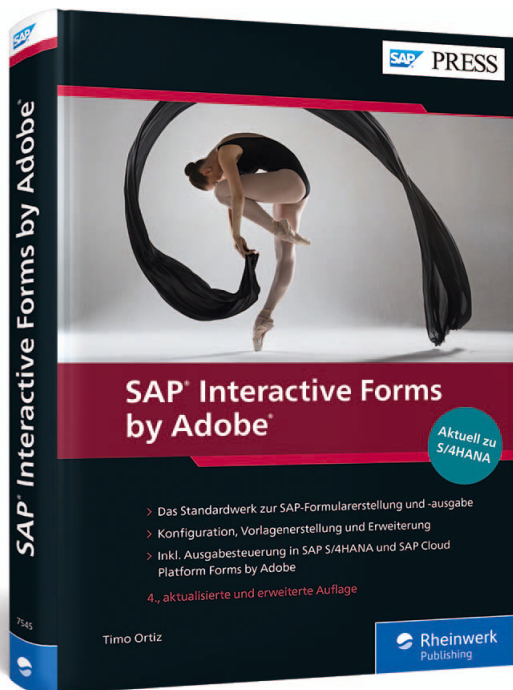


Timo Ortiz

SAP® Interactive Forms by Adobe®

Bonuskapitel »Interne Serviceanfrage«

© Rheinwerk Verlag GmbH, Bonn 2020
ISBN 978-3-8362-7545-3



Bonuskapitel

Interne Serviceanfrage

Die interne Serviceanfrage bietet die Möglichkeit, Antragsszenarien für innerbetriebliche Prozesse zur Abwicklung einer Serviceanfrage abzubilden. Das interaktive Adobe-Formular dient dabei als flexibel anpassbares Antragsformular.

In diesem Kapitel beschreiben wir die interne Serviceanfrage (Internal Service Request, ISR), die das interaktive Adobe-Formular als flexibel einsetzbares Antragsformular einbindet. Nach einer Einführung in die technischen Grundlagen der internen Serviceanfrage und die Rahmenbedingungen für mögliche betriebswirtschaftliche Antragsszenarien (siehe Abschnitt »Einführung«) zeigen wir Ihnen zunächst ein innerhalb von SAP ERP ausgeliefertes Szenario (siehe Abschnitt »Szenarien in SAP Enterprise Portal«). In Abschnitt »Getting started – ein einfaches Szenario erstellen«, wird erläutert, welche Überlegungen für die Erstellung eines Szenarios notwendig sind und wie es eingerichtet werden kann. Danach stellen wir die erforderlichen Customizing-Einstellungen sowie die Möglichkeiten programmtechnischer Erweiterungen zur Verfeinerung des Antragsszenarios im Detail dar. Der Abschnitt »Antragsformular als Adobe-Formular«, stellt dar, wie das Adobe-Formular im Rahmen des ISR definiert werden kann. Anschließend wird die mit Release SAP NetWeaver 7.0 EHP2 entwickelte neue Variante des ISR in der Anwendungsbasis (ABA) beschrieben.

Der Schwerpunkt des Kapitels liegt neben dem Customizing und den programmtechnischen Details des ISR auf der Beschreibung der Möglichkeiten des ISR anhand von Beispielen und hilfreichen Tipps. Die Lektüre dieses Kapitels soll Sie in die Lage versetzen, die Möglichkeiten des ISR-Frameworks zu überblicken und bedarfsgerechte, eigene formularbasierte Antragsszenarien zu implementieren.

Begriffe ISR und ISR-Szenario

Die verwendete Abkürzung ISR leitet sich vom englischen Begriff Internal Service Request ab. Ein ISR-Szenario umfasst alle Einstellungen in einem SAP-ERP-System, wie zum Beispiel SAP ERP 6.0, um eine Serviceanfrage für einen innerbetrieblichen Prozess abbilden zu können.



Einführung

Dieser Abschnitt beschreibt den betriebswirtschaftlichen Hintergrund und den technischen Rahmen für die Anwendung eines ISR-Szenarios innerhalb von SAP ERP sowie die erforderlichen systemtechnischen Rahmenbedingungen. Dabei verschafft Ihnen dieser Abschnitt einen Überblick über das ISR-Framework und benennt die wesentlichen Grundbegriffe.

Begriffe im ISR-Framework

Die interne Serviceanfrage dient der Eröffnung eines betriebsinternen Prozesses zur Abwicklung des vom Anwender gewünschten Service. Die Serviceanfrage ermöglicht es Gelegenheitsanwendern, in SAP Enterprise Portal über ein Online-Antragsformular einen beliebigen Service zu beantragen und den damit verbundenen erforderlichen internen Prozess abwickeln zu lassen.

Der Antragsteller kann dies ohne spezielle Kenntnisse des SAP-ERP-Systems durchführen. Die Hauptaufgabe ist, dem Anwender eine möglichst einfach zu bedienende Eingabemaske in Form eines Formulars anzubieten. Dies wird mit der Verwendung des interaktiven Adobe-Formulars gewährleistet. Aufgrund der freien und flexiblen Gestaltungsmöglichkeiten mit dem Form Designer kann das Formular für jedes ISR-Szenario passend entworfen werden.



Online-Formular

ISR unterstützt ausschließlich Online-Formulare. Das bedeutet, der Anwender muss das Formular im Portal ausfüllen und abschicken, der Offline-Modus wird generell nicht unterstützt.

Im Folgenden werden das ISR-Framework und die in diesem Rahmen wesentlichen Begriffe näher erläutert.

Innerbetrieblicher Prozess

Ein für eine Serviceanfrage geeigneter innerbetrieblicher Prozess wird in der Regel von vielen Anwendern immer wieder genutzt. Solche Prozesse erfordern oftmals die Durchführung von mitunter komplexen Transaktionen für Spezialisten oder Änderungen von geschützten Daten. Eine sinnvolle Anwendung für eine Serviceanfrage könnte beispielsweise die Beantragung eines Systembenutzers mit einem Passwort und den erforderlichen Berechtigungen für ein spezielles System sein.

Aufgrund der Komplexität solcher Transaktionen, der datenschutz- oder betriebskritischen Relevanz der Daten oder aus Gründen der Effizienz werden solche Transaktionen zumeist in kleinen, zentralen Verwaltungseinheiten betreut, in denen das

notwendige Spezialwissen vorgehalten wird. Ein komplizierter innerbetrieblicher Prozess kann aber auch die Beteiligung mehrerer Abteilungen erforderlich machen und einen oder mehrere Genehmigungsschritte beinhalten. Die Struktur der beteiligten Rollen in solchen Prozessen bleibt zumeist in ihren wesentlichen Bestandteilen gleich, unabhängig vom Anfordernden.

Der Antragsteller soll oder darf sogar diese Transaktionen nicht selbstständig durchführen, denn in der Regel fehlen dem Gelegenheitsanwender die Spezialkenntnisse und auch die notwendige Zugangsberechtigung zur Abwicklung seiner Serviceanfrage im System, wohingegen die Berechtigung zur Antragstellung durchaus gegeben ist. Häufig bleibt es dem Gelegenheitsanwender überlassen, mit mehr oder weniger ausführlicher Anleitung durch betriebsinterne Verfahrensanweisungen seine Serviceanfrage per E-Mail, telefonisch oder schriftlich zum Erfolg zu führen.

Kurzbeschreibung der Serviceanfrage

Durch das Ausfüllen und Absenden des Antragsformulars in SAP Enterprise Portal wird im SAP-ERP-System eine Meldung erzeugt, deren Bearbeiterfindung durch den SAP Business Workflow gesteuert werden kann. Aus der Meldung heraus kann der Bearbeiter direkt Aktivitäten ausführen (zum Beispiel E-Mails verschicken, Transaktionen ausführen). Die Antragsformulare und der daran anschließende Workflow sind individuell auf die zu realisierende Serviceanfrage und die Gegebenheiten im Unternehmen abstimmbaar.

Eigene ISR-Szenarien können schnell und problemlos aufgebaut werden. Durch die Integration mit der Controlling-Komponente können die betrieblichen Kosten der Anfrage direkt mit dem Kostenträger verrechnet, und mittels einiger Schnittstellen (BAdIs) können sowohl der Prozess und als auch das Formular jederzeit beeinflusst werden.

Technische Rollen

Im Rahmen des ISR sind unterschiedliche technische Rollen am Prozess für die Serviceanfrage beteiligt:

- **Antragsteller**
Der Antragsteller füllt bei Bedarf das interaktive Adobe-Formular zur Beantragung eines Service aus und schickt es ab. Er verfolgt den Status der Serviceanfrage.
- **Genehmigender**
Der Genehmigende genehmigt die Serviceanfrage oder lehnt sie ab.
- **Sachbearbeiter**
Der Sachbearbeiter sorgt für die Durchführung einer oder mehrerer betriebswirtschaftlicher Transaktionen in SAP ERP, die für die Umsetzung der Serviceanfrage notwendig geworden sind.

■ Administrator

Der Administrator ist für das betriebswirtschaftliche Szenario verantwortlich. Er verwaltet das szenariospezifische Customizing und die zugehörige ABAP-Programmierung.

Flexibles Framework

Die interne Serviceanfrage verbindet unterschiedliche Anwendungen aus SAP NetWeaver 7.0 und SAP ERP 6.0 und stellt flexible Konfigurationsmöglichkeiten mittels Customizing und geeigneten Programmschnittstellen zur Verfügung, um ein solches Szenario kundenspezifisch zu gestalten. Dazu gehört nicht nur die Abbildung des zu beantragenden Prozesses, sondern auch die Einbindung der SAP Interactive Forms by Adobe als Eingabemaske für den Antragsteller.

Das Adobe-Formular bietet neben den obligatorischen Eingabefeldern zusätzliche prozessspezifische Informationen und Hilfestellungen zur Erleichterung der Antragstellung an. Szenariospezifische Vorbelegungen von Eingabefeldern, Vorschlagswerte, Eingabehilfen, Online-Datenprüfung und beliebige Events erleichtern nicht nur das Ausfüllen des Formulars, sondern erzwingen bereits bei der Antragstellung eine hohe Datenqualität für die im Laufe des Prozesses auszuführenden Transaktionen. Des Weiteren ist es möglich, die genannten Funktionen und das Layout des Formulars anwenderspezifisch und damit dynamisch zu gestalten.

Einfache Szenarien bedürfen weniger Customizing-Einstellungen und Programmieraufwand als aufwendigere, komplexere Services. Wichtig ist, dass das Antragsformular für einen Gelegenheitsanwender verständlich bleibt und einfach zu bedienen ist. Dies ist unabhängig vom Grad der Komplexität mit dem ISR umsetzbar.

Voraussetzungen

Um interne Serviceanfragen mit interaktiven Adobe-Formularen nutzen zu können, müssen Sie mindestens SAP ERP 5.0 verwenden. Die Adobe-Formulare des ISR in SAP ERP 5.0 basieren noch auf der ACF-Technologie. Seit SAP ERP 5.0 steht Ihnen zudem die ISR-Oberfläche in Web Dynpro Java zur Verfügung. Für die Verwendung von interaktiven Adobe-Formularen benötigen Sie zudem den Adobe Reader der Version 6.0.2 oder höher und darüber hinaus mindestens SAP NetWeaver 6.40. Zudem ist das SAP Enterprise Portal erforderlich, um die ISR-Funktionalität vollständig anwenden zu können. Empfohlener Layouttyp für Adobe-Formulare ist das **ZCI Layout**.

Ab SAP ERP 6.0 SP 8 stellt der ISR seine Oberfläche auch in Web Dynpro ABAP zur Verfügung, die neben den Funktionen der Oberfläche in Web Dynpro Java noch einige Zusatzfunktionen aufweist. Technische Voraussetzungen sind hier SAP NetWeaver 7.0 SP 10 und der Adobe Reader 7.08 oder höher. Grundsätzlich ist zu empfehlen, immer die aktuellste Version des Adobe Readers zu verwenden.

Ausgelieferte Beispielszenarien des ISR in SAP Enterprise Portal finden Sie in verschiedenen Business-Paketen, wie zum Beispiel Manager Self-Services (MSS), Business Unit Analyst (BUA) und Employee Self-Services (ESS).

Um eine Serviceanfrage ausführen zu können, benötigt der Antragsteller zudem diverse Berechtigungen im SAP-ERP-System. Typische Fehlermeldungen beim Starten bzw. Versenden eines Formulars aufgrund fehlender Berechtigungen sind zum Beispiel:

- **Keine Berechtigung zum Anzeigen Benutzer (O1 495)**
Diese Fehlermeldung tritt bei fehlender Berechtigung für das Berechtigungsobjekt S_USER_GRP auf.
- **Ihnen fehlt die Berechtigung für Meldungsart... (QO 151)**
Diese Fehlermeldung tritt bei fehlender Berechtigung für das Berechtigungsobjekt B_NOBMP auf.
- **Bei der Bearbeitung der Meldung ist ein Fehler aufgetreten (IM 277)**
Diese Fehlermeldung tritt bei fehlender Berechtigung für das Berechtigungsobjekt I_QMEL auf.
- **Keine Berechtigung zu dieser Aktion (SBDS 106)**
Diese Fehlermeldung tritt bei fehlender Berechtigung für das Berechtigungsobjekt S_BDS_DS auf.

Alle Beschreibungen in diesem Kapitel basieren auf der mit SAP ERP 6.0 ausgelieferten Funktionalität, ältere Releases als SAP ERP 5.0 werden deshalb im Folgenden nicht erläutert. Insbesondere wird die technische Lösung des ISR unter der Verwendung von Web Dynpro ABAP beschrieben, daher wird auf die fast baugleiche Oberfläche in Web Dynpro Java nicht eingegangen.

Was kann konfiguriert und wo programmiert werden?

Wie in den folgenden Abschnitten dieses Kapitels detailliert ausgeführt wird, ist eine weitgehende Flexibilität in der Ausgestaltung einer Serviceanfrage das wichtigste Anliegen des ISR. Zahlreiche Einstellungen werden über das Customizing durchgeführt, für weitergehende Anpassungen, für die Sie die eigene ABAP-Programmierung verwenden möchten, wird eine Anzahl von BAdI-Schnittstellen zur Verfügung gestellt. Deren Implementierungen werden bei der Antragstellung und der anschließenden Abwicklung der Serviceanfrage zu geeigneten Zeitpunkten durch das ISR-Framework aufgerufen.

Abbildung 1 gibt einen Überblick über die Customizing-Einstellungen sowie die Schnittstellen und zeigt die Zusammenhänge.

- Dabei steht der Prozess von der Antragstellung bis zur Durchführung durch den Sachbearbeiter im Mittelpunkt der Schemazeichnung.

- Customizing-Einstellungen des ISR sind dunkelgrau markiert, die der Meldung mittelgrau, BADIs hellgrau, Funktionsbausteine bzw. ABAP-Klassen sind schwarz gezeichnet.

Das Szenario-Customizing bildet die Klammer um den gesamten Serviceantrag. Die einzelnen Komponenten werden in den nächsten Abschnitten näher vorgestellt.

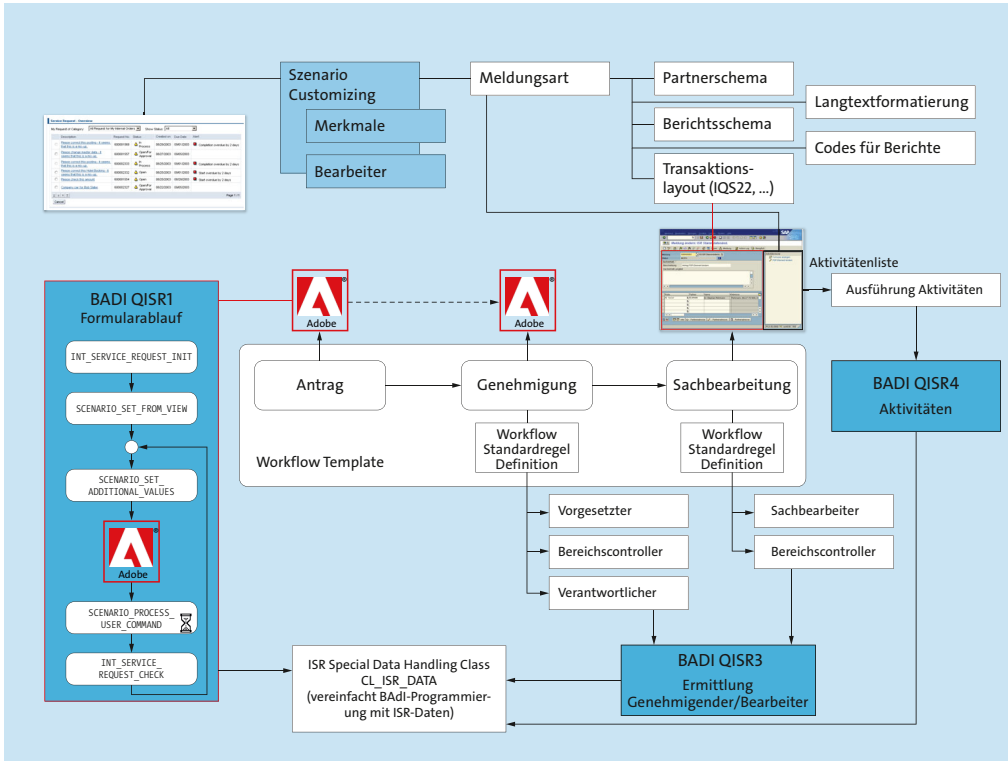


Abbildung 1 Schematische Darstellung der wechselseitigen Beziehungen bzw. Zuordnungen von Customizing und BAdI-Schnittstellen

Szenarien in SAP Enterprise Portal

Dieser Abschnitt bietet einen Überblick über bereits ausgelieferte Szenarien mit interaktivem Adobe-Formular und zeigt exemplarisch anhand einer ausgelieferten Beispielimplementierung eines ISR-Szenarios den Zweck und Nutzen des Frameworks. Im Rahmen dieses Beispielszenarios werden die wesentlichen Funktionen des ISR beschrieben und die Möglichkeiten der Einbindung ins Portal demonstriert.

Dieses Kapitel soll eine Übersicht über das Thema vermitteln und verzichtet daher auf eine vollständige Beschreibung der Funktionalität des ISR. Umfassende Infor-

mationen zu spezielleren Funktionalitäten finden Sie in den Abschnitten »Customizing und Programmierung eines Szenarios«, »Antragsformular als Adobe-Formular« sowie »Formularablauflogik«.

Gruppierung der ausgelieferten Szenarien

Mit SAP ERP werden verschiedene Szenarien ausgeliefert, die in der Regel als Anwendungsbeispiele und somit als Vorlage für kundeneigene Szenarien ausgelegt sind. Insbesondere das zugehörige Adobe-Formular wird allein aufgrund des Anspruchs kundenindividueller Gestaltung des Layouts zumeist angepasst. Trotzdem sind diese ausgelieferten Szenarien voll funktionsfähig.

Die Szenarien lassen sich in folgende betriebswirtschaftliche Themenblöcke gruppieren:

- Anträge auf das Anlegen bzw. die Änderung von Stammdaten aus SAP ERP, zum Beispiel Equipment, Kostenstelle, Innenauftrag, Profit-Center, PSP-Element, statistische Kennzahlen
- Anträge auf Änderung von Buchungen im Rechnungswesen
- Anträge auf Änderung von Budget
- Antrag auf neue Kostenberichte
- Planwerterfassung im Rahmen von Express Planning
- Anträge auf Änderung von Personalstammdaten von Angestellten
- HCM-Prozesse und Formulare
- Formulare zur Kenntnisnahme bzw. Bestätigung von speziellen Bestimmungen
- Formulare für Steuererklärungen (im Public Sector)
- Mitteilungen

Aufruf des Antragsformulars im Portal

Die im Abschnitt »Gruppierung der ausgelieferten Szenarien« genannten Gruppen von ISR-Szenarien unterscheiden sich nicht nur hinsichtlich ihres betriebswirtschaftlichen Hintergrundes, sondern auch in der Art und Weise, wie sie gestartet werden können. Die Formulare des ISR können dem Anwender in SAP Enterprise Portal prinzipiell auf drei verschiedene Arten angeboten werden:

■ Direkter Aufruf des Antragsformulars als Servicelink (per URL)

Das Antragsformular wird über einen Servicelink auf eine iView direkt gestartet (siehe Beispiel *Kostenstelle anlegen* im Abschnitt »Rolle Antragsteller«). Ausgelieferte generische iViews des ISR finden Sie im Portal unter **Portal Content • ERP Common Parts**.

■ **Indirekter Aufruf über das ISR Launchpad**

Das Antragsformular wird im Kontext einer anderen Anwendung über eine generierte Schaltfläche mithilfe des ISR Launchpads gestartet. Der Antrag bezieht den betriebswirtschaftlichen Kontext der Anwendung, aus dem er aufgerufen wird, mit ein (siehe Beispiel *Kostenstelle anlegen* im Abschnitt »Rolle Antragsteller«). Das ermöglicht es der umgebenden Anwendung, das Formular zumindest teilweise im Voraus auszufüllen. Das Anklicken des Auswahlknopfes führt analog zum Servicelink für den Start des Antragsformulars in einem neuen Browserfenster, eine direkte Kommunikationsmöglichkeit zwischen der ISR-Anwendung und der umgebenden Anwendung besteht nicht.

■ **Eingebunden in eine eigene Anwendung**

Das Antragsformular ist programmtechnisch in eine andere Web-Dynpro-Komponente eingebunden, wobei ein Datenaustausch zwischen den Web-Dynpro-Komponenten jederzeit möglich ist. Das Antragsformular wird im selben Browsermodus gestartet wie die gesamte Anwendung. Dies trifft beispielsweise auf die Antragsszenarien zur Änderung von Personalstammdaten zu (Personnel Change Requests, HCM-Prozesse und Formulare).

Die technischen Details der drei unterschiedlichen Verwendungsmöglichkeiten im Portal werden im Abschnitt »Verwendungsmöglichkeiten in SAP Enterprise Portal« näher beschrieben.

Rolle Antragsteller

Im folgenden Abschnitt stellen wir am Beispiel des ausgelieferten Szenarios *Kostenstelle anlegen* (technischer Name SMC1) die Funktionsweise des ISR von der Antragstellung bis zur Anlage der Kostenstelle im System dar.

In der Rolle **Bereichsanalyse** des Business Packages *Bereichscontroller* (BUA) kann das Antragsformular zum Szenario **Kostenstelle anlegen** neben den analog zu nutzenden Antragsszenarien **auftrag anlegen**, **Profitcenter anlegen** und **PSP-element anlegen** direkt als Servicelink aufgerufen werden (siehe Abbildung 2, linke Spalte, unter dem Ordner **Services**). Im Stammdatenmonitor des Bereichscontrollers (siehe Abbildung 2, iView **Stammdaten**) kann das entsprechende Antragsformular auch kontextabhängig aufgerufen werden. In beiden Fällen wird die Web-Dynpro-Anwendung QISR_UI_FORM in der iView gestartet (siehe Abschnitt »Verwendungsmöglichkeiten in SAP Enterprise Portal«). Das Szenario können Sie in Ihrem System vollständig nachvollziehen.

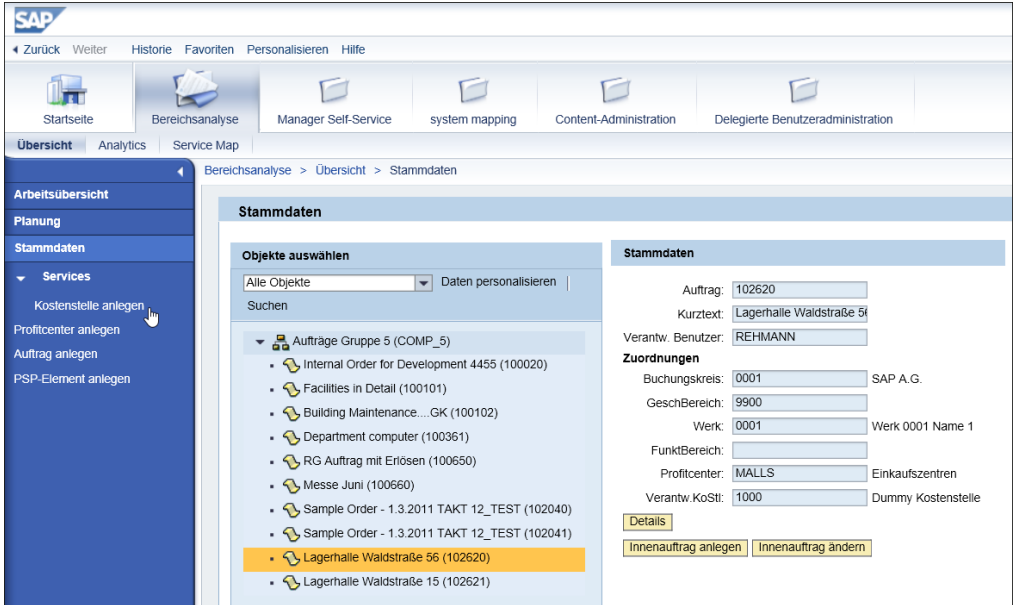


Abbildung 2 Aufruf »Kostenstelle anlegen« als Servicelink und im Stammdatenmonitor

Der Serviceantrag besteht aus drei Schritten; das Web-Dynpro-Bildschirmelement **Roadmap** über dem Formular dient der Visualisierung des aktuellen der insgesamt drei Antragschritte:

1. Im ersten Schritt wird das Formular ausgefüllt.
2. Im zweiten Schritt kann das ausgefüllte Formular noch einmal überprüft und ausgedruckt werden, bevor es abgeschickt wird.
3. Im dritten Schritt werden Sie über die erfolgreiche Versendung des Antrags informiert.

Diese Roadmap hat wohlgerne nichts mit dem nachgelagerten Prozess zur Durchführung des Serviceantrags zu tun.

Alternative

Wenn Sie die Web-Dynpro-Anwendung QISR_UI_FORM benutzen, sind diese drei Schritte festgelegt. Falls Sie jedoch das ISR-Antragsformular in eine eigene Web-Dynpro-Anwendung einbinden (siehe Abschnitt »Aufruf des Antragsformulars im Portal« und Abschnitt »Verwendungsmöglichkeiten in SAP Enterprise Portal«), können Sie auf die Roadmap verzichten.



1. Schritt: Formular bearbeiten

Abbildung 3 zeigt das Antragsformular, das aus einigen für die Beantragung einer neuen Kostenstelle notwendigen Eingabefeldern und aus einigen Anzeigefeldern besteht. Die Daten zur Person des Antragstellers (Name, Benutzerdaten etc.) werden automatisch ermittelt und angezeigt.

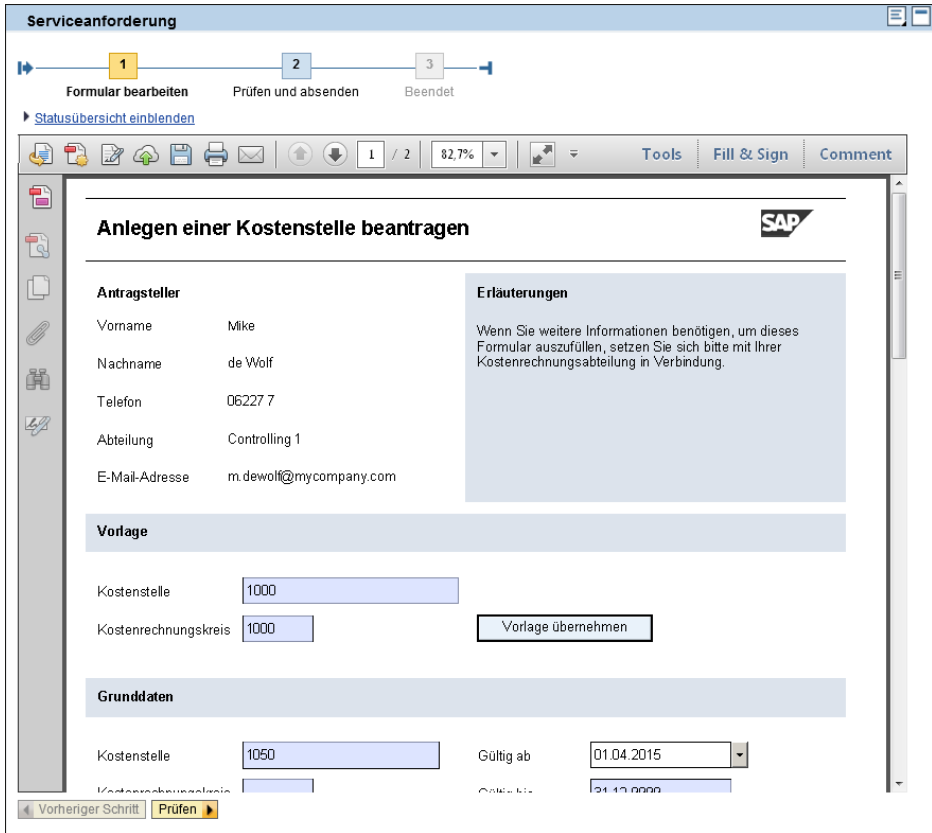


Abbildung 3 Antragsformular nach dem Start

Der Antragsteller hat die Möglichkeit, mithilfe der Schaltfläche **Vorlage übernehmen** eine existierende Kostenstelle als Vorlage für die neue Kostenstelle zu kopieren. Die Kostenstellenummer 1000 und der Kostenrechnungskreis 1000 wurden als Kopiervorlage aus der aktuell selektierten Kostenstelle des Stammdatenmonitors in das Formular übernommen.

Mit dem Drücken dieser Schaltfläche werden die Stammdaten der Kostenstelle 1000 in die entsprechenden Eingabefelder für die neue Kostenstelle in das Formular übernommen. Das erleichtert dem Antragsteller das Ausfüllen des Formulars, da – zumindest in diesem betriebswirtschaftlichen Kontext des Bereichscontrollers – Bezug auf eine existierende Kostenstelle genommen werden kann.

Im Unterschied dazu fehlt beim Starten des Antragsformulars als Servicelink der Kontext der Vorlagekostenstelle. In diesem Fall muss der Antragsteller die potenzielle Vorlagekostenstelle zunächst manuell eintragen, was in diesem einfachen Beispiel sicherlich als wenig problematisch erscheint. Allerdings ist es auch möglich, eine komplexere Dateninitialisierung aus dem Kontext einer umgebenden Anwendung vorzunehmen. Welche Art des Aufrufs Sie wählen, hängt demnach von den Anforderungen an ein Antragszenario ab. Vorstellbar wäre an dieser Stelle auch eine (programmierbare) Eingabehilfe.

Programmierbare Formularablauflogik mit BAdI QISR1

Die Funktion, die über eine Schaltfläche ausgeführt werden soll, wird im BAdI QISR1 programmiert. Weitere mögliche Funktionen neben Schaltflächen und Dateninitialisierungen auf einem Formular sind eine programmierbare Eingabehilfe, ein Event an einem Eingabefeld und die Prüfung der Formulardaten (siehe Abschnitt »Formularablauflogik«).

Am Ende des Formulars können Sie weitere Angaben oder Kommentare formlos in ein mehrzeiliges Langtextfeld eintragen (siehe Abbildung 4).

The screenshot shows a SAP form interface. At the top, there is a message icon and text: "Wenn Sie Felder erfassen möchten, die im Formular nicht angezeigt werden, geben Sie diese bitte im folgenden Textfeld ein." Below this, there is a label "Vorige Kommentare" and a text area. Further down, there is a label "Kommentare" and a larger text area containing the text: "Die Kostenstelle 1000 wurde aufgrund der Reorganisation aufgeteilt." At the bottom of the form, there are two buttons: "Vorheriger Schritt" with a left arrow and "Prüfen" with a right arrow.

Abbildung 4 Kommentareingabe und Prüfen im ersten Schritt

Nachdem Sie das Formular vervollständigt haben, drücken Sie die Schaltfläche **Prüfen** (oder **Nächster Schritt**), die unterhalb des Formulars angeboten wird. Zu diesem Zeitpunkt können Sie die Formulardaten vom System auf ihre Richtigkeit hin prüfen lassen. Diese Prüfungen werden im BAdI QISR1 programmiert. Sind einzelne oder mehrere Eingaben des Formulars fehlerhaft, kann die Fortführung des Antrags unterbrochen werden; hierzu können Sie auch einzelne oder mehrere Fehlermeldungen ausgeben, die oberhalb des Formulars angezeigt werden. Der Antrag bleibt in diesem Fall bei Schritt 1 stehen (siehe Abbildung 5). Nach der Korrektur der inkorrekten Feldwerte wechselt der Antrag zu Schritt 2 des Formulars, **Prüfen und absenden**.

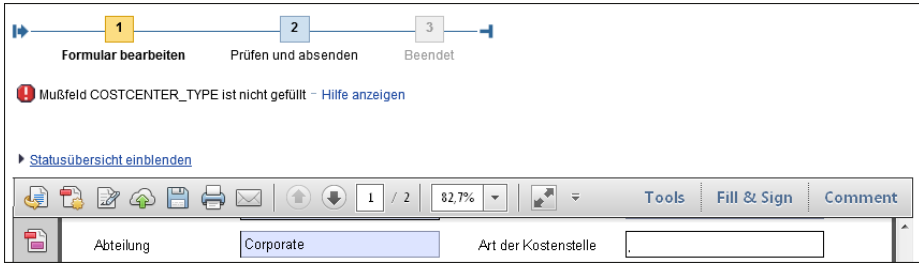


Abbildung 5 Fehlermeldung im ersten Schritt

2. Schritt: Prüfen und absenden

In Schritt 2 wird im Wesentlichen das Formular erneut angezeigt, es ist aber nicht mehr änderbar. Sie können das Formular mithilfe der Adobe-Reader-Funktionalität zu Dokumentationszwecken ausdrucken. Sind Eingaben nach Meinung des Antragstellers korrekturbedürftig, können Sie über die entsprechende Schaltfläche **Vorheriger Schritt** zu Schritt 1 zurückkehren (siehe Abbildung 6).

Abbildung 6 Zweiter Schritt zur Prüfung der Eingaben

Sind die Daten sowohl nach systemseitiger Prüfung als auch im Sinne des Antragstellers korrekt, können Sie den Antrag per Druck auf **Senden** (oder **Abschicken**) abschicken.

3. Schritt: Beendet – Ausgabe der Antragsnummer

Nachdem der Antrag im System erzeugt wurde, wird in Schritt 3 die jeweilige Antragsnummer bestätigt, die der Meldungsnummer im SAP-ERP-System (siehe [Abbildung 7](#)) entspricht. Außerdem können Sie den Antrag per Link (Web-Dynpro-Anwendung QISR_UI_DISPLAY_ONLY) aufrufen.

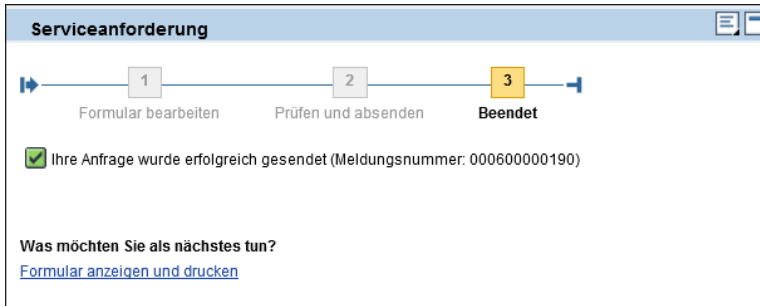


Abbildung 7 Dritter Schritt mit Meldungsnummer des Antrags

Weitere Schritte nach der Antragstellung

Der Antragsteller hat nun die Gelegenheit, den Status seiner Serviceanfrage mithilfe der ISR-Statusübersicht (die Web-Dynpro-Anwendung QISR_UI_STATUSOVERVIEW) zu verfolgen (siehe [Abbildung 8](#), die als separate iView dargestellt werden kann. Der Antragsteller kann die Statusübersicht aber auch über den Link **Statusübersicht einblenden** innerhalb eines Antragsformulars öffnen (siehe [Abbildung 3](#)). Von dort kann das Formular jederzeit im Anzeigemodus aufgerufen und gedruckt werden.

Statusübersicht Interner Service Anfragen										
Beschreibung	Meldung n.	Status	Angelegt am	Geändert am	Gew.Beginn	Gew.Ende	Zusätzliche Spalte 1	Zusätzliche Spalte 2	Warnung	Storno
Die Kostenstelle 1000 wurde wegen der Re	600000190	offen / zur Genehmigung	17.04.2015		17.04.2015	16.06.2015	1050	Corporate Services		stornieren
Antrag für Messeauftrag 2015/Q3	600000191	in Bearbeitung / genehmigt	17.04.2015	17.04.2015	17.04.2015	16.06.2015	199997	Messeauftrag 2015 Q3		stornieren
Umzug ins Gebäude 456	600000192	abgeschlossen / abgelehnt	17.04.2015	17.04.2015	17.04.2015	16.06.2015	1999966	Umzug ins Gebäude 456		stornieren
Obbligo fehlt im Bericht	600000193	offen	17.04.2015		17.04.2015	16.06.2015				stornieren

Daten vom: 17.04.2015 Uhrzeit: 13:10:46 [Aktualisieren](#)

Abbildung 8 Statusüberblick aller Anträge des Antragstellers

Je nach Einstellung ist es zumindest technisch möglich, dass der Antragsteller den Antrag nachträglich ändern kann. In diesem Beispiel ist dies nicht vorgesehen. Allerdings sieht der Antragsteller beim Starten des Formulars im Statusüberblick immer den aktuellen Stand seines Antragsformulars. Das bedeutet, dass er beispielsweise auch den entsprechenden Kommentar des Genehmigenden sehen kann. Solange der Antrag noch nicht von einem Sachbearbeiter oder einem Genehmigenden bearbeitet wird, ist der Status des Antrags **offen**. Anträge dieses Status können jederzeit über die passende Funktion im ISR-Statusüberblick storniert werden. Wechselt der Status auf **in Bearbeitung**, kann nicht mehr storniert werden. In zusätzlich konfigurierbaren Spalten können bis zu fünf ausgewählte Werte eines Formulars angezeigt werden

(siehe Abschnitt »Merkmale«). Im Fall des Antrags 1005183 in Abbildung 8 werden die gewünschte neue Kostenstellenummer und deren Bezeichnung angezeigt.

Rolle Genehmigender

Mit der erfolgreichen Antragstellung wurde eine Meldung im System im Status **offen** erzeugt. Da im Customizing der zugehörigen Meldungsart und des ISR die Verwendung des SAP Business Workflows vorgesehen ist, wurde mit der Antragstellung ein entsprechendes Workflow-Item erstellt; im Beispiel von *Kostenstelle anlegen* ist ein Genehmigungsverfahren vorgesehen. Dieses Workflow-Item erscheint beim im Customizing hinterlegten Genehmigenden sowohl in der Universal Worklist (UWL) in SAP Enterprise Portal als auch im Eingang vom **Arbeitsplatz** (Transaktion SBWP) in SAP ERP.

Der Genehmigende hat die Wahl, den Antrag über einen der beiden Eingänge zu bearbeiten. Beim Öffnen des Workflow-Items werden Ihnen jedoch unterschiedliche Benutzeroberflächen zur Verfügung gestellt. Es hängt vom verwendeten Workflow-Template ab, welche Art der Antragsbearbeitung für den Genehmigenden möglich ist (siehe Abschnitt »Bearbeiterfindung (BAdI QISR3) und Workflow«).

- **UWL: Start des Formulars im Portal mit Genehmigungsfunktionen**

Nimmt der Genehmigende den Antrag in der UWL entgegen, wird das Originalantragsformular im neuen Browserfenster gestartet (siehe Abbildung 9).

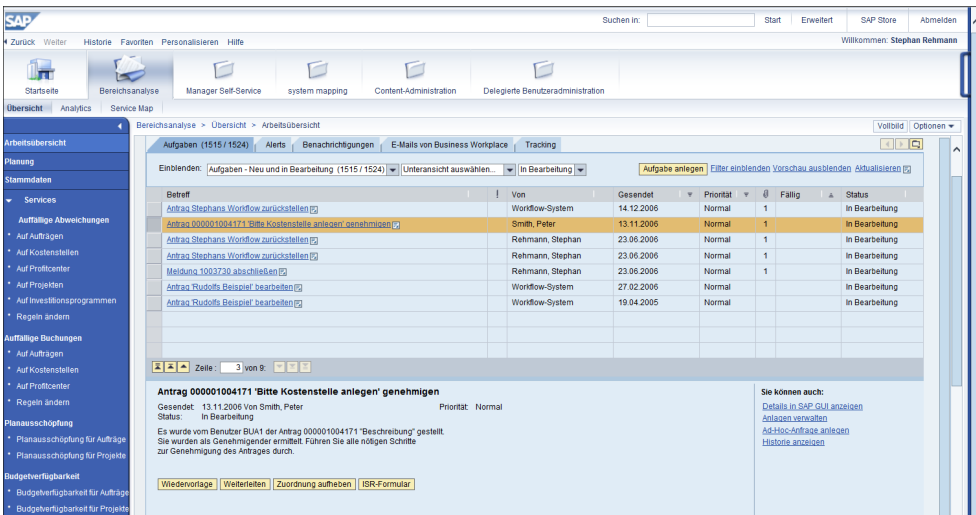


Abbildung 9 Workflow-Item in der UWL des Genehmigenden

Über die Funktionen **Genehmigen**, **Ablehnen** oder **Zurück zum Antragsteller** – dargestellt als Schaltflächen unter dem Formular – kann nunmehr über den Antrag entschieden werden (siehe Abbildung 10).

► Statusübersicht einblenden

Anrede

Name 1 Name 2

Name 3 Name 4

Straße Postfach

Ort Postleitzahl

Ortsteil PLZ-Postfach

Land Region

Jurisdiction

i Wenn Sie Felder erfassen möchten, die im Formular nicht angezeigt werden, geben Sie diese bitte im folgenden Textfeld ein.

Vorige Kommentare

* 04/17/2015 17:30:05 Stephan Rehmann (REHMANN) Tel. 06227 7 61466
Die Kostenstelle 1000 wurde wegen Reorganisation aufgeteilt

Kommentare

ist genehmigt

Genehmigen **Ablehnen** **Zurück zu Antragsteller**

Abbildung 10 Kommentareingabe und Funktionen des Genehmigenden auf dem Formular

Der Genehmigende hat in diesem Beispielszenario die Möglichkeit, seine Entscheidung zu kommentieren, die Antragsdaten können durch den Genehmigenden jedoch nicht geändert werden. Die Entscheidung **Ablehnen** führt auch zur automatischen Schließung der Meldung im System. **Zurück zu Antragsteller** führt dagegen zu einem neuen Workflow-Item im Eingang des Antragstellers, der nun den Antrag abändern und erneut abschicken kann.

Grundsätzlich erlaubt das ISR-Framework auch, die Antragsdaten durch den Genehmigenden ändern zu lassen (siehe Abschnitt »Remote Function Call »ISR PROCESS EVENT«). Ob das sinnvoll ist, hängt vom jeweiligen Antragszenario ab. Die Statusänderung aufgrund der Entscheidung des Genehmigenden und die Kommentierung kann der Antragsteller im ISR-Statusüberblick einsehen.

■ **Arbeitsplatz: Start von Transaktion IQS22 aus dem Workflow-Item**

Falls der Genehmigende den Antrag im Eingang des Arbeitsplatzes annimmt, wird Transaktion IQS22 (siehe Abbildung 11) aufgerufen. Dort hat der Genehmigende analog die Möglichkeit, den Antrag zu genehmigen oder abzulehnen. Die Funktion **Zurück zu Antragsteller** wird hier nicht unterstützt.

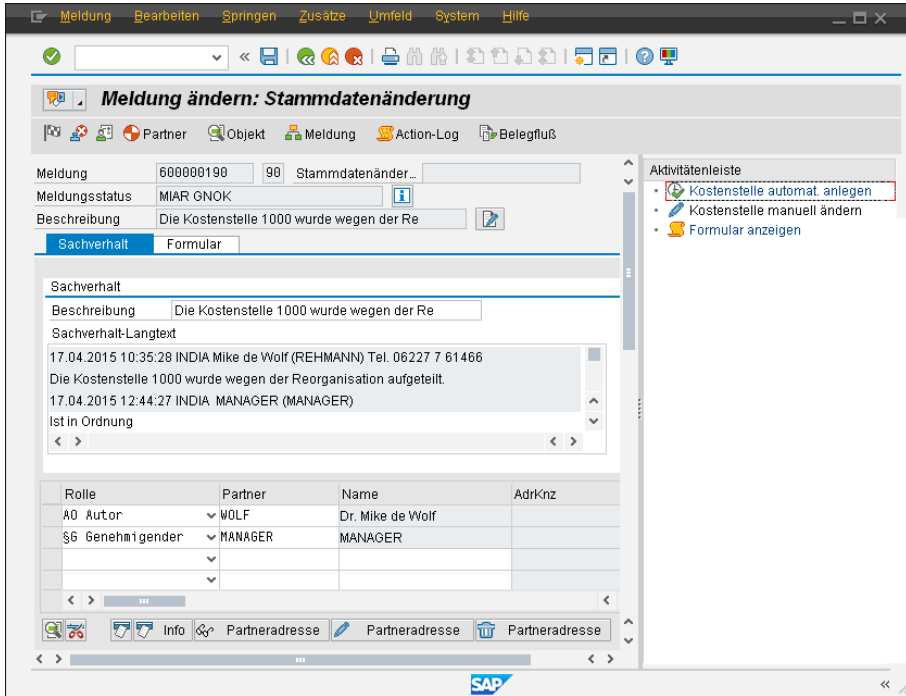


Abbildung 11 Meldungstransaktion IQS22

Wird der Antrag genehmigt, wird ein neues Workflow-Item für den zuständigen Sachbearbeiter erzeugt, der den Antrag ausschließlich mit der Transaktion IQS22 bearbeitet.

Rolle Sachbearbeiter

Der Sachbearbeiter kann das Formular mit den Originaldaten über eine Aktivität in der Meldungstransaktion IQS22 aufrufen und überprüfen sowie die Daten des Formulars in tabellarischer Form einsehen (siehe Abbildung 12).

Meldung ändern: Stammdatenänderung

Partner Objekt Meldung Action-Log Belegfluß

Meldung: 000000190 Stammdatenänder...

Meldungsstatus: MIAR GNOK

Beschreibung: Die Kostenstelle 1000 wurde wegen der Re

Sachverhalt Formular

Antragsdaten	
Kostenstellenstammdaten	
KostRechKreis	1000
Kostenstelle	1050
Gültig ab	01.04.2015
Gültig bis	31.12.9999
Verantwortl.	BUA
Abteilung	Corporate
KStellenart	F
Std-Hierarchie	1000
Buchungskreis	1000

Abbildung 12 Tabellarische Ansicht der Formulardaten in der Meldung

Anders als dem Genehmigenden werden dem Sachbearbeiter nun die verschiedenen Aktivitäten zur Durchführung der Serviceanfrage im System angeboten. Im Beispielszenario sind dies:

■ **Formular anzeigen**

Die Aktivität ruft die Web-Dynpro-Component QISR_UI_DISPLAY_ONLY in einem Browserfenster auf. Beachten Sie dazu das Customizing in Abschnitt »Aktivitäten und automatische Verbuchungen in der Meldung«.

■ **Kostenstelle automatisch anlegen**

Diese Aktivität führt ein BAPI zum Anlegen einer Kostenstelle aus, die Daten für das BAPI werden aus den Formulardaten übernommen.

■ **Kostenstelle manuell ändern**

Diese Aktivität führt zur Änderungstransaktion der Kostenstelle.

Diese Aktivitäten wurden im Szenario-Customizing festgelegt und im BAdI QISR4 ausprogrammiert. Zur Anzeige der Aktivität wurde ein Text für den Link in Transaktion IQS22 definiert. Eine wesentliche Einstellung für die Aktivität ist der Eintrag eines Funktionsbausteins zur Ausführung der Aktivität.

Automatische Ausführung einer Aktivität

Das ISR-Szenario kann so eingestellt werden, dass die Aktivität zum Anlegen der gewünschten Kostenstelle im Backend-System bereits automatisch mit der Geneh-



migung des Antrags auf dem Formular durchgeführt wird. In diesem Fall würde die Meldung automatisch geschlossen werden, und die Rolle des Sachbearbeiters wäre obsolet.

Noch ein Tipp: Programmieren Sie in der BAdI-Implementierung QISR4 der Aktivität das Versenden einer E-Mail zur automatischen Benachrichtigung des Antragstellers.

Hat der Sachbearbeiter die Aktivität(en) zum Anlegen der Kostenstelle erfolgreich durchgeführt, schließt er die Meldung über eine Menüfunktion ab. Optional kann er vorher noch einen Kommentar in das Langtextfeld eingeben. Der Status wechselt auf **Abgeschlossen**, somit wurde der Service erfolgreich ausgeführt und beendet. Der Antragsteller kann diesen Status im Statusüberblick überprüfen (siehe Abbildung 8).

Workflow oder Arbeitsvorrat?

Mit der Verwendung der Meldung als Datenträger des Antrags können Sie den Antrag per SAP Business Workflow an die zuständigen Stellen bzw. Personen weiterleiten, Sie können auf den Workflow aber auch verzichten. Die Meldung bietet prinzipiell mit den Transaktionen IQS8 und IQS9 sogenannte Arbeitsvorräte, mit deren Hilfe ein Sachbearbeiter oder Genehmigender die für ihn relevanten Anträge ermitteln kann. Von dort gelangt der Sachbearbeiter oder der Genehmigende zu der entsprechenden Transaktion (IQS22 bzw. IQS23) für die eigentliche Bearbeitung des Antrags. In Abbildung 11 wird die Oberfläche der Sachbearbeitertransaktion IQS22 gezeigt.

Getting started – ein einfaches Szenario erstellen

Nachdem Sie nun wissen, was Sie mit einem ISR-Szenario erreichen können, sollen im folgenden Abschnitt die notwendigen Vorüberlegungen und die Herangehensweise für die Erstellung Ihres eigenen Szenarios erläutert werden. Danach wird der ISR-Wizard zur Generierung eines einfachen Szenarios beschrieben. Haben Sie diesen Abschnitt Schritt für Schritt nachvollzogen, verfügen Sie über alle erforderlichen Grundkenntnisse, um eigene Szenarien entwerfen und implementieren zu können.

Vorüberlegungen für ein neues ISR-Szenario

Vor der Erstellung eines ISR-Szenarios müssen Sie zuerst einige Fragen zum betriebswirtschaftlichen Hintergrund des Serviceantrags klären. Tabelle 1 soll dazu dienen, Ihre Vorüberlegungen möglichst umfassend beschreiben zu können. Überlegen Sie

zunächst, wie der Gesamtprozess aussieht, der unterstützt werden soll. Erst wenn Sie eine Vorstellung dieses Prozesses haben, sollten Sie detaillierte Betrachtungen anstellen.

Themengebiet	Vorüberlegung
Antragsformular	<ul style="list-style-type: none"> ■ Welche Eingabefelder (Merkmale) sollen auf dem Antragsformular erscheinen? Welche Felder sind für die Erledigung der Serviceanfrage notwendig? ■ Welche Prüfungen, Dateninitialisierungen, Eingabehilfen oder spezielle Funktionen benötigen der Antragsteller oder andere Rollen auf dem Formular? ■ Wie sieht das Layout des Formulars aus? ■ Liegt Ihnen bereits ein Formular vor?
Rollen, Arbeitsvorrat und Workflow	<ul style="list-style-type: none"> ■ Welche Rollen sind für die Durchführung des Service vorgesehen? ■ Wollen Sie den Workflow für die Bearbeitung der Meldung verwenden, oder soll der Sachbearbeiter seine zu bearbeitenden Meldungen über einen Arbeitsvorrat selektieren? ■ Sind Genehmigungsschritte erforderlich und wie viele? ■ Müssen mehrere Bearbeiter gleichzeitig tätig werden, oder können der oder die Sachbearbeiter nacheinander aktiv werden? ■ Nach welchen Kriterien soll die Bearbeiterfindung erfolgen (fester Bearbeiter oder abhängig von Antragsfeldern, Organisationsstrukturen, Kosten, Nachnamen des Antragstellers etc.)?
Antrag im Backend (Meldung)	<ul style="list-style-type: none"> ■ Welche Aktivitäten (Aufruf von Transaktionen, Antwort-E-Mail etc.) sollen den Bearbeitern in der Aktivitätsliste der Meldungstransaktion zur Verfügung gestellt werden? ■ Sollen einzelne Aktivitäten automatisch (zum Beispiel durch den Antragsteller oder Genehmigenden) durchgeführt werden?
Auswertung	<ul style="list-style-type: none"> ■ Möchten Sie ein detailliertes Reporting über Ihr Antragsszenario erstellen?
Servicekosten	<ul style="list-style-type: none"> ■ Wollen Sie die Servicekosten Ihres Antragsszenarios erfassen und auswerten? ■ Genügen Ihnen einfache Pauschalkosten, oder möchten Sie differenzierte Servicekosten berechnen?

Tabelle 1 Checkliste für die Erstellung eines Szenarios

Haben Sie diese Fragen geklärt, können Sie anfangen, Ihr eigenes ISR-Szenario zu erstellen. Bevor Sie jedoch beginnen, das Customizing des ISR direkt zu erstellen, können Sie alternativ den ISR-Wizard verwenden. Dieser ist besonders hilfreich, wenn Sie das erste Mal mit dem ISR-Framework arbeiten.

ISR-Wizard

Mit dem ISR-Wizard kann ein Szenario Schritt für Schritt erstellt werden. Im IMG (Implementation Guide; Transaktion SPRO • **Projektbearbeitung** oder im SAP-Menü unter **Werkzeuge • Customizing • IMG**) des SAP-ERP-Systems finden Sie den ISR-Wizard über den Menüpfad **Anwendungsübergreifende Komponenten • Internet-/Intranetservices • Interne Serviceanfrage • Definition von Szenarios • ISR Wizard**.

- Die IMG-Aktivität **Szenario mit ISR Wizard definieren** (Transaktion QISRCONF) dient der Definition eines Szenarios.
- Die IMG-Aktivität **Übersicht der im ISR Wizard angelegten Szenarios** (Transaktion QISRLIST) zeigt Ihnen alle aktuellen, mit dem ISR-Wizard erstellten Szenarien an.

Innerhalb der Wizards begegnen Sie im Wesentlichen den gleichen Fragestellungen, wie in [Tabelle 1](#) beschrieben. Dabei wird versucht, die Fragen möglichst aus einem betriebswirtschaftlichen Blickwinkel zu formulieren und Fragen zu technischen Details soweit wie möglich zu vermeiden. Das heißt, tiefgehende technische Kenntnisse über die Komponenten des ISR (Meldung, Workflow, Adobe-Formular etc.) sind nicht erforderlich. Die Antworten aus den Teil-Wizards »übersetzt« der ISR-Wizard in technische Einstellungen, aus denen später die Customizing-Einstellungen zum eigentlichen ISR-Szenario generiert werden.

Teil-Wizards

Der ISR-Wizard besteht genau genommen aus sieben einzelnen Teil-Wizards zu unterschiedlichen Themengebieten; alle Schritte eines Teil-Wizards werden mit Status angezeigt. [Abbildung 13](#) zeigt zum Beispiel den aktuellen Stand des ISR-Wizards mit allen Teil-Wizards und deren Status für ein Szenario, das erst teilweise fertiggestellt ist.

Die einzelnen Teil-Wizards bauen zum Teil aufeinander auf und sind demzufolge voneinander abhängig. Ein abhängiger Teil-Wizard ist erst aufrufbar, wenn der oder die benötigten, vorhergehenden Teil-Wizard(s) fertiggestellt sind. Bis alle vorhergehenden, notwendigen Teil-Wizards abgearbeitet sind, ist der abhängige Teil-Wizard nicht aufrufbar, was entsprechend gekennzeichnet ist. Diese Abhängigkeiten werden innerhalb eines jeden Teil-Wizards erläutert. Abgesehen davon, können die Wizards in beliebiger Reihenfolge durchgeführt werden.

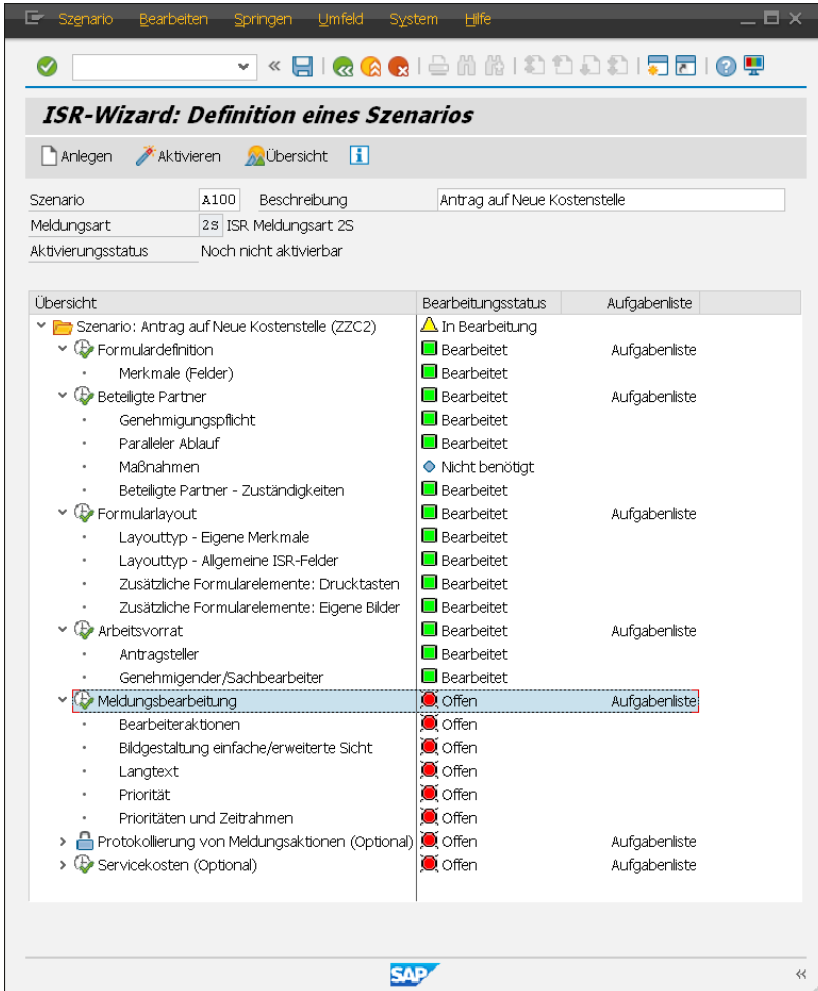


Abbildung 13 ISR-Wizard mit Statusinformationen

Nachbearbeitung eines Teil-Wizards

Wenn Sie einen Teil-Wizard nachbearbeiten – was jederzeit möglich ist –, müssen Sie auch die nachfolgenden, abhängigen Teil-Wizards nochmals durchgehen, um eventuell erforderliche Anpassungen durchzuführen.

Folgende Teil-Wizards stehen zur Verfügung, wobei ein Wizard einen bestimmten Themenschwerpunkt des ISR abdeckt. Die Customizing- bzw. Formulareinstellun-

gen, die durch jeden Teil-Wizard erzeugt werden, sind in den Abschnitten »Customizing und Programmierung eines Szenarios« und »Antragsformular als Adobe-Formular« detailliert erläutert.

- **Formulardefinition** (unabhängig von anderen Teil-Wizards ausführbar)
Hier definieren Sie die Merkmale des Formulars.
- **Beteiligte Partner** (unabhängig von anderen Teil-Wizards ausführbar)
Hier bestimmen Sie die am Geschäftsprozess der Serviceanfrage beteiligten Rollen und Partner und ordnen Benutzer- oder Workflow-Regeln zur Bestimmung von Bearbeitern zur Laufzeit zu.
- **Formularlayout** (abhängig vom ersten Wizard)
Hier bestimmen Sie das grobe Layout des Formulars. Sie legen z. B. die Positionierung der Merkmale fest, und zusätzlich können Sie Felder aus dem Pool **Allgemeine ISR Felder** auswählen und bestimmen, für welche Rolle das Merkmal eingabebereit ist.
- **Arbeitsvorrat** (abhängig vom zweiten Wizard)
Hier bestimmen Sie, wie die Bearbeiter ihre Serviceanfragen finden, und konfigurieren den ISR-Statusüberblick.
- **Meldungsbearbeitung** (unabhängig von anderen Teil-Wizards ausführbar)
Hier bestimmen Sie die Funktionalität der Meldungstransaktionen (zum Beispiel IQS22) zur Bearbeitung der Serviceanfrage, wie zum Beispiel Aktivitäten für den Sachbearbeiter.
- **Protokollierung von Meldungsaktionen**
(abhängig vom zweiten und fünften Wizard; optional)
Hier legen Sie fest, ob und wie Aktivitäten der Meldungsbearbeitung in der Meldung dokumentiert werden. Die Einstellungen sind für Auswertungen im Meldungsberichtswesen von Bedeutung.
- **Servicekosten**
(unabhängig von anderen Teil-Wizards ausführbar; optional)
Hier legen Sie fest, wie die Kosten einer Serviceanfrage erfasst werden.

Die beiden letzten Teil-Wizards sind optional, das heißt, deren Ausführung ist für die Aktivierung des Szenarios nicht erforderlich. Wurde ein Teil-Wizard noch nicht bearbeitet, hat er den Status **Offen**. Wurde ein Teil-Wizard partiell bearbeitet, aber noch nicht fertiggestellt, hat er den Status **In Bearbeitung**, nach dem Fertigstellen erhält er den Status **Bearbeitet**. Wenn alle Teil-Wizards, die nicht optional sind, vollständig bearbeitet wurden, ist das Szenario aktivierbar.

Aktivierung

Die aktuellen Einstellungen der Teil-Wizards werden zunächst nur im ISR-Wizard zwischengespeichert, auch wenn Sie nicht alle Teil-Wizards fertiggestellt haben. Sie können jeden Teil-Wizard beliebig verändern und nachbearbeiten, solange das Szenario nicht aktiviert ist. Erst wenn Sie das gesamte Szenario im ISR-Wizard aktivieren, werden die Einstellungen in die entsprechenden Customizing-Tabellen als tatsächlich verwendbares Szenario übertragen.

Die Generierung der Customizing-Einstellungen zum Szenario kann jedoch erst nach Durchführung aller nicht optionalen Teil-Wizards erfolgen, dazu muss die Funktion **Aktivieren** ausgeführt werden. Erst danach wird das eigentliche ISR-Szenario unter dem Namen des im ISR-Wizard definierten Szenarios erstellt und das Customizing in die betreffenden Customizing-Tabellen übertragen. Zusätzlich wird eine Meldungsart mit zugehörigen Customizing-Einstellungen generiert. Ob das Szenario aktivierbar ist, erkennen Sie am **Aktivierungsstatus**, der **Noch nicht aktivierbar** oder **Aktivierbar** anzeigt.

Mit dieser Aktivierung wird außerdem ein funktionsfähiges interaktives Adobe-Formular mit allen benötigten Eingabefeldern und allen erforderlichen technischen Eigenschaften – Schnittstelle, Kontext, Datenansicht, Skript, Felddbindung etc. – generiert (siehe [Abbildung 14](#)). Das generierte Formular kann anschließend beispielsweise für die Layoutgestaltung oder die nachträgliche Erweiterung um zusätzliche Felder überarbeitet werden (siehe Abschnitt »[Antragsformular als Adobe-Formular](#)«).

Der Name des Formulars bzw. der Schnittstelle setzt sich wie folgt zusammen (xxxx steht für den Namen des Szenarios):

- Formular: Z_ISR_FORM_xxxx
- Schnittstelle: Z_ISR_IF_xxxx

Die Customizing-Einstellungen und das generierte Adobe-Formular werden in den von Ihnen bereitgestellten Aufträgen gesammelt und können somit in andere Systeme transportiert werden. Zur Aktivierung benötigen Sie je einen Customizing- und einen Workbench-Auftrag (für die SAP Interactive Forms by Adobe). Sie werden außerdem nach der Paketuordnung für das Adobe-Formular und dessen Schnittstelle gefragt und müssen die zugehörigen Workbench-Objekte per Pop-up-Fenster aktivieren.

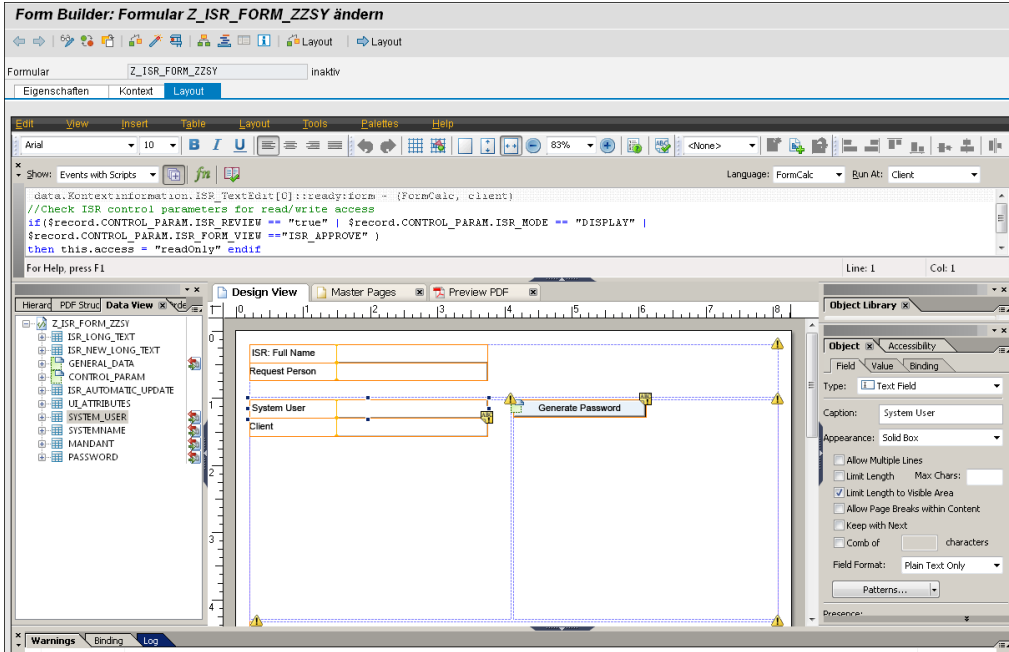


Abbildung 14 Generiertes Adobe-Formular mit Datenansicht, Layout der Felder, Skript

Aufgabenliste

Nach der Aktivierung werden im weiteren Verlauf der Implementierung eines neuen ISR-Szenarios sicherlich Änderungen am Customizing erforderlich sein. Aus diesem Grund wird mit der **Aufgabenliste** eine Zusammenfassung aller generierten Customizing-Einstellungen anhand eines Adobe-PDF-Druckformulars zur Verfügung gestellt (siehe [Abbildung 15](#), das Beispiel bezieht sich auf die Einstellungen des Beispielszenarios ZZSY). Darin ist unter anderem beschrieben, welche Steuerungsfunktion der betreffende generierte Customizing-Eintrag übernimmt und wo er im IMG zu finden ist.

Diese Liste dient auch der Dokumentation der Teil-Wizards, um die ursprünglichen Einträge in den Wizards nachvollziehen zu können. Denn nach der Aktivierung können Sie das Szenario nicht mehr im Wizard bearbeiten, sondern nur noch in den eigentlichen IMG-Aktivitäten zum ISR. Die Aufgabenliste zeigt den aktuellen Bearbeitungsstatus eines teilweise fertiggestellten und nicht aktivierten Szenarios an.

Vorschau Druckausgabe

SAP

ISR-Wizard Aufgabenliste

Szenario: ZZSY Beschreibung: Antrag für Systembenut
 Angelegt von: Dr. Stephan Rehmann Angelegt am: 18.04.2015
 Zuletzt geändert von: Dr. Stephan Rehmann Zuletzt geändert am: 18.04.2015
 Aktiviert: Nein
 Gesamtstatus: Bearbeitet
 Textfeldmeldungsart: 2U

In dieser Liste werden alle Aktivitäten angezeigt, die noch bearbeitet werden müssen oder bereits durch das Ausführen der Wizards vorgenommen wurden. Welche Aktivitäten noch zu bearbeiten sind bzw. welche Aktivitäten bereits durch den Wizard bearbeitet wurde ist an der Priorität der einzelnen Aktivitäten zu erkennen. Es gibt die folgenden drei Prioritäten:

- **Muss-Aktivität:** Aktivitäten, die diese Priorität haben, müssen noch ausgeführt werden. Es war nicht möglich, diese Aktivität durch einen Wizard auszuführen
- **Kann-Aktivität:** Aktivitäten, die ebenfalls nicht durch einen Wizard ausgeführt werden konnten, aber auch nicht unbedingt notwendig sind.
- **Info:** Diese Aktivitäten wurden durch den Wizard erfolgreich ausgeführt.
- **Offene Aktivität:** Diese Aktivität wird noch von Wizard ausgeführt.

In diesem Abschnitt nehmen Sie Einstellungen zur Formulardefinition vor.

Status: Bearbeitet

Formulardefinition				
Priorität	Beschreibung	Wert	IMG-Path	Aufgaben
Info				In diesem Abschnitt wird das Formular mit den wesentlichen Formularelementen definiert (obligatorisch).
Info	Merkmale (Felder)	Liste der Merkmale	Anwendungsübergreifende Komponenten -> Internet/ Intranetservices -> Interne Serviceanfrage (SR) -> Definition von Szenarios -> Szenarios definieren (Transaction QISCSBNAHQ) Unter: Grunddaten -> Version -> Merkmale	Die definierten Merkmale (Felder) sollen alle für den Prozess der Serviceanfrage des betreffenden Szenarios notwendigen Daten beinhalten. Alle oder nur ein Teil der Merkmale können auf dem Formular sichtbar bzw. eingabebereit sein. Jedes Merkmal kann mit oder ohne Bezug auf ein Datenelement oder eine Struktur aus dem Data Dictionary definiert werden.
Info	Merkmale (Felder)	SYSTEM_US		
Info	Merkmale (Felder)	SYSTEMNA		
Info	Merkmale (Felder)	ME		
Info	Merkmale (Felder)	MANDANT		
Info	Merkmale (Felder)	PASSWORD		

In diesem Abschnitt nehmen Sie Einstellungen zur Bestimmung der beteiligten Partner vor.

Status: Bearbeitet

Beteiligte Partner				
Priorität	Beschreibung	Wert	IMG-Path	Aufgaben
Info				In diesem Abschnitt werden die in der Serviceanfrage beteiligten (Partner) definiert (obligatorisch).
Info	Genehmigungspflicht	Mit Genehmigu	Anwendungsübergreifende Komponenten -> Internet/ Intranetservices -> Interne Serviceanfrage (SR) ->	Legt fest, ob in der Serviceanfrage ein Genehmigungsverfahren vorgesehen ist. Die Anzahl der

Page 1 of 5

Abbildung 15 Aufgabenliste des ISR-Wizards

In der Aufgabenliste sind außerdem weitere notwendige oder optionale Customizing-Aufgaben beschrieben, die nicht über den Wizard durchgeführt werden können, sondern nach der Aktivierung des Szenarios manuell ausgeführt werden. Beispielsweise muss die Zuordnung des Nummernkreises der Meldungsart im gewünschten Zielsystem in jedem Mandanten und für jede Meldungsart separat durchgeführt werden. Diese Einstellung kann nicht transportiert werden. Die Bestandteile des ISR-Szenarios, die programmiert werden müssen, wie zum Beispiel die Implementierung des BAdI QISR1, werden ebenfalls nicht mit dem ISR-Wizard generiert.

ISR-Wizard-Übersicht

Die **Szenarioliste** des ISR-Wizards bietet eine Übersicht über alle im System per ISR-Wizard angelegten Szenarien (siehe [Abbildung 16](#)), die zwischen aktivierten und offenen Szenarien unterscheidet. Sie können in dieser Übersicht neue Szenarien anlegen und die offenen Szenarien ändern oder löschen.

Liste von Szenarios	Sze...	Status	Angeleg...	Angelegt am	Aufgabenliste	ISR-Grundcustomizing	ISR Bad...
Offene Szenarios							
• Test: Adressänderung	A100	In Bearbeitung	Dr. Steph...	25.08.2008	Aufgabenliste		
• Antrag auf Urlaub	ZH12	Offen	Dr. Steph...	25.08.2008	Aufgabenliste		
Aktivierte Szenarios							
• Antrag Neue Kostenstelle (Version	Z4AA	Aktiviert	Dr. Steph...	17.04.2015	Aufgabenliste	ISR-Grundcustomizing	..BAdI-Imf
• Kostenstellenwechsel beantragen	Z234	Aktiviert	Dr. Steph...	25.08.2008	Aufgabenliste	ISR-Grundcustomizing	..BAdI-Imf
• Antrag auf Neue Kostenstelle	ZZC1	Aktiviert	Dr. Steph...	25.08.2008	Aufgabenliste	ISR-Grundcustomizing	..BAdI-Imf
• Antrag für Systembenutzer	ZZSY	Aktiviert	Dr. Steph...	25.08.2008	Aufgabenliste	ISR-Grundcustomizing	..BAdI-Imf

Abbildung 16 Liste mit offenen und aktivierten Szenarien des ISR-Wizards

Da ein aktiviertes Szenario nicht mehr im Wizard geändert werden kann, ist der ISR-Wizard für aktivierte Szenarien nicht mehr aufrufbar. Für ein aktiviertes Szenario steht jedoch die **Aufgabenliste** zur Verfügung. Außerdem können Sie direkt zum BAdI Builder für das BAdI QISR1 (siehe Abschnitt »[Formularablauflogik](#)«) und in das ISR-Grund-Customizing, IMG-Aktivität **Szenarios definieren**, abspringen (siehe Abschnitt »[Customizing und Programmierung eines Szenarios](#)«). Dort können Sie sehen, welche Einstellungen das System beim Aktivieren des Szenarios in die Customizing-Tabellen übertragen hat und diese gegebenenfalls ändern.



Aktiviertes Szenario

Nur aktivierte Szenarien sind im ISR-Grund-Customizing (Transaktion QISRSCENARIO) zu finden und können mit dem ISR-Wizard nicht mehr geändert oder gelöscht werden. Sie sind jedoch weiterhin in der ISR-Wizard-Übersicht (Transaktion QISRLIST) aufgelistet.

Definition eines Szenarios

Um ein Szenario mit dem ISR-Wizard zu generieren, können Sie sich an den folgenden Arbeitsschritten orientieren. Das Beispielszenario zum Antrag für Systembenutzer (ZZSY) soll dies veranschaulichen:

1. Legen Sie das Szenario ZZSY mit Transaktion QISRCONF an. Teil-Wizards mit dem Uhrsymbol sind ausführbar, Teil-Wizards mit dem Schlosssymbol sind noch nicht ausführbar. Die Nummer der **Meldungsart** wird automatisch ermittelt (siehe [Abbildung 17](#)).

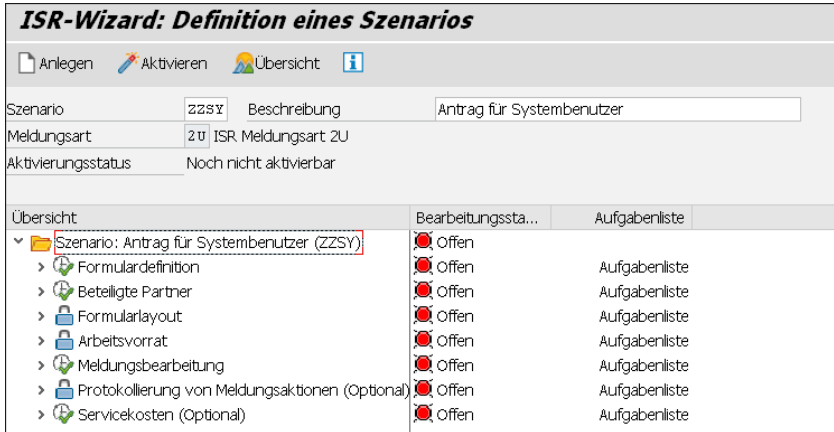


Abbildung 17 ISR-Wizard für ein neues Szenario

2. Bearbeiten Sie alle benötigten Teil-Wizards, bis das gesamte Szenario den Status **Bearbeitet** erhält. Alle Teil-Wizards sind analog aufgebaut und bieten weitgehend Vorschlagswerte für ein einfaches Szenario an. Das bedeutet, Sie können ein Szenario im Prinzip in kürzester Zeit erstellen, indem Sie die Vorschlagswerte in den Teil-Wizards akzeptieren.

Der erste Schritt enthält einleitende Erläuterungen zum Themengebiet. In [Abbildung 18](#) wird die Formulardefinition erläutert.

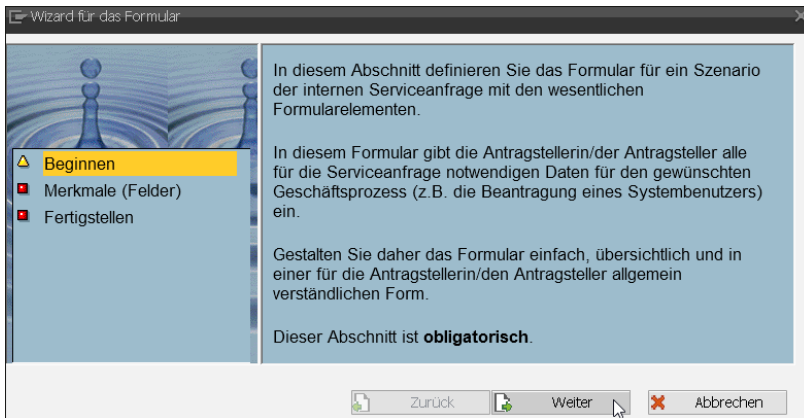


Abbildung 18 Schritt 1 des Teil-Wizards – Formulardefinition

3. In den nächsten Schritten des Teil-Wizards werden Einstellungen festgelegt. Im Schritt **Merkmale (Felder)** können die Merkmale des Szenarios eingetragen werden (siehe [Abbildung 19](#)). Wenn Sie den Bezug eines Merkmals zum Data Dictionary kennen, tragen Sie diesen hier ein. Im Beispiel werden der **Systembenutzer**, der **Systemname**, der **Mandant** und das **Passwort** als Merkmale definiert.

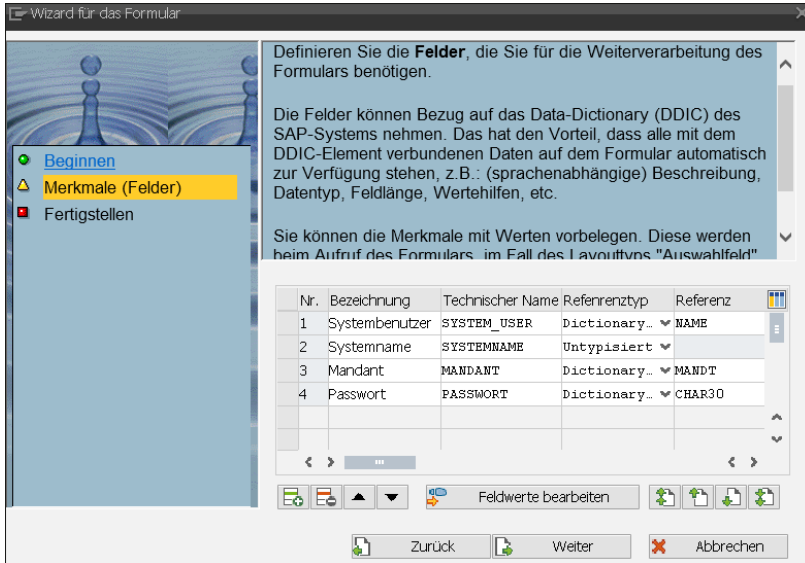


Abbildung 19 Schritt 2 des Teil-Wizards – Formulardefinition

4. Der letzte Schritt dient immer dem **Fertigstellen** des Teil-Wizards. Hier werden keine gesonderten Einstellungen mehr vorgenommen.
5. Die anderen Teil-Wizards, die nach dem gleichen Schema aufgebaut sind, führen Sie analog dazu aus. Nachdem alle relevanten Teil-Wizards durchgeführt wurden, klicken Sie auf die Schaltfläche **Aktivieren** (siehe [Abbildung 20](#)).

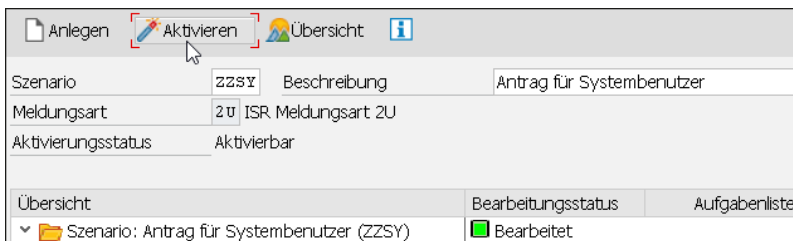


Abbildung 20 Aktivierung des ISR-Szenarios

6. Führen Sie die in der Aufgabenliste beschriebenen zusätzlichen Aufgaben aus.

Customizing und Programmierung eines Szenarios

Sie haben nun das Grundgerüst Ihres ISR-Szenarios mithilfe des ISR-Wizards erstellt. Zur Nachbearbeitung und Verfeinerung des *Szenario-Customizings* und für die szenariospezifische Programmierung der verfügbaren BADIs bedienen Sie sich nun der entsprechenden, im Folgenden beschriebenen IMG-Aktivitäten.

Natürlich können Sie ein ISR-Szenario auch ohne Wizard erstellen. Verfügen Sie bereits über Expertenwissen zum ISR oder wollen Sie ein bestehendes eigenes oder ein ausgeliefertes ISR-Szenario als Vorlage verwenden, bedienen Sie sich direkt der folgenden IMG-Aktivitäten.

Allgemeine Angaben

Die IMG-Aktivität **Szenarios definieren** (Transaktion QISRSCENARIO) ist der zentrale Einstieg in das Customizing eines ISR-Szenarios. Das Customizing des Szenarios wird häufig auch als ISR-Grund-Customizing bezeichnet. Sie finden das Customizing über den Menüpfad **Anwendungsübergreifende Komponenten • Internet-/Intranetservices • Interne Serviceanfrage • Definition von Szenarios**.

Dort sehen Sie eine Liste von Szenarien. Markieren Sie ein Szenario, gehen Sie auf **Grunddaten Ändern**, und Sie gelangen in das ISR-Grund-Customizing. Hier bearbeiten Sie die Einstellungen bezüglich der folgenden Themenbereiche (siehe [Abbildung 21](#)):

Sicht "Version" ändern: Detail

Neue Einträge

Dialogstruktur

- Grunddaten
 - Version**
 - Merkmale
 - Maßnahmen

Sie befinden sich im Szenario 225Y

Die Version lautet 0

Diese Szenarioversion ist aktiv.

Gültig ab 01.07.2015 bis 31.12.2020

Das Szenario enthält folgende allgemeine Angaben:

Beschreibung Antrag für Systembenutzer

Hier erfassen Sie einen Langtext

Applikation Standard Applikation

Meldungsart 2U ISR Meldungsart 2U

Bei dieser Meldungsart ist eine Genehmigung vorgesehen.

Anspruchspartner bei Fragen zum Formular ist REHMANN

Der Ansprechpartner hat folgende Partnerrolle Ansprechpartner ISR

Hilfen zur Bearbeitung der Anfrage definieren Sie in der Aktivitätenleiste

Aktivitäten zur automatischen Verbuchung pflegen Sie in Automat. Verbuchung

Das Szenario enthält folgende Angaben zur Erfassung der Anfrage:

Erfassungsart im Web Erfassung mit Adobe PDF

Formular 2_ISR_FORM_225Y

Schnittstelle 2_ISR_IF_225Y

Um eigene Ablauflogik zu definieren, wählen Sie

neues Business Add-In Business Add-In definieren

Business Add-In aus Szenario

Kosten

Das Szenario ist kostenverursachend.

Das Szenario enthält folgende Angaben zur Bearbeitung des Antrags:

Zusätzliche Daten zum Szenario

Abbildung 21 ISR-Grund-Customizing

■ **Allgemeine Angaben**

Die allgemeinen Angaben umfassen die Szenariobeschreibung, Version und Applikation (siehe Abschnitt »Applikation und Version«).

■ **Meldungsart**

Die Meldungsart umfasst das Customizing zu den Meldungstransaktionen des (optionalen) Genehmigenden und des Sachbearbeiters sowie deren Funktionen. Mit der Meldungsart legen Sie die Strukturprozessschritte des Service fest (siehe folgender Abschnitt »Meldungsart«).

■ **Erfassungsart**

Hier definieren Sie das interaktive Adobe-Formular als Erfassungsoberfläche für den Antragsteller (siehe Abschnitt »Erfassungsart«).

■ **Merkmale**

Hier definieren Sie die Eingabe- und Anzeigefelder des Formulars sowie weitere Felder für die Bearbeitung der Serviceanfrage (siehe Abschnitt »Merkmale«).

■ **Aktivitäten**

Hier definieren Sie manuell bzw. automatisch ausführbare Aktivitäten für die am Prozess beteiligten Rollen wie Sachbearbeiter und Genehmigender (siehe Abschnitte »Aktivitäten und automatische Verbuchungen in der Meldung« und »Aktivitäten für den Sachbearbeiter vom Formular aus«).

■ **Bearbeiterfindung**

Falls Sie den SAP Business Workflow einsetzen wollen, können Sie den oder die zuständigen Bearbeiter bzw. Regeln zur systemseitigen Bestimmung eines einzelnen oder mehrerer Bearbeiter festlegen (siehe Abschnitt »Bearbeiterfindung (BAdI QISR3) und Workflow«).

■ **Kostenerfassung**

Sie können einstellen, ob die internen Kosten der Serviceanfrage zusätzlich im Controlling des SAP-ERP-Systems ermittelt werden sollen (siehe Abschnitt »Servicekosten und BAdI QISR2«).

■ **Formularablauflogik**

Hier legen Sie fest, wie die Formularablauflogik per BAdI QISR1 beeinflusst werden soll. Die programmierbare Formularablauflogik ist eine der zentralen Funktionen des ISR und wird deshalb eigens im Abschnitt »Formularablauflogik«, ausführlich behandelt.

Alle Customizing-Einstellungen und Programmierungen zu einem Antragservice werden unter dem zentralen Begriff eines *Szenarios* zusammengefasst.



ISR-Szenario

Das technische Kürzel des ISR-Szenarios ist ein vierstelliges alphanumerisches Zeichen, hier ZZSY. Der Kundennamensraum wird durch das erste Zeichen festgelegt

und umfasst alle Buchstaben außer S und alle Zahlen von 0 bis 9. Die von SAP ausgelieferten Szenarien beginnen mit S. Das technische Schlüsselfeld SCENARIO dient der Identifizierung des ISR-Szenarios im gesamten ISR-Framework.

Applikation und Version

Das Szenario kann aus mehreren Versionen bestehen, von denen immer nur eine aktiv ist. Die Versionsnummer wird vom System, beginnend bei 0, automatisch ermittelt und für jede Szenarioversion kann ein Gültigkeitszeitraum festgelegt werden. Dieser Gültigkeitszeitraum einer Version muss eindeutig sein und darf sich nicht mit dem Gültigkeitszeitraum einer anderen Version überschneiden (siehe [Abbildung 22](#)). Ausgelieferte Szenarien haben ausschließlich die Version 0.

Sicht "Version" ändern: Übersicht

69 Neue Einträge

Dialogstruktur

- Grunddaten
 - Version
 - Merkmale
 - Maßnahmen

Version					
Szenario	Version	Aktiv	Gültig von	Gültig bis	
ZZSY	0	<input checked="" type="checkbox"/>		31.12.2020	
ZZSY	1	<input type="checkbox"/>	01.01.2020	31.12.2999	

Abbildung 22 Zwei Versionen eines Szenarios

Beim Starten des Formulars durch den Antragsteller kann durch den Parameter VALIDITY_DATE ein Gültigkeitsdatum mitgegeben werden, anhand dessen die gültige Version ermittelt wird. Alternativ kann die Version direkt über den Parameter VERSION ausgewählt werden. Das funktioniert natürlich nur, wenn diese Version zum aktuellen Zeitpunkt (lokale Zeit) gültig ist. Details zur Parameterübergabe finden Sie im Abschnitt »[Web-Dynpro-ABAP-Component und ihre Anwendung](#)« beschrieben.

Die Applikation dient der technischen Gruppierung mehrerer Szenarien und wird ausschließlich von bestimmten SAP-ERP-Anwendungen benutzt, um deren festgelegte Eigenschaften in Bezug auf den ISR steuern zu können. Eine Applikation kann nicht neu definiert werden.

Derzeit stellt SAP die folgenden Applikationen zur Verfügung:

- **Standardapplikation** (kein Eintrag im Feld **Applikation**)
Dies ist die Applikation für Standardszenarien der internen Serviceanfrage.
- **HCM-Prozesse und Formulare** (Eintrag H im Feld **Applikation**)
Ein Szenario mit dieser Applikation können Sie mit einem Formularszenario der HCM Administrative Services verknüpfen. Weitere Informationen finden Sie im

Customizing unter **Personalmanagement • HR Administrative Services • Formular-/Prozess-Konfiguration • Formularkonfiguration**.

■ **Einfache interaktive Formulare** (Eintrag S im Feld **Applikation**)

Dies ist die Applikation für einfache Formulare, die in der Anwendung Express Planning eingebunden werden können. Weitere Informationen finden Sie in der SAP-Bibliothek unter **Anwendungsübergreifende Komponenten • Express Planning • Planungsservices • Planungsservices mit enger Kopplung • Adobe-Formular-Service**.

■ **Manuelle Korrespondenz** (Eintrag MC im Feld **Applikation**)

Dies ist die Anwendung aus dem Public Sector.

■ **Steuer- und Einnahmenverwaltung – TRM** (Eintrag T im Feld **Applikation**)

Dies ist die Anwendung aus dem Public Sector zur Bearbeitung von Steuerformularen.

Die Applikation ist nur dann einzugeben, falls das Szenario in einer der genannten, speziellen SAP-ERP-Anwendungen eingebunden werden soll. Beachten Sie, dass ein solches Szenario ausschließlich im Rahmen dieser Applikationen funktionieren kann, da durch die Wahl einer Applikation bestimmte Funktionen wegfallen oder hinzukommen.

Wollen Sie die Applikation nutzen, sollten Sie sich zunächst über die gesonderte Dokumentation der jeweiligen SAP-ERP-Anwendung informieren. Wenn Sie ein Antragsszenario ohne Verbindung zu einer Applikation wollen, übergehen Sie diese Einstellung (siehe [Abbildung 23](#)). In der Regel ist die Applikation für beliebige, kundeneigene Szenarien nicht relevant. Außerdem wird die Szenariobeschreibung hier eingetragen.

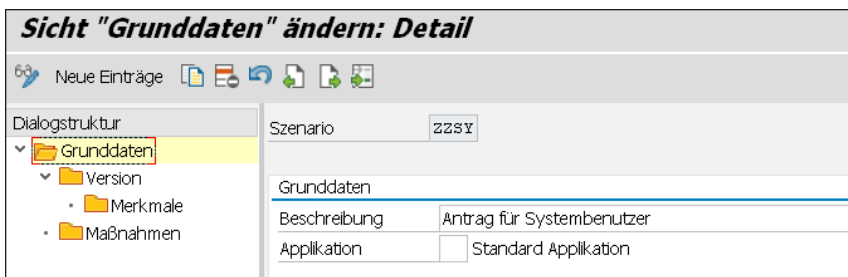


Abbildung 23 Einstieg in das ISR-Grund-Customizing

Meldungsart

Die Meldungsart wird dem Szenario zugeordnet (siehe [Abbildung 22](#)) und umfasst selbst wiederum das Customizing zu verschiedenen Bestandteilen der Meldung. Das Customizing zur Meldungsart definieren Sie im IMG unter **Anwendungsüber-**

greifende Komponenten • Internet-/Intranetservices • Interne Serviceanfrage • Definition von Szenarios • Vorbereiten der allgemeinen Meldung.

Für jeden der im Abschnitt »Gruppierung der ausgelieferten Szenarien« genannten Themenblöcke, der mehrere Szenarien umfasst, wird von SAP eine Meldungsart ausgeliefert. Genügen Ihnen die Einstellungen einer dieser Meldungsarten nicht, sollten Sie eine eigene Meldungsart definieren.

Folgende Einstellungen sind mit der Meldungsart verknüpft:

- Szenario grundsätzlich genehmigungspflichtig oder nicht
- die Definition des Layouts der Meldungstransaktion, die dem Sachbearbeiter zur Bearbeitung der Serviceanfrage dient
- die verfügbaren Aktivitäten

Für einfache Szenarien können Sie die Meldungsart O1 (genehmigungspflichtig) oder O2 (nicht genehmigungspflichtig) verwenden. Falls Sie eine eigene Meldungsart benötigen, empfiehlt es sich, Meldungsart O1 oder O2 zu kopieren und Ihre Szenarien mit einer eigenen Meldungsart aufzubauen. Sie finden die IMG-Aktivität unter **Anwendungsübergreifende Komponenten • Internet-/Intranetservices • Interne Serviceanfrage • Definition von Szenarios • Vorbereiten der allgemeinen Meldung • Meldungseröffnung • Meldungsart • Meldungsart definieren**. Falls die Meldungsarten O1 oder O2 nicht in Ihrem Mandanten enthalten sind, können Sie diese Einstellungen mit der Transaktion QISR_SM29 aus dem Mandanten 000 übernehmen.

In der Regel sollten Sie eine eigene Meldungsart definieren und als Vorlage eine ausgelieferte Meldungsart verwenden. Sollten Sie den ISR-Wizard zur Generierung eines ISR-Szenarios benutzt haben, wurde bereits eine neue Meldungsart generiert.

Nummernkreiszuordnung für die Meldungsart

Sie müssen die Meldungsart in jedem System einem Nummernkreis zuordnen. Dieses Customizing kann nicht transportiert werden. Gehen Sie dazu in die Transaktion IW20 oder in den IMG unter **Anwendungsübergreifende Komponenten • Internet-/Intranetservices • Interne Serviceanfrage • Definition von Szenarios • Vorbereiten der allgemeinen Meldung • Meldungseröffnung • Meldungsart • Nummernkreise festlegen**.

Sollte kein Nummernkreis zugeordnet sein, erhalten Sie beim Abschicken des Formulars die Fehlermeldung »Bei der Bearbeitung der Meldung ist ein Fehler aufgetreten« oder »Unerwarteter Fehler während der Prozessierung ...«.



Erfassungsart

Über die Erfassungsart legen Sie fest, mit welcher Art von Eingabemaske der Antragsteller die Serviceanfrage erfassen soll (siehe [Abbildung 21](#)). Aufgrund der Tatsache, dass der ISR bereits mit Release SAP R/3 4.6C erstmals ausgeliefert wurde, stehen grundsätzlich mehrere Erfassungsarten zur Verfügung. Tatsächlich ist seit SAP ERP 5.0 die **Erfassung mit Adobe PDF** die von SAP empfohlene Erfassungsart. Daher sollen die anderen Erfassungsarten hier nur aufgezählt, aber nicht weiter vertieft werden:

- **reine Texterfassung**
- **Erfassung mit ITS-Service**
- **Erfassung mit Meldungstransaktion**
- **Erfassung mit JSP-iView**
- **Erfassung mit Business Server Pages**

Nach der Auswahl von **Erfassung mit Adobe PDF** kann der Name des interaktiven Formulars eingetragen werden, und die zugehörige Schnittstelle des Formulars wird zu Informationszwecken angezeigt. Von dieser Stelle aus kann das Adobe-Formular neu generiert, geändert, angezeigt oder getestet werden (siehe Abschnitt [»Antragsformular als Adobe-Formular«](#)). Diese Funktionen werden über Schaltflächen neben dem Feld für das Formular zur Verfügung gestellt, so erreichen Sie durch das Drücken bestimmter Schaltflächen den Form Builder. Wichtig ist, dass das Formular einige ISR-spezifische Eigenschaften aufweisen muss, damit es im ISR-Framework verwendbar ist. Dies wird jedoch sichergestellt, wenn Sie das Formular mithilfe der Funktionen des ISR-Grund-Customizings bearbeiten.

Verwenden Sie für jedes Szenario ein eigenes Formular mit eigener Schnittstelle. Bei nachträglichen Änderungen könnten das Formular bzw. die Schnittstelle sonst für andere Szenarien unbrauchbar werden. Die Formulardefinition wird im Abschnitt [»Antragsformular als Adobe-Formular«](#) ausführlich erläutert.

Merkmale

Ein wesentlicher Bestandteil der Definition einer Serviceanfrage ist die Festlegung der Merkmale (Felder), die für die Serviceanfrage benötigt werden. An erster Stelle sind die erforderlichen Eingabe- oder Ausgabefelder auf dem Antragsformular zu nennen, die im Szenario-Customizing als Merkmale definiert werden. Im Prinzip müssen Sie jedes Feld, das Daten trägt, als Merkmal definieren. Dies können beispielsweise Felder für die Eingabe durch den Antragsteller sein.

Das bezieht jedoch auch Felder mit ein, die nur zur Anzeige von Daten aus dem Backend erforderlich sind oder die erst im Laufe der Servicebearbeitung angereichert werden. Felder, die nicht allen beteiligten Rollen zur Verfügung stehen sollen, die jedoch

für die weitere Bearbeitung der Serviceanfrage notwendig sind, müssen ebenso definiert werden. Ein Beispiel ist, dass auf dem Formular einer Anfrage für einen Systembenutzer das technische Feld für das initiale Passwort nicht benötigt wird. Der Wert dieses Feldes wird erst vom Sachbearbeiter generiert.

Die Merkmale des ISR-Customizings können wahlweise Bezug zum Data Dictionary haben oder folgendermaßen frei definiert werden (siehe [Abbildung 24](#)):

- Merkmale mit Referenz auf ein Datenelement
- Merkmale mit Referenz auf eine Struktur
- untypisierte Merkmale; diese haben den impliziten Typ CHAR250 ohne Eingabe- bzw. Ausgabekonvertierung und damit eine maximale Längen von 250 Zeichen.

Sicht "Merkmale" ändern: Übersicht

Neue Einträge

Dialogstruktur

- Grunddaten
 - Version
 - Merkmale**
 - Maßnahmen

N...	Merkmalsname	Merkmaltyp	Referenz	Bezeichnung
1	SYSTEM_USER	Dictionary-Typ	UNAME	Benutzername
2	SYSTEMNAME	Untypisiert		Systemname
3	CLIENT	Dictionary-Typ	MANDT	Mandant
4	PASSWORD	Dictionary-Typ	CHAR30	Passwort

Abbildung 24 Szenario-Customizing mit der Liste der Merkmale

Folgende Rahmenbedingungen gelten für die Definition von Merkmalen:

- Merkmalsnamen bzw. Strukturfeldnamen müssen innerhalb des Szenarios eindeutig sein.
- Die Länge der Merkmalsnamen bzw. Strukturfeldnamen sollte maximal 28 Zeichen betragen.
- Die Datenlänge der Merkmalsnamen bzw. Strukturfeldnamen ist generell auf 250 Zeichen begrenzt.
- Merkmalsnamen bzw. Strukturfeldnamen dürfen nicht mit »ISR_« beginnen.
- Die Merkmalsbezeichnung kann höchstens 40 Zeichen lang sein.

Neben diesen szenariospezifisch definierten Merkmalen steht Ihnen eine ganze Reihe von Standardmerkmalen zur Verfügung, die automatisch initialisiert und auf dem Antragsformular angezeigt werden können, darunter die allgemeinen ISR-Felder (siehe Abschnitt [»Aufbau der Schnittstelle und des Formularkontextes«](#)).

Allgemeine ISR-Felder

Das ISR-Framework stellt für jedes Szenario und Formular einen Pool sogenannter *allgemeiner Felder* zur Verfügung, die zur Laufzeit automatisch ermittelt werden. Die



Liste beinhaltet hauptsächlich Adressmerkmale der beteiligten Bearbeiter und technische Werte wie Szenario, Erfassungszeitpunkt der Meldung etc. Die vollständige Liste entnehmen Sie der DDIC-Struktur QISRSGENERAL_PARAM.

Mit der Merkmalsdefinition können Sie auch weitergehende Funktionalitäten festlegen:

■ **Programmierte Werteilfe**

Das Merkmal soll eine programmierte Werteilfe erhalten. Dazu müssen Sie am Merkmal hinterlegen, welchen technischen Namen Sie für den Schlüsselwert und die Beschreibung in der Programmierung der BAdI-Methode SCENARIO_SET_ADDITIONAL_VALUES des BAdI QISR1 verwenden (siehe Abschnitt »Formularablauflogik«).

■ **Langtextformatierung**

Das Merkmal soll als mehrzeiliges Langtextfeld dargestellt werden, und ein Fließtext soll eingegeben werden können. In diesem Fall markieren Sie das Ankreuzfeld **Langtextfeld** (siehe Abbildung 25). Alternativ können Sie auch das ISR-Standardfeld ISR_NEW_LONG_TEXT verwenden (siehe Abschnitt »Standardwerte für die Parameter MODE, USER COMMAND und FORM VIEW«).

N...	Merkmalsname	Platzhalter für Schlüsselwerte	Platzhalter für Vorschlagswerte	La...
1	SYSTEM_USER			<input type="checkbox"/>
2	SYSTEMNAME	SYSTEMNAME_KEY	SYSTEMNAME_LABEL	<input type="checkbox"/>
3	CLIENT			<input type="checkbox"/>
4	PASSWORD			<input type="checkbox"/>

Abbildung 25 Customizing für programmierte Werteilfe und Langtextformatierung

Die Merkmale für die zusätzlichen Spalten in der ISR-Statusübersicht (siehe Abbildung 8) definieren Sie über **Erweitertes Customizing zum Szenario**, das Sie über Drücken der gleichnamigen Schaltfläche im ISR-Grund-Customizing erreichen. Für die Definition der maximal fünf Spalten stehen alle Merkmale zur Verfügung (siehe Abbildung 26).

Zusätzliche Merkmale in der Statusübersicht	
1. Merkmal	SYSTEM_USER <input type="checkbox"/>
2. Merkmal	SYSTEMNAME
3. Merkmal	
4. Merkmal	
5. Merkmal	

Abbildung 26 Merkmale für die zusätzlichen Spalten der Statusübersicht

Aktivitäten und automatische Verbuchungen in der Meldung

In diesem Abschnitt werden zunächst die Aktivitäten beschrieben, die im Rahmen der Meldungstransaktionen durch den Genehmigenden oder den Sachbearbeiter ausgeführt werden können. Im Abschnitt »Aktivitäten für den Sachbearbeiter vom Formular aus« werden die Aktivitäten vorgestellt, die ein Sachbearbeiter vom Formular aus durchführen kann. Beide Arten von Aktivitäten haben unterschiedliche technische Voraussetzungen und benötigen ein unterschiedliches Customizing.

Aktivitäten für den Sachbearbeiter bzw. Genehmigenden in der Meldungstransaktion

Es wird unterschieden zwischen manuellen Aktivitäten für den Sachbearbeiter der Serviceanfrage über die Meldungstransaktionen und Aktivitäten zur automatischen Verbuchung ohne Verwendung einer Meldungstransaktion. Die automatische Aktivität wird nicht vom Sachbearbeiter ausgeführt oder angestoßen, sondern zu einem bestimmten Zeitpunkt automatisch vom System ausgeführt.

Um beide Arten von Aktivitäten zu definieren, drücken Sie im ISR-Grund-Customizing zunächst die Schaltfläche **Aktivitätenleiste** (siehe Abbildung 22). Aktivitäten werden jeweils für eine Meldungsart definiert und können innerhalb aller Szenarien verwendet werden, die dieser Meldungsart zugeordnet sind. Im Detailbildschirm des Customizings zu den Aktivitäten haben Sie dann folgende Einstellungsmöglichkeiten:

- Die Einstellungen im Abschnitt **Steuerung** sind für die Sichtbarkeit und damit Verwendbarkeit der Aktivität innerhalb der Meldungstransaktion von Bedeutung.
 - **Szenario** steuert die Sichtbarkeit der Aktivität in den jeweiligen Szenarien, die zu einer Meldungsart gehören.
 - **Sortiernummer** steuert die Reihenfolge, in der die Aktivitäten in der Aktivitätenleiste der Meldungstransaktionen angeboten werden.
 - **Verwendung in** steuert die Sichtbarkeit der Aktivitäten in der Meldung; Vorschlagswert: **Jede Sicht (anlegen, ändern, anzeigen)**.
 - **und** steuert die Sichtbarkeit in Meldung und/oder Maßnahme; Vorschlagswert: **Meldung und Massnahme**.
 - **Dokumentation** definiert, ob bei der Durchführung einer Aktivität diese in der Meldung dokumentiert werden soll. (Tipp: Wählen Sie der Einfachheit halber zuerst einmal **keine**.)
- Im Abschnitt **Funktion** können Sie den Namen des Funktionsbausteins, der beim Aufruf der Aktivität prozessiert wird, festlegen. Mit den ausgelieferten Szenarien erhalten Sie einige Beispiele solcher Funktionsbausteine.

- Im Abschnitt **Oberfläche** legen Sie das Symbol und den Text fest, der in der Aktivitätenleiste erscheint.
- Im Abschnitt **Regeln** legen Sie die Reihenfolge und Abhängigkeiten in der Ausführung der Aktivitäten fest.

Es ist zu empfehlen, einen der folgenden generischen Funktionsbausteine als **Funktion** zu hinterlegen, falls die Aktivität mit dem BAdI »Ausführen einer Funktion der Aktivitätenleiste« programmiert wird. Der Schlüsselbegriff **FUNKTION** der Aktivität ist für die zugehörige Implementierung des BAdI QISR4 erforderlich.

- `ISR_ACTIVITY_EXECUTE_TASKDOC`
- `ISR_ACTIVITY_EXECUTE_ACTIONDOC`
- `ISR_ACTIVITY_EXECUTE`

Alle drei Bausteine sind äquivalent verwendbar und rufen dieselbe Methode `EXECUTE_FUNCTION` des BAdI QISR4 auf. Welchen Funktionsbaustein Sie verwenden, hängt einzig von der Angabe im Feld **Dokumentation** ab. `ISR_ACTIVITY_EXECUTE` kann verwendet werden, wenn **Dokumentation** = **keine** gewählt wurde.



Programmierung einer Aktivität

Der Funktionsbaustein einer Aktivität muss eine festgelegte Schnittstelle aufweisen. Da diese generisch auf die Belange der Meldung zugeschnitten ist, bietet es sich an, den Funktionsbaustein `ISR_ACTIVITY_EXECUTE` einzutragen und die eigentliche Programmierung im BAdI QISR4 vorzunehmen. Das hat den Vorteil, dass die notwendigen Daten zum Ausführen der Aktivität in der ISR-typischen Form der Übergabestrukturen zur Verfügung stehen. Die Programmierung der Aktivität kann so analog zur Programmierung der Formularablauflogik im BAdI QISR1 erfolgen.

Aktivitäten für die automatische Verbuchung

Wollen Sie die automatische Verbuchung verwenden, drücken Sie die Schaltfläche **Automatische Verbuchung**. Prinzipiell stehen alle Aktivitäten des Szenarios für die automatische Verbuchung zur Verfügung. Einzig Aktivitäten, die eine Folgeaktivität bzw. abhängige Aktivität sind, das heißt, eine vorausgehende Aktivität benötigen, werden nicht angeboten. Sie geben an, ob durch den Antragsteller oder den Genehmigenden einzelne Aktivitäten zur automatischen Verbuchung ausgeführt werden. Dies geschieht mit dem Abschicken bzw. Genehmigen des Antrags.



Aktivität im BAdI QISR4 programmieren

Die automatische Aktivität muss im Backend ohne Interaktion, das heißt Ausgabe oder Anzeige auf Dynpros, ablaufen. Der Aufruf sollte über das BAdI QISR4 zum Bei-

spiel über BAPIs oder SELECT-Befehle realisiert werden, und es sollte keine Fehlermeldung über den RETURN-Parameter ausgegeben werden.

Die Schnittstelle der Methode EXECUTE_FUNCTION im BAdI QISR4 bietet die Parameter, die Sie Tabelle 2 entnehmen können.

Parameter	Art	Beschreibung
FLT_VAL	Importing	Szenarioname
QMART	Importing	Meldungsart
FUNCTION	Importing	Schlüssel einer Funktion der Aktivitätenleiste aus dem ISR-Customizing
RETURN	Exporting	Fehlermeldung
GENERAL_DATA	Changing	allgemeine Daten
SPECIAL_DATA	Changing	Daten der szenarioabhängigen Felder

Tabelle 2 Schnittstelle des BAdI »QISR4«

In der Regel wird nach der Auswertung des Szenarios die gewünschte Aktivität abhängig von der gewählten Funktion (FUNCTION) ausgeführt. Dazu ein Programmbeispiel: Mit Funktion MC1 soll eine Kostenstelle per BAPI angelegt werden, darüber hinaus soll diese Funktion als automatische Funktion ausführbar sein. Funktion MC3 führt eine Transaktion aus und ist somit nicht als automatische Aktivität verwendbar, trotzdem können beide Funktionen in einer Implementierung programmiert werden (siehe Listing 1). Die korrekte Anwendung der Funktionen muss im ISR-Customizing sichergestellt werden.

```
METHOD if_ex_qisr4~execute_function.
  DATA: ls_change_data  TYPE fcom_isr_cc,
         lt_return       TYPE TABLE OF bapiret2,
         lt_costcenter   TYPE TABLE OF
                        bapi0012_ccinputlist.
```

* Übertrage Formulardaten an lokale Struktur

```
CALL FUNCTION 'ISR_SPECIAL_DATA_TO_STRUC'
  EXPORTING
    it_special_data = special_data
  CHANGING
    cs_data         = ls_change_data.
```



```
CASE function.
* Kostenstelle automatisch anlegen
  WHEN 'MC1'.
* Übertrage spezielle Daten an BAPI-Struktur
*   ...
* Aufruf des BAPI, um Profitcenter anzulegen
  CALL FUNCTION 'BAPI_COSTCENTER_CREATEMULTIPLE'
    EXPORTING

      controllingarea = ls_change_data-co_area
      testrun         = space
  TABLES
      costcenterlist = lt_costcenter
      return         = lt_return.
  IF lt_return[] IS INITIAL.
    COMMIT WORK.
  ENDIF.

* Aufruf der Transaktion zur Kostenstelle ändern
  WHEN 'MC2'.
  SET PARAMETER ID:
    'CAC' FIELD ls_change_data-co_area,
    'KOS' FIELD ls_change_data-costcenter.
  CALL TRANSACTION 'KS02' AND SKIP FIRST SCREEN.
  WHEN OTHERS.
  ENDCASE.
ENDMETHOD.
```

Listing 1 Programmierung des BAdI »QISR4« für Aktivitäten

Aktivitäten für den Sachbearbeiter vom Formular aus

Der Sachbearbeiter einer Anfrage hat die Möglichkeit, mehrere Aktivitäten per Schaltfläche vom Formular aus durchzuführen. Die Aktivitäten werden im Serviceantrag oberhalb des Formulars als Reihe von Schaltflächen angeboten (siehe [Abbildung 27](#)) und können mit dem Link **Aktivitätenliste ausblenden** aus- und wieder eingeblendet werden.

Sie definieren die Aktivitäten im Customizing des Report Launchpads. Diese Aktivitäten können sinnvoll sein, wenn der Sachbearbeiter die Serviceanfrage ausschließlich in SAP Enterprise Portal bearbeiten und ganz auf die Meldungstransaktionen verzichtet werden soll.

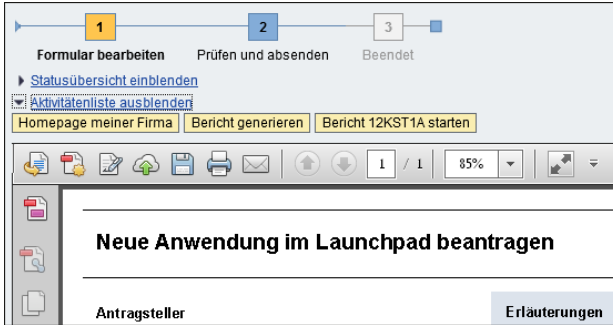


Abbildung 27 Aktivitätenliste des Sachbearbeiters über dem Formular

Technische Voraussetzung für die Verwendung der Aktivitäten ist SAP ERP 6.0 SP 12 und die Verwendung des SAP Enterprise Portals. Weitere Details dazu finden Sie in SAP-Hinweis 1103969. Sie müssen außerdem eine der Web-Dynpro-Anwendungen QISR_FORM oder QISR_UI_FORM_DISPLAY_ONLY benutzen.

Der Aufruf des Formulars erfolgt mit dem zusätzlichen URL-Parameter IFV=P, oder Sie setzen im Workflow den Container-Parameter FORM_VIEW = ISR_PROCESS. Die parametrisierte URL für einen Antrag mit der Meldungsnummer 000600000110 des Szenarios SRCR, die vom Sachbearbeiter im Modus CHANGE gestartet wird, sieht dann beispielsweise so aus:

http://<server>:<port>/sap/bc/webdynpro/sap/qISR_ui_form?sap-language=DE&SCENARIO=SRCR&MODE=CHANGE&NOBMP_NO=000600000110&IFV=P

Die Aktivitäten sind im Customizing des Report Launchpads über die Transaktion LPD_CUST konfigurierbar.

1. Um das Report Launchpad korrekt bedienen zu können, muss Ihr Systembenutzer den Parameter LPD_EXPERT auf 'X' gesetzt haben.
2. Rufen Sie die Transaktion LPD_CUST auf, und legen Sie ein neues Launchpad an.
3. Die Rolle muss immer »QISR« sein. Die Instanz entspricht Ihrem Szenario.
4. Markieren Sie das Kennzeichen **Action Launchpad** (siehe [Abbildung 28](#)). Im Launchpad selbst können Sie Aktivitäten mit unterschiedlichen Anwendungstypen definieren.



Abbildung 28 Customizing eines neuen Launchpads für Aktivitäten

5. In [Abbildung 29](#) sehen Sie einige Beispiele für Anwendungstypen, generelle Erläuterungen zum Report Launchpad entnehmen Sie der zugehörigen Dokumentation.

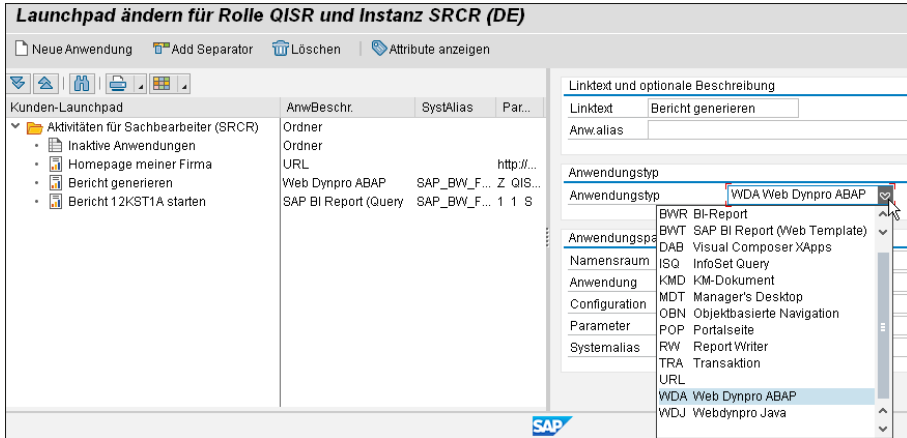


Abbildung 29 Customizing für Aktivitäten des Formulars

Sie können analog zu den Aktivitäten der Meldungstransaktion aus den vorangehenden Abschnitten das BAdI QISR4 verwenden.

1. Dazu definieren Sie eine Aktivität vom Anwendungstyp **Transaktion**.
2. Nehmen Sie eine Variante des Reports QISRACTIVITY in einer kundeneigenen Transaktion in das Launchpad auf.
3. Die Übergabe der Parameter QMNUM und SCENARIO für die Meldungsnummer und das Szenario erfolgt automatisch durch das ISR-Framework.
4. Den Schlüsselwert FUNCTION der ausgewählten Aktivität müssen Sie über die Reportvariante steuern. Dazu können Sie Transaktion QISRACTIVITY als Anhaltspunkt nehmen.

Bearbeiterfindung (BAdI QISR3) und Workflow

Die Bestimmung des zuständigen Sachbearbeiters bzw. Genehmigenden der Serviceanfrage kann auf unterschiedlichen Wegen erfolgen. Mit der Verwendung von SAP Business Workflow besteht die Möglichkeit, den Bearbeiter durch das System automatisch bestimmen zu lassen.

Verzichten Sie auf den SAP Business Workflow, kann der Sachbearbeiter die Serviceanfrage, für die er zuständig ist, über den Arbeitsvorrat finden. Dazu kann die Transaktion IQS8 verwendet werden, die sicherlich die Variante mit dem geringsten Implementierungsaufwand ist, aber den Nachteil hat, dass der Sachbearbeiter seine

Anträge aktiv suchen muss. In diesem Fall ist nicht durch das System sichergestellt, dass alle Serviceanfragen zugestellt werden.

Wenn Sie gänzlich auf die ausgelieferten Standard-Workflow-Templates verzichten und ein eigenes Workflow-Template verwenden wollen, müssen Sie zusätzlich im SAP Business Workflow eine Ereignistypkopplung für den Objekttyp BUS7051, das Ereignis CREATED und Ihr Workflow-Muster anlegen. Bei allen Ereignistypkopplungen für diese Kombination muss der Funktionsbaustein QGN01_EVENT_NOBMP_CREATED_ISR als **Check-Funktion** eingetragen werden.

Standard-Workflow-Template WS03100019

Das Business-Objekt BUS7051 ist für die Erzeugung des Laufzeitobjekts des Antrags verantwortlich, das bislang als *Meldung* bezeichnet wurde, und bietet eine Anbindung an den SAP Business Workflow. Dazu kann das Standard-Workflow-Template WS03100019 der Meldung verwendet werden. Dieses Workflow-Template unterstützt die Einbindung von beliebig vielen Sachbearbeitern, und Sie haben die Wahl, mit einem Genehmigungsschritt zu arbeiten oder diesen Schritt auszulassen.

Im Customizing des Workflows müssen Sie die Ereignistypkopplung für den Objekttyp BUS7051 aktivieren. Zuständige Bearbeiter können Sie entweder direkt im Customizing des Workflows hinterlegen oder durch die Regeln des Workflows (Transaktion PFAC) automatisch ermitteln lassen. Das Workflow-Template unterstützt im Genehmigungsfall nur die Variante mit der Backend-Transaktion IQS22.

ISR-Workflow-Template WS31000009

Möchten Sie *mehr* als einen Genehmigungsschritt durchführen, so genügt das Standard-Workflow-Template der Meldung nicht. Sie können alternativ das im Rahmen des ISR-Frameworks ausgelieferte Workflow-Template WS31000009 verwenden, das zudem die beiden Möglichkeiten der Antragsbearbeitung durch den Genehmigenden sowohl im SAP-ERP-Backend-System als auch im Portal (siehe Abschnitt »Rolle Genehmigender«) unterstützt. Es muss im Grund-Customizing des ISR unter **Zusätzliche Daten zum Szenario** hinterlegt werden (siehe Abbildung 30).

Szenariospezifische Workflow-Muster	
Meldung Anlegen	WS31000009
Meldung in Arbeit	
Zuordnung der Bearbeiter	
Aktivierung Ereigniskopplung	

Abbildung 30 Erweitertes ISR-Customizing zum Workflow

Die Ereignistypkopplung können Sie im selben Bild über die Schaltfläche **Aktivierung Ereigniskopplung** aufrufen. Bearbeiternamen oder Workflow-Regeln zur automatischen Bestimmung von Sachbearbeitern oder Genehmigenden hinterlegen Sie direkt

im Customizing des ISR (siehe [Abbildung 31](#)). Drücken Sie dazu die Schaltfläche **Zuordnung der Bearbeiter**. Zu empfehlen ist das ISR-Workflow-Template WS31000009, da es flexibler und einfacher zu konfigurieren ist.

Die Verwendung des Workflow-Templates WS31000009 hat neben dem zentralen Customizing den Vorteil, dass der aktuelle Workflow-Container bei jeder Bearbeiterbestimmung durch eine Regel über eine szenariospezifische Implementierung des BAdI QISR3 geändert werden kann. Mit der Regel können Zuständigkeiten in Abhängigkeit von den Formulardaten ermittelt und mit dem BAdI QISR3 die Bearbeiterbestimmung zusätzlich gesteuert werden. Das BAdI hat nur eine Methode.

Sicht "Bearbeiterfindung" ändern: Übersicht			
Neue Einträge			
Szenario	SMC1		
Version	0		
Bearbeiterfindung			
Rolle	S...	Typ	Bearbeiter/Regel
AP Genehmigender	1	US Benutzer	MANAGER
AP Genehmigender	2	AC Rege1	95000174
PR Meldungsbearbeiter	1	AC Rege1	95000140

Abbildung 31 Bearbeiterzuordnung zu Rollen des ISR

Tabelle 3 beschreibt die Schnittstelle der einzigen Methode `CONTAINER_FOR_ACTOR_DET_FILL`. Über die Parameter `IS_VIQMEL`, `IS_TQ80` und `IS_VIQMSM` erhalten Sie zur Laufzeit Informationen zur aktuellen Meldung, `IT_SPECIAL_DATA` enthält die Formulardaten. Diese Daten können Sie nun verwenden, um den Workflow-Container `CT_ACT_CONTAINER` anzupassen. `IT_CONT_DEF` beinhaltet die Liste der Containerelemente aus dem Workflow-Template. Der Containerinhalt kann dann in einer Workflow-Regel für die Bearbeiterbestimmung ausgewertet werden.

Parameter	Art	Beschreibung
IT_CONT_DEF	Importing	Workflow-Containerelemente
IS_VIQMEL	Importing	Meldung
IS_TQ80	Importing	Meldungsart
IS_VIQMSM	Importing	Maßnahme
IT_SPECIAL_DATA	Importing	Formulardaten
CT_ACT_CONTAINER	Tables	Workflow-Container

Tabelle 3 Schnittstelle der Methode »CONTAINER_FOR_ACTOR_DET_FILL«

Dazu ein einfaches Beispiel, das die Bearbeiterermittlung mithilfe einer Workflow-Regel unter der Verwendung des BAdI QISR3 beschreibt: Der Bearbeiter der Anfrage wird über die Regel 95000140 ermittelt (siehe [Abbildung 32](#)). Gemäß dieser Regel sollen alle Anträge des Szenarios SMC1 mit dem Merkmalswert für den Kostenstellenverantwortlichen A* bis L* einem Bearbeiter Smith zugeordnet werden.

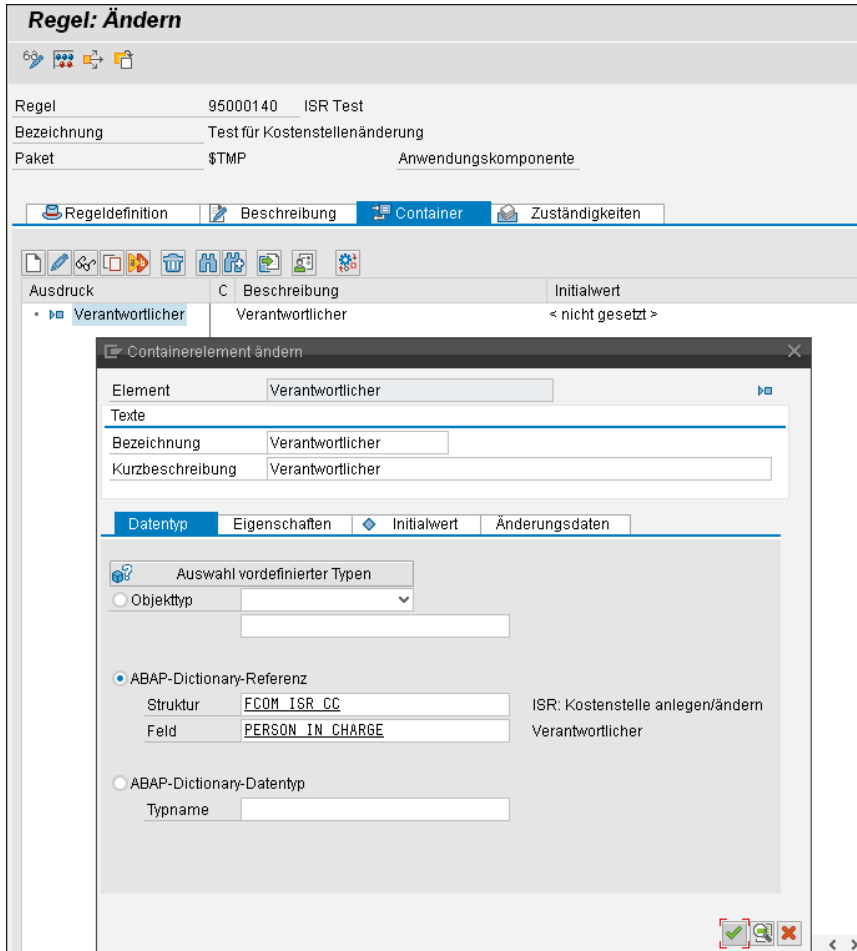


Abbildung 32 Containerelement der Regel mit Bezug auf ein Merkmal

Liegt der Merkmalswert im Intervall M* bis Z*, wird der Bearbeiter MSS1 ermittelt (siehe [Abbildung 33](#)). Das funktioniert bereits ohne das BAdI QISR3. Im BAdI können Sie zum Beispiel für bestimmte Kriterien den Kostenstellenverantwortlichen noch ändern. Sie sollten dabei das aktuelle ISR-Szenario abfragen, da diese BAdI-Implementierung für alle Szenarien durchlaufen wird, die die genannten Workflow-Templates verwenden (siehe [Listing 2](#)).

Bezeichnung	Priorität	Status	Kürzel	effektiv zug...	effektiv zugeo...
<ul style="list-style-type: none"> M bis Z <ul style="list-style-type: none"> BUA1 Peter Smith A bis L <ul style="list-style-type: none"> MSS1 		<ul style="list-style-type: none"> Zeitraum nicht... Zuständigkeit ... 	<ul style="list-style-type: none"> M bis Z BUA1 Smith A bis L MSS1 	<ul style="list-style-type: none"> 13.11.2006 19.04.2015 26.08.2008 	<ul style="list-style-type: none"> 31.12.2030 31.12.2030 31.12.2030

Abbildung 33 Definition der Zuständigkeiten der Regel

```
METHOD if_ex_qisr3~container_for_actor_det_fill.
DATA: ls_change_data TYPE fcom_isr_cc,
      ls_act_container TYPE swcont.
```

```
* Nur Szenario SMC1
CHECK is_viqmel-auswirk = 'SMC1'.
```

```
* Formulardaten in lokale Struktur
CALL FUNCTION 'ISR_SPECIAL_DATA_TO_STRUC'
EXPORTING
  it_special_data = it_special_data
CHANGING
  cs_data          = ls_change_data.
```

```
* Anpassung des WF-Containers
IF ls_change_data-department = 'A123'.
  READ TABLE ct_act_container INTO ls_act_container
  WITH KEY element = 'PERSON_INCHARGE'.
  ls_act_container-value = 'STANDARD_RESPONSIBLE'.
  MODIFY ct_act_container FROM ls_act_container.
ENDIF.
ENDMETHOD.
```

Listing 2 Programmierung des BAdI »QISR3«

Servicekosten und BAdI QISR2

Die interne Serviceanfrage bietet eine Integration in das Gemeinkostencontrolling von SAP ERP 6.0 an. Diese Funktion ermöglicht die kostenmäßige Kontrolle der zur Verfügung gestellten Serviceanfragen im Unternehmen. Stellt eine Abteilung Ihres Unternehmens einen oder mehrere Services für die Mitarbeiter zur Verfügung, können die Gemeinkosten, die dieser Abteilung entstehen, intern verrechnet werden.

Der ISR stellt drei Alternativen der Kostenermittlung zur Verfügung:

■ **Template-Verrechnung des Easy Cost Plannings**

Das Easy Cost Planning ermöglicht, eine komplexe Kostenermittlung mit der Serviceanfrage zu verbinden. Die Möglichkeiten und Funktionen dieses Tools werden hier nicht weiter erläutert, da sie den Rahmen dieses Kapitels überschreiten würden.

■ **Geschätzte Kosten**

Geschätzte Kosten führen zu einem Eintrag der Kosten in die Meldung. Diese Kosten werden jedoch nicht ins Gemeinkostencontrolling übernommen.

■ **Preisliste**

Die Preisliste ist eine einfache Möglichkeit, die Kosten eines Antrags zu ermitteln. In diesem Abschnitt soll das Verfahren mit der Preisliste erläutert werden.

Die Kosten werden automatisch mit dem Abschließen der durch die interne Serviceanfrage angelegten Meldung auf einem Kostenträgerobjekt, wie zum Beispiel einem Innenauftrag oder der Kostenstelle des Antragstellers, gebucht. Diese Kosten werden auf Basis eines ebenso einstellbaren Tarifs von der Kostenstelle dieser Serviceabteilung als Leistung aufgenommen.

Mit wenigen Einstellungen können Sie die automatische Kostenermittlung erreichen (siehe Abbildung 34):

1. Im ISR-Grund-Customizing markieren Sie das Kennzeichen **Das Szenario ist kostenverursachend**.
2. Dann tragen Sie den **Kostenrechnungskreis** ein.

Kosten

Das Szenario ist kostenverursachend.

Es enthält folgende Angaben:

Kostenrechnungskreis 1000 CO Europe

Zur Kalkulation der entstehenden Kosten hinterlegen Sie geschätzte Kosten oder eine Preisliste oder ein Template mit der Kalkulationsvariante und mit der Strategie

Geben Sie eines der folgenden "Fixen k...

Kostenstelle PSP-Element

EUR

Preisliste bearbeiten

Fixes Kontierungsobjekt

Fixes Kontierungsobjekt

Kostenstelle des Initiators 101924

PSP-Element des Initiators

User-Exit

Abbildung 34 ISR-Grund-Customizing für Servicekosten

3. Anschließend bestimmen Sie den Kostensender mithilfe der **Preisliste**. Markieren Sie dazu das Kennzeichen **Preisliste**, und klicken Sie auf **Preisliste bearbeiten**.
4. Sie können in der Preisliste die Kombination der Kostenstelle und der Leistungsart des Erbringers der Leistung (zum Beispiel Kostenstelle der Serviceabteilung) festlegen. Die »Kosten« eines Serviceantrags tragen Sie im Feld **Tarif ISR** ein.
5. Sie können auch mehrere Tarife anlegen. Das Kennzeichen **Default** steuert, welcher Tarif verwendet wird (siehe [Abbildung 35](#)).



Abbildung 35 Definition der Preisliste

6. Bestimmen Sie den Kostenempfänger (Strategie), indem Sie das sogenannte *Kontierungsobjekt* auswählen. Zur Auswahl stehen:
 - die Kostenstelle des Antragstellers, die aus den HR-spezifischen Benutzerstammdaten ermittelt wird
 - das PSP-Element des Antragstellers, das aus den HR-spezifischen Benutzerstammdaten ermittelt wird
 - Sie tragen einen Innenauftrag, eine Kostenstelle oder ein PSP-Element explizit ein
 - ein User-Exit, nämlich die BAdI-Implementierung des BAdI QISR2. [Tabelle 4](#) beschreibt die Schnittstelle der Methode `ISR_ACCOUNT_ASSIGNMENT_GET` des BAdI QISR2. Die Programmierung erfolgt analog zum BAdI QISR1. Über `OBJNR_REAL` geben Sie die Objektnummer für die echte Kontierung im Controlling mit. Über `OBJNR_STAT` kann das statistische Kontierungsobjekt festgelegt werden.

Parameter	Art	Beschreibung
FLT_VAL	Importing	Szenarioname
GENERAL_DATA	Importing	allgemeine ISR-Daten
SPECIAL_DATA	Importing	Formulardaten
OBJNR_REAL	Exporting	echte CO-Kontierung
OBJNR_STAT	Exporting	statistische CO-Kontierung

Tabelle 4 Methode »ISR_ACCOUNT_ASSIGNMENT_GET«

7. Haben Sie eine eigene Meldungsart definiert, prüfen Sie, ob eine Implementierung zum BAdI NOBMP_EVENT_POST mit der Meldungsart Ihres Szenarios als Filterwert existiert. Sollte keine Implementierung vorliegen, legen Sie in diesem Fall eine neue Implementierung mit Ihrer Meldungsart als Filterwert an. Als Vorlage sollten Sie die Implementierung ISR_CHECK_AT_POST verwenden.

Testen des Szenarios

Die Funktion **Testen** neben dem Feld **Formular** im ISR-Grund-Customizing (siehe Abbildung 22) bietet die Möglichkeit, das Formular schnell und direkt aufzurufen, um Änderungen am Formular oder im Customizing sofort überprüfen zu können.

1. Dazu müssen Sie in das Customizing **Zusätzliche Daten zum Szenario** wechseln und über die Schaltfläche **Server** einen passenden Server eintragen.
2. Falls noch kein Eintrag für die Erfassungsart im Web (**A – Erfassung mit Adobe PDF**) vorhanden ist, legen Sie einen neuen Eintrag an.
3. Im Feld **Server für den Aufruf der ISR Formulare** tragen Sie die URL für den Server in der folgenden Form ein: *http://<myserver>:<port>/sap/bc/webdynpro/sap/*. Den Platzhalter *<myserver>* ersetzen Sie dabei durch Ihren Server, und statt *<port>* tragen Sie den betreffenden Port ein.
4. Nehmen Sie außerdem die Einstellungen aus SAP-Hinweis 1110677 zur Aktivierung der Testbarkeit mit der Web-Dynpro-ABAP-Component des ISR im Customizing vor.
5. Zum Testen des Adobe-Formulars müssen Sie dann nur die Schaltfläche **Testen** drücken. Das System fügt die notwendigen URL-Parameter (zum Beispiel das Szenario) automatisch hinzu. Das Formular zum Anlegen eines Antrags wird in einem Browserfenster gestartet.

6. Eine weitere Alternative zum Testen bietet der Funktionsbaustein `ISR_PROCESS_EVENT`. Damit können alle Backend-Einstellungen ohne das Adobe-Formular getestet werden: das Szenario-Customizing, der Workflow und die BAdI-Implementierung. Eine detaillierte Beschreibung des Funktionsbausteins finden Sie im Abschnitt »[Remote Function Call »ISR_PROCESS_EVENT«](#)«.

Antragsformular als Adobe-Formular

Das interaktive Adobe-Formular im Rahmen des ISR-Frameworks muss bestimmte technische Bedingungen in Bezug auf die Schnittstelle, den Kontext des Formulars und die einzelnen Bildelemente erfüllen. Aus diesem Grund werden im ISR-Grund-Customizing zusätzliche Definitions- und Generierungshilfen zur Verfügung gestellt.

Spezifische Eigenschaften

Wie in Abschnitt »[Merkmale](#)«, erläutert wurde, definieren Sie die Merkmale im ISR-Customizing. Die Felder auf dem Adobe-Formular werden jedoch an den Kontext des Formulars gebunden und nicht an das Customizing des ISR. Wie wird demnach sichergestellt, dass das Formular die richtigen Merkmale enthält?

Insbesondere die Schnittstelle und der Kontext des Formulars müssen bestimmte Kriterien erfüllen. Die Möglichkeiten, die der Form Builder für die Definition der Formularschnittstelle zur Verfügung stellt, sind im Rahmen des ISR nicht vollständig einsetzbar. Grund dieser Einschränkung ist, dass die Definition des Formularkontextes gewissermaßen in das ISR-Customizing ausgelagert wird. Das bedeutet, dass Schnittstelle und Kontext konform mit den Merkmalen des Customizings sein müssen.

Für die Verwendbarkeit einer Schnittstelle im ISR gelten folgende Bedingungen:

- Es darf nur eine DDIC-Schnittstelle verwendet werden, eine XML-Schnittstelle zu verwenden ist nicht möglich.
- Die Schnittstelle muss eine ISR-spezifische Datenstruktur besitzen.
- Es werden ausschließlich Importparameter verwendet.
- Die Anbindung einzelner Elemente an spezielle DDIC-Elemente oder weitergehende Funktionen des Form Builders sind nicht möglich. So werden unter anderem Initialisierungen und Formroutinen nicht unterstützt.
- Dateninitialisierungen müssen Sie in einer passenden Implementierung des BAdI `QISR1` programmieren (siehe Abschnitt »[Formularablauflogik](#)«).

Für die Verwendbarkeit eines Formulars im ISR gelten die folgenden Bedingungen:

- Der Layouttyp muss **ZCI Layout** sein.
- Der Datenkontext hat eine ISR-spezifische Datenstruktur und muss 1:1 aus der Schnittstelle übernommen werden.
- Für die Nutzung von programmierten Wertehilfen ist das Skript der Bildelemente aus der ISR-Bibliothek obligatorisch. Gleiches gilt für die Anwendung von Formular-Events und wenn Sie das standardisierte Genehmigungsformular verwenden.

Zero Client Installation im ISR

Mit der Einführung der ZCI-Technologie (Zero Client Installation) wurden alle ausgelieferten Adobe-Formulare des ISR von der früheren ACF-Technologie auf ZCI umgestellt. Die Auslieferung in SAP ERP 6.0 erfolgte mit SP 5 (siehe SAP-Hinweis 947675). Grundsätzlich wird im Rahmen des ISR für kundeneigene Formulare die Verwendung des Layouttyps **ZCI Layout** empfohlen.



Aufbau der Schnittstelle und des Formulkontextes

Das ISR-Framework sorgt dafür, dass die Inhalte der Felder des Adobe-Formulars ins Backend gelangen, und umgekehrt, dass Feldinhalte, die im Backend ermittelt werden, korrekt im Formular angezeigt werden. Entscheidend dafür ist, dass der Formulkontext die Kriterien des ISR erfüllt. Der Kontext entspricht demnach exakt der zugeordneten Schnittstelle. Wenn Sie für die Generierung neuer und für die Anpassung bestehender Formulare ausschließlich die Funktionen des ISR-Customizings verwenden, wird diese Forderung automatisch erfüllt.

Um eine korrekte Datenanbindung des Adobe-Formulars zur Laufzeit zu gewährleisten, muss jedes Merkmal im Kontext des Adobe-Formulars die generischen Tabellentypen des ISR verwenden (siehe [Abbildung 36](#)):

- **QISR_TAB_TYPE_DATE**
Merkmale des Typs **Datum** (DDIC-Datentyp DATS)
- **QISR_TAB_TYPE_TEXT**
Merkmale für Fließtext
- **QISR_TAB_TYPE**
alle anderen Merkmale

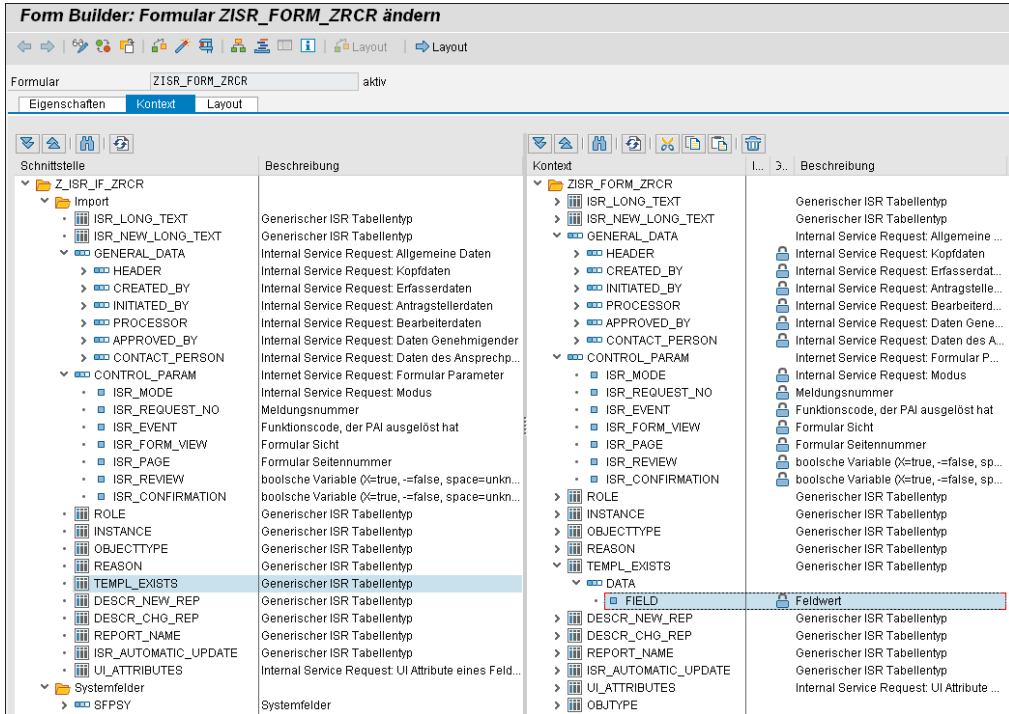


Abbildung 36 Schnittstelle und Kontext eines Adobe-Formulars im ISR

Alle drei Tabellentypen weisen die gleiche Datenstruktur auf: <Merkmal>-<DATA>-<FIELD>. Der Platzhalter <Merkmal> steht für den Namen des Merkmals aus dem ISR-Customizing. Aufgrund der generischen Programmierung der ISR-Web-Dynpro-Component ist diese tabellenartige Datenstruktur zwingend erforderlich und darf nicht verändert werden. Ausgenommen von dieser Regel sind die folgenden ISR-spezifischen Strukturen, da sie in allen Formularen und Schnittstellen des ISR vorkommen und deshalb nicht in der Web-Dynpro-Component generiert werden müssen:

- **Allgemeine ISR-Daten** (Struktur GENERAL_DATA)

Die Struktur enthält Informationen zu den beteiligten Bearbeitern und technische Informationen (siehe Abschnitt »Merkmale«).
- **Formularparameter** (Struktur CONTROL_PARAM)

Die Struktur enthält technische Informationen, die in einer Skriptprogrammierung verwendet werden können. Die Attribute entsprechen denen aus dem Abschnitt »Standardwerte für die Parameter MODE, USER COMMAND und FORM VIEW«. Das Attribut ISR_REVIEW zeigt an, ob sich der Anwender im zweiten Schritt (**Prüfen und absenden**) des Antragsformulars befindet, und Attribut ISR_CONFIRMATION informiert, dass sich der Anwender im dritten und letzten Schritt (**Beendet**) des Antragsformulars befindet.

- **UI-Attribute eines Feldes** (Struktur UI_ATTRIBUTES)
Diese Struktur beinhaltet die Layoutinformationen aus der BAdI-Implementierung QISR1.

Erstellen des Kontextes und der Schnittstelle

Sie sollten den Kontext eines Formulars grundsätzlich nur über die Funktionen **Generieren** und **Adobe-formular ändern** des ISR-Customizings anpassen. Die Schnittstelle wird dabei automatisch angepasst.



Generierung eines Adobe-Formulars aus den Merkmalen

Bei der Generierung eines Adobe-Formulars werden die Merkmale des Szenarios automatisch in die Schnittstelle und den Kontext des Formulars übernommen. Für die Erstellung eines Adobe-Formulars sind dabei verschiedene Konstellationen vorstellbar:

- **Sie haben bereits ein ISR-Szenario mit Merkmalen definiert. Das Formular wurde noch nicht erstellt.**
Sie generieren das Formular über die Schaltfläche **Generieren** (siehe [Abbildung 37](#)). Sie haben die Auswahl zwischen:
 - einem leeren Formular
 - einem Formular im SAP-Standard (Vorlage ist das Formular ISR_FORM_STANDARD)
 - einem Formular mit Vorlage eines anderen Szenarios

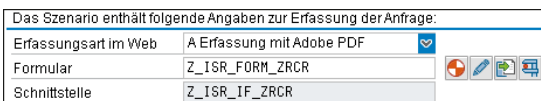


Abbildung 37 Funktionen zur Generierung und Bearbeitung eines Adobe-Formulars im ISR-Grund-Customizing

Im Form Builder entwerfen Sie dann das Layout. Zur Definition von Feldern mit ISR-spezifischer Skriptprogrammierung (siehe Abschnitt [»ISR-Bibliothek – spezielle Bildelemente«](#)) verwenden Sie zumindest am Anfang die ISR-Bibliothek. Zuletzt erzeugen Sie die Datenbindung zwischen Layoutelementen und den Kontextfeldern. Wenn Sie im Form Builder neue Felder im Layout definieren, für die noch kein Merkmal existiert, werden Sie beim Verlassen des Form Builders gefragt, ob diese Merkmale erzeugt werden sollen (siehe Abschnitt [»Generierung neuer Merkmale aus dem Formular«](#)).

- **Sie haben bereits ein ISR-Szenario mit Merkmalen definiert. Sie besitzen bereits ein PDF-Formular, aber noch kein Form-Builder-Formular.**
In diesem Fall können Sie verfahren, wie beschrieben, nur müssen Sie nach der

Generierung des PDF in den Form Builder laden und anschließend das Layout bearbeiten. Zuletzt nehmen Sie die Datenbindung vor.

- **Sie verfügen über ein im Form Builder definiertes Formular, das bereits die richtigen Formularfelder im Layout, jedoch nicht den für den ISR passenden Kontext hat.**

Hier gibt es zwei »sinnvolle« Alternativen, das Formular für den ISR anzupassen:

- Wenn Sie die zugehörigen Merkmale im ISR-Customizing bereits definiert haben, tragen Sie den Formularnamen im ISR-Customizing ein, und drücken Sie die Schaltfläche **Adobe-Formular ändern**. Damit gelangen Sie in den Form Builder, und die passende Schnittstelle und der passende Kontext werden generiert. Beachten Sie, dass der vorherige Kontext überschrieben wird. Nun müssen Sie noch die Datenbindung zwischen Layout und Kontext manuell herstellen.
- Wenn Sie die zugehörigen Merkmale im ISR-Customizing noch nicht definiert haben, tragen Sie den Formularnamen im ISR-Customizing ein, und drücken anschließend die Schaltfläche **Formularfelder importieren**. Damit liest das System die technischen Namen aller Layoutelemente des Formulars, die prinzipiell zur Dateneingabe geeignet sind, und generiert die Merkmale im ISR-Customizing. Zusätzlich wird automatisch der Kontext generiert und die Datenbindung im Formular hergestellt (siehe Abschnitt »Generierung neuer Merkmale aus dem Formular«).



Was geschieht bei der Generierung?

Bei der Generierung eines Formulars wird das Workbench-Objekt eines Formulars und einer ABAP-Dictionary-basierten Schnittstelle (DDIC) generiert. Die XML-basierte Schnittstelle kann im Rahmen des ISR nicht verwendet werden.

Den technischen Namen des Formulars und der Schnittstelle können Sie vorher in die gleichnamigen Felder des ISR-Customizings eintragen, wobei die Wahl der Namen im Rahmen des Kundennamensraums von Workbench-Objekten frei ist. Falls Sie keine Namen eintragen, werden diese Namen vom System automatisch ermittelt. Im Kundensystem wird dafür folgende Logik angewandt (xxxx steht für den vierstelligen Namen des Szenarios):

- Formular: ZISR_FORM_xxxx
- Schnittstelle: ZISR_IF_xxxx

Die im ISR-Customizing definierten Merkmale des Szenarios werden gleichzeitig in die DDIC-Schnittstelle und in den Kontext des Formulars übernommen. Analog dazu werden die im Abschnitt »Aufbau der Schnittstelle und des Formularkontextes« beschriebenen Standardfelder übernommen.

Generierung neuer Merkmale aus dem Formular

Für den Fall, dass Sie zunächst ein Formular definieren, ohne vorher ein ISR-Szenario mit Merkmalen angelegt zu haben oder wenn Sie im Layout neue Felder definiert haben, können Sie wie folgt vorgehen, um das Formular im ISR verwendbar zu machen:

1. Sie haben den Form Builder über die Schaltfläche **Adobe-formular ändern** aufgerufen. Haben Sie im Layout neue Felder definiert, werden diese beim Verlassen des Form Builders automatisch als Merkmale in das Szenario-Customizing übernommen. Dabei wird auch automatisch die Datenbindung der neuen Felder im Layout hinterlegt. Um dies zu erreichen, bestätigen Sie das zugehörige Pop-up-Fenster.
2. Sie können dies auch später nachholen. Im Grund-Customizing führen Sie die Funktion **Formularfelder importieren** per Schaltfläche aus. Dadurch werden aus den technischen Namen aller auf dem Formular definierten Bildelemente entsprechende Merkmale generiert. Zusätzlich werden die zugehörige Schnittstelle und der Kontext im Formular angepasst. Zuletzt wird die Datenbindung zwischen den Kontextelementen und den zugehörigen Bildelementen des Formulars erzeugt, all dies geschieht automatisch. Das Pop-up-Fenster mit der Frage, ob das Formular aktiviert werden soll, quittieren Sie mit **Ja**.
3. Die generierten Merkmale werden als **untypisiert** deklariert. Sie müssen daher das Customizing der Merkmale nachbearbeiten, um zum Beispiel für einzelne Merkmale den Bezug zum Data Dictionary herzustellen.
4. Haben Sie das Formular zunächst außerhalb des Szenario-Customizings erstellt, gehen Sie in das Customizing, und tragen Sie im Feld **Formular** das gewünschte Formular ein. Anschließend drücken Sie die Schaltfläche **Formularfelder importieren** rechts neben dem Eingabefeld.

ISR-Bibliothek – spezielle Bildelemente

Es ist zu empfehlen, die Bildelemente aus der ISR-Bibliothek für Formulare mit **ZCI Layout** zu verwenden (siehe [Abbildung 38](#)). Dabei ist nicht das Layout von Bedeutung, das Sie natürlich frei anpassen können, sondern insbesondere das Skript. Die Bibliothek wird mit dem SAP GUI ausgeliefert. Aktuelle Updates der ISR-Bibliothek finden Sie in SAP-Hinweis 947633, und die ISR-Bibliothek ist in SAP-Hinweis 741381 vollständig und detailliert dokumentiert.

Der Vorteil – oder besser die Notwendigkeit – der ISR-Bibliothek besteht darin, dass den einzelnen Elementen bereits ein ISR-spezifisches Skript (FormCalc oder JavaScript) mit speziellen Funktionen zugeordnet ist. Planen Sie beispielsweise, programmierte Eingabehilfen für ein Formularfeld zu realisieren, müssen Sie dieses Skript verwenden.

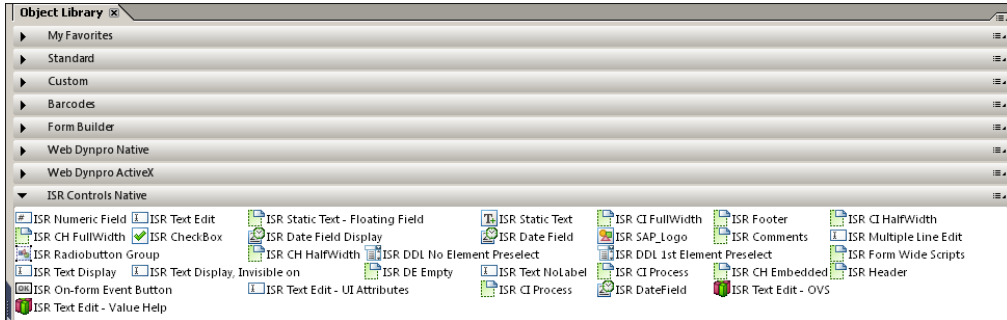


Abbildung 38 ISR-Bibliothek für Formulare mit dem Typ »ZCI Layout«

Wichtige spezielle Bildelemente mit eigenem Skript sind:

- **ISR DDL 1st Element Preselect (Eingabehilfe)**
Das erste Wertepaar der programmierten Werteliste wird automatisch als Feldwert übernommen. Ein Programmbeispiel für eine Dropdown-Listbox finden Sie im Abschnitt »[Programmbeispiele](#)«.
- **ISR DDL No Element Preselect (Eingabehilfe)**
Das Feld wird nur durch manuelle Auswahl aus der Dropdown-Listbox gefüllt.
- **ISR On-form Event Button (Schaltfläche)**
Hiermit wird eine Schaltfläche mit Event erzeugt, der im BAdI QISR1 ausgewertet werden kann.
- **ISR Text Edit – OVS (Eingabehilfe)**
Die Eingabe- und Ausgabefelder des Object Value Selectors (OVS) definieren Sie in der Customizing-Transaktion QISRSCENARIO_OVS. Die Programmierung dieser Eingabehilfe nehmen Sie in der Methode SET_ADDITIONAL_DATA Ihrer BAdI-Implementierung QISR1 vor.
- **ISR Text Edit – Value Help (Eingabehilfe)**
Details zu dieser DDIC-bezogenen Werthilfe finden Sie in SAP-Hinweis 1035630.
- **ISR Text Edit – UI Attributes für Layouteigenschaften (Skriptvorlage)**
Details zur Steuerung von Layouteigenschaften von einzelnen Feldern wie Mussfeld, Sichtbarkeit und Eingabebereitschaft über den BAdI QISR1 finden Sie in SAP-Hinweis 925657.
- **ISR Multiple Line Edit (mehrzeiliges Langtextfeld)**
Wenn Sie dieses Feld verwenden, müssen Sie das Kennzeichen **Langtext** für das entsprechende Merkmal im ISR-Grund-Customizing ankreuzen (siehe Abschnitt »[Merkmale](#)«).

Alternativ können Sie das Formularfeld an das ISR-Standardelement für Kommentare ISR_NEW_LONG_TEXT aus dem Formulkontext binden (siehe [Abbildung 39](#)). In diesem Fall wird jeder neu erfasste Text dem Sachverhalt-Langtext der Meldung

übergeben und ist in Transaktion IQS22 sichtbar und auch änderbar. In diesem Fall empfiehlt es sich, ein zweites Langtextfeld zu definieren und an das zweite ISR-Standardelement für Kommentare zur Anzeige ISR_LONG_TEXT zu binden. Damit werden bereits erfasste Texte angezeigt. Ein Beispiel für beide Langtextfelder sehen Sie im Formular in [Abbildung 10](#).

Im Folgenden werden Sie einfache Beispiele für ISR-spezifische Skripte kennenlernen.

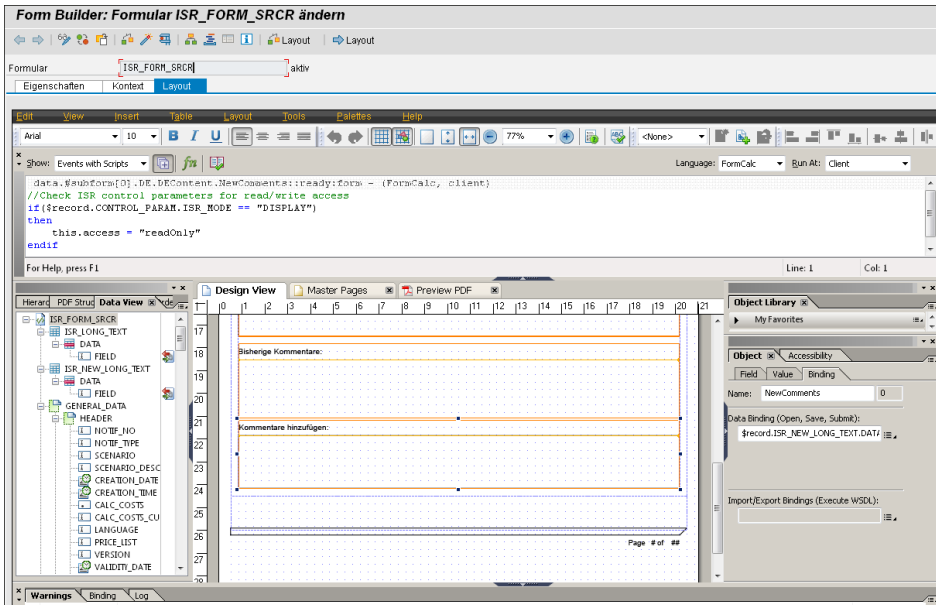


Abbildung 39 Verwendung der Standard-Langtextfelder »ISR_NEW_LONG_TEXT« und »ISR_LONG_TEXT« im Form Builder

Steuerung der Eingabebereitschaft von Feldern per Skript

Durch Anpassung der Skriptprogrammierung einzelner Felder im Form Builder können die Felder einfach eingabebereit geschaltet werden. Das ISR-Standardskript zur Steuerung der Eingabebereitschaft ist grundsätzlich im Ereignis `ready:form - (formcalc, client)` eines Feldes definiert (siehe [Listing 3](#)).

```
//Check ISR control parameters for read/write access
if($record.CONTROL_PARAM.ISR_MODE == "DISPLAY" |
    $record.CONTROL_PARAM.ISR_FORM_VIEW == "ISR_APPROVE" )
then
    this.access = "readOnly"
endif
```

Listing 3 Skript für Eingabebereitschaft

Entfernen Sie beispielsweise die Abfrage auf Gleichheit mit dem Event `ISR_APPROVE`, ist das entsprechende Feld für den Genehmigenden eingabebereit.

Haben Sie komplexere Anforderungen an die Layouteigenschaften, verwenden Sie das dynamische Skript der Skriptvorlage **ISR Text Edit – UI Attributes** aus der ISR-Bibliothek.

Erzeugung eines Events per Skript

Durch die Anpassung der Skriptprogrammierung einer Schaltfläche kann diese mit einem Event verbunden werden. Wird diese Schaltfläche gedrückt, wird die BAdI-Implementierung des BAdI `QISR1` durchlaufen und das `EVENT` des Formulars an den Parameter `USER_COMMAND` übergeben. Das ISR-Standardskript für den Namen des Events ist im Ereignis `mouseDown - (FormCalc, client)` der Schaltfläche durch folgendes Coding definiert:

```
//Set ISR_EVENT for BAdI processing in backend
$record.CONTROL_PARAM.ISR_EVENT = "CUSTOM_EVENT"
```

Zusätzlich muss der Mechanismus des Events auf dem Adobe-Formular im Ereignis `click - (JavaScript, client)` eines Feldes definiert werden (siehe [Listing 4](#)):

```
//Trigger call to backend for
//BAdI user command processing
ContainerFoundation_JS.SendMessageToContainer(event.
    target, "submit", "", "", "", "");
this.access = "readOnly"
```

Listing 4 Skript für Event

Erweiterte Werthilfe

Mit SAP ERP 6.0 SP 12 (`SAP_APPL`) können ein oder mehrere Felder eines Adobe-Formulars im ISR mit einer erweiterten Werthilfe (OVS) ausgestattet werden. Details zum OVS finden Sie in SAP-Hinweis 906950. Folgende Arbeitsschritte müssen Sie durchführen, um eine erweiterte Werthilfe zu implementieren:

1. Zugehöriges ISR-Customizing

In der Customizing-Transaktion `QISRSCENARIO_OVS` müssen Sie die Attribute zur Werthilfe definieren (siehe [Abbildung 40](#)). Wählen Sie ein Szenario aus, und gehen Sie zu **Merkmale mit Erweiterter Suchhilfe**. Definieren Sie ein Merkmal als Startfeld für eine Werthilfe, und setzen Sie unbedingt das **Aktiv**-Kennzeichen.

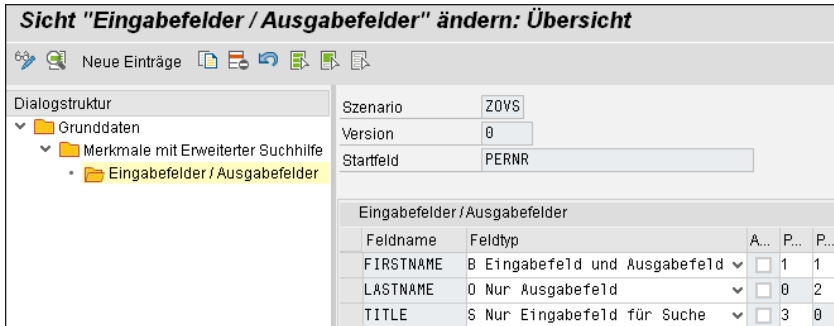


Abbildung 40 Customizing der erweiterten Wertehilfe

Definieren Sie für das gewählte Startfeld die Eingabe- und Ausgabefelder, verwendet werden können alle Merkmale des Szenarios, und legen Sie fest, welchen Typ und welche Position das jeweilige Feld haben soll:

- **B** Eingabefeld und Ausgabefeld
- **S** Nur Eingabefeld für Suche
- **O** Nur Ausgabefeld

Ausgabefelder können ausgeblendet werden. Das bedeutet, das Feld wird nicht in der Ergebnisliste der Wertehilfe angezeigt. Der Wert dieses Feldes wird jedoch nach Auswahl des Anwenders auf dem Pop-up-Fenster der Wertehilfe wie alle anderen ausgewählten Werte in das Formular übernommen.

2. Bildelement auf dem Formular

Verwenden Sie für die erweiterte Wertehilfe das Bildelement **ISR Text Edit – OVS** oder **ISR Text Edit – Value Help** aus der ISR-Bibliothek.

Beachten Sie, dass Sie zusätzlich zur üblichen Datenbindung und zu dem Skript der Eingabehilfe im Action-Handler `Click` den entsprechenden Feldnamen eintragen müssen. Suchen Sie nach der folgenden Zeile:

```
var fieldName = "FieldNameToBeReplaced";
```

Ersetzen Sie die Variable `FieldNameToBeReplaced` mit dem Feldnamen, den Sie unter dem Karteireiter **Objekt** unterhalb des untergeordneten Karteireiters **Bindung** finden. Der Feldname ist dort als **Name** aufgeführt.

Wenn nach Benutzereingabe in der Eingabehilfe eine automatische Prüfung des eingegebenen Wertes im SAP-ERP-Backend-System stattfinden soll, müssen im Ereignis `Exit` des Bildelements folgende Zeilen im Skripteditor des Adobe-Formulars eingefügt werden (siehe [Listing 5](#)):

```
xfa.record.CONTROL_PARAM.ISR_EVENT.value =  
    "<EVENT_NAME>";  
xfa.record.GENERAL_DATA.HEADER.EXT_REF_NUMBER.value =  
    "VALUE_HELP";
```

Listing 5 Zusätzliches Skript für Wertehilfe

Den Platzhalter <EVENT_NAME> müssen Sie durch einen beliebigen Ereignisnamen ersetzen. Sie können auch das Standard-Event CHECK verwenden.

3. Programmierung der Wertehilfe

Die Programmierung erfolgt in der szenariospezifischen Implementierung des BAdI QISR1 in der Methode SCENARIO_SET_ADDITIONAL_VALUES bzw. INT_SERVICE_REQUEST_INIT.

In die Übergabetabelle ADDITIONAL_DATA fügen Sie alle möglichen Ausgabewerte der Ausgabefelder ein. Diese Daten dienen als Grundlage für die Suche. Die Ergebnisliste wird dann zur Laufzeit, basierend auf der Eingabe, in die Suchfelder des OVS als Teilmenge der Einträge aus der Tabelle ADDITIONAL_DATA ermittelt. Die Tabelle enthält folgende Felder:

- Feld FIELDINDEX: Zähler, beginnend mit 1
- Feld FIELDNAME: Merkmalsname
- Feld FIELDVALUE: Wert

4. Tipps für die Implementierung im BAdI QISR1

Die Sortierung der Zeilen der Ergebnisliste wird über den Zähler FIELDINDEX gesteuert. In das Feld FIELDNAME muss der exakte Merkmalsname eingetragen werden, das heißt, ohne eine zusätzliche Endung, wie zum Beispiel KEY, wie es bei der Programmierung zur Dropdown-Listbox notwendig ist (siehe Abschnitt »Programmbeispiele«).

Andere Felder auf dem Formular können als Dropdown-Listbox programmiert werden. Der Funktionsbaustein ISR_CUST_SCENARIO_VALHLP_GET liefert alle Informationen aus dem Customizing zur erweiterten Wertehilfe.

In Listing 6 wird ein Programmbeispiel der Methode SCENARIO_SET_ADDITIONAL_VALUES aus dem BAdI QISR1 für eine erweiterte Wertehilfe gezeigt. Das Merkmal PERSON soll als OVS eingerichtet werden, und die maximale Ergebnisliste, in der gesucht werden soll, besteht aus den Ausgabefeldern FIRSTNAME, LASTNAME und TITLE. Die Einträge in der Tabelle ADDITIONAL_DATA könnten dann wie folgt aussehen (die Liste enthält zwei Datensätze):

```
METHOD if_ex_qisr1~SCENARIO_SET_ADDITIONAL_VALUES  
    DATA: ls_additional_value TYPE qisrsspecial_param.  
    ls_additional_value-fieldindex = 1.
```

```

ls_additional_value-fieldname = 'FIRSTNAME'.
ls_additional_value-fieldvalue = 'Jim'.
APPEND ls_additional_value TO additional_data.
ls_additional_value-fieldname = 'LASTNAME'.
ls_additional_value-fieldvalue = 'Smith' .
APPEND ls_additional_value TO additional_data.
ls_additional_value-fieldname = 'TITLE'.
ls_additional_value-fieldvalue = 'Mr.'.
APPEND ls_additional_value TO additional_data.
CLEAR ls_additional_value.

ls_additional_value-fieldindex = 2.
ls_additional_value-fieldname = 'FIRSTNAME'.
ls_additional_value-fieldvalue = 'Linda'.
APPEND ls_additional_value TO additional_data.
ls_additional_value-fieldname = 'LASTNAME'.
ls_additional_value-fieldvalue = 'Green' .
APPEND ls_additional_value TO additional_data.
ls_additional_value-fieldname = 'TITLE'.
ls_additional_value-fieldvalue = 'Mrs.'.
APPEND ls_additional_value TO additional_data.
CLEAR ls_additional_value.

```

Listing 6 Programmierung für eine erweiterte Wertheilfe (OVS)

Format von Beträgen und Mengen

Mit SAP ERP 6.0 Support Package 16 (SP 16, SAP_APPL) kann das Format eines Betragsfeldwertes in Abhängigkeit von einer zugehörigen Währung dargestellt werden. Ebenso kann das Format eines Wertes eines Mengenfeldes in Abhängigkeit von einer zugehörigen Mengeneinheit dargestellt werden. Beispielsweise werden viele Währungen mit zwei Nachkommastellen dargestellt. Japanische Yen verwenden jedoch drei Nachkommastellen. Details zu Betrags- und Mengenfeldern im ISR finden Sie in SAP-Hinweis 1350278.

In der QISR1-BAdI-Implementierung muss dazu ein sogenanntes Referenzfeld über die Tabelle `EXTERNAL_DATA` bereitgestellt werden. Dieses Referenzfeld stellt den Datentyp – ob Währung oder Mengeneinheit – sowie den konkreten Wert für die Bestimmung des korrekten Formats des Betrags- oder Mengenfeldes bereit. Da sich das Format der Währung bzw. der Mengeneinheit nach einer Änderung des Wertes auf dem Formular ändern kann, muss auch die Formatierung bei jedem Roundtrip des Formulars angepasst werden.

In [Listing 7](#) wird ein Programmbeispiel der Methode `SCENARIO_INT_SERVICE_REQUEST_CHECK` aus dem BAdI `QISR1` für ein Betragsfeld gezeigt. Der Wert des Merkmals `REVENUE` soll im Format gemäß der Referenzwährung `JPY` (Japanische Yen) korrekt dargestellt werden. Die Einträge in der Tabelle `EXTERNAL_DATA` sollten wie folgt aussehen: Das Feld `QISR_FIELDNAME_REF_TYPE` für den Referenztyp muss für ein Betragsfeld den Wert `C` erhalten. Im Falle eines Mengenfeldes wäre das der Wert `Q`. Die Programmierung kann in allen Methoden ausgeführt werden, die die Tabelle `EXTERNAL_DATA` als Schnittstellenparameter anbieten.

```
METHOD if_ex_qisr1~INT_SERVICE_REQUEST_CHECK
* lokale Daten
DATA: ls_external_value TYPE qisrsspecial_param.

ls_external_value-qisr_fieldname = 'REVENUE'.
ls_external_value-qisr_fieldname_ref = 'JPY'.
ls_external_value-qisr_fieldname_ref_type = 'C'.
APPEND ls_external_value TO external_data.
CLEAR ls_external_value.
```

Listing 7 Programmierung für die Formatierung eines Betragsfeldes



Tabelle »SPECIAL_DATA« verwenden

Falls die zugehörige Währung im Formular eingegeben wurde, können Sie den Wert der Währung aus dem entsprechenden Feld der Tabelle `SPECIAL_DATA` entnehmen. Beachten Sie, dass als Referenz nur die passenden Schlüssel aus dem SAP-Customizing verwendet werden können:

- Die Schlüssel für die Währungen finden Sie in der Transaktion `OY04` (ISO-Codes).
- Die Schlüssel für die Mengeneinheiten finden Sie in der Transaktion `CUNI`.

Sollte die Eingabe auf dem Formular nicht exakt diesem Schlüssel entsprechen, müssen Sie in Ihrer Implementierung ein passendes Datenmapping durchführen.

Beispiel: Auf dem Formular wurde `Yen` als erlaubter Wert eingegeben. Sie müssen diesen Wert vor der in [Listing 7](#) beschriebenen Übergabe an den Schnittstellenparameter `EXTERNAL_DATA` in `JPY` umwandeln.

Formularablauflogik

Zusätzlich zur Definition des Customizings und des Adobe-Formulars können Sie nun die Ablauflogik des Formulars programmatisch steuern. Das ISR-Framework stellt hierfür die vordefinierte BAdI-Definition `QISR1` bereit. Ein wesentlicher Be-

standteil des ISR ist die Flexibilität in der Beeinflussung der Ablauflogik des Antragsformulars. In diesem Abschnitt erfahren Sie anhand von einfachen, typischen Programmbeispielen, wie diese BAdI-Implementierung genutzt werden kann.

Um eine eigene Verarbeitungslogik zu einem Formular hinzuzufügen, wählen Sie im ISR-Grund-Customizing **Business-Add-In definieren** und können nun eine Implementierung anlegen, verändern oder löschen. Standardmäßig kann jedem Szenario eine eigene BAdI-Implementierung der BAdI-Definition QISR1 zugeordnet werden. Bei Bedarf ist es jedoch möglich, eine BAdI-Implementierung für mehrere Szenarien zu nutzen. Entweder Sie ordnen das betreffende zusätzliche Szenario direkt einer BAdI-Implementierung als weiteren Filterwert hinzu, oder Sie tragen unter **Business-Add-In aus Szenario** das Szenario ein, dessen BAdI-Implementierung Sie in dem aktuellen Szenario verwenden wollen. Diese Einstellung gilt nur für BAdI QISR1.

Mehrfache Zuordnung einer BAdI-Implementierung zum Szenario

Die BAdI-Implementierung kann in mehreren Szenarien verwendet werden. Die Verknüpfung eines Szenarios mit der BAdI-Implementierung eines anderen Szenarios im Grund-Customizing ist zu empfehlen, wenn Sie mehrere ähnliche Szenarien haben oder keine Modifikation an der bestehenden BAdI-Implementierung durchführen möchten.

Die Ablauflogik des Formulars mit den Zeitpunkten der Aufrufe der einzelnen Methoden des BAdI QISR1 ist in [Abbildung 41](#) schematisch dargestellt. Weitere Erläuterungen finden Sie in SAP-Hinweis 105505.

Es stehen insgesamt neun BAdI-Methoden zur Verfügung:

- **INT_SERVICE_REQUEST_INIT** (Initialisierung)

Diese Methode wird einmalig beim Aufruf eines Antragsformulars für einen neuen Antrag als erste Methode durchlaufen und für Dateninitialisierungen auf dem Formular verwendet. Folgende Parameterwerte sind dabei bereits festgelegt:

- MODE = "CREATE"
- FORM_VIEW = "ISR_REQUEST"
- EVENT (leer)

SAP-Hinweis 883718 erläutert, wie die Methode auch im Modus **Ändern** bzw. **Anzeigen** verwendet werden kann.

- **SCENARIO_SET_FORM_VIEW** (Formularsicht festlegen)

Diese Methode wird einmalig beim Aufruf eines Antragsformulars nach der Methode INT_SERVICE_REQUEST_INIT durchlaufen. Die Standardwerte für den Parameter FORM_VIEW werden im Abschnitt [»Standardwerte für die Parameter MODE, USER_COMMAND und FORM_VIEW«](#), beschrieben.

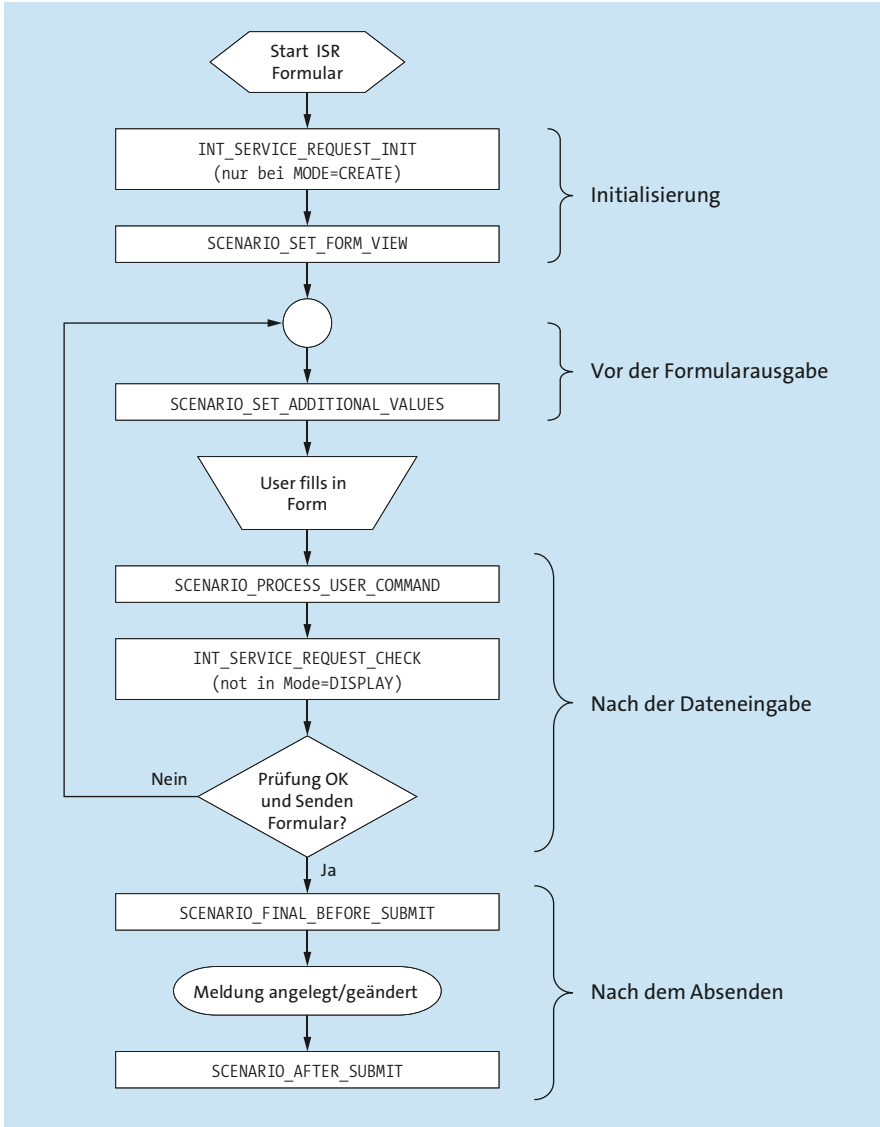


Abbildung 41 Zeitpunkte der Formularablauflogik mit BAdI »QISR1«

■ **SCENARIO_SET_ADDITIONAL_VALUES** (Wertehilfen)

Diese Methode wird beim Starten eines Antragsformulars und vor jeder erneuten Ausgabe des Formulars, zum Beispiel nach einem EVENT, durchlaufen. Füllen Sie in dieser Methode die interne Tabelle `ADDITIONAL_DATA` mit Werten, die nur zur Laufzeit des Formulars benötigt werden, jedoch nicht im Antrag abgespeichert werden sollen. Das heißt, hier können Wertelisten für Dropdown-Listboxen und Steuerparameter gesetzt werden. Die interne Tabelle `ADDITIONAL_DATA` wird nicht auto-

matisch gelöscht, sodass einmal gesetzte Daten bis zum Absenden des Formulars erhalten bleiben.

- **SCENARIO_PROCESS_USER_COMMAND** (eigene User Commands verarbeiten)
Diese Methode wird beim Auslösen eines beliebigen Events auf dem Antragsformular (**Prüfen**, **Absenden**, **User Command**) als Erstes durchlaufen.
- **INT_SERVICE_REQUEST_CHECK** (Prüfen)
Diese Methode wird beim Auslösen eines Standard-Events auf dem Antragsformular (**Prüfen** oder **Absenden** eines Formulars) nach der Methode `SCENARIO_PROCESS_USER_COMMAND` durchlaufen. Die Prüfung der Antragsdaten erfolgt somit immer vor der Fortschreibung der neuen bzw. geänderten Meldung.
- **SCENARIO_FINAL_BEFORE_SUBMIT** (Prüfen vor Submit-Event)
Diese Methode wird unmittelbar vor dem Anlegen der Meldung aufgerufen.
- **SCENARIO_AFTER_SUBMIT** (Exit nach Submit-Event)
Diese Methode wird nach dem Verbuchen der Formulardaten und nach dem Anlegen der Meldung aufgerufen.
- **POST_NOBMP_POSITION** (Daten für Position zur Meldung bestimmen)
Diese Methode dient der Ableitung von Meldungspositionen anhand der Meldungsdaten kurz vor der Verbuchung der Meldung. Sie ist nur relevant, wenn Sie Meldungspositionen verwenden wollen.
- **REVERSE_NOBMP_EXTERNAL** (externe Daten beim Stornieren einer Meldung separat löschen)
Diese Methode bildet insofern eine Ausnahme, als sie nicht durch das Formular prozessiert, sondern beim Stornieren einer Serviceanfrage in der ISR-Statusübersicht genutzt wird, um antrags- und szenariospezifische Datenänderungen außerhalb des eigentlichen Antrags vornehmen zu können.
Das ist beispielsweise der Fall, wenn Sie bei der Antragstellung in Ihrem Szenario zusätzliche Informationen zum Antrag in eine kundeneigene Tabelle geschrieben haben, um später eine eigene Auswertung darüber erstellen zu können. Mit dem Stornieren des Antrags können Sie diese Daten anpassen.

Datenstrukturen innerhalb des ISR-Frameworks

In Tabelle 5 werden die wichtigsten Datenstrukturen für die Implementierung der BAdI-Methoden aufgezählt. Diese werden in den meisten Methoden des BAdI QISR1 und analog auch in den anderen BAdIs des ISR angeboten (siehe Abschnitt »Customizing und Programmierung eines Szenarios«). Die Liste ist jedoch nicht vollständig, da die weiteren Parameter für die meisten Anwendungen ohne Bedeutung sind. Ihre Bedeutung können Sie der Dokumentation des BAdI QISR1 entnehmen.

Parameter	Typ	Beschreibung
FLT_VAL	QSCENARIO	aktuelles Szenario
MODE	QISRDMODE	beschreibt den Bearbeitungsmodus
FORM_VIEW	QISRDFORMVIEW	aktuelle Formularsicht
USER_COMMAND	SYUCOMM	Event (Standard-Events und szenariospezifische Events)
RETURN	BAPIRET1	Rückgabestruktur für Fehlermeldungen
MESSAGE_LIST	QISRTRETURN	Liste mit Meldungen unterschiedlichen Fehlertyps
UI_ATTRIBUTES	QISRTUI_ATTRIBUTES	UI-Eigenschaften eines Formularfeldes
GENERAL_DATA	QISRSGENERAL_PARAM	allgemeine ISR-Daten (Antragsteller etc.)
SPECIAL_DATA	QISRTSPECIAL_PARAM	Formulardaten
ADDITIONAL_DATA	QISRTSPECIAL_PARAM	Werte für Eingabehilfen

Tabelle 5 Schnittstellenparameter der Methoden des BAdI »QISR1«

Standardwerte für die Parameter **MODE**, **USER_COMMAND** und **FORM_VIEW**

Die folgende Aufzählung beschreibt einige wesentliche Parameter aus den BAdIs und die Bedeutung der zugehörigen erlaubten Werte:

■ **MODE**

MODE beschreibt den aktuellen Bearbeitungsmodus. Mithilfe des Parameters **MODE** können Sie Ihre Programmlogik in Abhängigkeit vom aktuellen Modus gestalten:

- **CREATE**, falls der Aufruf des ISR-Formulars für einen neuen Antrag erfolgen soll
- **CHANGE**, falls der Aufruf des ISR-Formulars mit Meldungsnummer zum Ändern erfolgen soll
- **DISPLAY**, falls der Aufruf des ISR-Formulars mit Meldungsnummer zum Anzeigen erfolgen soll

■ **FORM_VIEW**

FORM_VIEW beschreibt die aktuelle Bearbeiterrolle:

- **ISR_APPROVE**, falls der Genehmigende das Formular bearbeitet
- **ISR_REQUEST**, falls der Antragsteller das Formular bearbeitet
- **ISR_PROCESS**, falls der Sachbearbeiter das Formular bzw. die Meldung bearbeitet

■ USER_COMMAND

Die Standard-ISR-Events in USER_COMMAND sind wie folgt definiert:

- SEND, falls das Formular für eine neue Anfrage/Meldung abgeschickt wird
- SAVE, falls das Formular für eine existierende Anfrage/Meldung gesichert wird
- CHECK, falls auf dem Formular die Schaltfläche **Prüfen** gedrückt wird
- ACCEPT, falls auf dem Genehmigungsformular die Schaltfläche **Genehmigt** gedrückt wird
- REJECT, falls auf dem Genehmigungsformular die Schaltfläche **Ablehnen** gedrückt wird
- BACK, falls auf dem Genehmigungsformular die Schaltfläche **Zurück zu Antragsteller** gedrückt wird

Im kundeneigenen Formular können beliebige zusätzliche Events definiert werden, die in den Methoden des BAdI QISR1 zur Laufzeit über den Parameter USER_COMMAND abgefragt werden können (siehe folgender Abschnitt).

Programmbeispiele

In diesem Abschnitt werden einige einfache Programmbeispiele vorgestellt. Sie sollen einen Einstieg in die programmierte Formularablauflogik bieten und dabei die grundlegenden Möglichkeiten des BAdI QISR1 aufzeigen.

Programmbeispiel für eine Feldinitialisierung

Das Merkmal NEW_COSTCENTER wird mit dem ISR-Standardparameter INITIATED_BY-MASTER_CCTR initialisiert (siehe [Listing 8](#)):

```
METHOD if_ex_qisr1~int_service_request_init
* lokale Daten
DATA: ls_special_data TYPE qisrsspecial_param.

LOOP AT special_data INTO ls_special_data.
    CASE ls_special_data-fieldname.
        WHEN 'NEW_COSTCENTER'.
            ls_special_data-fieldvalue =
                general_data-initiated_by-master_cctr.
        WHEN OTHERS.
    ENDCASE.
    MODIFY special_data FROM ls_special_data.
ENDLOOP.
```

Listing 8 Feldinitialisierung

Programmbeispiel für eine Prüfung

Eine Fehlermeldung wird nach einer Prüfung ausgegeben (siehe [Listing 9](#)):

```
METHOD if_ex_qisr1~int_service_request_check
  IF general_data-initiated_by-lastname is initial.
    return-type   = 'E'.
    return-id     = 'QISR'.
    return-number = '046'.
  EXIT.
ENDIF.
```

Listing 9 Programmierung einer Fehlermeldung

Programmbeispiel für ein Formular-Event

Der USER_COMMAND bzw. das EVENT URLB des Formulars wird verarbeitet, falls es sich um einen Neuantrag handelt (siehe [Listing 10](#)):

```
METHOD if_ex_qisr1~scenario_process_user_command.
  DATA:   ls_special_data type qisrsspecial_param.
  DATA:   lv_urlaub(1)   type c.

* nur für Neuanträge
  CHECK mode = 'CREATE'.

* Nutzerbefehle bearbeiten
  CASE user_command.
    WHEN 'URLB'.

* Spezielle Daten URLAUB lesen
    LOOP AT special_data INTO ls_special_data.
      IF ls_special_data-fieldname = 'URLAUB'.
        lv_urlaub = ls_special_data-fieldvalue.
      ENDLLOOP.
    READ TABLE special_data INTO ls_special_data
      WITH KEY fieldname = 'ESS_COSTS'.
    IF sy-subrc EQ 0 AND lv_urlaub eq 'X'.
* Lösche 'ESS_COSTS', falls Urlaub angekreuzt ist
      CLEAR ls_special_data-fieldvalue.
    ENDF.
  ENDF.
```

```

MODIFY special_data FROM ls_special_data
                        INDEX sy-tabix.
ENDCASE.

```

Listing 10 Programmierung eines Events (User Command)

Programmbeispiel für eine Werthilfe

Um eine Werthilfe für ein Merkmal (zum Beispiel SYSTEMNAME) im BAdI QISR1 zu realisieren, müssen die interne Tabelle ADDITIONAL_DATA der Methode SCENARIO_SET_ADDITIONAL_VALUES für den Schlüsselwert (zum Beispiel SYSTEMNAME_KEY) und die zugehörige Beschreibung (zum Beispiel SYSTEMNAME_LABEL) gefüllt werden. Mehrfache Einträge in der Werthilfe werden durch die fortlaufende Nummerierung im Feld FIELDINDEX erreicht.

Der technische Name und die Beschreibung des Schlüsselwertes müssen im Customizing der Merkmale hinterlegt werden (siehe [Abbildung 25](#)). Die korrekte Zuordnung der Eingabewerte erfolgt auf Basis der technischen Namen, deshalb können in der Tabelle ADDITIONAL_DATA die Werte mehrerer Werthilfen gleichzeitig enthalten sein.

Auf dem Formular verwenden Sie entweder das Layoutelement **ISR_DDL_NoSelected** oder **ISR_DDL_1stSelected** aus der ISR-Bibliothek (siehe Abschnitt »[ISR-Bibliothek – spezielle Bildelemente](#)«). In [Abbildung 42](#) sehen Sie das Ergebnis des Programmbeispiels im Adobe-Formular bei Verwendung des Elements **ISR_DDL_NoSelected**. Wie Sie die Dropdown-Listbox als Extended Value Selector (EVS) darstellen können, ist in SAP-Hinweis 1133359 beschrieben.

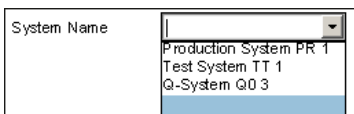


Abbildung 42 Programmierte Werthilfe zur Laufzeit im Adobe-Formular

Für die Definition einer erweiterten Werthilfe verwenden Sie die separate Customizing-Transaktion QISRSCENARIO_OVS statt des ISR-Grund-Customizings. Die Programmierung der Werteliste findet in der Methode SCENARIO_PROCESS_USER_COMMAND statt. Ein Programmbeispiel finden Sie in SAP-Hinweis 906950. Auf dem Adobe-Formular verwenden Sie das Layoutelement **ISR Text Edit – OVS** aus der ISR-Bibliothek.

Für die Anwendung der Werthilfe **ISR Text Edit – Value Help** muss diese Methode nicht implementiert werden. Mit dieser Werthilfe erhalten Sie automatisch Domänenfestwerte bzw. eine Werthilfe, die direkt einem Datenelement zugeordnet ist. Voraussetzung ist, dass das Merkmal im ISR-Customizing dem entsprechenden Datenelement zugeordnet ist.

Eine Werthilfe mit zwei Auswahlwerten für das Merkmal System wird dann wie in [Listing 11](#) programmiert:

```
if_ex_qisr1~scenario_set_additional_values
  DATA: ls_additional_value TYPE qisrsspecial_param.

* erster DDLB-Eintrag
  ls_additional_value-fieldindex = 1.
  ls_additional_value-fieldname  = 'SYSTEMNAME_KEY'.
  ls_additional_value-fieldvalue = 'PR1'.
  APPEND ls_additional_value TO additional_data.
  ls_additional_value-fieldname  = 'SYSTEMNAME_LABEL'.
  ls_additional_value-fieldvalue = 'Produktivsystem PR 1'.
  APPEND ls_additional_value TO additional_data.

* zweiter DDLB-Eintrag
  ls_additional_value-fieldindex = 2.
  ls_additional_value-fieldname  = 'SYSTEMNAME_KEY'.
  ls_additional_value-fieldvalue = 'TT1'.
  APPEND ls_additional_value TO additional_data.
  ls_additional_value-fieldname  = 'SYSTEMNAME_LABEL'.
  ls_additional_value-fieldvalue = 'Testsystem TT 1'.
  APPEND ls_additional_value TO additional_data.
```

Listing 11 Programmierung einer Werthilfe für Dropdown-Listbox

Lesen und Setzen der speziellen Antragsfelder in BAdI-Methoden

Innerhalb der BAdI-Implementierungen kann auf die allgemeinen Antragsdaten bequem zugegriffen werden, da diese als Struktur GENERAL_DATA in die Methoden gereicht werden. Die im ISR-Customizing als Merkmale definierten speziellen Antragsdaten hingegen liegen als Tabelle vor, und für jedes Merkmal enthält die Tabelle SPECIAL_DATA eine Zeile.

Wie Sie in den einfachen Beispielen gesehen haben, wird das Coding bei komplexeren Programmieraufgaben schnell unübersichtlich, wenn Sie einzelne Feldwerte immer über den ABAP-Befehl LOOP AT special_data finden müssen. Deshalb stehen zwei Funktionsbausteine zur Verfügung, um diese tabellarischen Daten am Anfang einer BAdI-Methode in entsprechende lokale Strukturen umzufüllen und am Ende der Methode wieder zurück in die Tabelle zu schreiben:

■ ISR_SPECIAL_DATA_TO_STRUC

Dieser Funktionsbaustein dient dem Umfüllen der speziellen Antragsdaten von der Tabelle SPECIAL_DATA in eine Struktur. Dabei werden diejenigen Antragsdaten

aus der Tabelle in die Struktur übertragen, deren Name mit einem Feldnamen der Struktur übereinstimmt (siehe [Listing 12](#)).

```
DATA: ls_master_data TYPE bapi2075_7.
* spezielle Antragsdaten in Struktur füllen
CALL FUNCTION 'ISR_SPECIAL_DATA_TO_STRUC'
  EXPORTING
    it_special_data = special_data
  CHANGING
    cs_data         = ls_master_data.
* weitere Verarbeitungsschritte
* ...
* Zurückfüllen der speziellen Antragsdaten
CALL FUNCTION 'ISR_STRUC_TO_SPECIAL_DATA'
  EXPORTING
    is_data         = ls_master_data
  CHANGING
    ct_special_data = special_data.
```

Listing 12 Programmierung der Überführung von Daten

■ ISR_STRUC_TO_SPECIAL_DATA

Dieser Funktionsbaustein funktioniert analog zu ISR_SPECIAL_DATA_TO_STRUC, nur in die andere Richtung.

Dieselben Funktionsbausteine können Sie natürlich auch zum Lesen und Schreiben der Übergabetabelle ADDITIONAL_DATA für die Wertheilfe verwenden.

Alternativ zu diesen beiden Funktionsbausteinen können Sie die Methoden der Hilfsklasse CL_ISR_DATA einsetzen. Diese Klasse bietet neben den genannten Möglichkeiten zum Datentransfer in die lokale Struktur zusätzlich die Option, einfache Datenmanipulationen wie Anfügen, Einfügen und Entfernen vorzunehmen (siehe [Listing 13](#)).

METHOD if_ex_qisr1~int_service_request_check.

```
DATA: BEGIN OF ls_characteristics,
      company_code TYPE bukrs,
      cost_center  TYPE kostl,
    END OF ls_characteristics.

* Übernahme der Formulardaten in Klassenspeicher
* CL_ISR_DATA
CALL METHOD cl_isr_data=>set
  EXPORTING
```



```
it_data = special_data.
```

* Lesen der Formulardaten in lokale Struktur

```
CALL METHOD cl_isr_data=>read
EXPORTING
  id_index      = 1
IMPORTING
  es_data_record = ls_characteristics.
```

* Datenänderungen

```
IF ls_characteristics-cost_center = '12345678'.
  ls_characteristics-company_code = '1000'.
ELSE.
  ls_characteristics-company_code = '0001'.
ENDIF.
```

* Ersetzen der Daten

```
CALL METHOD cl_isr_data=>insert
EXPORTING
  is_data_record = ls_characteristics
  id_index      = 1
RECEIVING
  rt_data      = special_data.
```

```
ENDMETHOD.
```

Listing 13 Programmierung mit Hilfsklasse »CL_ISR_DATA«

Behandlung von Fehlern bei der Prüfung

Sie können innerhalb der Methoden `INT_SERVICE_REQUEST_CHECK` und `SCENARIO_PROCESS_USER_COMMAND` Fehlermeldungen ausgeben, die die Weiterbearbeitung der Serviceanfrage unterbinden. Dazu füllen Sie die Übergabestruktur `RETURN` mit einer entsprechenden Fehlermeldung. Möchten Sie mehrere Meldungen ausgeben, verwenden Sie die Übergabetabelle `MESSAGE_LIST`.



Die Fehlerbehandlung steuert die weitere Bearbeitung des Antrags

Bei der Verwendung der Web-Dynpro-Applikationen im ISR können Sie die Weiterbearbeitung des Antrags durch den Antragsteller steuern. Enthält die Übergabetabelle `MESSAGE_LIST` eine oder mehrere Meldungen, so wird eine gleichzeitig ausgegebene Fehlermeldung aus der Struktur `RETURN` nicht angezeigt, die Meldungen der `MESSAGE_LIST` werden dagegen immer angezeigt. Die Fehlermeldung in der

Struktur RETURN führt jedoch dazu, dass der Antrag nicht abgeschickt werden kann. Bleibt die Struktur RETURN leer, kann der Antrag trotz Meldungen in der Übergabetabelle MESSAGE_LIST unabhängig von deren Meldungstyp abgeschickt werden.

Feld-Layout-Steuerung

Im ISR-Framework können die Layouteigenschaften von einzelnen Formularfeldern wie Mussfeld, Sichtbarkeit und Eingabebereitschaft gesteuert werden. Voraussetzung ist das Support Package 4 von SAP ERP 6.0. Nähere Informationen dazu finden Sie in SAP-Hinweis 925657. Die Steuerung der Layouteigenschaften ist in den folgenden Methoden möglich:

- INT_SERVICE_REQUEST_CHECK
- INT_SERVICE_REQUEST_INIT
- SCENARIO_SET_ADDITIONAL_VALUES
- SCENARIO_PROCESS_USER_COMMAND

Sie verfügen über den Schnittstellenparameter UI_ATTRIBUTES. Diese Tabelle ist analog zur Tabelle ADDITIONAL_DATA, die während der ISR-Laufzeit grundsätzlich leer ist und explizit für die entsprechenden Merkmale (FIELDNAME) gefüllt werden muss. Sie können jedem Feld des Formulars über das Feld UI_ATTRIBUTE die folgende Eigenschaft zuweisen:

- **M** Mussfeld und Eingabebereit
- **I** Eingabebereit
- **R** Sichtbar und nicht eingabebereit
- **H** Nicht Sichtbar

Diese Eigenschaft kann nach jedem Standard-Event oder einer Aktion auf dem Formular, bei der ein Event ausgeführt wird, angepasst werden.

ISR-Architektur

Dieser Abschnitt veranschaulicht schematisch die Architektur des ISR-Frameworks und gibt Ihnen einen Überblick über die verwendeten SAP-Technologien und ihre jeweilige Einbindung in das ISR-Framework. Darüber hinaus beschreibt es die Kommunikation zwischen den einzelnen Frontend-Komponenten und dem Backend sowie die wesentlichen Schnittstellen des ISR, insbesondere die Web-Dynpro-Anwendungen.

ISR als Toolset

Wie bereits erläutert, bedient sich der ISR verschiedener Anwendungen aus SAP Net-Weaver und SAP ERP und verbindet diese zu einem Framework, das formularbasierte Antragsszenarien und den damit verbundenen Prozess zur Durchführung des gewünschten Service ermöglicht. Für die Abwicklung des Prozesses, das heißt den Transport der Anfrage zu den zuständigen Bearbeitern, wird eine Integration mit SAP Business Workflow angeboten. Das wird durch die Verwendung des Business-Objekts BUS7051 der Qualitätsmeldung – oder kurz Meldung – als Laufzeitobjekt des Serviceantrags erreicht.

Darüber hinaus ist das Antragsszenario in das SAP-Controlling zur Bestimmung direkter und indirekter Gemeinkosten der Serviceanfrage integriert. ISR verbindet diese verschiedenen Anwendungen und stellt einige Schnittstellen zur Verfügung, um das Antragsformular und den Prozess flexibel programmatisch steuern zu können.

Abbildung 43 zeigt schematisch die Kommunikation zwischen der Web-Dynpro-Component QISR_UI_FORM und dem Backend. Die Web-Dynpro-Component kommuniziert dabei über RFC ISR_PROCESS_EVENT mit der ISR-Laufzeit im Backend. Innerhalb der ISR-Laufzeit werden unter anderem die Methoden des BAdI QISR1 (siehe Abschnitt »Datenstrukturen innerhalb des ISR-Frameworks«) zu verschiedenen Zeitpunkten aufgerufen.

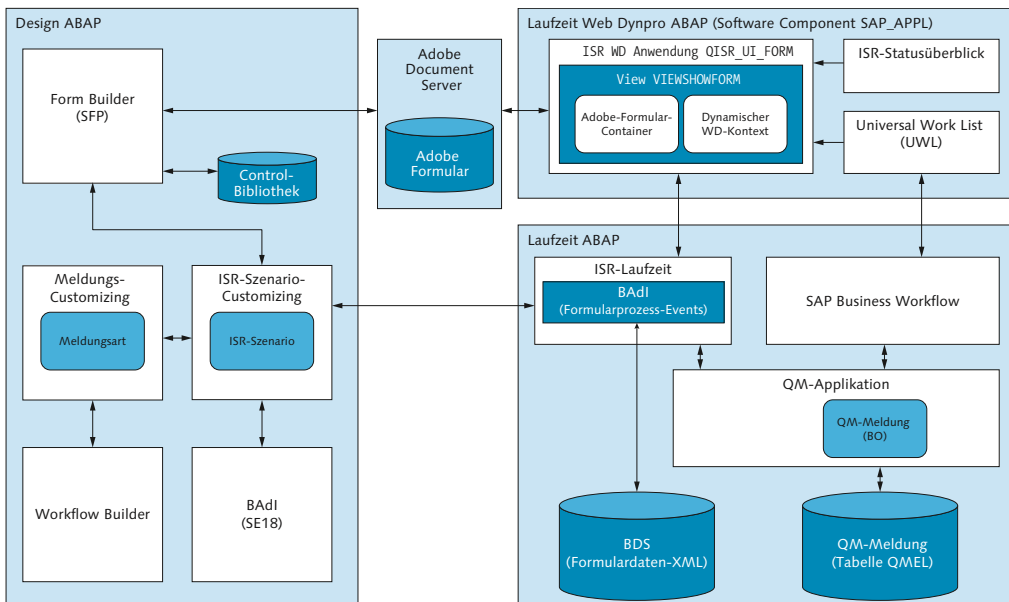


Abbildung 43 Kommunikation zwischen ABAP-Designzeit und ABAP-Laufzeit

Beim Abschicken des Formulars werden die Daten des Formulars im Business Document Service (BDS), der Dokumentenverwaltung des Knowledge Providers, separat abgespeichert, und die Meldung wird angelegt. Die Daten werden demnach nicht gemeinsam mit dem Formular als PDF gespeichert. Dieses Programmdesign ist auch der Grund dafür, dass der ISR keine Offline-Erfassung des Antrags unterstützen kann. Das Adobe-Formular dient im Rahmen des ISR nur als Oberflächenlayout zur Erfassung des Antrags und wird zur Laufzeit immer mit den aktuellen Daten abgemischt.

Datenhaltung getrennt vom Layout

Nur die Formulardaten werden abgespeichert, nicht das PDF. Das Adobe-Formular dient technisch gesehen ausschließlich als Eingabemaske. Natürlich kann der Anwender immer noch im Adobe Reader ein aktuelles PDF abspeichern.



Beim Anlegen der Meldung wird das Event `CREATED` des Business-Objekts ausgelöst, was zu einem entsprechenden Workitem in der Universal Worklist (im Portal) oder im SAP Business Workplace (Transaktion SBWP) führt.

Web-Dynpro-ABAP-Component und ihre Anwendung

Wie im Abschnitt »Erfassungsart« beschrieben wurde, ist die UI-Technologie zur Erfassung des Antrags im Prinzip austauschbar. Neben dem seit SAP ERP 5.0 verfügbaren interaktiven Adobe-Formular besteht grundsätzlich die Möglichkeit, andere Formulartechnologien wie ITS Services, BSP-Formulare oder JSP-Formulare einzusetzen. Für alle Antragsformulare seit SAP ERP 5.0 ist jedoch das Adobe-Formular die Standardlösung des ISR und in eine Web-Dynpro-Component eingebettet (siehe Abbildung 44).

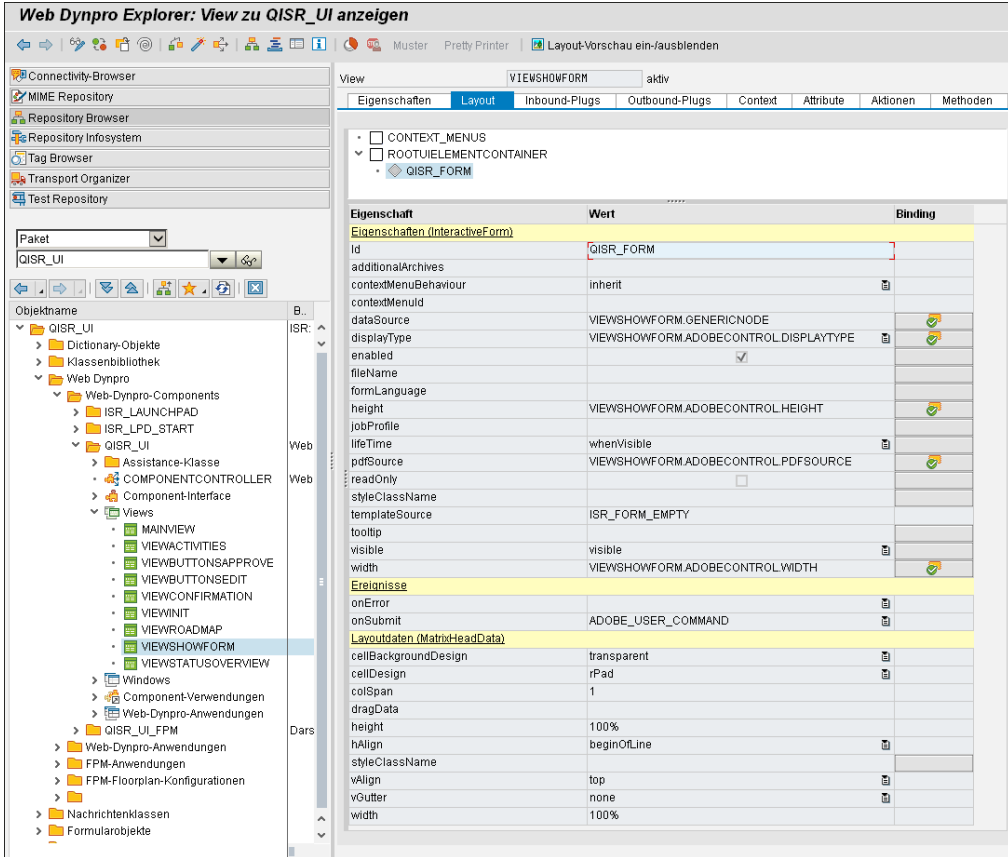


Abbildung 44 Web-Dynpro-Component »QISR_UI«

Grundlagen der Web-Dynpro-Component des ISR

Für die Integration des Adobe-Formulars stehen Web-Dynpro-Anwendungen, basierend auf Web Dynpro ABAP und Web Dynpro Java, zur Verfügung:

- Web-Dynpro-ABAP-Component QISR_UI
- Web-Dynpro-Java-Component PCUI_GP~ISR

Beide Lösungen bieten in wesentlichen Teilen die gleichen Funktionen, insbesondere bei der Darstellung des Formulars gibt es keine Unterschiede.

In beiden Technologien existieren jeweils vier Web-Dynpro-Anwendungen mit den gleichen Parametern. Die Web-Dynpro-ABAP-Component QISR_UI steht ab Support Package 8 von SAP ERP 6.0 zur Verfügung. Zu empfehlen ist jedoch mindestens die Verwendung von SAP ERP SAP_APPL SP 12. Die Web-Dynpro-Java-Component gibt es bereits seit SAP ERP 5.0, und sie wird in der Softwarekomponente PCUI_GP ausgeliefert.



Auswahlmöglichkeit für Web Dynpro ABAP bzw. Java

Sie werden sich vielleicht schon gefragt haben, warum es im ISR-Customizing zwar eine Erfassungsart **Adobe PDF** gibt, aber keine Auswahlmöglichkeit für Web Dynpro ABAP oder Java.

Jedes Szenario, das im Customizing die Erfassungsart **Adobe PDF** verwendet und dem ein interaktives Adobe-Formular zugeordnet ist, ist grundsätzlich mit beiden Web-Dynpro-UI-Technologien – ohne weitere Einstellungen im Backend – lauffähig. Aus diesem Grund ist für die Unterscheidung zwischen Web Dynpro ABAP und Web Dynpro Java kein Customizing-Eintrag möglich und nötig. Sie entscheiden im Prinzip durch Ihre iView oder Ihre Web-Dynpro-Component, welche Art von Web-Dynpro-Technologie Sie verwenden möchten.

Im Folgenden wird nur die Lösung Web Dynpro ABAP erläutert. Die meisten ausgelieferten ISR-Szenarien werden im Übrigen mit der Web-Dynpro-ABAP-Component `QISR_UI` gestartet. Nur die Anwendung Personnel Change Request (PCR) benutzt die Komponente Web Dynpro Java des ISR in SAP ERP 6.0.

Grundprinzip des ISR-Frameworks ist es, dass die Formulardaten getrennt vom Formular verarbeitet und gespeichert werden. Das interaktive Formular dient zur Laufzeit sozusagen nur als Eingabemaske. Die Grundlage für eine korrekte Datenweitergabe zwischen den verschiedenen Komponenten des ISR ist deshalb die namensgleiche Definition der Formularfelder im ISR-Customizing und im Kontext des Formulars (siehe Abschnitt »Aufbau der Schnittstelle und des Formularkontextes«).

Die Web-Dynpro-Component `QISR_UI` verwendet als Bildelement `InteractiveForm`. Sowohl das Attribut `dataSource` für die Formulardaten als auch das Attribut `pdfSource` sind jeweils an ein Kontextelement der Web-Dynpro-Component gebunden, die erst beim Starten der Komponente gefüllt wird.

Der Kontext der Web-Dynpro-Component für die Daten des Formulars wird zur Laufzeit generiert. Grundlage der Generierung ist die Liste der Merkmale aus dem ISR-Customizing, die mit einem Remote Function Call (RFC) ausgelesen wird, aus dem der passende Kontext für das Formular generiert wird. Zu Beginn der Formularlaufzeit wird über eine passende Web-Dynpro-Schnittstelle der Name des Formulars (zum Beispiel `ISR_FORM_SRCCR`) bestimmt und an das Web-Dynpro-Bildelement `InteractiveForm` übergeben. Das zugehörige interaktive PDF wird von den ADS gelesen und zusammen mit den Daten des Kontextes angezeigt.

Web-Dynpro-Anwendungen des ISR

Die Standardoberfläche des ISR wird durch vier Web-Dynpro-Anwendungen bereitgestellt, die Sie unabhängig voneinander in Ihre Portalumgebung einbauen können.

Beachten Sie, dass alle Anwendungen mit bestimmten Parametern gestartet werden müssen.

■ **QISR_UI_FORM**

Die Web-Dynpro-Anwendung QISR_UI_FORM in Form einer aus drei Schritten bestehenden Roadmap und den SAP Interactive Forms by Adobe wird für die Antragstellung verwendet (siehe [Abbildung 3](#)). Notwendige Parameter sind:

- SCENARIO
ISR-Szenario, immer anzugeben
- NOBMP_NO
Meldungsnummer
- MODE
Modus

Die Meldungsnummer muss natürlich nur im Fall der Änderung (Parameter MODE = CHANGE) oder Anzeige (Parameter MODE = DISPLAY) eines existierenden Antrags angegeben werden. Im Fall eines Neuantrags (Parameter MODE = CREATE) bleibt NOBMP_NO leer.

■ **QISR_UI_FORM_APPROVE**

Die Anwendung QISR_UI_FORM_APPROVE besteht nur aus den SAP Interactive Forms by Adobe und wird für die Genehmigung des Antrags verwendet. Eine Roadmap wird nicht benutzt.

Diese Anwendung ist natürlich nur dann sinnvoll verwendbar, wenn Sie eine Meldungsart mit Genehmigung verwenden. Dem Genehmigenden stehen die Schaltflächen **Genehmigen**, **Ablehnen** und **Zurück zu Antragsteller** zur Verfügung (siehe [Abbildung 10](#)).

Der einzig notwendige Parameter lautet: WF_ID (technische Nummer des Workflow-Items des Genehmigenden).

In der Standardauslieferung der ISR-Szenarien sind alle Formularfelder dieser Anwendung nicht eingabebereit, durch die Anpassung der Skriptprogrammierung einzelner Felder im Form Builder können beliebige Felder jedoch eingabebereit gemacht werden (siehe Abschnitt [»ISR-Bibliothek – spezielle Bildschirmelemente«](#)).

■ **QISR_UI_FORM_DISPLAY_ONLY**

Die Anwendung QISR_UI_FORM_DISPLAY_ONLY dient ausschließlich der Anzeige des aktuellen Status eines ISR-Formulars. Sie beinhaltet keine Roadmap und keine Funktionen in Gestalt von Schaltflächen, sondern nur das Formular ohne Eingabemöglichkeit. Notwendige Parameter sind:

- SCENARIO – ISR-Szenario
- NOBMP_NO – Meldungsnummer

■ QISR_UI_STATUSOVERVIEW

Die Anwendung QISR_UI_STATUSOVERVIEW gibt dem Antragsteller eine Übersicht über all seine Anträge der letzten 365 Tage und ihres Status (siehe [Abbildung 8](#)). Die Anwendung ist ohne zusätzlichen Parameter aufrufbar.

ISR_SCENARIO_PARAMS für beliebige Parameter

Mithilfe des Parameters ISR_SCENARIO_PARAMS können einzelne oder mehrere der im Szenario definierten Merkmale initialisiert werden. Dieser Parameter wird zum Beispiel auch verwendet, um die Version des Szenarios festlegen zu können (siehe Abschnitt »[Applikation und Version](#)«).

Zur Initialisierung mehrerer Felder trennen Sie die jeweiligen Kombinationen von Feld und Wert durch das Dollarzeichen (\$). Lassen Sie keine Leerzeichen zwischen diesen Kombinationen, sondern übergeben Sie die Werte als fortlaufenden String. Voraussetzung ist, dass der Parameter als Merkmal im ISR-Customizing definiert sein muss, er muss aber kein Formularfeld sein. Sie können ISR_SCENARIO_PARAMS bei allen drei Verwendungsmöglichkeiten des ISR in SAP Enterprise Portal (siehe folgender Abschnitt) einsetzen – je nach Technik als URL-Parameter (iView) oder über entsprechende Interface-Methoden des ISR (Web-Dynpro-Component).

Dazu ein Beispiel: Das Merkmal ORDER soll mit dem Wert 000000123456 und das Merkmal ACCOUNT mit dem Wert 400100 initialisiert werden. Daraus ergibt sich folgender, zusammengefügt Parameter für ISR_SCENARIO_PARAMS: ISR_SCENARIO_PARAMS=ORDER=000000123456\$ACCOUNT=400100.



Verwendungsmöglichkeiten in SAP Enterprise Portal

Die Formulare des ISR können dem Anwender in SAP Enterprise Portal auf unterschiedliche Weise angeboten werden:

- direkte Verwendung einer ISR-Web-Dynpro-Anwendung (zum Beispiel QISR_UI_FORM) per Link
- ISR Launchpad, eingebunden in eigene Web-Dynpro-Component
- Component-Verwendung der Web-Dynpro-Component QISR_UI

ISR-Web-Dynpro-Anwendung in einem iView

Bei der Verwendung einer ISR-Web-Dynpro-Anwendung kann der Antrag als direkt aufrufbarer Service im Portal angeboten werden. Das entspricht in etwa einem Link auf einen iView, der die Anwendung in Web Dynpro ABAP (oder Web Dynpro Java) des ISR startet. Die Anwendung muss mit den Parametern SCENARIO und MODE gestartet werden, wobei SCENARIO dem vierstelligen Szenarionamen entspricht, zum Beispiel SMC1, für einen Neuantrag muss MODE = CREATE angegeben werden. Gültige Werte

für den Parameter `MODE` können Sie Abschnitt »Standardwerte für die Parameter `MODE`, `USER_COMMAND` und `FORM_VIEW`«, entnehmen.

Eine Möglichkeit der Darstellung im Portal ist es, eine der zugehörigen Web-Dynpro-Applikationen direkt in einen iView zu inkludieren. Sowohl die Web-Dynpro-Component des ISR-Statusüberblicks als auch die Universal Worklist greifen auf diese Web-Dynpro-Anwendungen des ISR zu (siehe Abbildung 45).

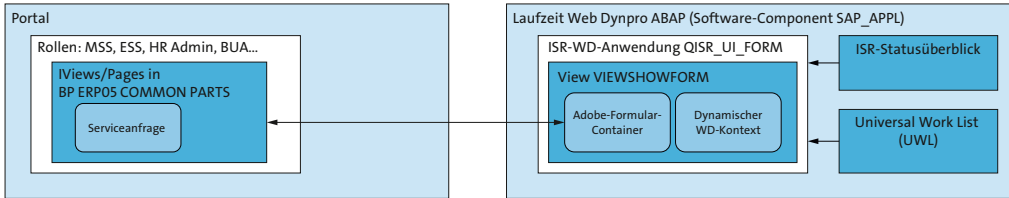


Abbildung 45 Kommunikation zwischen SAP Enterprise Portal und Web-Dynpro-Laufzeit

Ausgelieferte generische iViews des ISR finden Sie in SAP Enterprise Portal unter **Portal Content • ERP Common Parts**.

ISR Launchpad

Ein oder mehrere Antragsformulare können Sie über eine Reihe von Schaltflächen innerhalb einer eigenen Komponente in Web Dynpro ABAP aufrufen (siehe Abbildung 46).

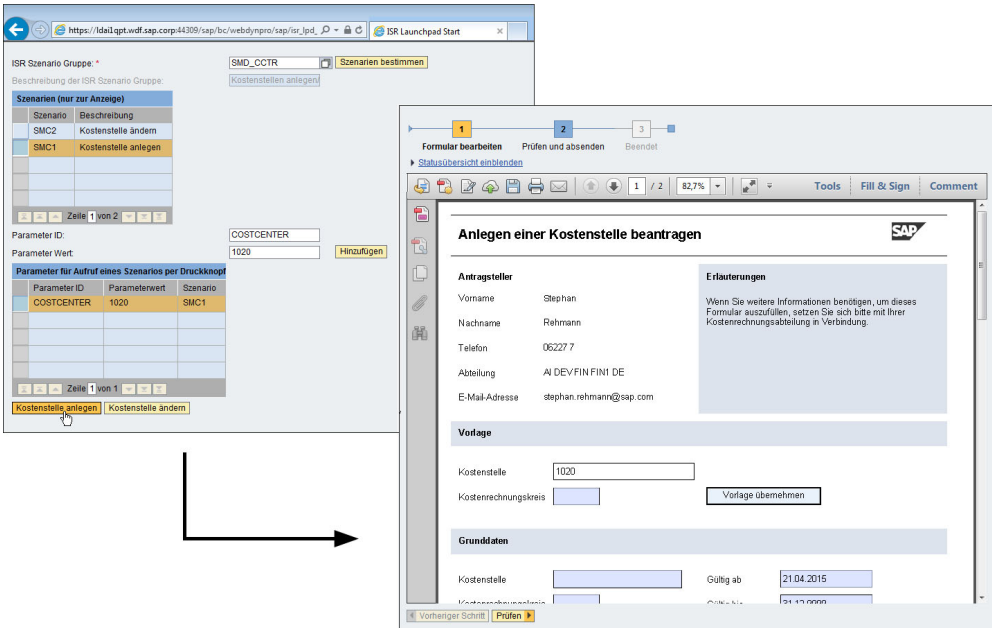


Abbildung 46 Start eines ISR-Formulars aus der Testanwendung »ISR_LPD_START«

Der Vorteil des ISR Launchpads ist, dass beim Starten des Antragsformulars beliebige URL-Parameter mitgegeben werden können, die in der Initialisierungsmethode der BAdI-Implementierung QISR1 ausgewertet werden können. Außerdem kann vor dem Starten des Formulars eine Prüfung im Backend durchgeführt werden, zum Beispiel ob der Anwender berechtigt ist, das Antragsformular zu starten.

Die Methode SCENARIO_CHECK des BAdI QISR6 wird im Funktionsbaustein ISR_LAUNCHPAD_SCENARIO_CHECK aufgerufen. Dieser Funktionsbaustein wird vor dem Start eines Antragsformulars im ISR Launchpad durchlaufen, dies gilt sowohl für die Variante Web Dynpro Java als auch Web Dynpro ABAP des ISR. Diese Prüfung dient dazu, das Starten des Formulars, zum Beispiel aus Gründen fehlender Berechtigung, unterbinden zu können.

Wenn Sie die dritte Variante der Einbindung des ISR verwenden (QISR_UI als Component-Verwendung), können Sie natürlich diese Prüfungsmöglichkeit auch in der eigenen Web-Dynpro-Component aufnehmen. Bauen Sie dazu einfach den Funktionsbaustein ISR_LAUNCHPAD_SCENARIO_CHECK ein.

Programmatische Einbindung des ISR Launchpads

Diese Art der Einbindung der ISR-Anwendung (siehe [Abbildung 47](#)) ist beispielhaft in der Web-Dynpro-Component ISR_LPD_START ausprogrammiert und kann über die gleichnamige Anwendung in Web Dynpro ABAP gestartet werden.

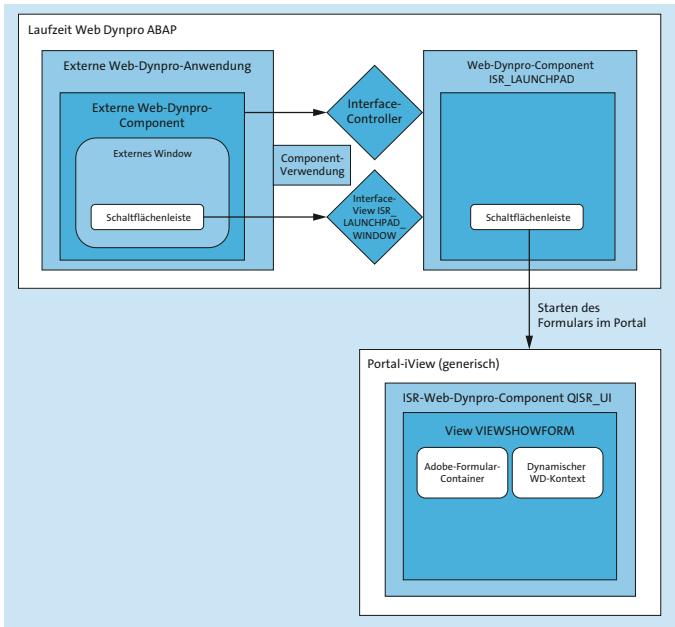


Abbildung 47 Schematische Darstellung des ISR Launchpads in eigener Web-Dynpro-Anwendung

Im Folgenden finden Sie die Schritte zur Einbindung des ISR Launchpads:

1. Zuerst tragen Sie in der Transaktion SE80 in der umgebenden Web-Dynpro-Component die Web-Dynpro-Component ISR_LAUNCHPAD als **Component** ein (siehe Abbildung 48).

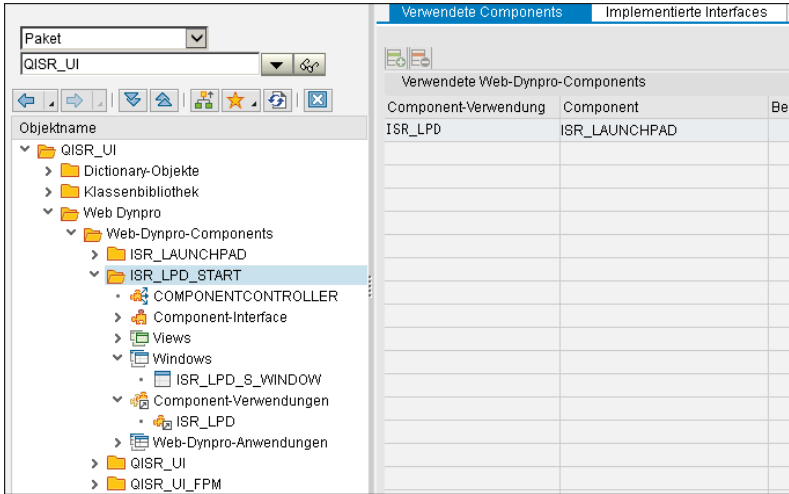


Abbildung 48 ISR_LAUNCHPAD als Component in der Beispielkomponente »ISR_LPD_START«

2. In den VIEWCONTAINER eines passenden WINDOW binden Sie dann die Interface-View ISR_LAUNCHPAD_WINDOW ein (siehe Abbildung 49).

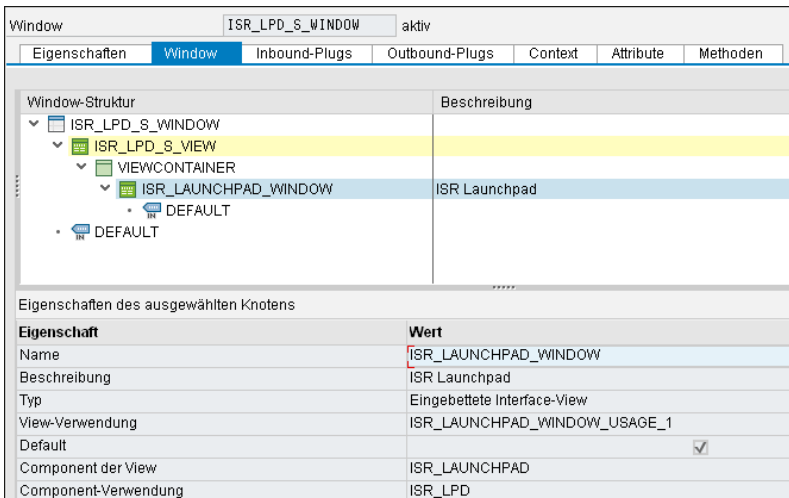


Abbildung 49 Interface-View »ISR_LAUNCHPAD_WINDOW«

- Legen Sie eine neue ABAP-Klasse (Provider-Klasse) an, die das Interface IF_ISR_LPD_PROVIDER implementiert. Diese Klasse benötigen Sie, um das Launchpad mit den gewünschten Parametern starten zu können (siehe [Abbildung 50](#)).

Klasse/Interface		CL_ISR_LPD_START_PROVIDER		realisiert / Aktiv																							
Eigenschaften	Interfaces	Friends	Attribute	Methoden	Ereignisse	Typen	Aliasse																				
<table border="1"> <thead> <tr> <th>Methode</th> <th>Art</th> <th>Sichtbark..</th> <th>M...</th> <th>Beschreibung</th> </tr> </thead> <tbody> <tr> <td>IF_ISR_LPD_PROVIDER-GETPARAMETERS</td> <td>Instance Method Public</td> <td></td> <td></td> <td>Parameter für ISR Aufruf</td> </tr> <tr> <td>IF_ISR_LPD_PROVIDER-GETSYSTEMALIAS</td> <td>Instance Method Public</td> <td></td> <td></td> <td>System Alias</td> </tr> <tr> <td>CONSTRUCTOR</td> <td>Instance Method Public</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>								Methode	Art	Sichtbark..	M...	Beschreibung	IF_ISR_LPD_PROVIDER-GETPARAMETERS	Instance Method Public			Parameter für ISR Aufruf	IF_ISR_LPD_PROVIDER-GETSYSTEMALIAS	Instance Method Public			System Alias	CONSTRUCTOR	Instance Method Public			
Methode	Art	Sichtbark..	M...	Beschreibung																							
IF_ISR_LPD_PROVIDER-GETPARAMETERS	Instance Method Public			Parameter für ISR Aufruf																							
IF_ISR_LPD_PROVIDER-GETSYSTEMALIAS	Instance Method Public			System Alias																							
CONSTRUCTOR	Instance Method Public																										

Abbildung 50 Provider-Klasse mit Interface

- Erzeugen Sie dann eine Instanz der Provider-Klasse, und initialisieren Sie das ISR Launchpad mit der benötigten Szenariogruppe in der View der umgebenden Anwendung. In der exemplarischen Web-Dynpro-Component ISR_LPD_START wurde zu diesem Zweck eine Methode INIT im COMPONENTCONTROLLER angelegt, die in der Methode WDDOMODIFY der View ISR_LPD_S_VIEW aufgerufen wird. Natürlich kann die Implementierung der Methode INIT in Ihrer Web-Dynpro-Component stattdessen auch direkt in der Methode WDDOINIT im COMPONENTCONTROLLER definiert werden, wenn sich die Szenariogruppe während der Anwendung nicht ändert (siehe [Abbildung 51](#)).

Component-Controller		COMPONENTCONTROLLER		aktiv																	
Eigenschaften	Context	Attribute	Ereignisse	Methoden																	
<table border="1"> <thead> <tr> <th>Methoden-Liste</th> <th>Methode</th> </tr> </thead> <tbody> <tr> <td>Methode</td> <td>INIT</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>Parameter</th> <th>Dekl...</th> <th>Reffo</th> <th>Opt</th> <th>Bezugstyp</th> <th>Default-Wert</th> </tr> </thead> <tbody> <tr> <td>ID_SCENARIOGROUP</td> <td>Impor...</td> <td><input type="checkbox"/></td> <td><input type="checkbox"/></td> <td>QISRSCENARIOGROUP</td> <td></td> </tr> </tbody> </table> <pre> 1 METHOD init . 2 3 * provider class 4 DATA: lr_provider TYPE REF TO ci_isr_lpd_start_provider. 5 6 * ISR Launchpad 7 DATA: l_intf_controller TYPE REF TO iwcs_isr_launchpad, 8 lr_object TYPE REF TO object. 9 10 11 12 * 1. Initialize reference to provider class 13 * transfer the CONTEXT of the view to the provider class 14 CREATE OBJECT lr_provider EXPORTING ir_context_node = wd_context. 15 16 * 2. Initialize ISR launchpad 17 wd_this->l_usage = wd_this->wd_cpuse_isr_lpd(). 18 19 lr_object = wd_this->l_usage->get_interface_controller(). 20 21 l_intf_controller ?= lr_object. 22 23 l_intf_controller->init_lpd(id_scenariogroup = id_scenariogroup 24 ir_provider = lr_provider). 25 26 ENDMETHOD. </pre>						Methoden-Liste	Methode	Methode	INIT	Parameter	Dekl...	Reffo	Opt	Bezugstyp	Default-Wert	ID_SCENARIOGROUP	Impor...	<input type="checkbox"/>	<input type="checkbox"/>	QISRSCENARIOGROUP	
Methoden-Liste	Methode																				
Methode	INIT																				
Parameter	Dekl...	Reffo	Opt	Bezugstyp	Default-Wert																
ID_SCENARIOGROUP	Impor...	<input type="checkbox"/>	<input type="checkbox"/>	QISRSCENARIOGROUP																	

Abbildung 51 Initialisierung des ISR Launchpads im »COMPONENTCONTROLLER«

5. Implementieren Sie die Methode `IF_ISR_LPD_PROVIDER~GET_PARAMETERS`, um die gewünschten Aufrufparameter an den ISR-Antrag zu übergeben. Die Methode hat den `EXPORTING`-Parameter `ET_PARAMS` vom Typ `QISRTPD_SCEN_PARAM`, der das Feld `SCENARIO` enthält, das wiederum eine Tabelle mit folgenden Feldern beinhaltet:

- PARAM_INDEX
- PARAM_KEY
- PARAM_VALUE

Somit können Sie für ein oder mehrere Szenarien eine Liste von selbst definierten Parametern bestimmen. `PARAM_KEY` entspricht dem technischen Namen des Parameters, `PARAM_VALUE` dem Wert des Parameters. Wollen Sie nur einen Parameterwert übergeben, geben Sie `PARAM_INDEX` mit '1' an.

Customizing des ISR Launchpads

Im Customizing zum ISR Launchpad gehen Sie folgendermaßen vor:

1. Zunächst definieren Sie eine sogenannte **ISR-Szenariogruppe** (siehe [Abbildung 52](#)).

Szenariogruppe	
ISR-Szenariogruppe	Beschreibung Gruppe
CCAC_EXP	
SBUDGETCONSUMPTIONO	Budget Consumption für Aufträge
SBUDGETCONSUMPTIONP	Budget Consumption für Projekte
SCOSTCENTERREVIEW	Kostenstellenmonitor
SCOSTREVIEW	Kostenmonitor
SEQUIEMONITOR	Szenarien Equipmentmonitor
SEQUIEMONITORREVIEW	Szenarien Equipmentmonitor Exp-planning
SINTERNALORDERREVIEW	Innenauftragsmonitor
SLINEITEMMONITORC	Auffällige Buchungen für Kostenstellen
SLINEITEMMONITORO	Auffällige Buchungen für Innenaufträge
SMASTERDATAMONITOR	Stammdatenmonitor
SMD_ALL	Stammdaten anlegen
SMD_CCTR	Kostenstellen anlegen/ändern
SMD_ORDER	Innenaufträge anlegen/ändern
SMD_PDEF	Projektdefinition anlegen

Abbildung 52 Customizing der Szenariogruppe für das ISR Launchpad

2. Auf Basis dieser Gruppe definieren Sie für Ihr ISR Launchpad die Liste der ISR-Szenarien, die gestartet werden können (siehe [Abbildung 53](#)).

Zuordnung Szenarien		Pos. A...	von	bis	Version
SMC1	Kostenstelle anlegen	1	<input checked="" type="checkbox"/>	31.12..	0
SMC2	Kostenstelle ändern	2	<input checked="" type="checkbox"/>		0

Abbildung 53 Zugeordnete Szenarien einer Szenariogruppe

3. Schließlich legen Sie die Reihenfolge sowie die Texte der Schaltflächen fest. Das Customizing des ISR Launchpads dazu finden Sie unter **Anwendungsbügreifende Komponenten • Internet-/Intranetservices • Interne Serviceanfrage • Definition von Szenarios • Szenariogruppen definieren** oder direkt über den View-Cluster VC_ISRGRP.

Web-Dynpro-Component QISR_UI als Component-Verwendung

Die dritte Möglichkeit der Verwendung der ISR-Web-Dynpro-Component ist die direkte Einbindung in eine eigene Web-Dynpro-Component. In diesem Fall wird die Web-Dynpro-Component QISR_UI als **Verwendete Component** hinterlegt (siehe [Abbildung 54](#)).

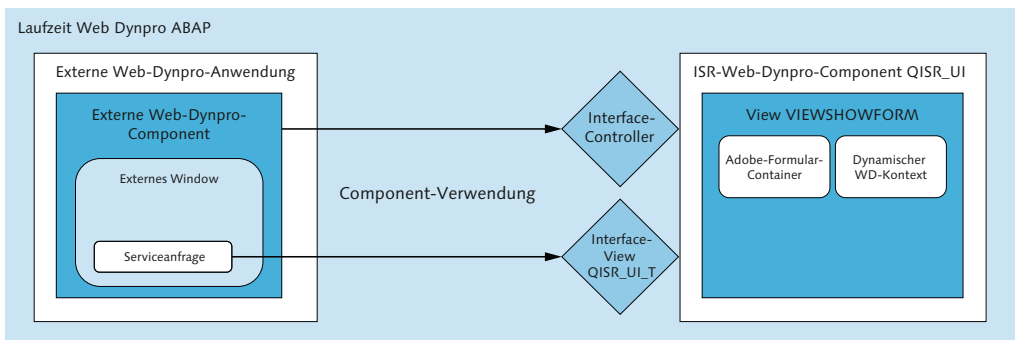


Abbildung 54 Schematische Darstellung der Web-Dynpro-Component »QISR_UI« als direkte Component-Verwendung

Die Kommunikation mit der Komponente QISR_UI erfolgt über die Methoden des Interface-Controllers. Dieser stellt eine Reihe von SET-/GET-Methoden zur Verfügung, um die Daten des Formulars und weitere technische Steuerungsparameter zu ermitteln bzw. zu verändern. Die zentrale Methode CALL_ISR_PROCESS_EVENT wird verwendet, um über ein EVENT das ISR-Framework aufzurufen. Dieser Importparameter EVENT verarbeitet die Standard-Events des ISR-Frameworks (siehe Abschnitt [»Standardwerte für die Parameter MODE, USER_COMMAND und FORM_VIEW«](#)) bzw. beliebige, selbst definierte Events, die in der entsprechenden BAdI-Implementierung abgefragt werden.

Remote Function Call »ISR_PROCESS_EVENT«

Zusätzlich zu den genannten Web-Dynpro-Oberflächen existieren weitere technische Schnittstellen, die mitunter für die Implementierung eines Szenarios hilfreich sein können. Darüber hinaus werden einige für die Implementierung der BAdIs hilfreiche Servicebausteine zur Verfügung gestellt.

Der Remote Function Call `ISR_PROCESS_EVENT` ist die zentrale Kommunikationschnittstelle des ISR-Frameworks zwischen dem ISR-Frontend und dem Backend. Er kann dazu verwendet werden, das Anlegen, Ändern oder Auslesen einer Serviceanfrage *ohne* Verwendung des ISR-Frontends (Web-Dynpro-Anwendung, Adobe-Formular) zu simulieren. Dabei wird die Implementierung des BAdI `QISR1` durchlaufen. Die Bedeutung der einzelnen Felder der Schnittstelle können Sie [Tabelle 6](#) entnehmen.

Parameter	Art	Werte	Beschreibung
SCENARIO	Importing		vierstelliger Szenarioname
MODE	Importing	CREATE, CHANGE, DISPLAY	beschreibt den Editiermodus
EVENT	Importing	SEND, SAVE, CHECK, APPROVE, etc.	Event (Standard-Events und szenariospezifische Events)
NOBMP_NO	Importing		Meldungsnummer, falls <code>MODE = CHANGE</code> oder <code>MODE = DISPLAY</code>
NOBMP_NO_OUT	Exporting		Meldungsnummer wird ausgegeben, falls Meldung angelegt wurde
RETURN	Exporting		Fehlermeldung
DATA	Tables		Formulardaten und allgemeine Daten
ADDITIONAL_DATA	Tables		Werte für Eingabehilfen

Tabelle 6 Schnittstelle des »RFC `ISR_PROCESS_EVENT`«

Die Parameter `FLAG_INOUT_CONVERSION`, `FLAG_RESET`, `ISR_PAGE_IN`, `ISR_PAGE_OUT`, `ISR_FORM_VIEW` und weitere Tabellenparameter können für die Simulation unberücksichtigt bleiben.



Tabellenparameter »DATA« und »ADDITIONAL_DATA«

Die Tabellenparameter `DATA` und `ADDITIONAL_DATA` sind generisch aufgebaut und beinhalten jeweils die folgenden Felder:

- `FIELDINDEX`
- `FIELDNAME`
- `FIELDVALUE`

DATA enthält eine Liste aller Merkmale (FIELDNAME) mit den zugehörigen Merkmalswerten (FIELDVALUE). Da ein Merkmal mehrere Werte tragen kann, erhält jede Kombination aus FIELDNAME und FIELDVALUE einen Zähler im Feld FIELDINDEX, beginnend bei '1'.

ADDITIONAL_DATA enthält die Werte für Eingabehilfen. Die Datenaufbereitung wurde im Abschnitt »[Programmbeispiele](#)« detailliert beschrieben.

Die folgenden Beispiele beschreiben die mögliche bzw. notwendige Parametrisierung des Remote Function Calls `ISR_PROCESS_EVENT` für unterschiedliche Anwendungsfälle:

■ **Beispiel 1: Starten eines Antrags**

Um den Start einer neuen Serviceanfrage zu simulieren, starten Sie den RFC mit den Parametern `SCENARIO = xxxx` (xxxx entspricht dem Szenarionamen) und `MODE = CREATE`. Der Parameter `EVENT` bleibt leer. Die Tabellen `DATA` und `ADDITIONAL_DATA` enthalten dann die initialisierten Merkmale und eventuelle Werte für die programmierten Eingabehilfen.

■ **Beispiel 2: Abschicken eines Antrags**

Möchten Sie eine Serviceanfrage abschicken, starten Sie den RFC mit den Parametern `SCENARIO = xxxx`, `MODE = CREATE` und `EVENT = SEND`. In der Tabelle `DATA` übergeben Sie die Daten, die die Serviceanfrage enthalten soll. Der Rückgabeparameter `NOBMP_NO_OUT` beinhaltet die Meldungsnummer.

Wenn Sie die Daten vor dem Abschicken prüfen wollen, geben Sie einfach nur für den Parameter `EVENT = CHECK` an. Falls die Daten nicht korrekt sind, erhalten Sie eine Fehlermeldung in der Rückgabestruktur `RETURN`. Das trifft auch auf Beispiel 3 zu.

■ **Beispiel 3: Lesen eines existierenden Antrags**

Möchten Sie eine Serviceanfrage lesen, starten Sie den RFC mit den Parametern `SCENARIO = xxxx`, `MODE = DISPLAY` und `NOBMP_NO` mit der entsprechenden Meldungsnummer. Der Parameter `EVENT` bleibt leer. Die Tabellen `DATA` und `ADDITIONAL_DATA` enthalten dann die Liste der Merkmalswerte und eventuelle Werte für die programmierten Eingabehilfen.

■ **Beispiel 4: Ändern eines existierenden Antrags**

Möchten Sie eine Serviceanfrage ändern, starten Sie den RFC mit den Parametern `SCENARIO = xxxx`, `MODE = CHANGE`, sowie `EVENT = SAVE` und `NOBMP_NO` mit der zugehörigen Meldungsnummer. In der Tabelle `DATA` übergeben Sie die Daten, die die Serviceanfrage enthalten soll. Beachten Sie, dass die Übergabe von Merkmalen ohne Merkmalswert zum Löschen eines existierenden Merkmalswertes führt.

Weitere Schnittstellen und Hilfsfunktionen

Weitere Funktionsbausteine und eine ABAP-Klasse, die für die Programmierung innerhalb der BAdI-Implementierung hilfreich sind, finden Sie in Tabelle 7.

Schnittstelle	Beschreibung
ISR_GENERAL_DATA_GET	Auslesen allgemeiner, szenario- und antragsabhängiger Daten eines Antrags von der Datenbank oder dem internen Funktionsgruppenspeicher
ISR_SPECIAL_DATA_GET	Auslesen szenario- und antragsabhängiger Merkmalsdaten eines Antrags von der Datenbank oder dem internen Funktionsgruppenspeicher
Klasse CL_ISR_DATA	Diese Hilfsklasse kann verwendet werden, um die Programmierung auf den relativ allgemein gehaltenen Schnittstellenparametern der BAdI-Methoden für die speziellen Merkmale eines Szenarios zu vereinfachen. Ein Programmierbeispiel finden Sie in Abschnitt » <u>Lesen und Setzen der speziellen Antragsfelder in BAdI-Methoden</u> «.
ISR_SPECIAL_DATA_TO_STRUC	Der Funktionsbaustein dient dazu, die Formulardaten aus den generisch aufbereiteten Übergabetabellen der BAdIs in eine passende, szenariospezifische Tabelle oder Struktur umzufüllen. Das erleichtert die nachfolgende Programmierung, basierend auf den Formulardaten. Der Funktionsbaustein kann alternativ zur Klasse CL_ISR_DATA verwendet werden.
ISR_STRUC_TO_SPECIAL_DATA	Möchten Sie die Formulardaten in einer BAdI-Methode ändern, hilft dieser Baustein, die Daten aus der szenariospezifischen Tabelle oder Struktur in die generische Übergabetabelle der BAdIs umzufüllen.
ISR_NOBMP_REVERSE	Der Funktionsbaustein dient dem Stornieren eines ISR-Antrags. Das ist so lange möglich, wie der Status der Meldung Offen ist. Der Funktionsbaustein ändert den Status auf Abgeschlossen . Im ISR-Statusüberblick wird zusätzlich der Status Storniert angezeigt. In einer Implementierung der Methode REVERSE_NOBMP_EXTERNAL des BAdI QISR1 können Sie eigene Verarbeitungen durchführen. Der Funktionsbaustein wird auch im ISR-Statusüberblick verwendet.

Tabelle 7 Hilfreiche Schnittstellen für die BAdI-Implementierung

Schnittstelle	Beschreibung
ISR_REQUEST_GET_RFC	Der Funktionsbaustein gibt eine Liste aller Serviceanträge eines beliebigen Antragstellers mit aktuellen Statusinformationen aus. Der Funktionsbaustein wird auch im ISR-Statusüberblick verwendet.
ISR_ACTOR_FOR_ROLE_GET	Der Funktionsbaustein kann benutzt werden, um zum Beispiel in einem kundeneigenen Workflow-Template anhand einer Regel den aktuellen Bearbeiter zu bestimmen. Außerdem wird die Implementierung der Methode CONTAINER_FOR_ACTOR_DET_FILL des BAdI QISR3 zur Änderung der Daten des Workflow-Containers durchlaufen.
Funktionsgruppe QISR5	Die Funktionsbausteine der Funktionsgruppe QISR5 können verwendet werden, um die Customizing-Einstellungen zu einem Szenario zu ermitteln.

Tabelle 7 Hilfreiche Schnittstellen für die BAdI-Implementierung (Forts.)

Die in [Tabelle 8](#) dargestellten BAdI-Schnittstellen können im Rahmen des ISR verwendet werden und wurden ausführlich anhand von Programmbeispielen in diesem Kapitel beschrieben. BAdI QISR5 existiert zwar, wird im Übrigen aber nicht verwendet.

BAdI-Schnittstelle	Verwendungszweck	Wo wird das erläutert?
QISR1	Programmierschnittstelle Formulare	Abschnitt » Formularablauflogik «
QISR2	Kontierungsfindung	Abschnitt » Servicekosten und BAdI QISR2 «
QISR3	Füllen des Containers zum Finden des Bearbeiters	Abschnitt » Bearbeiterfindung (BAdI QISR3) und Workflow «
QISR4	Ausführen einer Funktion der Aktivitätenleiste	Abschnitt » Aktivitäten und automatische Verbuchungen in der Meldung «
QISR6	ISR: Aufruf auf dem ISR Launchpad (Web Dynpro)	Abschnitt » Verwendungsmöglichkeiten in SAP Enterprise Portal «

Tabelle 8 BAdI-Schnittstellen des ISR

ISR in der Anwendungsbasis (ABA)

Dieser Abschnitt beschreibt die Variante des ISR in der sogenannten Anwendungsbasis (Softwarekomponente SAP_ABA), die mit Release SAP NetWeaver EHP2 zur Verfügung steht. Im Folgenden soll diese Variante ISR (ABA) genannt werden. Der in den vorhergehenden Abschnitten beschriebene ISR soll »ISR in ECC« genannt werden.

Grundkonzept

Die Idee zur Entwicklung des ISR (ABA) lässt sich im Wesentlichen aus zwei zentralen Überlegungen herleiten:

■ ISR nur als Formularframework

Eine Anwendung möchte zwar den ISR als Formularframework nutzen, dabei jedoch ein eigenes Laufzeitobjekt verwenden. Hintergrund hierfür ist, dass es im ISR des ECC bereits Anwendungen gibt, die den ISR nur als Formularframework nutzen, aber ihr eigenes Laufzeitobjekt verwenden: Das sind Szenarien der Applikationen T (»Steuer- und Einnahmenverwaltung – TRM«) und H (»HCM Processes and Forms«). Somit verwenden diese Anwendungen auch eigene Business-Workflow-Templates zur Durchführung des Antragsprozesses.

■ ISR außerhalb des ECC

Anwendungen außerhalb des ECC war eine Verwendung des ISR bislang nicht möglich, da der ISR in ECC in der Softwarekomponente SAP_APPL ausgeliefert wird. Mit ISR (ABA), der in der Softwarekomponente SAP_ABA entwickelt wurde, ist somit eine Verwendung in Anwendungen wie »SAP Master Data Governance« möglich.

Aus diesem Grund wurde die Funktionalität des ISR (ABA) reduziert. Dabei wurde aber das Konzept des ISR beibehalten und komplett von den entsprechenden Komponenten des ISR übernommen, wie er in ECC 6.0 zur Verfügung steht. Die Konsequenz daraus und deshalb ein wesentlicher Unterschied zwischen den beiden Varianten des ISR ist, dass im ISR (ABA) auf die Anbindung an SAP ERP verzichtet wird.

Der ISR (ABA) wurde komplett neu programmiert. Das heißt, Customizing-Tabellen und -Views sowie die Laufzeitumgebung sind durchweg neue Workbench-Objekte. Das technische Design ist weitestgehend analog zum ISR in ECC 6.0 gehalten. Das Augenmerk dieses Abschnitts soll darauf liegen, die Unterschiede zwischen ISR (ABA) und ISR in ECC zu erläutern. Eine Gegenüberstellung zwischen beiden bezüglich ihrer wesentlichen technischen Grundlagen bzw. Funktionen bietet Tabelle 9.

	ISR in ERP und ECC (Standard- anwendung)	ISR in ECC für Appli- kation T (TRM)	ISR in ABA
Software- Component	SAP_APPL, seit Release 4.6C	IS-PS-CA	SAP_ABA, seit Release 7.02
Formular	Adobe Form (seit ECC 5.0) ITS (seit 4.6C) JSP und BSP	Adobe Form	Adobe Form
Formular- eigenschaften	<ul style="list-style-type: none"> ■ Initialisierung ■ Prüfungen ■ eigene Events ■ Werthilfe (EVS, OVS) 	<ul style="list-style-type: none"> ■ Initialisierung ■ Prüfungen ■ eigene Events ■ Werthilfe (EVS, OVS) 	<ul style="list-style-type: none"> ■ Initialisierung ■ Prüfungen ■ eigene Events ■ Werthilfe (EVS, OVS)
Formular- schnittstelle	DDIC-basierend	DDIC-basierend	XML-Schema- basiert
FPM-Applikation	QISR_UI_FPM_GAF (seit ECC 6.0, SP 8)	TRM eigene Anwen- dung	APB_ISR_UI_FPM_ GAF
Formulardaten- ablage im BDS	ID = Meldungsnum- mer	ID = generierte OBJECTID	ID = generierte OBJECTID
Laufzeit BADI/Lauf- zeit API	<ul style="list-style-type: none"> ■ QISR1 ■ ISR_PROCESS_ EVENT 	<ul style="list-style-type: none"> ■ QISR1 ■ ISR_PROCESS_ EVENT 	<ul style="list-style-type: none"> ■ APB_ISR_PRO- CESS (analoge Struktur wie QISR1) ■ APB_ISR_PRO- CESS_EVENT
SAP Business Work- flow	<ul style="list-style-type: none"> ■ ja, generisch durch die Mel- dung ■ mehrfache Genehmigungs- schritte 	ja, TRM-spezifisch	nein

Tabelle 9 Vergleich: ISR in ECC (SAP_APPL), ISR für TRM und ISR in ABA (SAP_ABA)
(Quelle: SAP)

	ISR in ERP und ECC (Standard- anwendung)	ISR in ECC für Appli- kation T (TRM)	ISR in ABA
(QM)-Meldung als Laufzeitobjekt	ja	nein	nein
Statusüberblick	ja	nein	nein
Folgeaktivitäten	ja	nein	nein
Customizing	<ul style="list-style-type: none"> ■ QISRSCENARIO (Transaktion) ■ Wizard für neues Szenario 	Views des VC_ SCENARIO (aus Transaktion QISR- SCENARIO) werden verwendet	APB_ISR_CUST (Transaktion)
Framework Integration	siehe Abschnitt 9.4.2	TRM (Abschnitt 9.4.2)	SAP Master Data Governance (optional per Appli- kation)

Tabelle 9 Vergleich: ISR in ECC (SAP_APPL), ISR für TRM und ISR in ABA (SAP_ABA) (Quelle: SAP) (Forts.)

Definition eines Szenarios

Das Grundprinzip des Customizings von ISR (ABA) entspricht dem des ISR in ECC. Alle relevanten Einstellungen sind weiterhin für ein Szenario vorzunehmen.

Die IMG-Aktivität **Szenarios definieren** (Transaktion APB_ISR_CUST) ist der zentrale Einstieg in das Customizing eines ISR-(ABA-)Szenarios. Sie finden das Customizing unter **Anwendungsübergreifende Komponenten • Allgemeine Anwendungsfunktionen • Generische Funktionen der SAP Business Suite • Elementare interne Serviceanfrage • Szenarios definieren**. Dort sehen Sie eine Liste von Szenarien. Markieren Sie ein Szenario, gehen Sie auf **Grunddaten Ändern**, und Sie gelangen in das ISR-(ABA-) Grund-Customizing (siehe [Abbildung 55](#)).

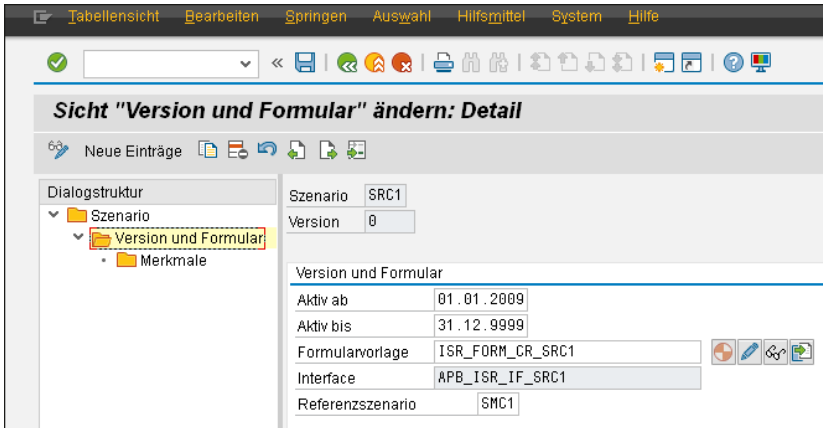


Abbildung 55 ISR-(ABA-)Grund-Customizing

ISR-(ABA-)Szenario

Die Formatierung (vierstellig), der Kundennamensraum sowie das technische Schlüsselfeld SCENARIO entsprechen dem des ISR-Szenarios in ECC.

ISR (ABA) und ISR in ECC sind technisch vollkommen voneinander unabhängig, da der ISR (ABA) mit neuen, unabhängigen Tabellen und Programmen entwickelt wurde. Somit sind auch die Definitionen der Szenarien voneinander unabhängig. Sie können sogar die gleichen technischen Namen in beiden ISR-Varianten verwenden.

Im Gegensatz zum ISR in ECC finden Sie im ISR (ABA) nur noch sehr wenige Einstellungen, die jedoch analog zum ISR in ECC gehandhabt werden:

■ Applikation

Sie müssen einem Szenario eine Applikation zuordnen. Folgende Applikationen stehen zur Verfügung:

- *Standardanwendung (kein Wert im Feld »Applikation«)*
Diese Applikation verwenden Sie, wenn Sie den ISR (ABA) in Ihrer eigenen Web-Dynpro-Anwendung integrieren möchten oder falls Sie die Web-Dynpro-Anwendung des ISR (ABA) separat verwenden möchten.
- *SAP Master Data Governance (Wert FM im Feld »Applikation«)*
Diese Applikation kann nur im Rahmen des gleichnamigen Frameworks zur Behandlung von Änderungsanträgen von Stammdaten verwendet werden.

■ Version und Gültigkeitszeitraum

Die Version und der Gültigkeitszeitraum werden genauso gehandhabt wie im ISR in ECC (siehe Abschnitt »[Applikation und Version](#)«). Geben Sie als Versionsnummer unter **Version und Formular** einen eindeutigen Wert an.

■ **Formularvorlage und Interface**

Sie können entweder ein vorhandenes interaktives Adobe-Formular eintragen oder ein neues Formular anlegen.

- Legen Sie ein neues Formular an, erfassen Sie im Feld **Formularvorlage** einen Namen für das Formular, oder Sie lassen das Feld leer. In diesem Fall wird der Formularname automatisch in folgendem Format generiert: APB_ISR_FORM_XXXX (XXXX = technischer Name des Szenarios). Wählen Sie **Generieren**, um das neue Formular zu erzeugen. Der Form Builder wird geöffnet.
- Wenn Sie das Layout des Formulars ändern möchten, wählen Sie **Ändern**. Der Form Builder wird geöffnet. Wenn Sie im Layout neue Felder definieren, können Sie diese beim Verlassen des Form Builders automatisch als Merkmale in das Szenario übertragen. Dabei wird auch die Datenbindung des neuen Feldes automatisch im Layout gespeichert.

Die Schnittstelle des Formulars (Interface) wird beim Anlegen des Formulars automatisch erzeugt und vom System im Feld **Interface** eingetragen. Der Name dieser Schnittstelle ist danach nicht mehr manuell änderbar. Wenn die Schnittstelle im Formular ausgetauscht wird, wird der Name der Schnittstelle automatisch im Szenario-Customizing angepasst.

Das Prinzip beim Erstellen des Formulars sowie die technischen Grundlagen des Formulars und des Formularkontextes entsprechen vollständig dem ISR in ECC (siehe Abschnitt »Antragsformular als Adobe-Formular«). Da ISR (ABA) nur das Adobe-Formular als Eingabemaske unterstützt, ist ein **Erfassungstyp** nicht notwendig, anders als im ISR in ECC.



ISR-(ABA-)Formular Schnittstellentyp

Im ISR (ABA) kann nur eine XML-Schema-basierte Schnittstelle verwendet werden (siehe Abbildung 56), während beim ISR in ECC nur die DDIC-Schnittstelle angewendet werden darf.

Schnittstelle	Z_APB_ISR_IF_SRC1	aktiv
<div style="display: flex; border-bottom: 1px solid black;"> <div style="background-color: #0070C0; color: white; padding: 2px 5px; margin-right: 5px;">Eigenschaften</div> <div style="padding: 2px 5px;">Schnittstelle</div> </div>		
Beschreibung	Schnittstelle für ISR-Szenario ZRC1	
Schnittstellentyp	XML Schema basierte Schnittstelle	
XML Schema Quelle	Generiert	

Abbildung 56 ISR-(ABA-)Schnittstellentyp – XML-Schema

■ **Referenzszenario**

Das Referenzszenario verwenden Sie, wenn die Formularablauflogik (siehe folgender Abschnitt) durch das in diesem Feld hinterlegte Szenario gesteuert werden soll.

■ **Merkmale**

Hier definieren Sie die Felder des Formulars. Die Definition der Merkmale erfolgt nach den gleichen Regeln und Einschränkungen, wie im Abschnitt »Merkmale« beschrieben ist. Es stehen somit auch die Möglichkeiten für programmierte Eingabehilfen, Steuerung als Langtextfeld sowie die Referenzierung auf ein Datenelement oder eine Struktur aus dem Data Dictionary zur Verfügung. Insbesondere die programmierte Wertheilfe mittels OVS wird erstellt, wie bereits im Abschnitt »Programmbeispiele« und im Abschnitt »Erweiterte Wertheilfe« beschrieben.

Das Customizing zur erweiterten Wertheilfe pflegen Sie für ein Szenario entsprechend per Transaktion APB_ISR_CUST_OVS oder über die IMG-Aktivität **OVS Wertheilfe einrichten**. Diese finden Sie im IMG unter **Anwendungsübergreifende Komponenten • Allgemeine Anwendungsfunktionen • Generische Funktionen der SAP Business Suite • Elementare interne Serviceanfrage • OVS Wertheilfe einrichten**. Informationen zum Customizing des OVS für ein Szenario liefert der Funktionsbaustein APB_ISR_CUST_SCEN_VALHLP_GET.

Formularablauflogik

Ebenso wie mit dem ISR in ECC können Sie die Ablauflogik des Formulars mithilfe des BAdI APB_ISR_FORM_PROCESSING programmatisch steuern. Dieser BAdI kann über den Erweiterungsspot APB_ISR_PROCESS implementiert werden.

Um eine eigene Verarbeitungslogik zu einem Formular hinzuzufügen, wählen Sie die IMG-Aktivität **BAdI: Formularverarbeitung** unter **Anwendungsübergreifende Komponenten • Allgemeine Anwendungsfunktionen • Generische Funktionen der SAP Business Suite • Elementare interne Serviceanfrage • BAdI: Formularverarbeitung**.

Standardmäßig kann jedem Szenario eine eigene BAdI-Implementierung der BAdI-Definition APB_ISR_FORM_PROCESSING zugeordnet werden. Als Filterwert tragen Sie das betreffende Szenario ein. Die Methoden der BAdI-Definition APB_ISR_FORM_PROCESSING sind im Vergleich zu den äquivalenten Methoden des BAdI QISR1 gleichartig entwickelt und verwendbar. Der BAdI QISR1 hatte mehr Methoden, die aber für die Formularablauflogik im ISR (ABA) nicht von Bedeutung sind. Zum Vergleich sind die Methoden beider BAdI-Schnittstellen in Abbildung 57 schematisch gegenübergestellt.

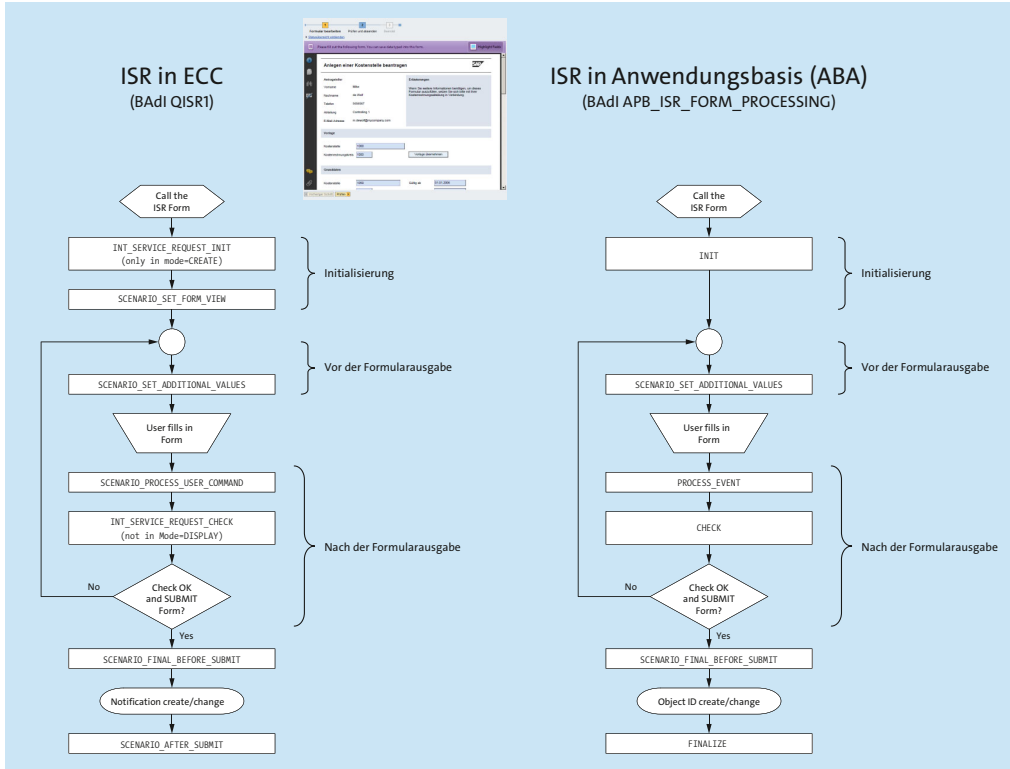


Abbildung 57 Zeitpunkte der Formularablaflglogik – Vergleich zwischen ISR in ECC und ISR (ABA)

Methoden

Es stehen insgesamt fünf BAdI-Methoden zur Verfügung:

- **INIT (Initialisierung)**
Diese Methode wird nur beim ersten Aufruf eines neuen Anfrageformulars aufgerufen. CT_SPECIAL_DATA dient dabei der Initialisierung der bereitgestellten Formularfelder. Die folgenden Parameter sind festgelegt:
 - IV_MODE = "CREATE"
 - IV_EVENT (leer)
- **SET_ADDITIONAL_VALUES (Wertehilfen)**
In dieser Methode können Sie die interne Tabelle CT_ADDITIONAL_DATA mit Wertelisten für Dropdown-Listboxen und Steuerungsparameter füllen. Diese Werte werden nicht in der Anfrage gesichert, stehen aber während der gesamten aktuellen Laufzeit des Adobe-Formulars zur Verfügung.
Die Werteliste muss alphabetisch nach den technischen Merkmalsnamen sortiert werden.

- **PROCESS_EVENT** (eigene Events verarbeiten)
Diese Methode wird beim Auslösen eines beliebigen Events auf dem Antragsformular (**Prüfen**, **Absenden**, **CHECK**, **IV_EVENT**) als Erstes durchlaufen. Der Event kann entsprechend abgearbeitet werden.
- **CHECK** (Prüfen)
Diese Methode wird beim Auslösen eines Standard-Events auf dem Antragsformular (**Prüfen** oder **Absenden** eines Formulars) nach der Methode SCENARIO_PROCESS_USER_COMMAND durchlaufen. Die Prüfung der Antragsdaten erfolgt somit immer vor der Fortschreibung der Anfrage. Falls Fehler auftreten, können Sie über den Exportparameter ES_RETURN eine entsprechende Fehlermeldung ausgeben. Falls Sie im Formular mehr als eine Meldung ausgeben möchten, können Sie den Parameter CT_MESSAGE_LIST verwenden.
- **FINALIZE** (nach Datenspeicherung finalisieren)
Diese Methode wird verwendet, nachdem die Formulardaten in der Datenbank gesichert wurden und die Anfragenummer (IV_OBJECTID) erstellt wurde, jedoch bevor die Anfragenummer auf dem Bild bereitgestellt wird. Sie können die Methode verwenden, um eine Folgeaktion auszuführen. Falls Sie das Anlegen der Anfrage auf der Datenbank zurücksetzen möchten, können Sie über den Exportparameter ES_RETURN eine entsprechende Fehlermeldung ausgeben. Der Datenbankeintrag der Anfrage wird dadurch zurückgesetzt.

In Tabelle 10 werden alle Datenstrukturen für die Implementierung der Methoden des BAdI APB_ISR_FORM_PROCESSING aufgelistet. Die Bedeutung und Verwendung der jeweiligen Datenstruktur entspricht der (fast) gleichnamigen Datenstruktur des BAdI QISR1 (siehe Abschnitt »Datenstrukturen innerhalb des ISR-Frameworks«). Die Programmierbeispiele aus dem Abschnitt »Formularablauflogik« können wieder verwendet werden, ersetzen Sie einfach die äquivalenten Datenstrukturen entsprechend.

Parameter	Typ	Beschreibung
IV_SCENARIO	APB_ISR_SCENARIO	aktuelles ABA-Szenario – entspricht FLT_VAL in QISR1
IV_MODE	APB_ISR_MODE	Beschreibt den Bearbeitungsmodus.

Tabelle 10 Schnittstellenparameter der Methoden des BAdI »APB_ISR_FORM_PROCESSING«

Parameter	Typ	Beschreibung
IV_EVENT	APB_ISR_EVENT	Event (Standard-Events und szenariospezifische Events) – entspricht USER_COMMAND in QISR1
IV_OBJECTID	APB_ISR_OBJECTID	Objekt-ID
ES_RETURN	BAPIRET1	Rückgabestruktur für Fehlermeldungen
CT_MESSAGE_LIST	BAPIRET2_T	Liste mit Meldungen unterschiedlichen Fehlertyps
CT_UI_ATTRIBUTES	APB_ISR_T_UI_ATTRIBUTE_PARAM	UI-Eigenschaften eines Formularfeldes
CS_GENERAL_DATA	APB_ISR_S_GENERAL_PARAM	allgemeine ISR-Daten
CT_SPECIAL_DATA	APB_ISR_T_SPECIAL_PARAM	Formulardaten
CT_ADDITIONAL_DATA	APB_ISR_T_SPECIAL_PARAM	Werte für Eingabehilfen
CT_EXTERNAL_DATA	APB_ISR_T_SPECIAL_PARAM	Werte für Steuerungsfunktionen

Tabelle 10 Schnittstellenparameter der Methoden des BAdI »APB_ISR_FORM_PROCESSING«

Standardwerte für die Parameter »MODE« und »IV_EVENT«

Die folgende kurze Aufzählung beschreibt die beiden Parameter Mode und Event und die Bedeutung der zugehörigen erlaubten Werte:

■ **MODE**

MODE beschreibt den aktuellen Bearbeitungsmodus. Mithilfe des Parameters MODE können Sie Ihre Programmlogik abhängig vom aktuellen Modus gestalten:

- CREATE, falls der Aufruf des ISR-Formulars für einen neuen Antrag erfolgen soll
- CHANGE, falls der Aufruf des ISR-Formulars mit Objekt-ID zum Ändern erfolgen soll
- DISPLAY, falls der Aufruf des ISR-Formulars mit Objekt-ID zum Anzeigen erfolgen soll

■ IV_EVENT

IV_EVENT entspricht dem Parameter USER_COMMAND in QISR1. Die Standard-ISR-Events des Parameters IV_EVENT sind wie folgt definiert:

- SEND, für den Fall, das das Formular für eine neue Anfrage abgeschickt wird
- SAVE, für den Fall, das das Formular für eine existierende Anfrage gesichert wird
- CHECK, für den Fall, das auf dem Formular die Schaltfläche **Prüfen** gedrückt wird

Im kundeneigenen Formular können beliebige zusätzliche Events definiert werden, die über APB_ISR_FORM_PROCESSING zur Laufzeit über den Parameter IV_EVENT abgefragt werden können.

Architektur

Der ISR (ABA) unterstützt formularbasierte Antragszenarien mit den in den vorigen Kapiteln beschriebenen Möglichkeiten der dynamischen Programmierung von Werthilfen, Initialisierungen, Prüfungen und eigener Events. Der ISR (ABA) stellt weiterhin eine BAdI-Schnittstelle zur Verfügung, um das Antragsformular flexibel steuern zu können.

Der mit dem Antragszenario verbundene Prozess zur Durchführung des gewünschten Service wird jedoch nicht mehr unterstützt. So fehlt die Anbindung des Business-Objekts BUS7051 als Laufzeitobjekt des Serviceantrags und das zugehörige Customizing. Auch die Anbindung an den Business Workflow ist demzufolge nicht mehr automatisch unterstützt. Das Laufzeitobjekt des ISR (ABA) muss von der Anwendung zur Verfügung gestellt werden und kann über die BAdI-Implementierung angesteuert werden (siehe Abschnitt »[Formularablauflogik](#)«).

[Abbildung 58](#) zeigt schematisch die Kommunikation zwischen der Web-Dynpro-Component APB_ISR_UI_FORM und dem Backend auf der einen Seite. Auf der anderen Seite wird die Verbindung zwischen ABAP-Designzeit und ABAP-Laufzeit skizziert. Diese schematische Darstellung bezieht sich auf [Abbildung 43](#) als Entsprechung für den ISR in ECC. Die Bestandteile der Architektur, die im ISR (ABA) im Vergleich zum ISR in ECC weggefallen sind, sind grau hinterlegt. Übrig bleiben die Komponenten des ISR (ABA) mit den jeweiligen neuen Objektnamen.

Die Web-Dynpro-Component der Laufzeit kommuniziert über den RFC APB_ISR_PROCESS_EVENT mit der ISR-(ABA)-Laufzeit im Backend. Innerhalb der ISR-(ABA)-Laufzeit werden unter anderem die Methoden des BAdI APB_ISR_FORM_PROCESSING (siehe [Abbildung 57](#)) zu verschiedenen Zeitpunkten aufgerufen.

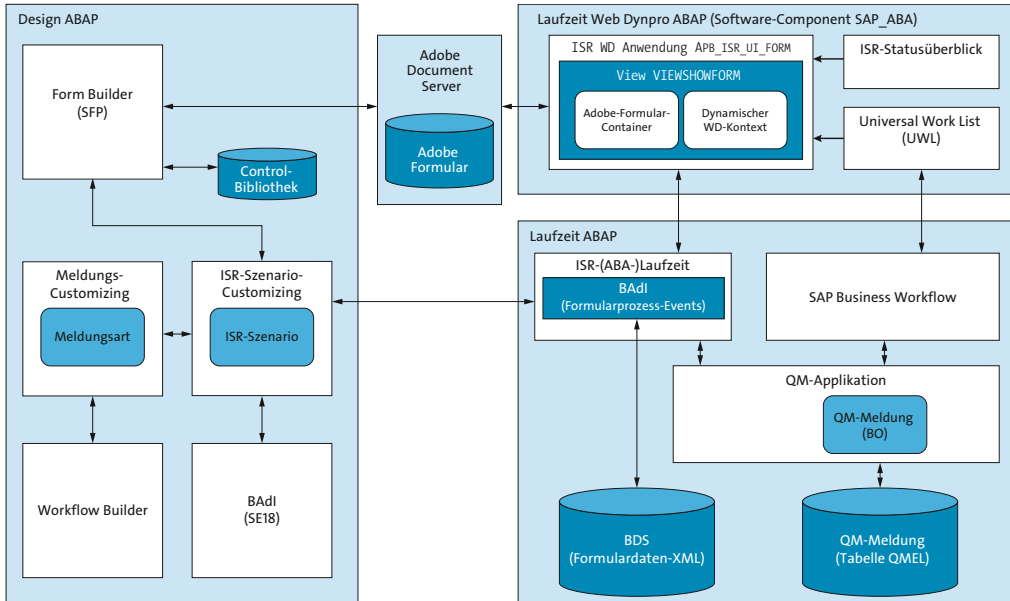


Abbildung 58 Kommunikation zwischen ABAP-Designzeit und ABAP-Laufzeit des ISR (ABA)

Beim Abschicken des Formulars werden die Daten des Formulars weiterhin im Business Document Service (BDS) der Dokumentenverwaltung des Knowledge Providers mit einer generierten, 32-stelligen OBJECTID abgespeichert. Die Daten werden auch im ISR (ABA) nicht gemeinsam mit dem Formular als PDF gespeichert. Das Adobe-Formular dient also nur als Oberflächenlayout zur Erfassung des Antrags und wird zur Laufzeit immer mit den aktuellen Daten abgemischt.



»Sprechende« Objekt-ID

Natürlich ist es möglich, über die BAdI-Implementierung zusätzlich eine eigene, »sprechende« Objekt-ID (maximale Länge 20 Zeichen) zu vergeben. Diese kann in der Struktur GENERAL_DATA-HEADER im Feld EXT_REF_NUMBER übergeben werden. Beim Abspeichern wird diese ebenso fortgeschrieben.

Für die Integration des Adobe-Formulars stehen die Web-Dynpro-Anwendungen APB_ISR_UI und APB_ISR_UI_FPM, basierend auf den gleichnamigen Web-Dynpro-ABAP-Komponenten, zur Verfügung:

- **Web-Dynpro-ABAP-Component APB_ISR_UI**

Die Web-Dynpro-Komponente APB_ISR_UI ist baugleich zur Web-Dynpro-Komponente QISR_UI entworfen und programmiert und kann somit auch genauso als

Verwendete Component in ihre eigene Web-Dynpro-Komponente integriert werden (siehe Abschnitt »Verwendungsmöglichkeiten in SAP Enterprise Portal«). Abbildung 59 veranschaulicht die Verwendung der Web-Dynpro-ABAP-Component APB_ISR_UI.

Die notwendigen Parameter für den Aufruf der Web-Dynpro-Anwendung sind SCENARIO, OBJECTID und MODE. Die Bedeutung der Parameter entnehmen Sie Tabelle 10.

■ **Web-Dynpro-ABAP-Component APB_ISR_UI_FPM**

Die Komponente APB_ISR_UI_FPM basiert auf dem sogenannten Floorplan Manager (FPM) und dient nur Demonstrationszwecken. Es wird empfohlen, die Web-Dynpro-Komponente APB_ISR_UI zu verwenden und eventuell in eine eigene FPM-basierte Anwendung zu integrieren.

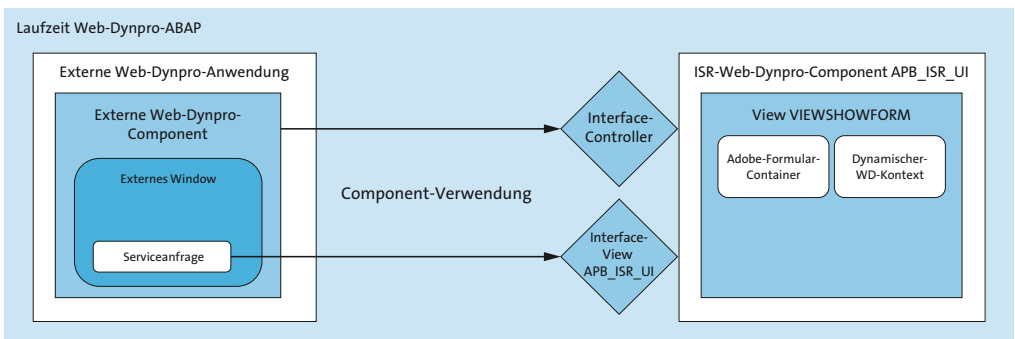


Abbildung 59 Schematische Darstellung der Web-Dynpro-Component »APB_ISR_UI« als direkte Component-Verwendung

Vergleichbar zum ISR in ECC steht im ISR (ABA) ein Remote Function Call APB_ISR_PROCESS_EVENT zur Verfügung. Er ist die zentrale Kommunikationsschnittstelle des ISR-(ABA)-Frameworks zwischen dem ISR-Frontend und dem Backend. Er kann aber auch dazu verwendet werden, das Anlegen, Ändern oder Auslesen einer Serviceanfrage *ohne* Verwendung des ISR-Frontends (Web-Dynpro-Anwendung, Adobe-Formular) zu simulieren. Dabei wird die Implementierung des BAdI APB_ISR_FORM_PROCESSING durchlaufen. Die Schnittstelle des RFC entspricht der des RFC ISR_PROCESS_EVENT (siehe Abschnitt »Remote Function Call »ISR PROCESS EVENT««). Mit Hilfe von Tabelle 6 können Sie deren Bedeutung leicht verstehen. Die beiden RFCs stimmen in ihrer Funktionsweise überein.

Zusammenfassung

In diesem Kapitel haben Sie das Konzept der internen Serviceanfrage als Anwendung der interaktiven Adobe-Formulare innerhalb von SAP ERP kennengelernt. Sie wissen nun, dass ISR ein flexibel zu konfigurierendes Framework ist, das mehrere SAP-Komponenten verbindet, und kennen die generellen Voraussetzungen zur Anwendung des ISR. Anhand eines Beispielszenarios wurde die Verwendungsmöglichkeit des ISR dargestellt. Sie haben erfahren, wie Sie mit dem ISR-Wizard ein einfaches ISR-Szenario zusammen mit einem Adobe-Formular erstellen.

Des Weiteren wurden das Customizing und die BAdI-Schnittstellen als Modellierungswerkzeuge des Szenarios ausführlich vorgestellt. Damit sind Sie in der Lage, auch komplexere Anforderungen an das Szenario zu implementieren. Sie haben gelernt, wie Sie ein interaktives Adobe-Formular für das ISR-Framework generieren und erweitern. Dies ist besonders wichtig, da das ISR-Framework eine bestimmte Schnittstellen- und Kontextstruktur voraussetzt. Darüber hinaus haben Sie den Nutzen der ISR-Bibliothek gesehen.

Danach haben Sie die technische Grobarchitektur des ISR kennengelernt. Insbesondere die genannten ABAP-Schnittstellen können helfen, die Programmierung der BAdI-Implementierung eines Szenarios effizienter zu gestalten.

Zuletzt wurde der ISR in der Anwendungsbasis (ABA) erläutert. In Kapitel 10, das sich vor allem an fortgeschrittene Entwickler von Formularen wendet, erfahren Sie mehr über das ABAP-PDF-Objekt.