperren** bestimmt das Sperrverhalten des Objekts. Aktualisiert ein Benutzer aktiv einen Datensatz, dann lässt das System nicht zu, dass derselbe Datensatz von einem anderen Benutzer aktualisiert wird.

Die Feldbeschreibungen für das Register **Sperren** lauten folgendermaßen:

Feld	Beschreibung
Sperre verwenden <i>use.locking</i>	Aktivieren Sie dieses Kontrollkästchen, um die Sperre zu aktivieren.
Bei Anzeige sperren <i>lock.on.display</i>	Lautet der Wert true oder liegt eine Bedingung vor, die mit true ausgewertet wird, versucht Service Manager, den Datensatz zu sperren, sobald dieser angezeigt wird. Sperre verwenden muss

Feld	Beschreibung
	aktiviert sein, damit dieses Feld funktioniert.
Übergeordneten Datensatz sperren <i>lock.parent</i>	Spernt den aktuellen Datensatz und den ihm übergeordneten Datensatz. Wenn beispielsweise dieses Kontrollkästchen aktiviert ist und jemand eine Änderungsaufgabe aktualisiert, wird auch die Änderungsanforderung für diese Aufgabe gesperrt.
Feld "Übergeordnete ID" <i>parent.id</i>	Geben Sie in den aktuellen Datensatz einen Feldnamen ein, der die ID des zu sperrenden übergeordneten Datensatzes enthält.
Übergeordneter Dateiname/Objekt <i>parent.object</i>	Der Name der Tabelle, die den übergeordneten Datensatz enthält.
Überwachungsvariablen <i>watch.variables</i>	Überwachungsvariablen werden verwendet, wenn das Dokumentenmodul überprüft, ob ein Datensatz geändert wurde oder nicht. Der Wert für Überwachungsvariablen muss NULL lauten, wenn der Datensatz zum ersten Mal angezeigt wird.

Feldbeschreibungen des Registers "Revisionen"

Das Register **Revisionen** bestimmt das Revisionsverhalten des Objekts.

Die Feldbeschreibungen für das Register lauten folgendermaßen:

Feldname	Beschreibung
Revisionstabellenname <i>revision.file</i>	Gibt den Namen der Tabelle an, in der Sie die Revisionen der Datensätze dieses Objekts speichern möchten.
Höchstanzahl Revisionen <i>max.revisions</i>	Gibt die zulässige Gesamtanzahl an Revisionen für dieses Objekt an. Wenn das Feld leer bleibt, gibt es keine Obergrenze.

Feldbeschreibungen des Registers "Variablen/Globale Listen"

Das Register **Variablen/Globale Listen** beschreibt lokale und globale Variablen, die vom Objekt verwendet werden.

Hinweis: Globale Variablen werden im Speicher erstellt und gespeichert.

Die Feldbeschreibungen für das Register **Variablen/Globale Listen** lauten folgendermaßen:

Feldname	Beschreibung
Lokale Variablen <i>local.variables</i>	Geben Sie eine Liste lokaler Variablen ein, die in Prozessen verwendet werden können. Lokale Variablen werden in der

Feldname	Beschreibung
	Anwendung definiert. Lokale Variablen werden dem Objekt zugewiesen, das Sie erstellen, und stehen allen dem Objekt zugeordneten Prozessen und Zuständen zur Verfügung. Lokale Variablen stehen anderen Objekten nicht zur Verfügung.
Globale Listen <i>global.lists</i>	Sobald globale Listen erstellt sind, stehen sie allen Prozessen zur Verfügung. Globale Listen können beim Anmelden erstellt werden, wenn sie in der Systemstartliste aufgeführt sind. Globale Listen stehen bei jedem Zugriff auf das Objekt zur Verfügung.

Feldbeschreibungen des Registers "Aktivitäten"

Das Register **Aktivitäten** definiert die Aktualisierungsprotokollierung (Aktivitätsprotokollierung) für das Objekt.

Die Feldbeschreibungen für das Register **Aktivitäten** lauten folgendermaßen:

Feldname	Beschreibung
Aktivitätsprotokolltabelle <i>activitylog.file.name</i>	Der Name der Tabelle, die die Aktivitätsprotokolleinträge für das Objekt enthält.
Auswahllistenvariable <i>activity.selection.var</i>	Die auf einem Aktualisierungsformular zu verwendende Variable für die Anzeige von Aktivitätstypen, die beim Ausführen einer Aktualisierung eines Datensatzes für ein bestimmtes Objekt vom Bearbeiter ausgewählt werden können.
Übertragungs-Link <i>activity.post.link</i>	Der Name des zum Übertragen von Informationen verwendeten Links.
Aktualisierung erforderlich, wenn Aktivitätsdatensatz NICHT erzeugt wird? <i>activity.mandatory</i>	Ein Kontrollkästchen, mit dem angegeben wird, dass eine Aktivitätsaktualisierung erforderlich ist, wenn es aktiviert ist.
Aktualisierungsfeld <i>update.field.var</i>	Das Feld oder die Variable, das bzw. die die Aktivitätsaktualisierung auf dem Formular enthält. Dieses Feld wird nur angezeigt, wenn das Kontrollkästchen Aktualisierung erforderlich, wenn Aktivitätsdatensatz NICHT erzeugt wird? aktiviert ist.
Meldung anzeigen <i>activity.mandatory.msg</i>	Die Meldung, die anzeigt, dass eine Aktivitätsaktualisierung erforderlich ist. Dieses Feld wird nur angezeigt, wenn das Kontrollkästchen Aktualisierung erforderlich, wenn Aktivitätsdatensatz NICHT erzeugt wird? aktiviert ist.

Feldbeschreibungen des Registers "Alerts"

Das Register **Alerts** definiert, wo Alerts und Bedingungen zum Erstellen von Alerts festgelegt werden. Diese Alerts können von jedem Objekt mit einem eindeutigen Schlüssel verwendet werden.

Die Feldbeschreibungen für das Register **Alerts** lauten folgendermaßen:

Feldname	Beschreibung
Alert für <i>alert.location</i>	<p>Der Alert-Speicherort definiert, wo der Name der auszuführenden Alert-Definition gefunden oder gespeichert wurde. Geben Sie an, wo die Alert-Definition gespeichert wird:</p> <p>Datensatz: Speichert Alerts im Datensatz selbst.</p> <p>Kategorie: Speichert Alerts in der im Register Objektinformationen definierten Kategoriedatei.</p> <p>Phase: Speichert Alerts in dem im Register Objektinformationen definierten Phasendatensatz.</p> <p>Objekt: Speichert Alerts im Objektdatensatz. Wenn diese Option ausgewählt ist, wird eine Alerts-Array-Tabelle angezeigt, um Ihnen die Eingabe der zu verwendenden Alerts zu ermöglichen.</p> <p>Hinweis: Die Tabelle des Objekts muss über einen eindeutigen Schlüssel für die verbundenen Alerts verfügen. Eine Tabelle, die statt eines eindeutigen Schlüssels einen Keine Nullwerte-Schlüssel enthält, kann keine Alerts verwenden.</p>
Alert-Bedingung <i>alert.condition</i>	Geben Sie eine Bedingung an, um zu bestimmen, ob der Alert verarbeitet werden soll. Beispiel: open in \$L.file~=false .
Alert-Feldname <i>alert.field.name</i>	Geben Sie den Feldnamen an, der den tatsächlichen Alert-Namen enthält, wie vom Alert-Standort definiert.
Alert-Statusfeld <i>alert.status.field</i>	Gibt das Feld im aktuellen Datensatz an, in dem der Alert-Status nach dem Verarbeiten des Alerts angezeigt wird.
Alert-Aktualisierungsprozess <i>alert.update.process</i>	Gibt den Namen des Prozessdatensatzes für zusätzliche Funktionen an, die das System nach dem Ausführen des Alerts ausführt.
Alerts protokollieren? <i>log.alerts</i>	Wenn dieses Kontrollkästchen aktiviert wird, werden die Alerts nach der Verarbeitung in die Datei Alertlog verschoben, um einen Alert-Verlauf zu generieren.
Alerts übergeordneter Elemente verarbeiten? <i>alerts.against.parent</i>	Wenn Sie im Register Sperren/Revisionen das Kontrollkästchen Übergeordneten Datensatz sperren aktiviert haben und Sie dieses Kontrollkästchen aktivieren, wird dieser Alert auch gegen den übergeordneten Datensatz registriert.

Feldname	Beschreibung
Alerts neu berechnen, wenn <i>alert.recalc</i>	Geben Sie Bedingungen an, die festlegen, ob Bedingungen für vorhandene Alerts neu berechnet werden sollen.
Alerts zurücksetzen, wenn <i>alert.reset</i>	Bestimmt, wann vorhandene Alerts gelöscht und sämtliche Bedingungen neu berechnet werden sollen.

Feldbeschreibungen des Registers "Genehmigungen"

Das Register **Genehmigungen** legt Genehmigungsoptionen und zugehörige Benachrichtigungen für das Objekt fest. Genehmigungen werden in der Datei **ApprovalDef** definiert.

Die Feldbeschreibungen für das Register **Genehmigungen** lauten folgendermaßen:

Feldname	Beschreibung
Genehmigungsbedingung <i>approval.condition</i>	Wenn die Genehmigungsbedingungen mit true evaluiert wird, werden für die Datensätze des Objekts Genehmigungen verwendet.
Genehmigung für <i>approval.location</i>	Gibt an, wo die Genehmigungsinformationen gespeichert werden: Datensatz, Phase, Objekt oder Kategorie .
Genehmigungsfeldname <i>approval.field.name</i>	Der Feldname, der die tatsächliche Genehmigungsbezeichnung in der Tabelle enthält, die unter Genehmigung für definiert ist.
Feld "Genehmigungsstatus" <i>approval.status.field</i>	Das Feld ist der aktuelle Datensatz, in dem der Genehmigungsstatus gespeichert werden soll.
Genehmigungsgruppen <i>approval.groups</i>	Speichert eine Variable, die die Gruppen enthält, zu denen der aktuelle Benutzer gehören muss, damit Genehmigungen für dieses Objekt ausgegeben werden können.
Genehmigungstyp <i>appr.cond.type</i>	Es gibt vier vordefinierte Genehmigungstypen: Alle Genehmigungen: Alle unter der Genehmigungsdefinition definierten Gruppen/Bearbeiter müssen eine Genehmigung ausgeben, bevor das System den Status des Datensatzes auf Genehmigt festlegt. Ist dies nicht der Fall, wird der Status des Datensatzes in Anstehend geändert. Eine Genehmigung: Die Genehmigung des Datensatzes erfolgt bei Genehmigung durch ein beliebiges Mitglied der genehmigenden Gruppe bzw. durch den genehmigenden Bearbeiter. Mehrheit: Der Datensatz gilt als genehmigt, wenn die Mehrheit der Mitglieder der Genehmigungsgruppe die Genehmigung erteilt.

Feldname	Beschreibung
	Alle Genehmigungen - sofortige Ablehnung: Alle Gruppen/Bearbeiter müssen den Datensatz genehmigen. Bei Ablehnung durch einen Genehmiger wird der Status sofort in Ablehnen geändert. Alle anderen Genehmiger müssen keine weiteren Schritte ausführen.
Genehmigungsbenachrichtigung <i>single.notify.approval</i>	Wählen Sie die Benachrichtigung aus, die ausgeführt werden soll, wenn die Anforderung genehmigt wird.
Ablehnungsbenachrichtigung <i>single.notify.denial</i>	Wählen Sie die Benachrichtigung aus, die angezeigt werden soll, wenn ein Genehmiger die Anforderung verweigert.
Rücknahmebenachrichtigung <i>single.notify.retraction</i>	Wählen Sie die Benachrichtigung aus, die ausgeführt werden soll, wenn eine vorherige Aktion zurückgenommen wird.
Endgültige Genehmigungsbenachrichtigung <i>final.notify.approval</i>	Wählen Sie die Benachrichtigung aus, die gesendet werden soll, sobald die endgültige Genehmigung erteilt wird.
Endgültige Ablehnungsbenachrichtigung <i>final.notify.denial</i>	Wählen Sie die Benachrichtigung aus, die gesendet werden soll, wenn die Anforderung abgelehnt wird.
Genehmigungs-FC <i>appr.fc</i>	Gibt den Namen des Format Control-Datensatzes an, der bei Genehmigung ausgeführt werden soll.
Genehmigungsprozess <i>approval.process</i>	Wählen Sie den Prozess aus, der ausgeführt werden soll, wenn der Datensatz genehmigt wird.
Ablehnungsprozess <i>denial.process</i>	Wählen Sie den Prozess aus, der ausgeführt werden soll, wenn der Datensatz abgelehnt wird.

Feldname	Beschreibung
Beim Öffnen vorab genehmigen <i>preapprove.cond</i>	Bestimmt, ob der Datensatz automatisch genehmigt werden soll. Wenn die Bedingung true ist und der Benutzer einer der Gruppen zur anstehenden Genehmigung angehört, wird die Genehmigung automatisch verarbeitet. Wenn der Benutzer nicht zu einer der Gruppen zur anstehenden Genehmigung gehört, wird die Genehmigung nicht automatisch vorgenommen, sondern es ist der reguläre Genehmigungsprozess anzuwenden. Es gilt standardmäßig true .
Genehmigungen protokollieren? <i>log.approvals</i>	Aktivieren Sie dieses Kontrollkästchen, um den Verlauf von Genehmigungen in der ApprovalLog-Tabelle zu protokollieren.
Genehmigungskommentare erforderlich? <i>approval.comments</i>	Wenn das Kontrollkästchen aktiviert ist, sind vom Genehmiger Genehmigungskommentare erforderlich.
Sammelgenehmigungen? <i>aggregate.approvals</i>	Wenn das Kontrollkästchen aktiviert ist, werden Genehmigungen zusammengefasst.
Genehmigungen neu berechnen, wenn <i>approval.recalc</i>	Gibt die Bedingungen an, die bestimmen, ob die Bedingungen für die vorhandenen Genehmigungen neu berechnet werden sollen.
Genehmigungen zurücksetzen, wenn <i>approval.reset</i>	Bestimmt, wann vorhandene Genehmigungen gelöscht und sämtliche Bedingungen neu berechnet werden sollen.

Feldbeschreibungen des Registers "Warteschlangen verwalten"

Das Register **Warteschlangen verwalten** steuert, wie Warteschlangen und Ansichten angezeigt werden. Darüber hinaus steuert es die Thread-Verarbeitung. Dieselben Felder stehen in Datenrichtlinien für Dateien zur Verfügung, die über keinen zugeordneten Objektdatensatz verfügen. Oder sie sind virtuell in den datadict-Datensatz des Objektdatensatzes eingebunden, sofern vorhanden.

Die Feldbeschreibungen für das Register **Warteschlangen verwalten** lauten folgendermaßen:

Feldname	Beschreibung
Manage-Bedingung <i>scm.condition</i>	Gibt eine Bedingung an, die nur bestimmten Benutzern das Anzeigen der Warteschlangen ermöglicht, die Datensätze dieses Objektes anzeigen. Durchsuchen Sie beispielsweise \$G.pm.environment .
Manage-Anzeigeformat <i>scm.manage.screen</i>	Wählen Sie das Format aus, das zum Anzeigen der Ansicht verwendet werden soll. Der vordefinierte Service Manager verfügt

Feldname	Beschreibung
	über ein Standardanzeigeformat: sc.manage.generic – dieses wird verwendet, wenn kein anderes Formular ausgewählt wird. HP empfiehlt, das sc.manage.generic-Format nicht zu ändern.
Manage-Standardansicht <i>scm.inbox</i>	Wählen Sie die Standardansicht für diese Warteschlange aus. Wenn Sie eine Benutzeransicht für einen bestimmten Benutzer angeben, können Sie eine Liste von Ansichten für die HP Service Manager-Option Warteschlangen verwalten einrichten. Wenn für einen Benutzer keine bestimmte Ansicht definiert ist, wird die Standardansicht verwendet.
Manage-Standardabfrage <i>scm.query</i>	Gibt eine Standardabfrage an, die ausgeführt wird, wenn keine Standardansicht ausgewählt ist.
Standardabfrage-Beschreibung <i>scm.query.name</i>	Gibt einen Namen für das oben genannte Feld an. Sie können diesem Feld eine Meldung zuordnen. Beispiel: scmsg(491, "us") .
Thread-Ansicht -> Suchen? <i>scm.thread.list.edit</i>	Gibt true oder einen Ausdruck an, der mit true ausgewertet wird, um beim Bewerten einer Suche einen neuen Thread zu öffnen.
Suchformat (falls erforderlich) <i>scm.search.format</i>	Wählen Sie ein Standardsuchformat aus.
Thread-Suche -> Liste? <i>scm.thread.search.list</i>	Gibt für einen neuen Thread true oder einen Ausdruck an, der mit true ausgewertet wird, wenn ein Benutzer eine Liste mit anzuzeigenden Datensätzen findet.
Thread-Liste -> Bearbeiten? <i>scm.thread.list.edit</i>	Gibt für einen neuen Thread true oder einen Ausdruck an, der mit true ausgewertet wird, wenn ein Benutzer aus einer Liste von Datensätzen einen Datensatz zur Ansicht auswählt.
Thread-Ansicht -> Bearbeiten? <i>scm.thread.inbox.edit</i>	Gibt true oder einen Ausdruck an, der mit true ausgewertet wird, um einen neuen Thread zu öffnen, wenn der Benutzer einen vorhandenen Datensatz aus der Warteschlange zur Ansicht auswählt.
Bedingung hinzufügen zulassen <i>scm.add.condition</i>	Gibt eine Bedingung an, die die Möglichkeit eines Bearbeiters zum Hinzufügen eines Datensatzes bewertet.
Anwendung hinzufügen/öffnen <i>scm.add.appl</i>	Gibt den Namen der Anwendung an, die aufgerufen werden soll, wenn ein Datensatz hinzugefügt oder geöffnet wird.

Feldname	Beschreibung
Parameter - Namen <i>scm.add.names</i>	Gibt die Parameternamen an, die an die im Feld Anw. hinzufügen/öffnen angegebene Anwendung übertragen werden sollen.
Parameter - Werte <i>scm.add.values</i>	Gibt die Parameterwerte an, die an die im Feld Anw. hinzufügen/öffnen angegebene Anwendung übertragen werden sollen.

Feldbeschreibungen des Registers "Ansichten/Vorlagen"

Das Register **Ansichten/Vorlagen** definiert, ob ein Benutzer globale und persönliche Ansichten sowie Vorlagen und Vorlagenunterstützungen erstellen kann.

Die Feldbeschreibungen für das Register lauten folgendermaßen:

Feldname	Beschreibung
Darf persönliche Ansichten erstellen <i>personal.inbox</i>	Gibt eine Bedingung an, die mit true oder false ausgewertet wird. Bei true kann der Benutzer persönliche Ansichten erstellen.
Darf Systemansichten erstellen <i>global.inbox</i>	Gibt eine Bedingungen an, die mit true oder false ausgewertet wird, um zu bestimmen, ob der Benutzer globale Ansichten erstellen kann.
Standardvorlage <i>default.template</i>	Gibt den Namen der Standardvorlage an, die für Datensätze in dieser Tabelle verwendet werden soll.
Unterstützt Vorlagen? <i>supportTemplates</i>	Aktivieren Sie dieses Kontrollkästchen, um die Unterstützung von Vorlagen für das Objekt zu ermöglichen.

Feldbeschreibungen des Registers "Benachrichtigungen"

Das Register **Benachrichtigungen** legt Benachrichtigungen fest, die für Hinzufügungs-, Aktualisierungs- oder Lösungsaktivitäten für das Objekt automatisch gesendet werden.

Die Feldbeschreibungen für das Register **Benachrichtigungen** lauten folgendermaßen:

Feldname	Beschreibung
Hinzufügen <i>notification.add</i>	Wählen Sie die Benachrichtigung aus, die automatisch gesendet wird, sobald der Tabelle ein Datensatz hinzugefügt wird.
Aktualisieren <i>notification.update</i>	Wählen Sie die Benachrichtigung aus, die automatisch gesendet wird, sobald ein Datensatz in der Tabelle aktualisiert wird.

Feldname	Beschreibung
Löschen <i>notification.delete</i>	Wählen Sie die Benachrichtigung aus, die automatisch gesendet wird, sobald ein Datensatz aus der Tabelle gelöscht wird.

Feldbeschreibungen des Registers "Suchkonfiguration"

Das Register **Suchkonfiguration** steuert die verfügbaren Optionen im Register **Weitere Optionen** der Suchbildschirme.

Die Feldbeschreibungen für das Register **Suchkonfiguration** lauten folgendermaßen:

Feldname	Beschreibung
Tabellenname <i>tablename</i>	Der Name der Tabelle, die abgefragt wird.
Suchformat <i>searchFormat</i>	Der Name des Unterformats, das für das Register Weitere Optionen verwendet wird.
Initialisierungsprozess <i>init.process</i>	Der Name eines Prozesses, der vor der Anzeige des Suchformulars ausgeführt wird.
Erweiterte Suche zulassen <i>allowAdvAccess</i>	Definiert die Bedingungen, die bestimmen, ob eine erweiterte Suche zugelassen wird oder nicht.

Register "Definierte Abfragen"

Das Register **Definierte Abfragen** definiert die Abfrageanweisungen und Labels, die im Register **Weitere Optionen** des Suchbildschirms (z. B. **Weitere Optionen** im Formular **Incidents suchen**) verwendet werden sollen. Die Felder werden in der Tabelle **SearchConfig** definiert.

Die Feldbeschreibungen für das Register **Definierte Abfragen** lauten folgendermaßen:

Feldname	Beschreibung
ID <i>id</i>	Gibt die eindeutige ID für die Abfrage an und darf keine Sonderzeichen (einschließlich Leerzeichen) enthalten.
Abfrage <i>query</i>	Ein Abfrageausdruck, der die Systemsprachesyntax verwendet. Beispiel: <code>assignee.name=operator()</code> .

Feldname	Beschreibung
Beschreibung <i>description</i>	Gibt das Label für das Kontrollkästchen im Register Weitere Optionen an.

Register "Bereiche"

Das Register "Bereiche" ermöglicht Ihnen das einfache Einrichten einer Suche nach Start- und Enddatumsbereichen. Klicken Sie hierzu auf den Link **Konfiguration ändern**, um eine Variable für den Start und das Ende zu definieren und als Eingabe in das Formular zu verwenden. Geben Sie diese Variable in die Spalte **Variable 1** ein und verwenden Sie die Spalten **Feld** und **Operator 1**, um die Abfrage zu definieren, die ausgeführt wird. Änderungen werden über den Link **Konfiguration ändern** ebenso vorgenommen wie im Register **Definierte Abfragen**.

Die Feldbeschreibungen für das Register **Bereiche** lauten folgendermaßen:

Feldname	Beschreibung
Feld <i>fieldName</i>	Gibt das Feld in der Tabelle an, das in der Abfrage verwendet werden soll.
Operator 1 <i>operator1</i>	Der Vergleichsoperator in der Abfrage. (Beispiel: >=)
Variable 1 <i>variable1</i>	Gibt die Variable an, die als Eingabe in das Formular verwendet wird.
Sondertyp <i>specialType</i>	Derzeit nicht verwendet.

Kapitel 3

Zustände

Zustände werden von Objekten aufgerufen und von Prozessen definiert. Der Zustandsdatensatz enthält Informationen zur Beschaffenheit eines Datensatzes zu einem bestimmten Zeitpunkt.

Eine Zustandsdatensatzdefinition kann folgende Informationen enthalten:

- Den Zustandsnamen
- Den zum Anzeigen des Datensatzes oder der Datensätze zu verwendenden Anzeigebildschirm
- Den Initialisierungsprozess, der beim ersten Anzeigen einer Liste verwendet wird
- Das Format, das zum Anzeigen der Daten verwendet werden soll
- Die Eingabebedingung, die angibt, ob ein Benutzer den Datensatz bearbeiten kann oder nicht
- Den Prozess, der ausgeführt werden soll, sobald ein Benutzer eine bestimmte Anzeigoption auslöst (Anzeigeaktion)

Welcher Zustandsdatensatz verwendet wird, hängt davon ab, wie viele Datensätze der Benutzer anzeigt. Für die Suche nach Datensätzen, das Anzeigen einer Liste, das Anzeigen eines einzelnen Datensatzes und das Durchsuchen eines Datensatzes gibt es vordefinierte Zustandsdatensätze.

Suche

Wenn in der aktuellen Dateivariablen keine Datensätze vorhanden sind, wird davon ausgegangen, dass sich der Benutzer im Suchmodus befindet. Es wird der Zustand verwendet, der im Objektdatensatz der Tabelle als Suchzustand definiert wurde. Der Standardsuchzustand lautet **db.search**.

Datensatzlisten

Bei der Anzeige einer Datensatzliste wird der im Objektdatensatz der Tabelle definierte Listenzustand verwendet. Der Standardlistenzustand lautet **db.list**.

Hinweis: In einigen älteren Service Manager-Versionen werden Datensatzlisten auch als QBE-Listen bezeichnet.

Anzeigen eines einzelnen Datensatzes

Bei der Anzeige eines einzelnen Datensatzes wird der Datensatz zuerst daraufhin geprüft, ob sein DBDict ein Zustandsfeld enthält. Wenn das Feld vorhanden und ausgefüllt ist, wird der Inhalt dieses Feldes als aktueller Zustand des Datensatzes verwendet. Wenn das Feld nicht vorhanden oder leer ist, wird der im Objektdatensatz der Tabelle verwendete Standardzustand verwendet. Der Standardwert für den Standardzustand lautet **db.view**.

Durchsuchen eines Datensatzes

Beim Durchsuchen eines Datensatzes wird der im Objektdatensatz der Tabelle definierte Durchsuchen-Zustand verwendet. Beachten Sie, dass nur Dateien, die Sperren verwenden, einen Durchsuchen-Zustand erfordern. Das Anzeigen eines Datensatzes ohne Aktualisierungsrechte

sieht ähnlich aus wie der Durchsuchenbildschirm, jedoch verwendet dieser den Standardzustand, der Durchsuchenbildschirm jedoch den Durchsuchen-Zustand. Es gibt keinen standardmäßigen Durchsuchen-Zustand.

Integrationsstipps für die Display-Anwendung

Wenn einem Anzeigebildschirm keine Anzeigeeignisse zugeordnet sind, verwendet das System automatisch das Anzeigeeignis **se.default**. Im Fall von **OnFormModified** führt dies **se.lock.object** aus. Wenn für einen Anzeigedatensatz das Kontrollkästchen **Ändert Datensatz** aktiviert ist, wird der Datensatz gesperrt, sobald auf die Schaltfläche geklickt wird.

Erstellen und Aktualisieren von Zuständen

So erstellen Sie einen neuen Zustand:

1. Greifen Sie auf das Dokumentenmodul zu. Siehe [Zugriff auf das Dokumentenmodul auf Seite 10](#).
2. Füllen Sie im Formular **Zustandsdefinition** die Felder aus, die erforderlich sind, um einen Zustand zu erstellen, der die von Ihnen gewünschten Funktionen ausführt. Informationen zu den Feldbeschreibungen finden Sie unter [Formular "Zustandsdefinition" und Felder auf Seite 28](#).

So ändern Sie einen vorhandenen Zustand:

1. Greifen Sie auf das Dokumentenmodul zu. Siehe [Zugriff auf das Dokumentenmodul auf Seite 10](#).
2. Geben Sie in das Feld **Zustand** den Namen des Zustands ein, den Sie ändern möchten, oder klicken Sie auf **Suchen**, um nach dem Zustand zu suchen.

Formular "Zustandsdefinition" und Felder

Der Zustandsdefinitionsdatensatz definiert das Verhalten und die Anzeige von Spezifikationen für einen Datensatz zu einem bestimmten Zeitpunkt.

Im Zustandsdatensatz können Nicht-Basismethoden definiert oder das Verhalten der Basismethoden geändert werden. Basismethoden werden in den Abschnitten zu den Basisfunktionen in **se.search.engine**, **se.view.engine** und **se.list.engine** beschrieben. Zu diesen zählen Funktionen wie Speichern, Finden, Füllen, OK, Abbrechen und Suchen.

Die Feldbeschreibungen für das Formular **Zustandsdefinition** lauten folgendermaßen:

Feldname	Beschreibung
Zustand <i>state</i>	Gibt den Namen des Zustands an. (Erforderlich)
Anzeigebildschirm <i>display.screen</i>	Der dem Zustand zugeordnete Anzeigebildschirm.

Feldname	Beschreibung
Initialisierungsprozess <i>init.process</i>	Der Name des Prozesses, der vor Beginn des Zustands ausgeführt werden soll.
Format <i>format.name</i>	Gibt das Format an, in dem der Zustandsdatensatz geöffnet wird. Dieses Formular wird in einem Datensatz gespeichert und kann variabel oder hartcodiert sein. Der Name des Formats kann hartcodiert sein, sich in einer Variablen befinden oder kann mit Hilfe eines Systemsprachausdrucks aus einem Datensatz abgerufen werden.
Eingabebedingung <i>input.condition</i>	Die Eingabebedingung wird für Anzeigezustände ausgewertet und bestimmt, ob der Datensatz schreibgeschützt ist oder nicht. Geben Sie false an für schreibgeschützt und true für bearbeitbar. Sie können auch einen Ausdruck eingeben, der mit true oder false ausgewertet wird. Geben Sie andernfalls true ein oder geben Sie einen Ausdruck ein, der mit true oder false ausgewertet wird.
Nicht-Basismethoden	
Anzeigeaktion <i>process.label</i>	Gibt den Aktionsparameter an, der von einer Anzeigeaktion nach Eingabe durch einen Benutzer ausgegeben wird. Die Anzeigeaktion stammt aus dem Aktionsfeld in der Anzeigeeoption einer Schaltfläche oder eines Menübefehls, auf die bzw. den ein Benutzer klickt.
Prozessname <i>valid.process</i>	Gibt den Namen des Prozesses an, der als Ergebnis der Aktion ausgeführt wird.
Bedingung <i>process.condition</i>	Gibt einen Ausdruck an, der für den Prozess mit true ausgewertet wird, der im aufzurufenden Prozessnamen angegeben ist.
Zuerst speichern <i>run.save.before</i>	Legen Sie fest, dass der Speicherprozess ausgeführt werden soll, bevor Sie diesen Prozess ausführen. Geben Sie true ein, wenn zuerst gespeichert werden soll, andernfalls false . Die Standardeinstellung ist false .

Kapitel 4

Prozesse

Prozesse sind die kleinsten diskreten Arbeitseinheiten, die dem Dokumentenmodul zur Verfügung stehen. Die Daten werden auf Prozessebene bearbeitet. Benutzer können ihre eigenen Prozesse erstellen oder einen der über 700 Prozesse im Lieferumfang von Service Manager verwenden. Die Prozessdefinitionen bestehen aus Anfangsausdrücken, RAD und Endausdrücken, wie diese in die Register **Anfangsausdrücke**, **Anfangs-Javascript**, **RAD**, **Endausdrücke**, **End-Javascript** und **Nächster Prozess** eingegeben wurden, sowie aus einem optionalen Aufruf des nächsten Prozesses. Ausdrücke werden unter Verwendung der standardmäßigen RAD-Ausdrücke und/oder von Javascript geschrieben.

Erstellen und Ändern eines Prozesses

So erstellen Sie einen Prozess:

1. Greifen Sie auf das Dokumentenmodul zu. Siehe [Zugriff auf das Dokumentenmodul auf Seite 10](#).
2. Verwenden Sie die Register unter **Prozesse**, um die Felder auszufüllen, die erforderlich sind, um einen Prozess zu erstellen, der die von Ihnen gewünschten Funktionen ausführt. Weitere Informationen finden Sie unter [Formular "Prozessdefinition" und Felder auf Seite 31](#).

So ändern Sie einen Prozess:

1. Greifen Sie auf das Dokumentenmodul zu. Siehe [Zugriff auf das Dokumentenmodul auf Seite 10](#).
2. Geben Sie in das Feld **Prozessname** den Namen des Prozesses ein, den Sie ändern möchten, oder klicken Sie auf **Suchen**, um nach dem Prozess zu suchen.

Formular "Prozessdefinition" und Felder

Auf dem Formular **Prozessdefinition** definieren Sie neue Prozesse oder bearbeiten vorhandene Prozesse. Prozesse führen Code oder Ausdrücke aus, um die vom Benutzer ausgewählten Aktionen auszuführen.

Die Feldbeschreibungen für das Formular **Prozessdefinition** lauten folgendermaßen:

Feldname	Beschreibung
Prozessname <i>process</i>	Gibt den Namen des Prozesses an. (Erforderlich)
Cursorposition speichern? <i>save.cursor.position</i>	Aktivieren Sie dieses Kontrollkästchen, wenn Sie nach der Aktion zu derselben Cursorposition zurückkehren möchten

Feldname	Beschreibung
	(Beispiel: Füllen).
Nach Abschluss Standardprozess ausführen? <i>run.standard</i>	Wenn Sie diese Option auswählen, führt das System nach Abschluss der aktuellen Aktion oder des aktuellen Prozesses einen Standardprozess aus. Ein solcher Standardprozess kann beispielsweise das Speichern sein. Aktivieren Sie dieses Kontrollkästchen, wenn Sie einen Speicherprozess erstellt haben und nach Abschluss des von Ihnen definierten Speicherprozesses den im Umfang des Dokumentenmoduls enthaltenen Speicherprozess ausführen möchten. Ein Standardprozess wird innerhalb dieses Dokuments im Abschnitt zu den Basisfunktionen definiert.
In Fenster ausführen? <i>run.in.window</i>	Wenn dieses Kontrollkästchen aktiviert ist, wird der Prozess in einem eigenen Fenster ausgeführt.
Fenstertitel <i>window.name</i>	Geben Sie einen Titel für das Fenster an, wenn das Kontrollkästchen In Fenster ausführen? aktiviert ist. Für lokalisierte Fenstertitel kann ein scmsg-Ausdruck verwendet werden, beispielsweise scmsg(1980, "us") .

Zusätzlich zu den oben beschriebenen Feldern stehen auf diesem Formular auch Register zur Verfügung, die eine weitere Definition des Prozesses ermöglichen. Es handelt sich dabei um die folgenden Register:

- [Register "Anfangsausdrücke" auf Seite 32](#)
- [Register "Anfangs-Javascript" auf Seite 33](#)
- [Register "RAD" auf Seite 33](#)
- [Register "Endausdrücke" auf Seite 33](#)
- [Register "End-Javascript" auf Seite 34](#)
- [Register "Nächster Prozess" auf Seite 34](#)

Register "Anfangsausdrücke"

Das Register **Anfangsausdrücke** definiert die Anfangsausdrücke, die vor dem anfänglichen Javascript und vor den im Register **RAD** definierten RAD-Aufrufen ausgeführt werden. Die Anfangsausdrücke werden unter Verwendung standardmäßiger Service Manager-Ausdrücke geschrieben.

Register "Anfangs-Javascript"

Das Register **Anfangs-Javascript** definiert die Anfangs-Javascript-Ausdrücke, die vor der im Register **RAD** definierten RAD-Anwendung ausgeführt werden.

Register "RAD"

Das Register **RAD** definiert die Vor-RAD-Ausdrücke, RAD-Aufrufe und Nach-RAD-Ausdrücke, die als Teil des Prozesses ausgeführt werden.

Die Feldbeschreibungen für das Register **RAD** lauten folgendermaßen:

Hinweis: Diese Felder werden für jede RAD-Anwendung wiederholt, die in diesem Register definiert ist.

Feldname	Beschreibung
Vor RAD-Aufruf ausgewertete Ausdrücke <i>pre.rad.expressions</i>	Geben Sie den Ausdruck an, der vor der im Feld RAD-Anwendung definierten RAD-Anwendung ausgeführt werden soll. Alle der RAD-Anwendung übermittelten Parameterwerte müssen in Form von Variablen oder Ausdrücken formuliert sein. Die Werte der Variablen müssen in den Vor-RAD-Ausdrücken zugewiesen werden. Beispiel: \$L.value.name="Test" .
RAD-Anwendung <i>application</i>	Gibt den Namen der auszuführenden RAD-Anwendung an.
Bedingung <i>rad.condition</i>	Gibt an, unter welchen Bedingungen die RAD-Anwendung ausgeführt (Bedingung wird mit true ausgewertet) oder übersprungen werden soll (Bedingung wird mit false ausgewertet).
Parameternamen <i>names</i>	Gibt die Parameternamen an, die an die RAD-Anwendung übermittelt werden.
Parameterwerte <i>values</i>	Gibt die Parameterwerte an, die an die RAD-Anwendung übermittelt werden. Diese Werte müssen die Form von Variablen oder Ausdrücken aufweisen. Zeichenfolgewerte können übermittelt werden, wenn sie in doppelte Anführungszeichen eingeschlossen werden ("Assistentenname").
Nach-RAD-Ausdrücke <i>post.rad.expressions</i>	Gibt einen beliebigen RAD-Ausdruck an, der nach Beendigung der RAD-Anwendung ausgeführt wird.

Register "Endausdrücke"

Das Register **Endausdrücke** definiert die Endausdrücke, die nach Abschluss der Verarbeitung der Daten des Registers **RAD** ausgeführt werden. Die Endausdrücke werden unter Verwendung standardmäßiger Service Manager-Ausdrücke geschrieben.

Register "End-Javascript"

Das in diesem Register definierte Javascript wird nach den Endausdrücken und den RAD-Anwendungen im Register **RAD** ausgeführt.

Register "Nächster Prozess"

Dieses Register gibt die nächsten Prozesse an, die ausgeführt werden sollen, sobald der aktuelle Prozess beendet wird.

Die Feldbeschreibungen für das Register **Nächster Prozess** lauten folgendermaßen:

Feldname	Beschreibung
Nächster Prozess <i>next.process</i>	Gibt den Namen des nächsten auszuführenden Prozesses an.
Bedingung <i>process.condition</i>	Gibt eine dem im Feld Nächster Prozess genannten Prozess zugeordnete Bedingung an, die mit true oder false ausgewertet wird. Beispielsweise true für den Prozess sm.close .

Kapitel 5

Dokumentenmodulressourcen

Das Dokumentenmodul weist lokale Variablen und Basisfunktionen auf, die von jedem beliebigen Objekt verwendet werden können. Bei den Basisfunktionen handelt es sich um vordefinierte Standardaktionen, die die Benutzer ausführen können. Wenn beispielsweise ein Benutzer nach der Aktualisierung eines Datensatzes auf die Schaltfläche **Speichern** klickt, verarbeitet das Dokumentenmodul diese Anforderung und schreibt die Änderungen in die Datenbank. Zum Ausführen eines anderen Prozesses kann eine Basisfunktion im Zustandsdatensatz überschrieben werden.

DEFAULT-Objekt

Alle Dateien, auf die Database Manager zugreift, verwenden automatisch für das entsprechende Objekt definierte Regeln und Prozesse. Wenn für die Datei kein Objekt angegeben wurde, wird das DEFAULT-Objekt verwendet. Das DEFAULT-Objekt übernimmt die Funktionalität der früheren Database Manager-Versionen.

Ein Systemadministrator kann auch über das DEFAULT-Objekt auf eine Datei mit einem entsprechenden Objekt zugreifen. Hierzu greift er über Database Manager auf die Datei zu, wobei er das Kontrollkästchen **Verwaltungsmodus** aktiviert hat. Dieses Kontrollkästchen steht ausschließlich Systemadministratoren zur Verfügung. Ein Benutzer mit dem Berechtigungswort **AlwaysAdmin** verwendet immer das DEFAULT-Objekt, wenn er auf Informationen in Service Manager zugreift. Das Zuweisen des Berechtigungsworts zu einem Benutzer wird nicht empfohlen.

Das Dokumentenmodul verwendet Umgebungsprofile. In früheren Versionen des Systems wurden Systemadministratoren unabhängig von den Format Control-Einstellungen alle Rechte gewährt (zum Hinzufügen, Aktualisieren, Löschen). Das Dokumentenmodul stellt sicher, dass die gewährten Rechte mit denen übereinstimmen, die im Format Control-Datensatz definiert wurden. Administratoren, die die alte Methode zur Gewährung von Rechten vorziehen, können die Einstellungen zur Profilanwendung im DEFAULT-Objektdatensatz einfach von **d.environment** in **d.environment.sysadmin** ändern.

RAD-Anwendungen

Eine datenbankgesteuerte Anwendung ermöglicht es einem Benutzer, mit einer der drei folgenden Basisdatensatzgruppen zu arbeiten:

- **Ohne Datensätze** – Wenn Informationen gesucht werden.
- **Viele Datensätze** – Wenn eine Liste mit Informationen durchsucht wird.
- **Ein Datensatz** – Wenn ein einzelner Datensatz geändert wird.

Aus Benutzerperspektive treffen beim Arbeiten mit Daten folgende Aussagen zu:

- Sie zeigen entweder ein leeres Formular an, in das Sie Suchinformationen eingeben können (oder Sie erstellen einen neuen Datensatz).
- Oder Sie sind dabei, einen einzelnen Datensatz zu aktualisieren oder anzuzeigen.

- Oder Sie haben eine Suche ausgeführt und es wird Ihnen eine Liste mit einer Gruppe verschiedener Datensätze angezeigt.

Das Dokumentenmodul verwendet hierzu drei wichtige RAD-Anwendungen. Die verwendete RAD-Routine wird durch die Anzahl an Datensätzen bestimmt, die zu einem beliebigen Zeitpunkt angezeigt werden.

se.search.engine

Das Suchmodul wird verwendet, wenn keine Datensätze angezeigt werden. Der Hauptzweck dieser Routine besteht im Formulieren einer Abfrage und im Auswählen von Datensätzen aus der richtigen Tabelle. Diese Routine kann auch für den Anfangseintrag von Informationen in einen leeren Datensatz zum Hinzufügen eines neuen Datensatzes zu einer Datenbank verwendet werden. Der Standardzustand lautet **db.search**.

se.list.engine

Das Listenmodul zeigt mehrere Datensätze an. Mit dem Listenmodul kann ein Benutzer einen bestimmten Datensatz aus der Liste auswählen oder Aktionen für die gesamte Liste der Datensätze ausführen. Der Standardzustand lautet **db.list**.

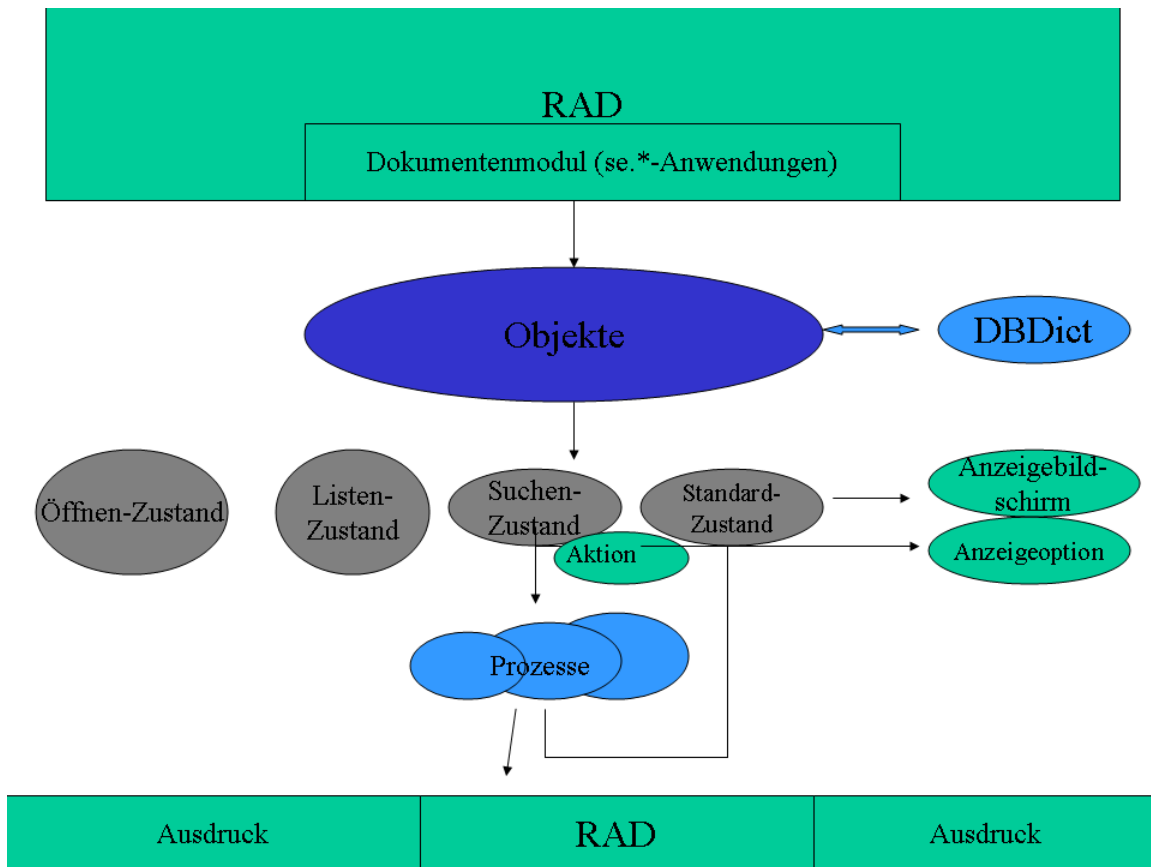
se.view.engine

Das Ansichtsmodul zeigt einen einzelnen Datensatz an. Diese Anwendung wird verwendet, um Aktionen (beispielsweise Aktualisierungs- oder Löschaktionen) für einen bestimmten Datensatz auszuführen. Der Standardzustand lautet **db.view**.

Hinweis: Wenn Sie die Datensatzlistenfunktion von Service Manager verwenden, wird das Ansichtsmodul sowohl für die Liste als auch für Informationen zu einzelnen Datensätzen verwendet.

RAD-Anwendungsfluss

Die folgende Abbildung bietet eine Übersicht der Interaktion der anderen Anpassungswerkzeuge mit dem Dokumentenmodul.



Hinweis: Alle mit dem Dokumentenmodul verbundenen RAD-Anwendungen beginnen mit "se".
Beispiel: **se.view.engine**. Das Dokumentenmodul behandelt jeden Datensatz als ein Dokument, das dem Benutzer zur Verfügung gestellt wird.

Wenn eine Datei mit dem Dokumentenmodul angezeigt wird, verwendet sie immer eine der drei Anwendungen, die weiter oben in diesem Dokument angegeben wurden. Der Anzeigebildschirm, die Anzeigeoptionen und das Format werden vom aktuellen Zustand des Datensatzes bestimmt. Wenn eine Anzeigeoption ausgelöst wird, nachdem die Standardanzeigefunktion ausgeführt wurde, überprüft das Modul die "Aktion" dieser Option anhand der verfügbaren Prozesse, die im aktuellen Zustandsdatensatz definiert sind. Wenn ein Prozess definiert wurde, der eine mit **true** ausgewertete Bedingungen aufweist, führt das Modul diesen Prozess anschließend für den aktuellen Datensatz (oder die aktuelle Datensatzgruppe) aus. Wenn die Aktion im Zustandsdatensatz nicht definiert ist, prüft das Modul, ob diese Aktion in der aktuellen Anwendung als Basisprozess definiert ist. Ist dies der Fall, führt das System diesen Basisprozess für den Datensatz aus. Wenn dies nicht der Fall ist, erfolgt keine Aktion.

Integrationsstipps für die Display-Anwendung

Wenn einem Anzeigebildschirm keine Anzeigereignisse zugeordnet sind, verwendet das System automatisch das Anzeigereignis **se.default**. Im Fall von **OnFormModified** führt dies **se.lock.object** aus.

Wenn für einen Anzeigeoptionsdatensatz das Kontrollkästchen **Ändert Datensatz** aktiviert ist, wird der Datensatz gesperrt, sobald auf die Schaltfläche geklickt wird.

Basisfunktionen in "se.search.engine"

Anzeigeaktion	Beschreibung
back	Beendet den Datensatz.
find	Zeigt detaillierte Informationen aus dem Datensatz an, der mit dem Feld verknüpft ist.
fill	Kopiert Informationen aus einem verknüpften Datensatz in einen Zieldatensatz. Fügt Daten aus einem Quelldatensatzfeld in ein Zielfeld ein.
advanced	Startet den Prozess Erweiterte Suche , der dem Benutzer weitere Suchoptionen bietet. Für gewöhnlich ist diese Option nur Systemadministratoren verfügbar. Diese Option kann jedoch auch anderen Benutzern zur Verfügung gestellt werden.
clear	Löscht den aktuellen Bildschirm.
openinbox/inbox	Fordert den Benutzer auf, eine der aktuellen Datei zugeordnete Ansicht zu öffnen. Dies ist eine vordefinierte Abfrage, die Ergebnisse in Form einer Ansicht anzeigt.
search	Führt eine Standardsuche aus, indem eine Suchabfrage unter Verwendung der angegebenen Informationen erstellt wird und anschließend die Ergebnisse in einer Datensatzliste angezeigt werden.
add	Führt die Format Control-Funktion add aus und versucht, der Datenbank einen Datensatz hinzuzufügen.
restore	Stellt den Inhalt des Bildschirms wieder her (nach Ausführen von clear).
irquery	Führt eine IR-Abfrage-Textsuche aus.
validitylookup	Führt die Service Manager-Funktion zum Nachsehen der Gültigkeit aus. Durch das Nachsehen der Gültigkeit wird anhand eines Eintrags in der Gültigkeitstabelle sichergestellt, dass ein bestimmter Wert für das definierte Feld gültig ist. Wenn Sie die Funktion zum Nachsehen der Gültigkeit verwenden, stellt das System sicher, dass ein vom Benutzer eingegebener Wert den Validierungsregeln entspricht.
expandarray	Durch das Erweitern eines Arrays werden zusätzliche Funktionen zum Bearbeiten von Arrays bereitgestellt, beispielsweise zum Einfügen und Löschen von Zeilen.
reset	Setzt die aktuelle Datei zurück. Diese Aktion löscht ALLE Datensätze für die Tabelle in der Datenbank.

Anzeigeaktion	Beschreibung
regen	Generiert die Indizes der aktuellen Datei neu. Diese Aktion ist nur auf Tabellen anwendbar, die über einen IR-Schlüssel verfügen. Bei dieser Aktion wird nur eine IR-Neugenerierung ausgeführt.
export/unload	Startet den Export-/Entladeprozess "Keine Datensätze". Diese Option wird ausschließlich zum Entladen leerer DBDict-Definitionen ohne Daten verwendet.
views	Wenn diese Option ausgewählt wird, zeigt das System dem Benutzer eine Liste alternativer Formulare an, in denen der aktuelle Datensatz angezeigt werden kann.
findrevision	Zeigt Revisionen für Datensätze in diesem Objekt an.
initrevision	Erstellt eine Revision eines Datensatzes in diesem Objekt.
newsite	
newview	
newTable	
addFilter	Ausgelöst von einem Kunden, der im Standardsuchbildschirm eine Suche mit erweitertem Filter ausführt.
editFilter	Ausgelöst von einem Kunden, der im Standardsuchbildschirm eine Suche mit erweitertem Filter ausführt.
addCompound	Ausgelöst von einem Kunden, der im Standardsuchbildschirm eine Suche mit erweitertem Filter ausführt.
removeSelection	Ausgelöst von einem Kunden, der im Standardsuchbildschirm eine Suche mit erweitertem Filter ausführt.

Basisfunktionen in "se.view.engine"

Anzeigeaktion	Beschreibung
save	Führt die Format Control-Funktion update aus und aktualisiert den Datensatz in der Datenbank, wenn die Aktualisierung gültig ist.
add	Führt die Format Control-Funktion add aus und fügt der Datenbank einen neuen Datensatz hinzu, wenn der neue Datensatz gültig ist.
ok	Speichert die Daten, wenn der Datensatz geändert wurde, oder schließt den Datensatz andernfalls.

Anzeigeaktion	Beschreibung
reselect	Wählt den aktuellen Datensatz aus der Datenbank erneut aus (für den Fall, dass dieser geändert wurde).
fill	Kopiert Informationen aus einem verknüpften Datensatz in einen Zieldatensatz. Fügt Daten aus einem Quellfeld in ein Zielfeld ein.
find	Zeigt detaillierte Informationen aus dem Datensatz an, der mit dem Feld verknüpft ist.
next	Wechselt zum nächsten Datensatz in der Liste (nachdem überprüft wurde, ob Änderungen vorliegen).
previous	Wechselt zum vorherigen Datensatz in der Liste (nachdem überprüft wurde, ob Änderungen vorliegen).
back	Beendet den Vorgang und ruft die Liste oder das Suchformular auf.
menu	Beendet den Vorgang und ruft das Aufrufmenü auf.
delete	Führt die Format Control-Funktion delete aus und versucht, nach der Validierung der Anforderung den Datensatz aus der Datenbank zu löschen.
views	Nach dem Auswählen zeigt das System dem Benutzer eine Liste alternativer Formulare an, in denen der aktuelle Datensatz angezeigt werden kann.
print	Druckt den aktuellen Datensatz.
printlist*	Druckt die aktuelle Liste der Datensätze.
validitylookup	Führt die Service Manager-Funktion zum Nachsehen der Gültigkeit aus. Durch das Nachsehen der Gültigkeit wird anhand eines Eintrags in der Gültigkeitstabelle sichergestellt, dass ein bestimmter Wert für das definierte Feld gültig ist. Wenn Sie die Funktion zum Nachsehen der Gültigkeit verwenden, stellt das System sicher, dass ein vom Benutzer eingegebener Wert den Validierungsregeln entspricht.
export/unload	Standardmäßige Export-/Entladefunktion des Service Managers (mehrere Datensätze).
massunload*	Bietet Entladefunktionen für eine Liste von Datensätzen.
massadd*	Fügt eine neue Datensatzgruppe hinzu, die eine vorhandene Datensatzgruppe exakt dupliziert. Die einzige Ausnahme besteht darin, dass die neue Datensatzgruppe über einen eindeutigen Schlüsselwert verfügt, der mit neuen Daten aktualisiert wird. Sobald die Verarbeitung abgeschlossen ist, weist die Tabelle doppelt so viele Datensätze auf wie zuvor.

Anzeigeaktion	Beschreibung
massupdate*	Stellt dieselben Aktualisierungen für eine Gruppe angegebener Felder in einer Liste von Datensätzen bereit.
massdelete*	Löscht eine angegebene Datensatzgruppe aus einer Tabelle.
irquery	Führt eine IR-Abfrage-Textsuche aus.
expandarray	Durch das Erweitern eines Arrays werden zusätzliche Funktionen zum Bearbeiten von Arrays bereitgestellt, beispielsweise zum Einfügen und Löschen von Zeilen.
count*	Zählt Datensätze in der Liste der Datensätze und zeigt die Gesamtanzahl an.
audithistory	Ruft die standardmäßige Prüfungsverlaufsroutine auf. Dieser Prüfungsverlauf wird von den zu prüfenden Feldern bestimmt, die in der Tabelle auditdef definiert wurden.
inbox.save/inbox*	Speichert die aktuelle Abfrage als Ansicht.
approval.log	Zeigt den Genehmigungsverlauf für den aktuellen Datensatz an.
alert.log	Zeigt den Alert-Verlauf für den aktuellen Datensatz an.
alerts	Zeigt die aktuellen und geplanten Alerts für den aktuellen Datensatz an.
pagelist/listpages	Die Seitenliste zeigt den vollständigen Verlauf zu einem Datensatz an. Wenn Paging aktiviert ist, wird bei jeder Aktualisierung eine Kopie des Datensatzes in die Seitendatei geschrieben. Die Seitenliste zeigt den vollständigen Verlauf zu einem Datensatz basierend auf der Paging-Datei an.
clocks	Zeigt diesem Datensatz zugeordnete Uhrendatensätze an.
xmlfill	Bearbeitet XML-Felder, z. B. die Benutzeroptionen in Service Catalog.
createTemplate	Erstellt auf der Grundlage des aktuellen Datensatzes einen Vorlagendatensatz.
applyTemplate	Wendet eine vorhandene Vorlage auf den aktuellen Datensatz an.

Hinweis: * Diese Befehle sind nur anwendbar, wenn die Datensatzlistenfunktion verwendet wurde.

Basisfunktionen in "se.list.engine"

Anzeigeaktion	Beschreibung
exit (oder back)	Wechselt zurück zum Suchformular (oder Aufrufformular).
inbox.save/inbox	Fordert den Benutzer auf, die aktuelle Abfrage als neue Ansicht zu speichern.
count	Führt Standardzählfunktionen aus und zeigt die Gesamtanzahl an Datensätzen an.
refresh	Aktualisiert die Liste unter Verwendung der aktuellen Abfrage.
big.green	Beendet das aktuelle Modul vollständig. "Großer grüner Pfeil"
print	Druckt die Liste der angezeigten Datensätze.
views	Wenn diese Option ausgewählt wird, zeigt das System dem Benutzer eine Liste alternativer Formulare an, in denen der aktuelle Datensatz angezeigt werden kann.
export/unload	Standardmäßige Export-/Entladefunktion des Service Managers (mehrere Datensätze).
massadd	Fügt eine neue Datensatzgruppe hinzu, die eine vorhandene Datensatzgruppe exakt dupliziert. Die einzige Ausnahme besteht darin, dass die neue Datensatzgruppe über einen eindeutigen Schlüsselwert verfügt, der mit neuen Daten aktualisiert wird. Sobald die Verarbeitung abgeschlossen ist, weist die Tabelle doppelt so viele Datensätze auf wie zuvor.
massupdate	Stellt dieselben Aktualisierungen für eine Gruppe angegebener Felder in einer Liste von Datensätzen bereit.
massdelete	Löscht eine angegebene Datensatzgruppe aus einer Tabelle.

Lokale Variablen

Lokale Variablen beginnen mit der Zeichenfolge **\$L.** und bestehen nur in der aktuell ausgeführten RAD-Anwendung. Sie werden vom Server automatisch gelöscht, wenn die RAD-Anwendung beendet wird.

Im Folgenden sind die vom Dokumentenmodul verwendeten Standardvariablen aufgeführt:

\$L.action – Der Anzeigeaktionswert der Anzeigeeoption

\$L.bg – Hintergrundkennzeichen

\$L.category – Der Kategoriedatensatz (falls verfügbar)

\$L.env – Der aktuelle Umgebungsdatensatz

\$L.exit – Der interne exit-Parameter

\$L.file – Die aktuelle Dateivariablen

\$L.file.save – Eine Kopie des Datensatzes in seinem ursprünglichen Zustand

\$L.format – Der Name des zum Anzeigen des Datensatzes verwendeten Formats

\$irspread – Bestimmt die IR-Erkennungsoptionen: 0 = oberflächliche Suche, 2 = tiefe Suche, 4 = vollständige Übereinstimmung

\$L.mode – Der Modus des angezeigten Datensatzes, normalerweise Hinzufügen zum Erstellen eines neuen Datensatzes, Aktualisieren zum Ändern eines vorhandenen Datensatzes oder Schließen zum Abschließen der Verarbeitung eines vorhandenen Datensatzes

\$L.mult – Ein Kennzeichen, dessen Wert **true** lautet, wenn die Variable **\$L.file** mehrere Datensätze aufweist

\$L.object – Der Objektdatensatz

\$L.phase – Der Phasendatensatz (falls verfügbar)

\$L.sql oder **\$L.query** – Die aktuelle Abfrage

\$L.sort – Die aktuelle Sortierreihenfolge

\$L.state – Der vom System verwendete Zustandsdatensatz (= der Zustand, in dem sich der Datensatz befindet)

Im Anzeigemodus verfügbare Variablen (bei der Anzeige eines einzelnen Datensatzes)

\$L.fc – Eine Kopie des Format Control-Detailldatensatzes

\$L.fc.master – Eine Kopie des Basis-Format Control-Datensatzes

Kapitel 6

Übersicht über die Fehlerbehebung

Für eine erfolgreiche Fehlerbehebung für das Dokumentenmodul müssen Sie die folgenden Informationen zusammenstellen:

- Verwendetes DBDict und verwendetes Objekt
- Zustand des Datensatzes
- Aufgerufener Prozess
- Befolgte Schritte zum Reproduzieren (STR) des Problems

Untersuchen des Anwendungspfads durch das Dokumentenmodul

Geben Sie zur Fehlerbehebung für das Dokumentenmodul sowie für jede Service Manager-Anwendung **RTM:3** und **debugdbquery:999** in die Service Manager-Datei `sm.ini` ein und starten Sie anschließend eine neue Clientverbindung. Wenn es sich bei diesem Benutzerprozess nicht um den ersten Prozess handelt, der die zu debuggenden Dokumentenmodulprozesse aufruft, wird die Auswahl der Zustands- und Prozessdatensätze in der Datei **sm.log** beim Ausführen dieser Verfolgung möglicherweise nicht angezeigt. Jedoch erhalten Sie hilfreiche Hinweise zum aufgerufenen Prozess.

Auffinden des verwendeten DBDicts oder Objekts

Durchsuchen Sie die Protokolldatei, um zu bestimmen, welches DBDict oder Objekt verwendet wurde.

Protokollbeispiel:

```
1320 07/18/2006 11:00:36 RADTRACE 20 [ 1] se.get.object get.object select CPU( 0 1411 )
1320 07/18/2006 11:00:36 (0x0129AC08) DBACCESS - Cache Find against file Object found 1
record, query: file.name="pcsoftware"
1320 07/18/2006 11:00:36 RADTRACE 20 [ 1] se.get.object set.access process CPU( 0 1411 )
```

Auffinden des Zustands für die Datensätze

Als Nächstes stellt sich die Frage nach dem Zustand der Datensätze. Um diese Informationen zu finden, suchen Sie im Verfolgungsprotokoll **sm.log** nach Folgendem:

Protokollbeispiel:

```
1320 07/18/2006 11:00:48 RADTRACE 10 [ 1] se.get.state select.state select CPU( 0 1491 )
1320 07/18/2006 11:00:48 (0x01292FB0) DBACCESS - Cache Find against file States found 1
record, query: state="pcs.list"
1320 07/18/2006 11:00:48 RADTRACE 10 [ 1] se.get.state exit.normal process CPU( 0 1491 )
```

Auffinden des Namens des Prozesses

Der Name des Prozesses kann ebenfalls gefunden werden, indem die Datei `sm.log` durchsucht wird, die die im folgenden Beispiel gezeigte Verfolgung enthält.

Protokollbeispiel:

```
1320 07/18/2006 11:00:50 RADTRACE 20 [ 1] se.call.process select.process select CPU( 0
1542 )
1320 07/18/2006 11:00:50 (0x00B56810) DBACCESS - Cache Find against file Process found 1
record, query: process="upgrade.pcs"
1320 07/18/2006 11:00:50 RADTRACE 20 [ 1] se.call.process run.pre.exp process CPU( 0 1542
)
```

Untersuchen von Anwendungsfehlern

Prozesse rufen eine Reihe von RAD-Anwendungen auf und führen eine Reihe von Ausdrücken aus. Dabei ist es möglich, dass anschließend weitere Prozesse aufgerufen werden. Wenn eine der Anwendungen oder einer der Ausdrücke aufgrund fehlerhafter Syntax oder Logik einen fehlerhaften Exit verursacht hat, können die Informationen dazu der Datei `sm.log` entnommen werden.

Protokollbeispiel:

```
Prozess-Panel run.pre.exp in RAD se.call.process hat Fehler in Zeile 1 erkannt
(se.call.process,run.pre.exp)
Ausdruck kann nicht ausgewertet werden (se.call.process,run.pre.exp)
Ausdruck kann nicht ausgewertet werden (se.call.process,run.pre.exp)
Fehlerhaftes Argument(2) Operator = (se.call.process,run.pre.exp)
Ausdruck kann nicht ausgewertet werden (se.call.process,run.pre.exp)
Ausdruck kann nicht ausgewertet werden (se.call.process,run.pre.exp)
Fehlerhaftes Argument(2) Operator = (se.call.process,run.pre.exp)
Ausdruck kann nicht ausgewertet werden (se.call.process,run.pre.exp)
Fehlerhaftes Argument(2) Operator nullsub (se.call.process,run.pre.exp)
Ausdruck kann nicht ausgewertet werden (se.call.process,run.pre.exp)
Fehlerhaftes Argument(2) Operator in (se.call.process,run.pre.exp)
Ausdruck kann nicht ausgewertet werden (se.call.process,run.pre.exp)
Unbehebbarer Fehler in Anwendung: se.search.objects, Feld call.list.engine
Unbehebbarer Fehler in Anwendung: se.list.engine, Feld call.process.1
Unbehebbarer Fehler in Anwendung: se.call.process, Feld run.pre.exp
```

In diesem Beispiel (nicht vordefiniert) trat der Fehler in `se.call.process, run.pre.exp`, bzw. beim Auswerten der Anfangsausdrücke des Prozesses auf. Um herauszufinden, welcher Prozess das Problem verursacht hat, führen Sie die zuvor beschriebenen Schritte aus und beachten Sie den Prozess der folgenden Zeile:

```
DBACCESS - Cache Find against file Process found 1 record, query: process="upgrade.pcs"
```

Rufen Sie den Prozessdatensatz mit dem Namen **upgrade.pcs** auf und suchen Sie im Register **Anfangsausdrücke** nach Anweisungen. In diesem bestimmten Fall enthält der Ausdruck das Word **nullsub**. Der Ausdruck für diesen Test konnte beispielsweise folgendermaßen lauten:

```
$L.icount=nullsub($L.icount, anynumberIwant)
```

Die Variable (**anynumberIwant**) ist kein gültiges Feld (Literal oder Variable) und muss geändert werden, um dieses Problem zu vermeiden.

Druckwerte von Variablen oder Ergebnisse von Ausdrücken

Im Dokumentenmodul wird der Pfad durch einen Workflow oft durch das Zuweisen eines Werts zu einem Feld oder einer Variablen bestimmt. Verwenden Sie die Funktion `JavaScript print()` oder `log rtecall ($L.void=rtecall("log", $L.rc, "message"))` in den RAD-Ausdrücken, um den Wert zu bestimmen, der einem Feld oder einer Variablen mit Einfluss auf den Workflow zugewiesen ist. Bei der Meldung kann es sich um eine verkettete Zeichenfolge handeln wie beispielsweise:

```
$L.message="The value of $L.test is " + $L.test
```

wobei `$L.test` eine Variable ist, der ein Zeichenwert zugewiesen wurde.

Kapitel 7

Übersicht über das Arbeitsauftragsbeispiel

Was ist ein Arbeitsauftrag? Ein Arbeitsauftrag ist ein bestimmter Auftrag, der einem einzelnen Techniker zugewiesen ist und zum Lösen eines Incidents erforderliche Aktivitäten definiert oder möglicherweise einem anderen Modul zugewiesen ist, wenn das Arbeitsauftragssystem so erweitert wurde, dass es Änderungen, Probleme und bekannte Fehler verarbeitet. Das folgende Beispiel basiert auf Service Manager, Version 7.11, und wurde geschrieben, um Benutzern das Erstellen, Aktualisieren und Schließen von Arbeitsaufträgen für Incident Management zu ermöglichen. Es kann einfach für andere Service Manager-Anwendungen geändert werden.

In diesem Beispiel wird ein Arbeitsauftragssystem erstellt, um die Verwendung des Dokumentenmoduls zu demonstrieren. Dieses Arbeitsauftragssystem ermöglicht es Benutzern, Arbeitsaufträge für einen Incident zu erstellen und den Status dieser Arbeitsaufträge anzuzeigen. Das Arbeitsauftragssystem ermöglicht es den Benutzern außerdem, einen Arbeitsauftrag von Incident Management aus anzuzeigen und zu aktualisieren. Alle Arbeitsaufträge sind einem bestimmten Incident im System zugeordnet. Der Incident kann erst geschlossen werden, wenn alle Arbeitsaufträge für den Incident geschlossen sind.

Dieses Beispiel richtet sich an Benutzer, die das System anpassen. Sie sollten über gute Kenntnisse zu den folgenden Anpassungsfunktionen verfügen:

- Database Dictionary zum Erstellen einer neuen Tabelle
- Forms Designer zum Ändern von vordefinierten Formularen und zum Erstellen neuer Formulare
- Werkzeug zum Erstellen eines Assistenten

In diesem Arbeitsauftragsbeispiel werden Sie durch die folgenden Schritte geführt:

- Erstellen eines neuen Database Dictionarys (DBDict) mit dem DBDict-Dienstprogramm
- Angeben der Schlüsselfelder für die Tabelle
- Erstellen eines EXWorkOrder-Formulars für die EXWorkOrder-Tabelle mit Forms Designer
- Erstellen einer Datei für Sequenznummern
- Ändern des Formulars zum Hinzufügen von Dropdown-Listen
- Erstellen eines Links für das EXWorkOrder-Formular
- Erstellen eines Formulars für den Assistenten zum Sammeln von Informationen für den Arbeitsauftrag
- Ändern der Formulare zum Schließen und zum Aktualisieren von Incidents
- Erstellen von Aliassen zum Verbinden von Arbeitsaufträgen mit Incidents
- Erstellen von Zustandsdefinitionen
- Erstellen von Bildschirmdefinitionen für die Display-Anwendung zum Öffnen, Schließen und Anzeigen

- Erstellen einer Optionsdefinition für die Display-Anwendung **Hinzufügen**, **Abbrechen**, **Füllen** und **Finden** zum Öffnen, Schließen oder Anzeigen
- Ändern des Prozessdefinitionsdatensatzes **im.set.close**
- Testen des Arbeitsauftragsbeispiels

Erstellen einer Tabelle

Verwenden Sie das DBDict-Dienstprogramm, um eine neue Tabelle zu erstellen. In diesem Beispiel erstellen Sie eine Tabelle mit dem Namen **EXWorkOrder**. Bevor Sie diese Tabelle erstellen können, müssen Sie wissen, welche Felder erforderlich sind. Darüber hinaus müssen Sie die Attribute dieser Felder kennen. In diesem Beispiel werden in der Tabelle die Daten für Arbeitsaufträge gespeichert, die Incidents zugeordnet sind.

So erstellen Sie eine Tabelle:

1. Klicken Sie im Systemnavigator auf **Anpassung > Database Dictionary**. Das Formular **Database Dictionary** wird geöffnet.
2. Geben Sie unter **Dateiname** den Namen `EXWorkOrder` ein.
3. Klicken Sie auf **Neu**.
4. Klicken Sie auf **Neues Feld/Schlüssel** und geben Sie anschließend die folgenden Informationen im Register **Felder** ein.

Feld	Beschreibung
ID	Wird von der Nummerndatei ausgefüllt. Typ: Zeichen
RelatedID	Wird von der eindeutigen ID des entsprechenden Datensatzes ausgefüllt. Beispiel: Incident-Nummer Typ: Zeichen
status	Wird zum Speichern des Status des Arbeitsauftrags verwendet. Typ: Zeichen
initiator	Der Bearbeiter, der den Arbeitsauftrag geöffnet hat. Typ: Zeichen
assignee.name	Der Bearbeiter, der dem Arbeitsauftrag zugewiesen ist. Typ: Zeichen
description	Dieses Feld enthält die Beschreibung der im Rahmen des Arbeitsauftrags auszuführenden Arbeit. Typ: Array von Zeichen
category	Dieses Feld wird von der Kategorie des zugehörigen

	Datensatzes ausgefüllt. Typ: Zeichen
RelatedCIs	Dieses Feld wird für die Liste der mit diesem Arbeitsauftrag verknüpften Konfigurationselementen (CIs) verwendet. Typ: Array von Zeichen
impact	Typ: Zeichen
urgency	Typ: Zeichen
priority	Typ: Zeichen
closure.code	Typ: Zeichen
deadline	Datum, bis zu dem der Arbeitsauftrag beim Öffnen ausgeführt und ausgefüllt sein muss. Typ: Datum/Uhrzeit
est.finish	Der vom Sachbearbeiter geschätzte Zeitpunkt der Fertigstellung. Typ: Datum/Uhrzeit
update.action	. Typ: Array von Zeichen
closure.comments	. Typ: Array von Zeichen

5. Nachdem Sie der Tabelle die Felder hinzugefügt haben, müssen Sie der Tabelle Schlüssel hinzufügen. Sie sollten das Dienstprogramm Database Dictionary erst beenden, wenn Sie der Tabelle **EXWorkOrder** alle erforderlichen Elemente hinzugefügt haben. Detaillierte Informationen hierzu finden Sie unter [Hinzufügen von Schlüsselfeldern zu einer Tabelle](#).

Hinzufügen von Schlüsselfeldern zu einer Tabelle

Verwenden Sie das DBDict-Dienstprogramm, um einer neuen Tabelle Schlüssel hinzuzufügen. Nachdem Sie eine Tabelle mit der Benennung **EXWorkOrder** erstellt haben, müssen Sie der Tabelle Schlüssel hinzufügen. Schlüssel werden verwendet, um indizierte Suchvorgänge zu ermöglichen und die Datenkonsistenz sicherzustellen.

So fügen Sie Schlüssel zu einer Tabelle hinzu:

1. Klicken Sie im **Database Dictionary** auf das Register **Schlüssel**.
2. Wählen Sie den ersten verfügbaren Eintrag für ein neues Schlüsselfeld aus.
3. Klicken Sie auf **Neues Feld/Schlüssel**.
4. Geben Sie für jeden Schlüssel, den Sie erstellen oder bearbeiten, die folgenden Informationen ein.

Feld	Beschreibung
ID	Typ: Eindeutig
RelatedID	Typ: Keine Nullwerte
RelatedCIs	Typ: Nullwerte und Duplikate
assignee.name	Typ: Nullwerte und Duplikate

5. Klicken Sie auf **OK**.

Erstellen eines Formulars

Rufen Sie Forms Designer auf, um ein Formular für die Tabelle **EXWorkOrder** zu erstellen. Wenn Sie das Formular unter Verwendung des Assistenten erstellen, wählen Sie **Datensatzdetails** aus und fahren Sie fort. Ein Basisformular mit sämtlichen Feldern wird erstellt und kann wie erforderlich geändert werden. Im Folgenden wird ein Beispiel für das Formular **EXWorkOrder** angezeigt.

So erstellen Sie ein Formular:

1. Klicken Sie im Systemnavigator auf **Anpassung > Forms Designer**.
Forms Designer wird geöffnet.
2. Geben Sie unter **Formular** die Zeichenfolge `EXWorkOrder` ein.
3. Klicken Sie auf **Neu**.
4. Klicken Sie auf **Ja**, um den Forms Designer-Assistenten aufzurufen.
5. Geben Sie `EXWorkOrder` als Namen der Tabelle ein, für die Sie ein Formular erstellen möchten.
6. Wählen Sie als Typ des Formulars, das Sie erstellen möchten, **Datensatzdetails** aus.
7. Klicken Sie auf **OK**.
8. Klicken Sie auf **Fortfahren**, um die Standardwerte für die im Formular zu verwendenden Felder zu übernehmen.
9. Verwenden Sie das Forms Designer-Werkzeug, um das Layout des Formulars zu ändern. Im Folgenden wird ein Beispiel für das Formular **EXWorkOrder** angezeigt:
10. Die Werte für **Status** lauten folgendermaßen:
 - New (Neu) - Standardstatus
 - Open (Geöffnet)
 - Ready (Bereit)
 - Closed (Geschlossen)
11. Die Werte für den Abschlusscode lauten folgendermaßen:
 - Implemented (Implementiert)
 - Canceled (Abgebrochen)

- Rollback

Nachfolgend wird ein Beispiel für das Formular **EXWorkOrder** angezeigt.

The screenshot shows a web browser window with the title 'WorkOrder'. The browser's address bar and menu bar are visible. The form itself has a header with the HP logo and a title bar. Below the title bar, there are several input fields: 'ID:', 'Verbundene ID:', 'Kategorie:', and 'Status:'. Below these, there are three tabs: 'Allgemein', 'Asset-Informationen', and 'Aktualisierung / Abschluss'. The 'Allgemein' tab is selected and contains the following fields: 'Initiator:', 'Sachbearbeitername:', 'Deadline:', 'Gesch. Sonstige:', 'Auswirkung:', 'Dringlichkeit:', and 'Priorität:'. The 'Beschreibung:' field is a large text area at the bottom of the form.

Erstellen einer Kopie eines Formulars

Erstellen Sie in Forms Designer das Formular **sc.manage.WorkOrder**, indem Sie eine Kopie von **sc.manage.problem** erstellen. Ändern Sie die Tabellenspalteneinträge, um Felder aus der Tabelle **EXWorkOrder** zu verwenden.

So erstellen Sie ein neues Formular aus einer Kopie eines Formulars:

1. Klicken Sie im Systemnavigator auf **Anpassung > Forms Designer**.
Forms Designer wird geöffnet.
2. Geben Sie `sc.manage.problem` in das Formular ein.
3. Klicken Sie auf **Suchen**.
4. Wählen Sie **sc.manage.problem.g** aus.
5. Klicken Sie im Menü **Weitere Aktionen** auf **Kopieren/Umbenennen**.
6. Geben Sie unter **Neuer Name** die Zeichenfolge `sc.manage.WorkOrder` ein.
7. Klicken Sie auf **OK**.

8. Aktualisieren Sie in Forms Designer mit Hilfe der Tabelle **EXWorkOrder** die Eingabefelder für die Spalten.
9.
 - Incident-ID - ID
 - Kategorie - category
 - Verbundene ID - RelatedID
 - Status - status
 - Sachbearbeiter - assignee.name
 - Beschreibung - description, 1
 - Priorität - priority
 - Anfängliche Auswirkung - impact
 - Dringlichkeit - urgency
10. Klicken Sie auf **OK**.

Erstellen eines Links für das Arbeitsauftragsformular

Ein Vorteil der relationalen Datenbank besteht darin, dass redundante Daten entfernt werden können. Hierzu müssen Daten zu einem bestimmten Thema in einer Tabelle gespeichert werden, die Links zu anderen Themen enthält. Bei Links handelt es sich um eine Kombination aus Daten und Link-Definitionen sowie Gruppen von Bedingungen, die die Beziehungen verknüpfter Daten enthalten.

Sie erstellen Sie einen Link für das Arbeitsauftragsformular:

1. Klicken Sie im Systemnavigator auf **Anpassung** > **Systemanpassungswerkzeuge** > **Links**. Das Formular **Link-Datei** wird geöffnet.
2. Geben Sie unter **Name** die Zeichenfolge `EXWorkOrder` ein.
3. Geben Sie unter **Beschreibung** eine Beschreibung ein.
4. Klicken Sie auf **Neu**.
5. Geben Sie die folgenden Informationen ein.

Name Quellfeld	Name Zieldatei	Name Zielfeld
initiator	operator	name
assignee.name	operator	name
RelatedCIs	device	logical.name

6. Markieren Sie die Initiatorzeile, klicken Sie auf das Menü **Weitere Aktionen** und wählen Sie **Zeile auswählen** aus.
7. Geben Sie in das Formular **link.structure.g** die folgenden Informationen für den Initiator ein. Wiederholen Sie anschließend den Schritt 6 für **assignee.name** und **RelatedCIs** und

verwenden Sie die folgenden Informationen für diese Felder.

Quellfeld (Einfügen in/Übertragen von)	Zielfeld (Einfügen von/Übertragen nach)
Initiator	name
assignee.name	name
RelatedCIs	logical.name

8. Klicken Sie auf **Speichern**.
9. Klicken Sie auf **OK**.

Erstellen einer Datei für Sequenznummern

Erstellen Sie eine Datei für Sequenznummern, um Sequenznummern für die Datensätze in der Tabelle **EXWorkOrder** zu generieren.

So erstellen Sie eine Datei für Sequenznummern:

1. Klicken Sie im Systemnavigator auf **Anpassung > Systemanpassungswerkzeuge > Sequenznummern**.

Das Formular **Datei für Sequenznummern** wird geöffnet.

2. Geben Sie unter **Klasse** die Zeichenfolge `EXWorkOrder` ein.
3. Geben Sie unter **Letzte Nummer** den Wert `1` ein.
4. Geben Sie unter **Beschreibung** die Beschreibung `Nummer des Arbeitsauftrags` ein.
5. Geben Sie unter **Länge** den Wert `5` ein.
6. Geben Sie unter **Präfix** die Zeichenfolge `WO` ein.
7. Klicken Sie auf **Hinzufügen**.

Erstellen einer Objektdefinition

Der Zweck dieses Objekts ist die Definition der Eigenschaften und des Verhaltens des Objekts `EXWorkOrder`, die bestimmen, welche Daten in einem Arbeitsauftragsdatensatz enthalten sein müssen und wie das System Arbeitsaufträge verarbeitet.

Hinweis: Beschreibungen der Felder des Formulars **Beschreibungen zum Formular Objektdefinition** finden Sie unter [Formular "Objektdefinition" und Felder auf Seite 12](#).

So erstellen Sie eine Objektdefinition:

1. Klicken Sie im Systemnavigator auf **Anpassung > Dokumentenmodul > Objekte**.
Das Formular **Objektdefinition** wird geöffnet.
2. Geben Sie unter **Dateiname** den Namen `EXWorkOrder` ein.
3. Klicken Sie auf **Hinzufügen**, um den Objektdatensatz zu erstellen.

4. Geben Sie im Register **Objektinfo** die folgenden Informationen ein:

Feld	Wert
Eindeutiger Schlüssel	Wird automatisch eingefügt (ID)
Allgemeiner Name	Wird automatisch eingefügt (EXWorkOrder)
Beschreibungsfeld	Tabelle mit Arbeitsaufträgen zu beliebigen Modulen
Profilanwendung	db.environment
Profilvariable	\$L.env
Nummerdatensatzname	EXWorkOrder
Kategorietabellenname	category
Basis-Format Control	EXWorkOrder
Statusfeld	status
Zugewiesen an-Felder	assignee.name
Öffnen-Zustand	EXWorkOrder.open
Schließen-Zustand	EXWorkOrder.close
Standardzustand	EXWorkOrder.view
Suchen-Zustand	EXWorkOrder.search

5. Klicken Sie auf **Speichern**.
6. Wechseln Sie zum Register **Warteschlangen verwalten**.
7. Geben Sie im Register **Warteschlangen verwalten** die folgenden Informationen ein:

Feld	Wert
Manage-Bedingung	true
Manage-Anzeigeformat	sc.manage.WorkOrder
Manage-Standardabfrage	assignee.name=operator()
Bedingung hinzufügen zulassen	false
Standardabfrage-Beschreibung	Meine Arbeitsaufträge

8. Klicken Sie auf **Speichern**.
9. Klicken Sie auf **OK**.

Erstellen einer Initialisierungsprozessdefinition

Diese Prozessdefinitionen für das Arbeitsauftragsbeispiel geben an, welche Anfangsausdrücke und RAD-Anwendungen verwendet werden sollen, wenn ein Benutzer einen Arbeitsauftragsdatensatz öffnet.

So erstellen Sie eine Prozessdefinition für die Initialisierung:

1. Klicken Sie im Systemnavigator auf **Anpassung > Dokumentenmodul > Prozesse**. Das Formular **Prozessdefinition** wird geöffnet.
2. Geben Sie unter **Prozessname** die Zeichenfolge `EXWorkOrder.open.initial` ein.
3. Klicken Sie auf **Hinzufügen**.
4. Geben Sie im Register **Anfangsausdrücke** die Zeichenfolge `$.number.format="EXWorkOrder"` ein.
5. Klicken Sie auf **Speichern**.
6. Geben Sie im Register **RAD** die folgenden Informationen ein.

Feld	Wert
Vor RAD-Aufruf ausgewertete Ausdrücke	<code>\$.number.record="EXWorkOrder";\$.number.type="string"</code>
RAD-Anwendung	<code>getnumb</code>
Parameternamen	Parameterwerte
name	<code>\$.number.record</code>
index	ID in <code>\$.file</code>
text	<code>\$.number.type</code>

7. Klicken Sie auf **Speichern**.

Erstellen von Bildschirmdefinitionen für die Display-Anwendung

Sie können Bildschirmdefinitionen für die Display-Anwendung zum Öffnen, Schließen und Anzeigen von Bildschirmen für die Arbeitsauftragsformulare erstellen, die es Benutzern ermöglichen, Arbeitsauftragsdatensätze zu öffnen, zu schließen und anzuzeigen.

So erstellen Sie eine Bildschirmanwendungsdefinition zum Öffnen, Schließen und Anzeigen:

1. Klicken Sie im Systemnavigator auf **Anpassung > Systemanpassungswerkzeuge > Anzeigebildschirme**. Das Formular **Bildschirmdefinition für Display-Anwendung** wird geöffnet.

2. Geben Sie unter **Bildschirm-ID** die Zeichenfolge `EXWorkOrder.open` ein.
3. Geben Sie die folgenden Informationen ein:

Feld	Wert
Bildschirm-ID	EXWorkOrder.open
Titel	Neuen Arbeitsauftrag öffnen
Format	\$L.format
I/O (f. RIO)	true
Bei Eingabe:	Bildschirm neu aufbauen
Sprache	ENG

4. Klicken Sie auf **Hinzufügen** und dann auf **OK**.

Für dieses Beispiel müssen Sie auch für den Bildschirm zum Schließen Bildschirmdefinitionen für die Display-Anwendung erstellen.

1. Klicken Sie im Systemnavigator auf **Anpassung > Systemanpassungswerkzeuge > Anzeigebildschirme**.
Das Formular **Bildschirmdefinition für Display-Anwendung** wird geöffnet.
2. Verwenden Sie die folgenden Informationen:

Feld	Wert
Bildschirm-ID	EXWorkOrder.close
Titel	Arbeitsauftrag schließen
Format	\$L.format
I/O (f. RIO)	true
Bei Eingabe:	Bildschirm neu aufbauen
Sprache	ENG

3. Klicken Sie auf **Hinzufügen** und dann auf **OK**.

Für dieses Beispiel müssen Sie auch für den Bildschirm zum Anzeigen Bildschirmdefinitionen für die Display-Anwendung erstellen.

1. Klicken Sie im Systemnavigator auf **Anpassung > Systemanpassungswerkzeuge > Anzeigebildschirme**.
Das Formular **Bildschirmdefinition für Display-Anwendung** wird geöffnet.
2. Verwenden Sie die folgenden Informationen:

Feld	Wert
Bildschirm-ID	EXWorkOrder.view
Titel	Neuen Arbeitsauftrag anzeigen
Format	\$L.format
I/O (f. RIO)	true
Bei Eingabe:	Bildschirm neu aufbauen
Sprache	ENG

3. Klicken Sie auf **Hinzufügen** und dann auf **OK**.

Erstellen von Optionsdefinitionen für die Display-Anwendung

Dieses Verfahren bietet eine Schritt-für-Schritt-Anleitung zum Erstellen, Hinzufügen, Abbrechen, Ausfüllen und Finden von Optionsdefinitionen für die Display-Anwendung für die folgenden WorkOrder-Definitionen:

- open
- close
- view

Hierzu wiederholen Sie für jede WorkOrder-Bildschirmdefinition (**open**, **close**, **view**) viermal dieselben Schritte. Dabei geben Sie jedoch jedes Mal unterschiedliche Werte in die Felder des Formulars **Optionsdefinition für Display-Anwendung** ein. Die Tabellen unter den Schritten stellen die Werte bereit, die Sie zum Konfigurieren der Display-Anwendungsoptionen benötigen.

Weitere Informationen zu den Optionen der Display-Anwendung finden Sie in den Themen der Online-Hilfe zur Display-Anwendung auf dem HP Service Manager-Online-Hilfe-Server.

Nachdem Sie die Bildschirmdefinitionen für die Display-Anwendung zum Öffnen, Schließen und Anzeigen erstellt haben, müssen Sie Optionen für die Display-Anwendung zum Öffnen, Schließen und Anzeigen von Bildschirmen erstellen.

So erstellen Sie eine Optionsdefinition für die Display-Anwendung:

1. Klicken Sie im Systemnavigator auf **Anpassung > Systemanpassungswerkzeuge > Anzeigoptionen**.
Das Formular **Optionsdefinition für Display-Anwendung** wird geöffnet.
2. Geben Sie unter **Bildschirm-ID** die Zeichenfolge `EXWorkOrder.open` ein.
3. Geben Sie die folgenden Informationen ein.

Feld	Wert
------	------

Eindeutige ID	Automatisch erstellt (EXWorkOrder.open_add)
Aktion	add
GUI-Option	4
Textoption	4
Bank	1
Bedingung	true
Standardlabel	Hinzufügen

4. Klicken Sie auf **Hinzufügen**.
5. Klicken Sie auf **OK**.
6. Wiederholen Sie die Schritte 1 bis 6 für jede der folgenden Display-Optionsdefinitionen. Verwenden Sie dazu die in der Tabelle angegebenen Werte.

Hinweis: Mitunter steht der für das Feld **Aktion** erforderliche Wert nicht in der Dropdown-Liste zur Verfügung. Sie können den entsprechenden Wert jedoch in das Feld eingeben.

Öffnen - Abbrechen

Feld	Wert
Bildschirm-ID	EXWorkOrder.open
Eindeutige ID	Vom System erstellt. (EXWorkOrder.open_cancel)
Aktion	back
GUI-Option	3
Textoption	3
Bank	1
Bedingung	true
Standardlabel	Abbrechen

Öffnen - Füllen

Feld	Wert
Bildschirm-ID	EXWorkOrder.open
Eindeutige ID	Vom System erstellt. (EXWorkOrder.open_fill)

Aktion	fill
GUI-Option	9
Textoption	9
Bank	1
Bedingung	true
Standardlabel	Füllen

Öffnen - Finden

Feld	Wert
Bildschirm-ID	EXWorkOrder.open
Eindeutige ID	Vom System erstellt. (EXWorkOrder.open_find)
Aktion	find
GUI-Option	8
Textoption	8
Bank	1
Bedingung	true
Standardlabel	Finden

Schließen - Abbrechen

Feld	Wert
Bildschirm-ID	EXWorkOrder.close
Eindeutige ID	Vom System erstellt. (EXWorkOrder.close_cancel)
Aktion	back
GUI-Option	3
Textoption	3
Bank	1
Bedingung	true
Standardlabel	Abbrechen

Schließen - Schließen

Feld	Wert
Bildschirm-ID	EXWorkOrder.close
Eindeutige ID	Vom System erstellt. (EXWorkOrder.close_close)
Aktion	close
GUI-Option	5
Textoption	5
Bank	1
Bedingung	true
Standardlabel	Schließen

Schließen - Füllen

Feld	Wert
Bildschirm-ID	EXWorkOrder.close
Eindeutige ID	Vom System erstellt. (EXWorkOrder.close_fill)
Aktion	fill
GUI-Option	9
Textoption	9
Bank	1
Bedingung	true
Standardlabel	Füllen

Schließen - Finden

Feld	Wert
Bildschirm-ID	EXWorkOrder.close
Eindeutige ID	Vom System erstellt. (EXWorkOrder.close_find)
Aktion	find
GUI-Option	8

Textoption	8
Bank	1
Bedingung	true
Standardlabel	Finden

Anzeigen - Abbrechen

Feld	Wert
Bildschirm-ID	EXWorkOrder.view
Eindeutige ID	Vom System erstellt. (EXWorkOrder.view_cancel)
Aktion	back
GUI-Option	3
Textoption	3
Bank	1
Bedingung	true
Standardlabel	Abbrechen

Anzeigen - Füllen

Feld	Wert
Bildschirm-ID	EXWorkOrder.view
Eindeutige ID	Vom System erstellt. (EXWorkOrder.view_fill)
Aktion	fill
GUI-Option	9
Textoption	9
Bank	1
Bedingung	true
Standardlabel	Füllen

Anzeigen - Finden

Standardlabel	Finden
---------------	--------

Bildschirm-ID	EXWorkOrder.view
Eindeutige ID	Vom System erstellt. (EXWorkOrder.view_find)
Aktion	find
GUI-Option	8
Textoption	8
Bank	1
Bedingung	true
Standardlabel	Finden

Anzeigen - Speichern

Feld	Wert
Bildschirm-ID	EXWorkOrder.view
Eindeutige ID	Vom System erstellt. (EXWorkOrder.view_save)
Aktion	save
GUI-Option	4
Textoption	4
Bank	1
Bedingung	true
Standardlabel	Speichern

Erstellen einer Zustandsdefinition

Diese Zustandsdefinitionen für das Arbeitsauftragsbeispiel legen fest, welcher Prozess verwendet werden soll und welche Aktionen zulässig sind, wenn ein Benutzer einen Arbeitsauftragsdatensatz öffnet, schließt oder anzeigt.

Hinweis: Beschreibungen der Felder des Formulars **Zustandsdefinition** finden Sie unter [Formular "Zustandsdefinition"](#) und [Felder auf Seite 28](#).

So erstellen Sie eine Zustandsdefinition für das Öffnen:

1. Klicken Sie im Systemnavigator auf **Anpassung** > **Dokumentenmodul** > **Zustände**.
Das Formular **Zustandsdefinition** wird geöffnet.
2. Geben Sie die folgenden Informationen ein:

Feld	Wert
Zustand	EXWorkOrder.open
Anzeigebildschirm	EXWorkOrder.open
Initialisierungsprozess	EXWorkOrder.open.initial
Format	\$.format
Eingabebedingung	true

3. Klicken Sie auf **Hinzufügen**.

So erstellen Sie eine Zustandsdefinition für das Schließen:

1. Klicken Sie im Systemnavigator auf **Anpassung > Dokumentenmodul > Zustände**. Das Formular **Zustandsdefinition** wird geöffnet.
2. Geben Sie die folgenden Informationen ein:

Feld	Wert
Zustand	EXWorkOrder.close
Anzeigebildschirm	EXWorkOrder.close
Initialisierungsprozess	EXWorkOrder.close.initial
Format	\$.format
Eingabebedingung	true

3. Klicken Sie auf **Hinzufügen**.

So erstellen Sie eine Zustandsdefinition für das Anzeigen:

1. Klicken Sie im Systemnavigator auf **Anpassung > Dokumentenmodul > Zustände**. Das Formular **Zustandsdefinition** wird geöffnet.
2. Geben Sie die folgenden Informationen ein:

Feld	Wert
Zustand	EXWorkOrder.view
Anzeigebildschirm	EXWorkOrder.view
Initialisierungsprozess	EXWorkOrder.view.initial

Feld	Wert
Format	\$.format
Eingabebedingung	true

3. Klicken Sie auf **Hinzufügen**.

So erstellen Sie eine Zustandsdefinition für das Suchen:

1. Klicken Sie im Systemnavigator auf **Anpassung > Dokumentenmodul > Zustände**. Das Formular **Zustandsdefinition** wird geöffnet.
2. Geben Sie die folgenden Informationen ein:

Feld	Wert
Zustand	EXWorkOrder.search
Anzeigebildschirm	db.search
Initialisierungsprozess	EXWorkOrder.view.search
Format	\$.format
Eingabebedingung	true

3. Klicken Sie auf **OK**.

Hinzufügen einer Schaltfläche zum Schließen von Arbeitsaufträgen

Sobald ein Benutzer die Aufgabe oder Aufgaben abschließt, sollte der Arbeitsauftrag geschlossen werden. Mit dem hier beschriebenen Verfahren wird dem Formular **EXWorkOrder** eine Schaltfläche zum Schließen hinzugefügt und der Status des zu schließenden Arbeitsauftrags aktualisiert. Es werden eine Optionsdefinition für eine Display-Anwendung, eine Zustandsdefinition für **EXWorkOrder.view** und eine Prozessdefinition für **EXWorkOrder.close** hinzugefügt.

Hinweis: Dieses Verfahren erklärt nicht, wie eine Validierung für diese Aktivität hinzugefügt werden kann. Zum Hinzufügen einer Validierung verwenden Sie **Format Control**.

So erstellen Sie eine Optionsdefinition für eine Display-Anwendung für **EXWorkOrder.view**:

1. Klicken Sie im Systemnavigator auf **Anpassung > Systemanpassungswerkzeuge > Anzeigeeoptionen**. Das Formular **Optionsdefinition für Display-Anwendung** wird geöffnet.
2. Geben Sie in das Feld **Bildschirm-ID** die Zeichenfolge `EXWorkOrder.view` ein.
3. Geben Sie die folgenden Informationen ein.

Feld	Wert
Eindeutige ID	Automatisch erstellt (EXWorkOrder.view_close)
Aktion	close
GUI-Option	5
Textoption	5
Bank	1
Bedingung	true
Standardlabel	Schließen

4. Klicken Sie auf **Hinzufügen**.
5. Klicken Sie auf **OK**.

So aktualisieren Sie die Zustandsdefinition **EXWorkOrder.view**:

1. Klicken Sie im Systemnavigator auf **Anpassung > Dokumentenmodul > Zustände**. Das Formular **Zustandsdefinition** wird geöffnet.
2. Geben Sie **EXWorkOrder.view** im Feld **Zustand** ein und klicken Sie auf **Suchen**. Das Definitionsformular für den Zustand **EXWorkOrder.views** wird geöffnet.
3. Fügen Sie Folgendes hinzu, um das Formular für die Zustandsdefinition zu aktualisieren.

Feld	Wert
Anzeigeaktion	Schließen
Prozessname	EXWorkOrder.close
Bedingung	true

4. Klicken Sie auf **Speichern** und dann auf **OK**.

So fügen Sie einen Prozessdefinitionsdatensatz für **EXWorkOrder.close** hinzu:

1. Klicken Sie im Systemnavigator auf **Anpassung > Dokumentenmodul > Prozesse**. Das Formular **Prozessdefinition** wird geöffnet.
2. Geben Sie unter **Prozessname** die Zeichenfolge `EXWorkOrder.close` ein.
3. Klicken Sie auf **Hinzufügen**.
4. Geben Sie im Register **Anfangsausdrücke** den folgenden Ausdruck ein:
 - `status in $L.file="Closed"`
 - `$L.mode="closed"`
5. Klicken Sie auf **Speichern**.
6. Geben Sie im Register **RAD** die folgenden Informationen ein:

Feld	Wert
Vor RAD-Aufruf ausgewertete Ausdrücke	\$L.EXaction="update"
RAD-Anwendung	se.base.method
Bedingung	true
Parameter - Namen	Parameterwerte
file	\$L.file
prompt	\$L.EXaction
second file	\$L.file.save
record	\$L.fc
second.record	\$L.object
boolean1	false

7. Klicken Sie auf **Speichern** und dann auf **OK**.

Erstellen eines Assistenten für den Arbeitsauftrag

In diesem Beispiel wird ein Assistent zum Erstellen eines Arbeitsauftrags verwendet, da dieser die Aufgabe für den Benutzer vereinfacht und das System die Funktionalität bereits zur Verfügung stellt.

Mit Hilfe des Werkzeugs zum Erstellen von Assistenten wird ein Assistent erstellt, der es Benutzern ermöglicht, einen Arbeitsauftrag über das Modul Incident Management zu erstellen. Wenn ein Benutzer einen Arbeitsauftrag aus einem Incident erstellt, fordert der Assistent den Benutzer zur Bereitstellung von Informationen auf. Einige Felder des Arbeitsauftrags werden automatisch ausgefüllt.

Sie erstellen Sie einen Assistenten für das Arbeitsauftragsformular:

1. Klicken Sie im Systemnavigator auf **Anpassung > Assistenten**. Das Formular **Assistenteninformationen** wird geöffnet.
2. Geben Sie unter **Assistentenname** die Zeichenfolge `Create Workorder - 1` ein.
3. Klicken Sie auf **Hinzufügen**, um den Assistentendatensatz zu erstellen.
4. Geben Sie im Register **Assistenteninfo** die folgenden Informationen ein:

Feld	Wert
Kurzbeschreibung	Erstellen eines neuen Arbeitsauftrags
Fenstertitel	Create Workorder

Titel	Create Workorder
Anfangsknoten?	Wenn Sie über mehrere Assistentendatensätze verfügen, aktivieren Sie das Kontrollkästchen für den Anfangsknoten (auf true festgelegt), um anzuzeigen, dass dies der erste Assistent einer Reihe von Assistenten ist.

5. Klicken Sie auf **Speichern**.
6. Wechseln Sie auf dem Formular **Assistenteninformationen** im Register **Dateiauswahl** zum Register "**\$L.file**" **auswählen**.
7. Geben Sie im Register "**\$L.file**" **auswählen** die folgenden Informationen ein:

Feld	Wert
Datensatz erstellen	true
vom Typ	EXWorkOrder

8. Klicken Sie auf **Speichern**.
9. Geben Sie im Register **Verwendung** die folgenden Informationen ein:

Feld	Wert
Assistentenverwendung	Klicken Sie im Abschnitt Assistentenverwendung auf Benutzereingabe anfordern .
Anzuzeigendes Unterformat	Geben Sie createWO.assigneeAndCIs ein.
Anzeigebildschirm	Geben Sie wizard.display ein.
Option "Fertig stellen" aktivieren?	Aktivieren Sie das Kontrollkästchen, damit auf dem Assistentenformular die Schaltfläche Fertig stellen zur Verfügung steht.

10. Klicken Sie auf **Speichern**.
11. Geben Sie im Register **Aktionen** die folgenden Informationen ein:

Feld	Wert
Aktionen ausführen für	Wählen Sie Aktuelle Datei (\$L.file) aus.

Durchzuführende Aktionen > Ausdrücke	initiator in \$.file=operator() status in \$.file="New" RelatedID in \$.file=number in \$relatedRec category in \$.file=category in \$relatedRec impact in \$.file=initial.impact in \$relatedRec urgency in \$.file=severity in \$relatedRec priority in \$.file=priority.code in \$relatedRec
Datensätze bei Abschluss anzeigen?	Aktivieren Sie das Kontrollkästchen Datensätze bei Abschluss anzeigen?
Modus	Wählen Sie Hinzufügen aus.

12. Klicken Sie auf **Speichern**.
13. Geben Sie im Register **Ausdrücke bei Abbruch** die folgenden Informationen ein:

Feld	Wert
Bei Abbruch ausgeführte Ausdrücke	cleanup(\$relatedRec)

14. Klicken Sie auf **Speichern** und dann auf **OK**.

Assistenteninformationen

Assistenteninfo | Dateiauswahl | Verwendung | Aktionen | Meldungen | Variablen | Nächster Assistent >>2

Assistentenname: Create Workorder - 1 Anfangsknoten?

Kurzbeschreibung: Erstellen eines neuen Arbeitsauftrags

Fenstertitel: Create Workorder

Titel: Create Workorder

Eingabeaufforderung:

Bitmap:

Globale Listen:

Hinzufügen eines Prozessdefinitions-Datensatzes

Der Prozessdefinitions-Datensatz legt fest, wie das System auf eine Benutzeraktion reagiert. Die Prozessdefinition verwendet RAD-Ausdrücke, Javascript und Aufrufe vorhandener RAD-Anwendungen, um die Aktionen für den aktuellen Datensatz, in diesem Fall ein Arbeitsauftragsdatensatz, auszuführen.

So fügen Sie einen Prozessdefinitions-Datensatz hinzu:

1. Klicken Sie im Systemnavigator auf **Anpassung > Dokumentenmodul > Prozesse**. Das Formular **Prozessdefinition** wird geöffnet.
2. Geben Sie unter **Prozessname** die Zeichenfolge **create.WorkOrder** ein.
3. Klicken Sie auf **Hinzufügen**.
4. Geben Sie im Register **Anfangsausdrücke** den folgenden Ausdruck ein:
 - `$L.void=fduplicate($relatedRec, $L.file)`
 - `$relatedCIs={}`
5. Klicken Sie auf **Speichern**.
6. Geben Sie im Register **Anfangs-Javascript** den folgenden Ausdruck ein:

```
system.vars.$relatedCIs=system.library.BSGFunctions.getMembers(system.vars.$L-  
_file.affected_item, false, 3)
```

Hinweis: Der Ausdruck muss in einer Zeile eingegeben werden.
7. Klicken Sie auf **Speichern**.

8. Geben Sie im Register **RAD** die folgenden Informationen ein:

Feld	Wert
Vor RAD-Aufruf ausgewertete Ausdrücke	Geben Sie die folgenden zwei Ausdrücke ein: <code>\$L.wiz.name="Create Workorder-1"</code> <code>if (not null(logical.name in \$L.file)) then (\$relatedCIs=insert (\$RelatedCIs, 1, 1, logical.name in \$L.file))</code> Hinweis: Achten Sie darauf, diesen Ausdruck in einer Zeile einzugeben. Beachten Sie darüber hinaus, dass beim Eingeben des Ausdrucks kein Leerzeichen nach dem Wort insert eingefügt wird. Beispielsweise wird der oben aufgeführte Ausdruck nach dem Wort insert folgendermaßen fortgesetzt: <code>insert(\$RelatedCIs, 1, 1, logical.name in \$L.file))</code>
RAD-Anwendung	wizard.run
Parameternamen	Name
Parameterwerte	\$L.wiz.name
Parameternamen	text
Parameterwerte	\$L.exit
Bedingung	true

9. Klicken Sie auf **Speichern**.
10. Geben Sie im Register **Endausdrücke** die Zeichenfolge `cleanup($relatedRec)` ein.
11. Klicken Sie auf **Speichern** und dann auf **OK**.

Erstellen eines Eingabeformulars für den Assistenten

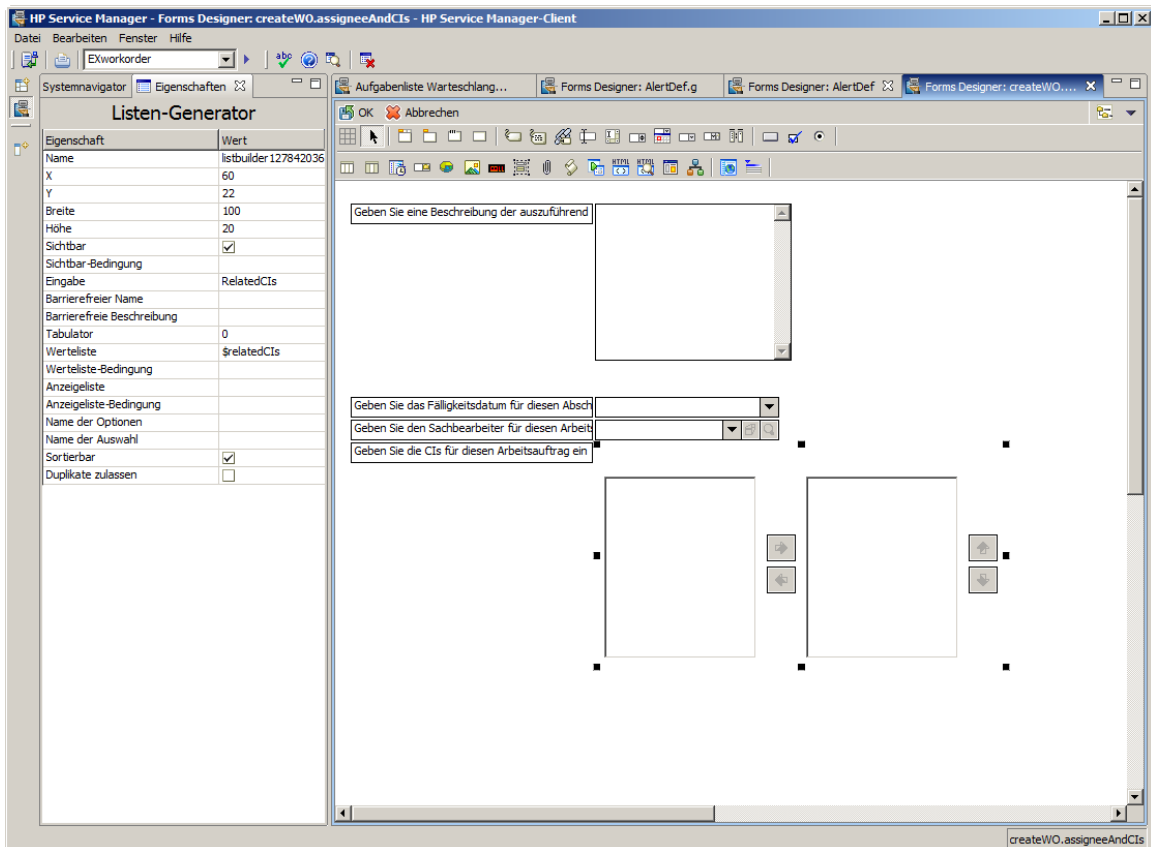
Dieses Formular ist das erste, das im Arbeitsauftragsassistenten angezeigt wird. Der Benutzer gibt die Informationen für den Assistenten ein, die zum Erstellen des Arbeitsauftragsdatensatzes erforderlich sind. Verwenden Sie zum Erstellen dieses Formulars den Forms Designer. In diesem Beispiel lautet der Formulaname **createWO.assigneeAndCIs**.

So erstellen Sie ein Eingabeformular für den Assistenten:

1. Klicken Sie im Systemnavigator auf **Anpassung > Forms Designer**.
 Das Forms Designer-Formular zum Suchen/Erstellen neuer Formulare wird geöffnet.

2. Geben Sie im Formular **createWO.assigneeAndCIs** als Formularnamen ein.
3. Klicken Sie auf **Neu**. Für dieses Formular müssen Sie den Forms Designer-Assistenten nicht verwenden.
4. Erstellen Sie die folgenden Eingabefelder auf dem Formular:
 - Beschreibung (Geben Sie eine Beschreibung der auszuführenden Arbeit ein.)
 - Fälligkeitsdatum (Geben Sie das Fälligkeitsdatum für diesen Arbeitsauftrag ein.)
 - Sachbearbeiter (Geben Sie den Sachbearbeiter für diesen Arbeitsauftrag ein.)
 - CIs (Geben Sie die CIs für diesen Arbeitsauftrag ein.)
5. Dieses Formular sollte die folgenden Eigenschaften aufweisen:
 - Eingabe: **RelatedCIs**
 - Werteliste: **\$relatedCIs**
 - Sortierbar: aktiviert
6. Klicken Sie auf **Speichern**.

Die folgende Abbildung zeigt ein Beispiel für das Formular **createWO.assigneeAndCIs** für das Eingabeformular des Assistenten **EXWorkorder** an.



Ändern der Formulare zum Schließen und zum Aktualisieren von Incidents

Im Falle des Arbeitsauftragsbeispiels müssen Sie die Formulare zum Schließen und zum Aktualisieren eines Incidents so ändern, dass Sie die dem Incident zugewiesenen Arbeitsaufträge anzeigen und auf sie doppelklicken können, um den Arbeitsauftrag vom Incident aus lesen oder bearbeiten zu können. Verwenden Sie Forms Designer, um die Formulare **IM.update.incident** und **IM.close.incident** zu aktualisieren. Hierzu müssen Sie in beiden Formularen (**IM.update.incident** und **IM.close.incident**) ein Register hinzufügen und anschließend ein Formular im Register.

So ändern Sie das Formular zum Aktualisieren des Incidents:

Hinweis: Verwenden Sie dieselben Schritte, um auch das Formular zum Schließen des Incidents zu aktualisieren.

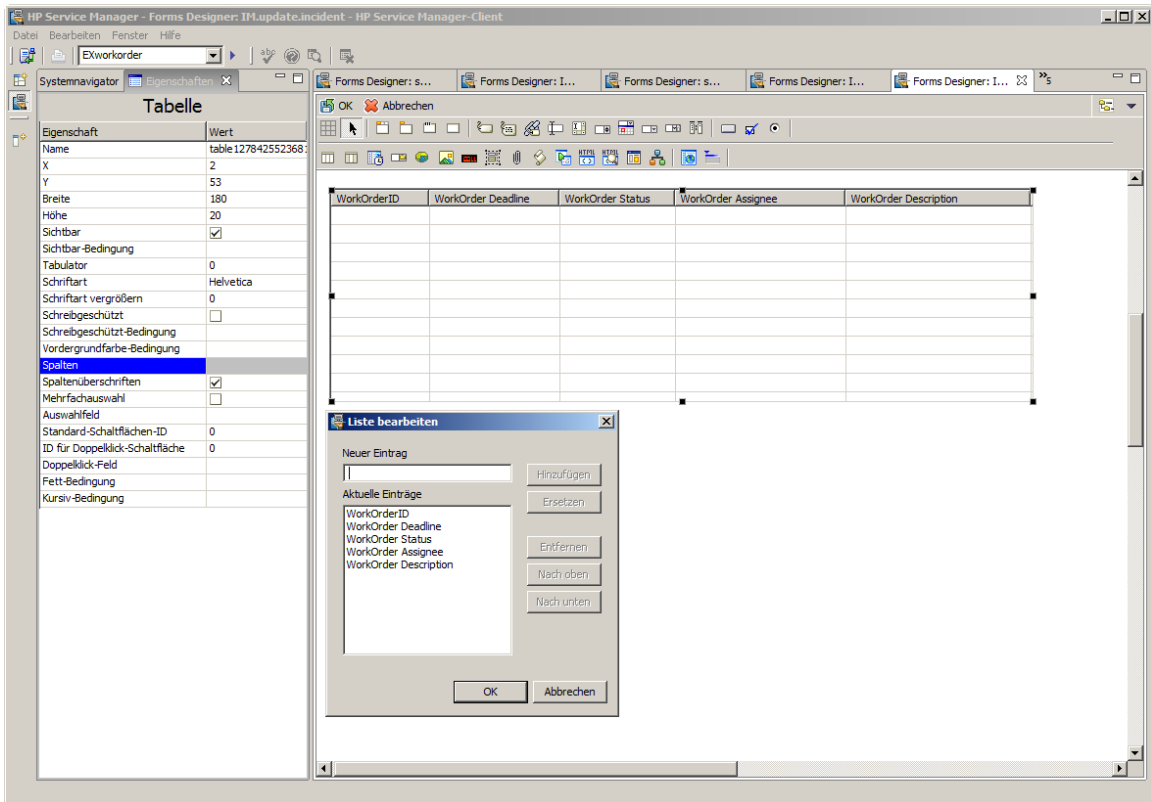
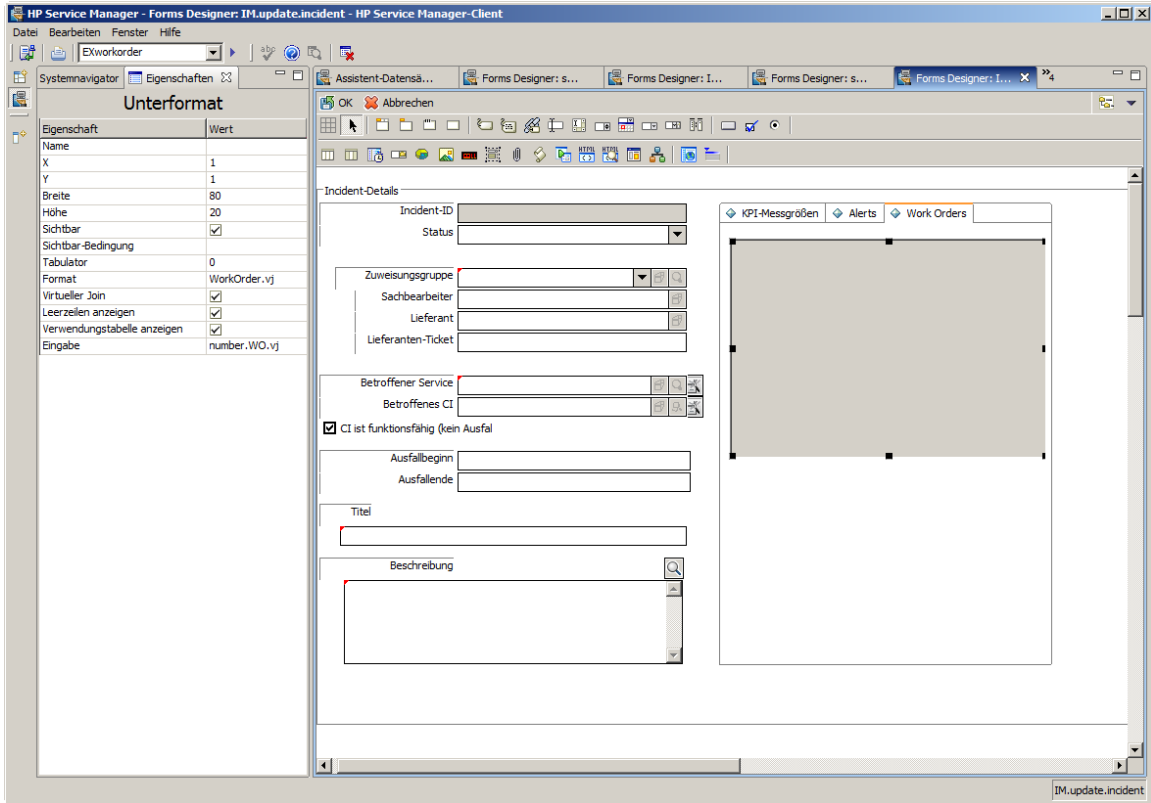
1. Klicken Sie im Systemnavigator auf **Anpassung > Forms Designer**. Das Suchen-Formular von Forms Designer wird geöffnet.
2. Geben Sie in das Formular **IM.update.incident** als Formularnamen ein.
3. Klicken Sie auf **Suchen**.
4. Klicken Sie auf **Gestalten**.
5. Fügen Sie ein Register zu einem bestehenden Notizbuch hinzu.
6. Legen Sie die Beschriftung des Registers auf **WorkOrder** fest.
7. Fügen Sie ein Unterformat mit den folgenden Eigenschaften hinzu:
 - Sichtbar: aktiviert
 - Format: WorkOrder.vj
 - Virtueller Join: aktiviert
 - Leerzeilen anzeigen: aktiviert
 - Tabelle für Anzeige verwenden: aktiviert
 - Eingabe: number.WO.vj
8. Klicken Sie auf **Speichern**.
9. Verwenden Sie das Forms Designer-Werkzeug, um das Formular **WorkOrder.vj** zu erstellen.
 - WorkOrder-ID: ID
 - WorkOrder-Deadline: Deadline
 - WorkOrder-Status: status
 - WorkOrder-Sachbearbeiter: assign.name
 - WorkOrder-Beschreibung: description
10. Wiederholen Sie die Schritte 1 bis 8 für das Formular **IM.close.incident**.

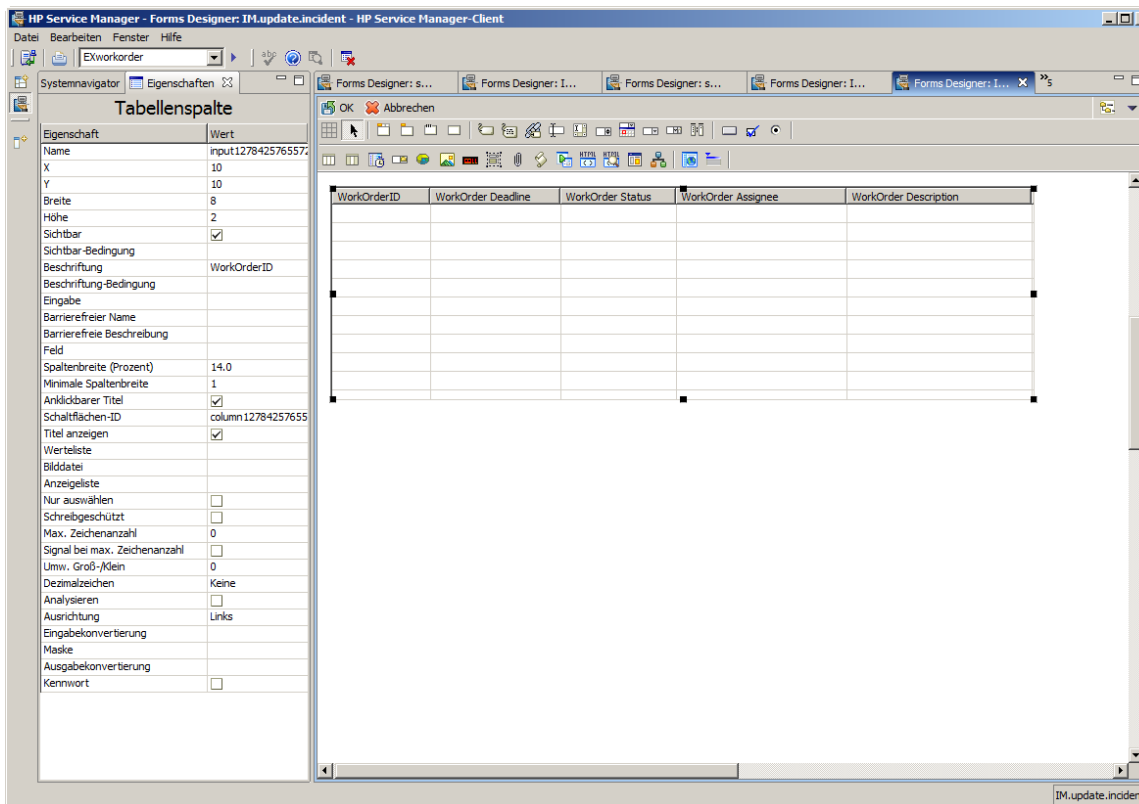
Hinweis: Sie können für den virtuellen Join das in Schritt 9 erstellte Formular **WorkOrder.vj** verwenden.

Die folgenden Abbildungen zeigen Beispiele der hinzuzufügenden Register.

Handbuch zum Dokumentenmodul

Kapitel 7: Übersicht über das Arbeitsauftragsbeispiel





Erstellen eines Alias zum Verknüpfen in der Tabelle "probsummary"

Im dem Arbeitsauftragsbeispiel müssen Benutzer von einem Incident-Datensatz aus auf einen Arbeitsauftrag zugreifen können, wenn mit dem Incident ein Arbeitsauftrag verknüpft ist. Hierzu müssen Sie in der Tabelle **probsummary** einen Alias erstellen und diesen Alias anschließend zum Erstellen eines Links zwischen der Tabelle **probsummary** und der Tabelle **EXWorkOrder** verwenden.

So fügen Sie der Tabelle **probsummary** einen Alias hinzu:

1. Klicken Sie im Systemnavigator auf **Anpassung > Database Dictionary**. Das Formular **Database Dictionary** wird geöffnet.
2. Geben Sie unter **Dateiname** die Zeichenfolge `probsummary` ein.
3. Klicken Sie auf **Suchen**.
4. Wählen Sie im Register **Felder** das Zahlenfeld aus und klicken Sie auf **Feld/Schlüssel bearbeiten**.
5. Klicken Sie auf **Alias erstellen** und geben Sie als Namen `number.WO.vj` ein und als Typ **Zeichen**.
6. Klicken Sie auf **OK**.

So verknüpfen Sie die Tabelle **EXWorkOrder** mit der Tabelle **probsummary**:

1. Klicken Sie im Systemnavigator auf **Anpassung** > **Systemanpassungswerkzeuge** > **Links**.

Das Formular **Link-Datei** wird geöffnet.

2. Geben Sie unter **Name** die Zeichenfolge `probsummary` ein.
3. Klicken Sie auf **Suchen**.
4. Klicken Sie im Anschluss an den letzten Eintrag, um eine leere Zeile für einen neuen Eintrag zu erstellen.
5. Geben Sie im Quellfeld `number.WO.vj` ein.
6. Wählen Sie die gesamte neue Zeile aus und klicken Sie anschließend auf **Zeile auswählen**.
7. Geben Sie die folgenden Informationen ein:

Feld	Wert
Feld (Von/Quelle)	number.WO.vj
Datei (Nach/Ziel)	EXWorkOrder
Feld (Nach/Ziel)	RelatedID
Abfrage	\$query
Ausdrücke	\$query="RelatedID=\""+number in \$File+"\""

8. Klicken Sie auf **Speichern** und dann auf **Zurück**.
9. Wiederholen Sie die Schritte, um eine Link-Zeile für das ID-Feld zu erstellen.
10. Geben Sie die folgenden Informationen ein.

Feld	Wert
Feld (Von/Quelle)	ID
Datei (Nach/Ziel)	EXWorkOrder
Feld (Nach/Ziel)	ID
Abfrage	\$query
Ausdrücke	\$query="ID=\""+nullsub(cursor.field.contents(), "xxx")+\""

11. Klicken Sie auf **Speichern** und dann auf **Zurück**.

Ändern der Definition des Prozesses "im.set.close"

Dieser Prozess muss geändert werden, damit der Benutzer einen Incident nicht schließen kann, wenn für diesen noch Arbeitsaufträge geöffnet sind.

So ändern Sie einen Prozessdefinitions-Datensatz:

1. Klicken Sie im Systemnavigator auf **Anpassung > Dokumentenmodul > Prozesse**. Das Formular **Prozessdefinition** wird geöffnet.
2. Geben Sie unter **Prozessname** den Namen **im.set.close** ein.
3. Klicken Sie auf **Suchen**.
4. Geben Sie im Register **Anfangs-Javascript** den folgenden JavaScript-Ausdruck ein:


```
var WO=new SCFile ("EXWorkOrder")

var FoundOpenWO=WO.doSelect ("RelatedID=\""+system.vars.$L_
file.number + "\""+ " and status ~=\"\" + "Closed" + "\"")

if (FoundOpenWO == RC_SUCCESS)

{

system.vars.$openWO=true;

}

else

{

system .vars.$openWO=false;

}


```
5. Klicken Sie auf **Speichern**.
6. Geben Sie im Register **RAD** die folgenden Informationen ein:

Feld	Wert
Hinweis: Einige der Felder im Register RAD weisen bereits Werte auf. Diese müssen Sie nicht ändern.	
RAD-Anwendung – Wählen Sie den leeren Abschnitt unter dem Abschnitt aus, der die RAD-Anwendung us.consume.wrapper aufruft, und geben Sie anschließend die folgenden Informationen ein:	
Vor RAD-Aufruf ausgewertete Ausdrücke	\$L.text="Es gibt noch geöffnete Arbeitsaufträge. Dieser Incident kann noch nicht geschlossen werden."
RAD-Anwendung	apm.mb.ok
Bedingung	\$openWO=true
Parameternamen	text
Parameterwerte	\$L.text

7. Klicken Sie auf **Speichern**.
8. Geben Sie im Register **Endausdrücke** die folgenden Informationen ein:

- if (\$openWO=true) then (\$L.exit="badval")
- \$L.exit="closestate"

9. Klicken Sie auf **Speichern** und dann auf **OK**.

Testen des Arbeitsauftragsbeispiels

Sobald Sie die Aufgaben zum Erstellen eines Arbeitsauftragssystems fertig gestellt haben, müssen Sie dessen ordnungsgemäße Funktion überprüfen. Führen Sie folgende Schritte aus, um die grundlegende Funktionalität des Arbeitsauftragsbeispiels zu überprüfen:

- Suchen Sie im Incident-Manager nach einen geöffneten Incident oder erstellen Sie einen.
- Verwenden Sie das Menü **Weitere Aktionen** oder die entsprechende Schaltfläche auf der Symbolleiste, um einen Arbeitsauftrag zu erstellen. Im Menü **Weitere Aktionen** sollte **Create Workorder** angezeigt werden.
- Geben Sie im Arbeitsauftragsassistenten das Datum ein, zu dem ein Arbeitsauftrag für den Incident erstellt werden soll.
- Fügen Sie den Arbeitsauftrag zu dem Incident hinzu und speichern Sie die Änderungen am Incident.
- Öffnen Sie den Incident zur Bearbeitung und bearbeiten Sie den Arbeitsauftrag vom Register **WorkOrder** aus.
- Speichern Sie Ihre Änderungen.
- Öffnen Sie den Incident erneut und bearbeiten Sie dieses Mal den Arbeitsauftrag und schließen Sie ihn.
- Jetzt sollten Sie den Incident schließen können.
- Wiederholen Sie diese Schritte, aber erstellen Sie dieses Mal zwei Arbeitsaufträge für einen Incident.
- Schließen Sie einen Arbeitsauftrag und versuchen Sie anschließend, den Incident zu schließen. Das System sollte eine Fehlermeldung erstellen, die anzeigt, dass der Incident noch immer offene Arbeitsaufträge aufweist. Deshalb kann er nicht geschlossen werden.
- Schließen Sie alle Arbeitsaufträge für den Incident. Jetzt sollten Sie den Incident schließen können.

