

Web Dynpro ABAP ist eine der Schlüssel-UI-Technologien von SAP. Auch in den ABAP Development Tools ist die Web-Dynpro-Entwicklung möglich.

4 Web Dynpro ABAP in Eclipse

Web Dynpro ABAP gehört neben dem *SAP UI Development Toolkit for HTML5* (SAPUI5) zu den Schlüssel-Oberflächentechnologien von SAP. Neue Anwendungen sollten daher in Web Dynpro ABAP oder SAPUI5 umgesetzt werden. Für die SAPUI5-Entwicklung steht ein eigenes Toolset in der Eclipse-Entwicklungsumgebung zur Verfügung, das Sie über die Seite <https://tools.hana.ondemand.com/#sapui5> beziehen können. Die Entwicklung mit Web Dynpro ABAP ist hingegen in den ABAP Development Tools for SAP NetWeaver (ADT) möglich und Thema dieses Kapitels.

Web Dynpro Java

Verwechseln Sie die Entwicklungsumgebung für Web Dynpro ABAP in den ADT nicht mit den Werkzeugen für Web Dynpro Java, die im *SAP NetWeaver Developer Studio* verfügbar sind. Bei den Web-Dynpro-ABAP-Werkzeugen handelt es sich um ein neues Toolset, d. h., Sie benötigen kein Developer Studio.

【◀】

Zur Einführung in die Web-Dynpro-Entwicklung in Eclipse entwerfen wir in diesem Kapitel eine Applikation, um unsere in Kapitel 2, »Funktionen der ABAP Development Tools«, eingeführte Aufgabenliste nach Projekten strukturiert auszugeben. Wir sehen uns außerdem an, wie Sie eine Anwendung auf Basis des *Floorplan Managers* anlegen können. Ziel dieses Kapitels ist es, Ihnen die Entwicklungsumgebung vorzustellen, Kenntnisse in Web Dynpro ABAP und des *Floorplan Managers* werden daher vorausgesetzt.

4.1 Controller-Programmierung

Um zu beginnen, müssen wir zunächst eine neue Web-Dynpro-ABAP-Komponente anlegen. Öffnen Sie dazu die ABAP Development Tools in Eclipse,

und wählen Sie Ihr aktuelles ABAP-Projekt. Sie benötigen kein neues ABAP-Projekt, um in Web Dynpro zu entwickeln. Gehen Sie dann wie folgt vor:

1. Markieren Sie das Paket Z_ISSUE_LIST im Project Explorer.
2. Starten Sie den Anlage-Assistenten im Kontextmenü über NEW • OTHER ABAP REPOSITORY OBJECT oder mithilfe der Tastenkombination **[Strg] + [N]**.
3. Wählen Sie aus der Liste der verfügbaren Assistenten ABAP • WEB DYNPRO • COMPONENT, und bestätigen Sie mit NEXT (siehe Abbildung 4.1).

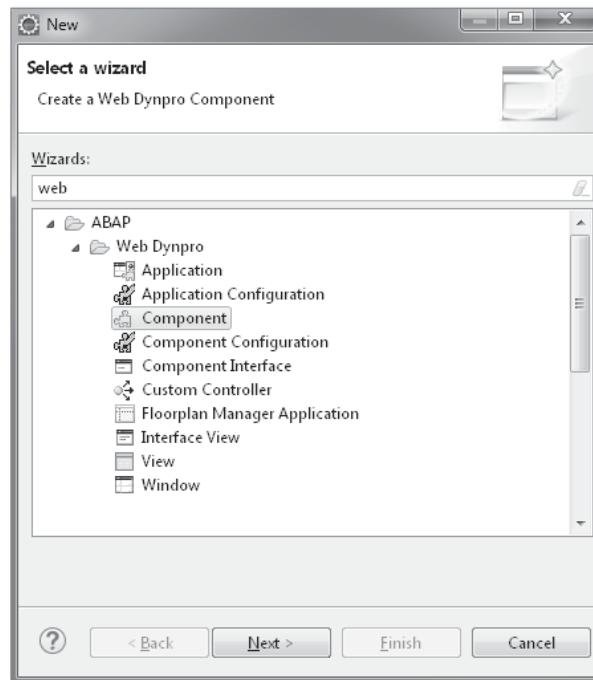


Abbildung 4.1 Assistenten zum Anlegen einer Web-Dynpro-Komponente aufrufen

4. Geben Sie im folgenden Dialog NEW WEB DYNPRO COMPONENT »ZWD_C_ISSUE_LIST« als Namen für die Komponente ein (siehe Abbildung 4.2).
5. Genau wie in der ABAP Workbench können Sie sich direkt einen initialen *Window*- und *View*-Controller anlegen lassen. Markieren Sie dazu die Option CREATE WINDOW, und tragen Sie »W_ISSUE_LIST« als Namen für das Window ein. Markieren Sie auch die Option CREATE VIEW. Als Namen für den View wählen Sie »V_MAIN«. Fahren Sie mit NEXT fort.
6. Wählen Sie den Transportauftrag, den Sie für unser Demoszenario angelegt haben, und bestätigen Sie die Anlage der Komponente mit FINISH.

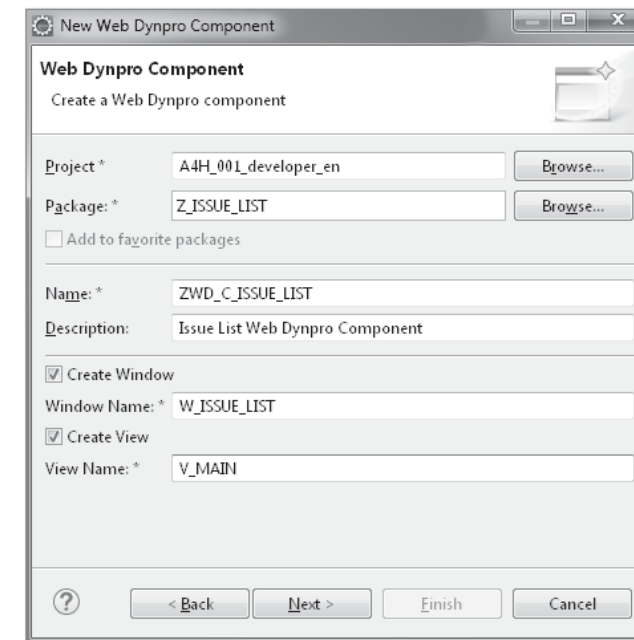


Abbildung 4.2 Web-Dynpro-Komponente anlegen

7. Der Assistent erstellt die neue Komponente und öffnet diese in einem SAP-GUI-Bildschirm. Aktivieren Sie hier Ihre Komponente und alle automatisch angelegten Controller mit der Tastenkombination **[Strg] + [F3]**, und schließen Sie die Registerkarte.

Das Ergebnis sehen Sie in Abbildung 4.3. Unterhalb des Namens der Web-Dynpro-Komponente werden die einzelnen Bestandteile aufgelistet – ähnlich wie in der ABAP Workbench.

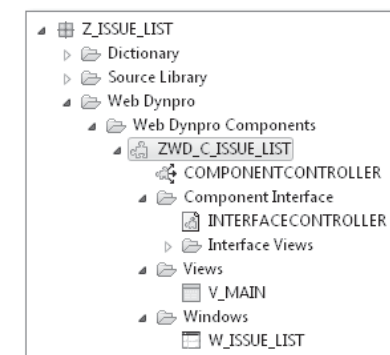


Abbildung 4.3 Struktur der neuen Web-Dynpro-Komponente

Die Eclipse-Editoren zur Web-Dynpro-Bearbeitung sind in mehrere Registerkarten aufgeteilt – ähnlich den Registerkarten in der ABAP Workbench. Der Editor für Komponenten-Controller unterteilt sich in die Bereiche OVERVIEW, CONTEXT, ATTRIBUTES, METHODS, EVENTS und IMPLEMENTATION. Die Register zur Navigation, die in Abbildung 4.4 dargestellt sind, befinden sich unterhalb des Editor-Fensters. Zusätzliche Informationen und Einstellungen zu den Elementen eines Controllers sind im View PROPERTIES zu finden.

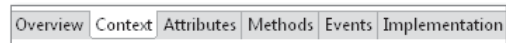


Abbildung 4.4 Registerkarten im Editor für Komponenten-Controller

4.1.1 Kontext-Editor

Als Nächstes müssen wir den Kontext unserer Applikation anlegen. Da wir unsere Aufgabenliste (Issue List) als Baum darstellen möchten, sollte sich diese Struktur im Kontext wiederfinden:

1. Öffnen Sie den Komponenten-Controller der Komponente ZWD_C_ISSUE_LIST. Die Registerkarte CONTEXT ist standardmäßig zuerst geöffnet.
2. Klicken Sie mit der rechten Maustaste auf den Wurzelknoten CONTEXT, und wählen Sie ADD NODE im Kontextmenü (siehe Abbildung 4.5), um einen neuen Knoten anzulegen. Alternativ können Sie das Tastenkürzel **Alt** + **O** verwenden.

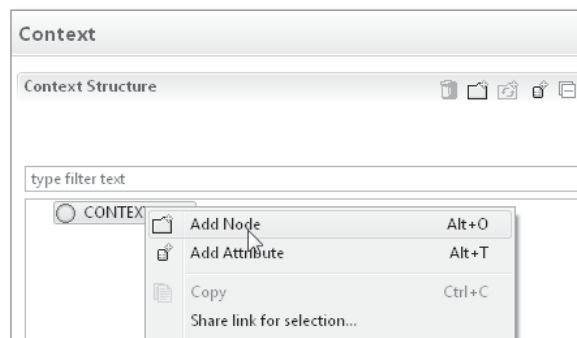


Abbildung 4.5 Kontextknoten einfügen

3. Markieren Sie den neuen Knoten, und drücken Sie **F2**, um ihn umzubenennen (Alternativ können Sie den Namen auch im PROPERTIES-View ändern.). Nennen Sie den neuen Knoten »PROJECTS«.

4. Öffnen Sie den PROPERTIES-View mit einem Doppelklick auf den Kontextknoten. Ändern Sie die CARDINALITY (Kardinalität) des Knotens PROJECTS auf 0..n.

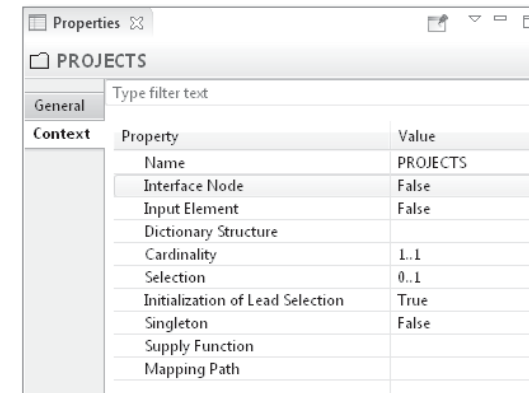


Abbildung 4.6 Eigenschaften des Knotens PROJECTS

5. Klicken Sie mit der rechten Maustaste auf den Knoten PROJECTS, und fügen Sie über die Funktion ADD ATTRIBUTE im Kontextmenü (oder per **Alt** + **T**) ein neues Attribut ein.
6. Drücken Sie die Funktionstaste **F2**, und benennen Sie das neue Attribut in »PROJECT_ID« um.
7. Ändern Sie den Typ des Attributs im PROPERTIES-View in »Z_PROJECT_ID«. Sie können in diesem Feld übrigens auch die Code-Vervollständigung verwenden oder über den Button mit den Pünktchen (...) in eine Suchmaske springen.
8. Legen Sie auf die gleiche Art und Weise ein weiteres Element PROJECT_NAME vom Typ Z_PROJECT_NAME an.
9. Markieren Sie den PROJECTS-Knoten, und fügen Sie mit der Tastenkombination **Alt** + **O** einen neuen Unterknoten ein. Nennen Sie ihn ISSUES, und vergeben Sie die Kardinalität 0..n.
10. Markieren Sie die Eigenschaft SUPPLY FUNCTION im PROPERTIES-View, und klicken Sie auf den Button CREATE (siehe Abbildung 4.7), um eine neue Methode anzulegen.
11. Nennen Sie die Methode »SUPPLY_ISSUES«, und vergeben Sie eine Beschreibung.

Fügen Sie die drei Attribute, die Sie in Abbildung 4.8 sehen, zu Ihrem neuen ISSUES-Knoten hinzu.

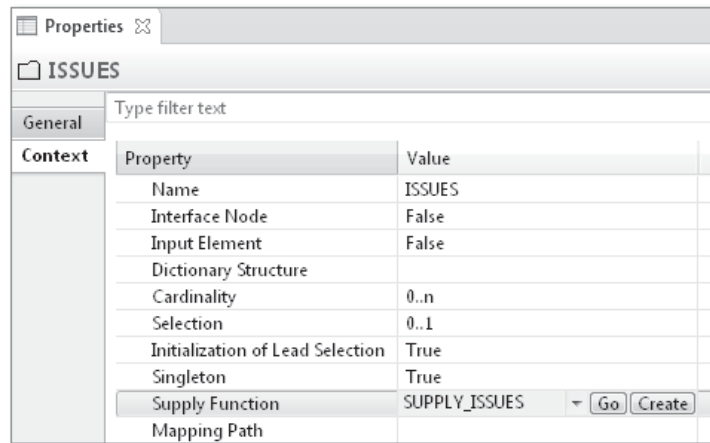


Abbildung 4.7 Eigenschaften des Knotens ISSUES

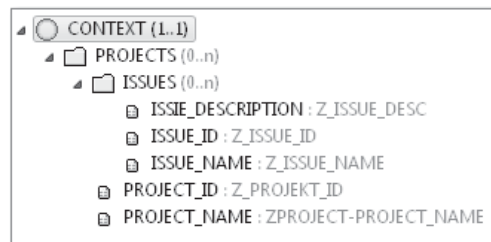


Abbildung 4.8 Fertige Struktur des Kontextes PROJECTS

Damit haben Sie den Kontext für die Navigation in der Aufgabenliste angelegt. Jetzt benötigen wir noch einen Speicher für die Detailanzeige. Die Kontextstruktur dazu soll jedoch nicht manuell, sondern mit Bezug auf ein ABAP-Dictionary-Objekt angelegt werden. Wir beschränken uns in diesem Beispiel auf die Anzeige von Projekten. Sie können die Lösung aber nach dem gleichen Prinzip um die Anzeige von Issues erweitern.

1. Fügen Sie dem Kontext mit der Tastenkombination **[Alt] + [0]** einen neuen Knoten hinzu, den Sie über die Funktionstaste **[F2]** in PROJECT umbenennen.
2. Im PROPERTIES-View geben Sie »ZPROJECT« als DICTIONARY STRUCTURE an. Auch dabei können Sie wieder mit der Code-Vervollständigung (**[Strg]** + Leertaste) arbeiten. Klicken Sie in derselbe Zeile auf den Button DERIVE (siehe Abbildung 4.9).
3. Ihnen wird ein Dialogfenster mit den Elementen der Struktur angezeigt. Belassen Sie es bei der Auswahl, und bestätigen Sie den Dialog mit FINISH.

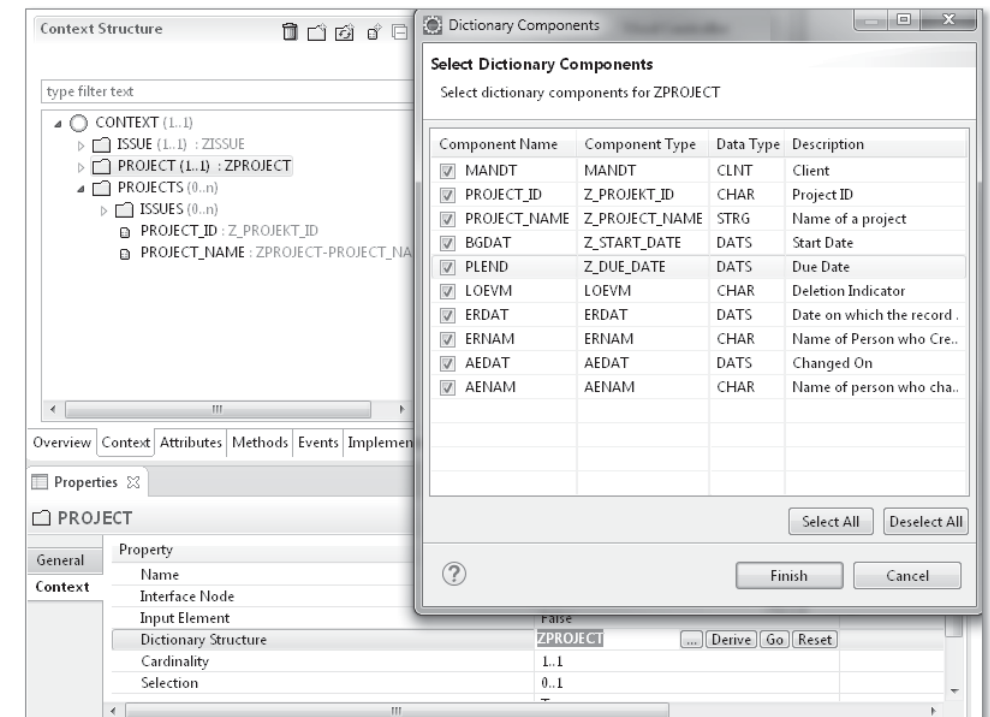


Abbildung 4.9 Kontext mit Bezug auf Dictionary-Objekt anlegen

4. Fügen Sie unterhalb des Knotens PROJECT noch einen Knoten PROJECT_ISSUES hinzu. Vergeben Sie die Kardinalität 0..n, und binden Sie ihn an die Dictionary-Struktur ZISSUE. Übernehmen Sie mit dem Button DERIVE alle Elemente aus der Struktur wieder in den Kontextknoten.

Der neue Kontext-Editor ähnelt zwar sehr dem aus der ABAP Workbench, allerdings wurden viele Kleinigkeiten geändert, die Ihre Arbeit sehr erleichtern können. So sind die wichtigsten Funktionen über Tastenkürzel zu erreichen und stehen zudem im Kontextmenü sowie über Funktionstasten zur Verfügung. Die wichtigsten Informationen zu Ihren Kontextknoten finden Sie nun direkt im Kontextbaum. Neben dem Namen werden auch die Kardinalität sowie der Knoten- bzw. Attributstyp angezeigt. Die Einstellungen können Sie, wie bereits gesehen, im PROPERTIES-View ändern. Die Anzeige können Sie durch ein Suchfeld oberhalb der Baumdarstellung filtern.

Sie können auch einen *Verwendungsnachweis* für Kontextelemente anfordern (**[Alt] + [Strg] + [G]**). Bei allen Controllern haben Sie außerdem die Möglichkeit, Methoden, Attribute und Kontextelemente zu verlinken, wozu Ihnen ADT-Links oder HTTP-Links zur Verfügung sehen.

» Design der Beispielkomponente

Das Design unserer Beispielkomponente wurde bewusst einfach ausgelegt und sieht nur eine Komponente vor. Bei einer produktiven Anwendung könnte man die Detailsicht in eine oder mehrere Komponenten verlagern und den Kontext anders aufbauen.

4.1.2 Attribute

Als Nächstes wollen wir ein Attribut anlegen, um eine Instanz unserer Klasse ZCL_ISSUE abzulegen. Diese Instanz dient uns später zur Bearbeitung der angezeigten Aufgabe. Attribute pflegen Sie wie folgt:


1. Wechseln Sie im Editor des Komponenten-Controllers auf die Registerkarte ATTRIBUTES.
2. Klicken Sie auf die Schaltfläche ADD ATTRIBUTE, oder drücken Sie **[Alt] + [T]**.
3. Geben Sie »MR_PROJECT« als Attributnamen im folgenden Dialogfenster ein. Wählen Sie zudem eine passende Beschreibung, und bestätigen Sie mit FINISH.
4. Geben Sie im Feld TYPE bei den ATTRIBUTE DETAILS »ZCL_PROJECT« ein. Markieren Sie zudem das Ankreuzfeld REF TO, um zu kennzeichnen, dass es sich um eine Referenzvariable handelt. Sie können im Feld TYPE auch mit der Tastenkombination **[Strg] + [Leertaste]** arbeiten oder über den Button BROWSE einen Suchdialog öffnen.

Attribute werden in Eclipse in einer filterbaren Liste dargestellt, in der Sie genau wie bei Kontextknoten schon die Typisierung des Attributs erkennen. Die Eigenschaften zu einem Attribut finden Sie rechts neben der Listendarstellung (nicht im PROPERTIES-View). Ist der Text TYPE:* blau hinterlegt (siehe Abbildung 4.10), wird damit eine Verlinkung angedeutet, über die Sie sich die Definition des angegebenen Typs anzeigen lassen können. Diese Navigationsform finden Sie in einigen Web-Dynpro-Editoren in Eclipse wieder. Leider ist es derzeit nicht möglich, Attribute umzubenennen. Sie können sie lediglich kopieren (**[Strg] + [C]**) und unter einem neuen Namen wieder einfügen (**[Strg] + [V]**).

4.1.3 Methoden

Nun wollen wir unserer Komponente ein wenig Intelligenz einhauchen und dazu Methoden anlegen. Die Controller-Programmierung unterscheidet sich

in diesem Punkt von der ABAP-Objects-Entwicklung, da die Definition von Methoden nicht im Quelltext, sondern in einer eigenen Maske erfolgt. Die Implementierung läuft aber sehr ähnlich ab.

1. Wechseln Sie auf die Registerkarte METHODS im Komponenten-Controller.
2. Klicken Sie auf den Button ADD METHOD. Alternativ können Sie auch mit den Tasten **[Alt] + [O]** eine neue Methode anlegen.
3. Geben Sie im Dialogfenster NEW METHOD »FILL_PROJECTS_NODE« als Namen sowie eine beliebige Beschreibung ein.
4. Damit haben Sie schon Ihre zweite Web-Dynpro-Methode in Eclipse erstellt. Die erste Methode war die Supply-Methode, die Sie beim Erstellen des Kontextes angelegt haben. Diese finden Sie auch in der Liste der Methoden. Ein Icon  symbolisiert, dass es sich hierbei um eine Supply-Methode handelt..

Als Nächstes benötigen wir eine Methode, mit der ein Projekt zur Anzeige im Detailbild ausgewählt wird.

1. Legen Sie eine weitere Methode mit dem Namen »SELECT_PROJECT« an.
2. Klicken Sie auf den Button ADD PARAMETER (alternativ **[Alt] + [T]**). Ein neuer Parameter wird zu Ihrer Methode hinzugefügt und als Element unterhalb des Methodennamens dargestellt. Auf der rechten Seite können Sie die Eigenschaften des Parameters ändern.
3. Geben Sie »IF_PROJECT_ID« als Namen dieses Parameters ein. Als TYPE tragen Sie »Z_PROJECT_ID« ein (siehe Abbildung 4.10). Nachdem Sie die Methoden angelegt haben, sollte die Übersicht in Ihrem Editor etwa wie in Abbildung 4.11 aussehen.

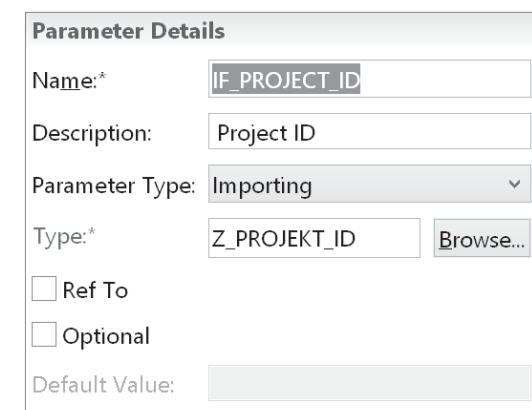


Abbildung 4.10 Eigenschaften eines Methodenparameters

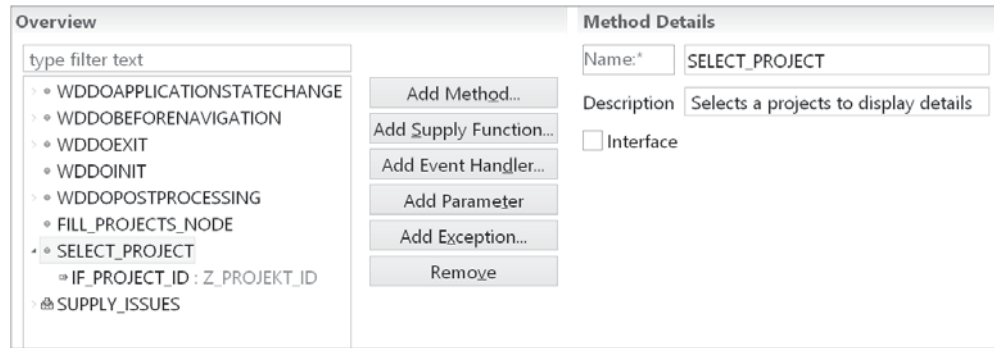


Abbildung 4.11 Übersicht und Details im Methoden-Editor

Auf ähnliche Art und Weise können Sie einer Methode über die Tastenkombination `[Alt] + [X]` eine Ausnahme hinzufügen. Wie Sie in Abbildung 4.11 sehen können, können Sie in dieser Sicht auch eine neue Supply-Methode (ADD SUPPLY FUNCTION) oder einen neuen Ereignisbehandler (ADD EVENT HANDLER) anlegen.

Als Nächstes implementieren wir unsere Methoden. Markieren Sie dazu die Methode `FILL_PROJECTS_NODE` in der Übersichtsliste. Anschließend klicken Sie auf der rechten Seite auf den Bezeichner `NAME*`. Der Link bringt Sie direkt in den Quelltext-Editor Ihres Controllers zur Implementierung der Methode.

Der Quelltext-Editor funktioniert sehr ähnlich wie der Editor anderer Entwicklungsobjekte in den ADT, mit dem Unterschied, dass Sie nicht uneingeschränkt ändern dürfen. Beim Code von Web-Dynpro-Controllern handelt es sich größtenteils um generierten Code, d. h., ein Grundgerüst wird Ihnen durch das Framework bereits vorgegeben und sollte nicht geändert werden. Sie haben lediglich die Möglichkeit, an geeigneten Stellen eignen Quelltext einzubringen. Daher sind große Teile des Web-Dynpro-Quelltextes grau hinterlegt und für Änderungen gesperrt. Methodenimplementierungen sind hingegen änderbar. Die Editor-Sicht für den Komponenten-Controller `ZWD_C_ISSUE_LIST` sehen Sie in Abbildung 4.12.

Zur Navigation innerhalb des Quelltextes steht Ihnen wieder der OUTLINE-View zur Verfügung (siehe Abbildung 4.13 und Abschnitt 2.2.4, »Klassen«). Sie sehen hier den Inhalt des generierten Includes zum aktuellen Controller. Daher werden Ihnen auch die beiden Interfaces `IG_COMPONENTCONTROLLER` und `IF_COMPONENTCONTROLLER` angezeigt.

```

4*[A4H] ZWD_C_ISSUE_LIST | COMPONENTCONTROLLER
194 method WDDOEXIT . "#EC NEEDED
195 endmethod.
196
197 method WDDOINIT . "#EC NEEDED
198 endmethod.
199
200 method WDDOPOSTPROCESSING . "#EC NEEDED
201 endmethod.
202
203 method FILL_PROJECTS_NODE . "#EC NEEDED
204 endmethod.
205
206 method SELECT_PROJECT . "#EC NEEDED
207 endmethod.
208
209 * +-----+
210 * | supply methods |
211 * +-----+
212
213 method SUPPLY_ISSUES . "#EC NEEDED
214 endmethod.
215
216 endclass.

```

Abbildung 4.12 Der Quelltext-Editor zum Komponenten-Controller ZWD_C_ISSUE_LIST

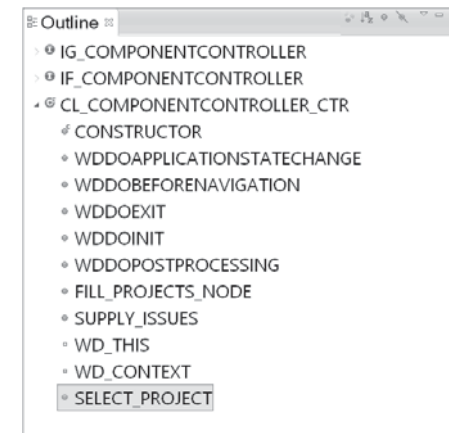


Abbildung 4.13 View »Outline« zum Komponenten-Controller ZWD_C_ISSUE_LIST

Für die Entwicklung in Web Dynpro stehen Ihnen spezielle Code-Templates zur Verfügung. Daneben können Sie eigene Web-Dynpro-Templates erstellen. Im Anlage-Dialog finden Sie eine Auswahlbox, um ein Template für Web Dynpro zu kennzeichnen. Auch auf den *Web Dynpro Code Wizard* zur Erzeugung von Kontextzugriffen müssen Sie nicht verzichten. Dieses aus der ABAP Workbench bekannte Tool versteckt sich nun hinter der Code-Vervollständigung (`[Strg] + Leertaste`).

Implementieren wir nun zunächst die Methode `FILL_PROJECTS_NODE` unserer Web-Dynpro-Komponente:

1. Wechseln Sie zur Registerkarte `IMPLEMENTATION` und zur Methode `FILL_PROJECTS_NODE`. (Entweder springen Sie von der Registerkarte `METHODS` ab oder vom `OUTLINE-View`.)
2. Fügen Sie nach der folgenden Code-Zeile eine Leerzeile ein:

```
METHOD fill_projects_node. "#EC NEEDED
```
3. Verwenden Sie die Tastenkombination `[Strg] + Leertaste`, um die Code-Vervollständigung aufzurufen.
4. Die Code-Vervollständigung wird wie in Abbildung 4.14 angezeigt. Wählen Sie die Option `CONTEXTWRITE` aus, um eine Set-Operation auf den Kontext auszuführen.

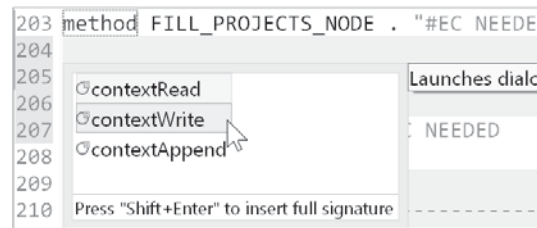


Abbildung 4.14 Kontext lesen und schreiben

5. Im nächsten Schritt können Sie, wie in Abbildung 4.15 gezeigt, einen Kontextknoten wählen. Ob Sie die statischen Attribute des Elements oder die ganze Tabelle auslesen möchten, steuern Sie über die Auswahlbox `CODE GENERATED AS TABLE OPERATIONS`.

Wählen Sie für unser Beispiel den Knoten `PROJECTS` aus, und bestätigen Sie mit `OK`.

6. Nun müssen wir selbst tätig werden und den generierten Code ein wenig anpassen. Die Methode `ZCL_PROJECTS=>LIST_PROJECTS` liefert eine Tabelle mit allen auf der Datenbank gespeicherten Projekten. Holen Sie sich die aktuelle Projektliste und speichern sie diese in eine Variable vom Typ `Z_PROJECT_T`.
7. Iterieren Sie mit einem `LOOP` über die Projektliste. Übertragen Sie die Einträge aus der Projektliste in die Kontextstruktur (Typ `WD_THIS->ELEMENT_PROJECTS`), und füllen Sie die Tabelle für den Knoten `PROJECTS` mit diesen Einträgen. Ein entsprechendes Feld `LT_PROJECTS` wurde schon vom Wizard angelegt. Verschieben Sie abschließend die Methode zum Binden

der Tabelle an den Kontext ans Ende der Methode `FILL_PROJECTS_NODE`. Die Methode sollte daraufhin ähnlich wie in Listing 4.1 aussehen.

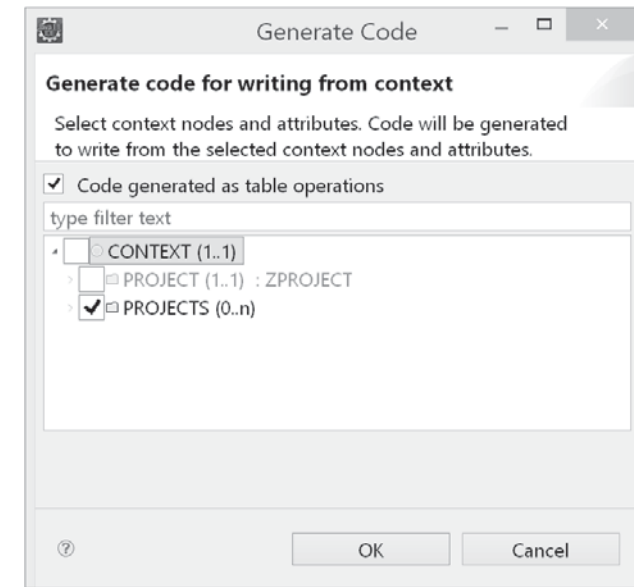


Abbildung 4.15 Auswahl eines Kontextknotens

Knoten mit Dictionary-Bezug

«

Wenn Sie einen Kontextknoten mit Bezug zu einem ABAP-Dictionary-Objekt anlegen, können Sie wie gewohnt Strukturen bzw. Tabellen dieses Typs direkt an den Knoten binden. In unserem Beispiel ist das nicht möglich, da wir die Knoten manuell aufgebaut haben und die Struktur nicht mit der auf der Datenbank kompatibel ist.

```
METHOD fill_projects_node.
  DATA: lo_nd_projects TYPE REF TO if_wd_context_node,
         lt_projects    TYPE wd_this->elements_projects,
         ls_project     TYPE wd_this->element_projects,
         lt_project_list TYPE z_project_t.

  lo_nd_projects = wd_context->get_child_node(
    name = wd_this->wdctx_projects ).
  lt_project_list = zcl_project=>list_projects( ).
  LOOP AT lt_project_list INTO DATA(ls_project_list).
    ls_project-project_id = ls_project_list-project_id.
    ls_project-project_name = ls_project_list-project_name.
    APPEND ls_project TO lt_projects.
```

```

ENDLOOP.
lo_nd_projects->bind_table( new_items = lt_projects
    set_initial_elements = abap_true ).
ENDMETHOD.

```

Listing 4.1 Implementierung der Methode FILL_PROJECTS_NODE

Nun kümmern wir uns um die Implementierung der verbleibenden Methoden des Komponenten-Controllers:

1. Drücken Sie die Tasten `[Strg] + [0]`, und wählen Sie aus dem QUICK OVERVIEW die Methode `SELECT_PROJECT`, um zur Implementierung dieser Methode zu springen.
2. Sie müssen nicht zurück zur Registerkarte `METHODS` springen, um sich die Methodensignatur ins Gedächtnis zu rufen. Setzen Sie den Cursor auf den Methodennamen, und drücken Sie `[F2]`. Die ABAP Element Info zeigt Ihnen die Signatur der Methode an (siehe Abbildung 4.16).

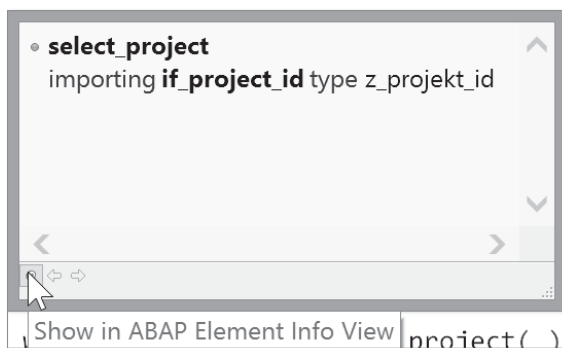


Abbildung 4.16 ABAP Element Info zur Methode SELECT_PROJECT

3. Fügen Sie den Code aus Listing 4.2 als Implementierung der Methode ein.

```

METHOD select_project.
    DATA lo_nd_project TYPE REF TO if_wd_context_node.
    DATA lo_el_project TYPE REF TO if_wd_context_element.
    DATA ls_project TYPE wd_this->element_project.
    DATA lo_nd_project_issues TYPE REF TO if_wd_context_node.
    DATA lt_project_issues TYPE
        wd_this->elements_project_issues.
    wd_this->mr_project = NEW #( if_project_id ).
    lo_nd_project = wd_context->get_child_node(
        name = wd_this->wdctx_project ).
    lo_el_project = lo_nd_project->get_element( ).
    ls_project = wd_this->mr_project->get_project( ).

```

```

lo_el_project->set_static_attributes(
    static_attributes = ls_project ).
* Issues setzen
lo_nd_project_issues = lo_nd_project->get_child_node(
    name = wd_this->wdctx_project_issues ).
lt_project_issues = wd_this->mr_project->get_issues( ).

lo_nd_project_issues->bind_table(
    new_items = lt_project_issues
    set_initial_elements = abap_true ).
ENDMETHOD.

```

Listing 4.2 Implementierung der Methode SELECT_PROJECT

4. Navigieren Sie zur Methode `SUPPLY_ISSUES`, und fügen Sie den Code aus Listing 4.3 als Implementierung ein.

```

METHOD supply_issues.
    DATA:
        ls_project TYPE zproject,
        lt_issues TYPE wd_this->elements_issues,
        ls_issues TYPE wd_this->element_issues,
        lt_issue_list TYPE z_issue_t,
        lr_project TYPE REF TO zcl_project.

    parent_element->get_static_attributes(
        IMPORTING
            static_attributes = ls_project
    ).
    IF ls_project-project_id IS NOT INITIAL.
        lr_project = NEW #( ls_project-project_id ).
        lt_issue_list = lr_project->get_issues( ).
        LOOP AT lt_issue_list INTO DATA(ls_issue_list).
            ls_issues-issue_id = ls_issue_list-issue_id.
            ls_issues-issue_name = ls_issue_list-name.
            ls_issues-issue_description =
                ls_issue_list-description.
            APPEND ls_issues TO lt_issues.
        ENDLOOP.
    ENDIF.
    node->bind_table( new_items = lt_issues
        set_initial_elements = abap_true ).
ENDMETHOD.

```

Listing 4.3 Implementierung der Methode SUPPLY_ISSUES

5. Zu guter Letzt müssen wir noch dafür sorgen, dass unsere Projektliste initial befüllt wird. Navigieren Sie daher zur Methode `WDDOINIT`, und fügen Sie den folgenden Aufruf ein:

```
wd_this->fill_projects( ).
```

Sie können dazu die Code-Vervollständigung für die Variable `WD_THIS` verwenden.

6. Aktivieren Sie Ihre Änderungen.

Damit sind unsere Arbeiten am Komponenten-Controller abgeschlossen.

4.2 Views und Windows

In diesem Abschnitt widmen wir uns den visuellen Bestandteilen unserer Web-Dynpro-Komponente. Zunächst erstellen wir die Views, die die Navigation und die Detailanzeige übernehmen sollen. Anschließend kombinieren wir diese Views in einem Window.

4.2.1 Views

Der View-Editor ist ähnlich aufgebaut wie der des Komponenten-Controllers. Die Registerkarten, die Sie aus der ABAP Workbench gewöhnt sind, wurden mehr oder weniger 1:1 nachempfunden und finden sich nun am unteren Rand des Editors. Wenn Sie im Project Explorer doppelt auf einen View klicken, landen Sie zunächst in der Kontextansicht. Diese ist genauso aufgebaut wie die Sicht, die Sie schon vom Komponenten-Controller kennen. Auf der linken Seite sehen Sie den (im Moment noch leeren) Kontext des Views. In unserem Fall sehen Sie auf der rechten Seite die Kontextknoten, die Sie im Komponenten-Controller angelegt haben.

Ordnen Sie nun die benötigten Knoten aus dem Komponenten-Controller dem View zu (*Mapping*).

1. Wechseln Sie zum View `V_MAIN` auf der Registerkarte `CONTEXT`.
2. Expandieren Sie den Kontextbaum auf der rechten Seite, bis Sie zur Ebene des Knotens `PROJECTS` gelangen.
3. Markieren Sie den Knoten `PROJECTS`, und ziehen Sie ihn per Drag & Drop auf den (Wurzel-)Knoten `CONTEXT` auf der linken Seite.

4. Der Dialog `CONTEXT MAPPING` öffnet sich (siehe Abbildung 4.17). Hier können Sie, anders als im SAP GUI, noch einmal genau differenzieren, welche Elemente Sie mappen möchten.

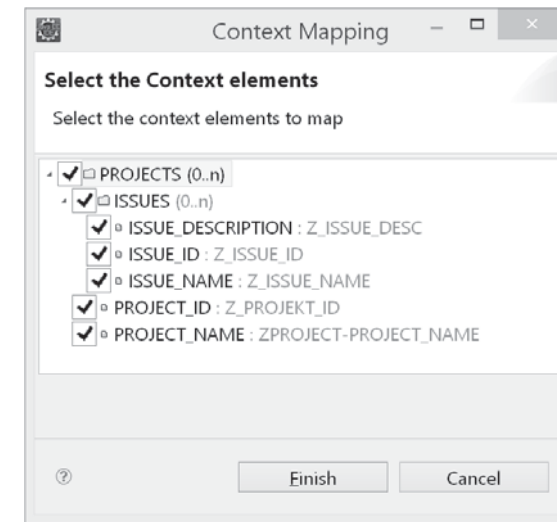


Abbildung 4.17 Elemente für das Kontext-Mapping auswählen

5. Der Knoten mit den gewählten Elementen wird in den View-Controller kopiert, und das Mapping wird angelegt. Es wird Ihnen durch Verbindungslinien angezeigt (siehe Abbildung 4.18).

6. Aktivieren Sie Ihre Änderungen.

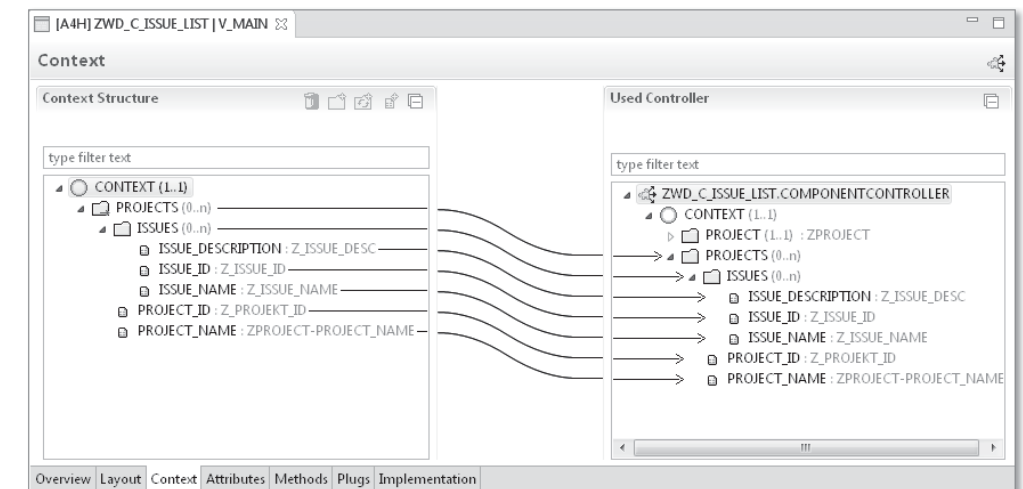


Abbildung 4.18 Anzeige des Kontext-Mappings

Als Nächstes bauen wir das *Layout* der Views auf. Der Layout-Editor lässt sich etwas anders bedienen als sein Verwandter im SAP GUI, das neue Bedienkonzept kann Ihnen die Arbeit aber erleichtern. Neben dem Layout-Editor arbeiten Sie mit den Views **OUTLINE** und **PROPERTIES**:

► **Layout-Editor**

Im grafischen Layout-Editor wird eine Vorschau angezeigt, die an die Layout-Vorschau der ABAP Workbench erinnert. Allerdings gibt es keine Palette mit UI-Elementen. Möchten Sie ein UI-Element im Layout-Editor einfügen, können Sie dies über das Kontextmenü tun (mit **INSERT • ELEMENT**). Ein Dialog stellt Ihnen die zum Einfügen verfügbaren UI-Elemente zur Auswahl. Mit dem Ankreuzfeld **HIGHLIGHT CONTAINERS** oben rechts können Sie die Anzeige der Umrisse sämtlicher Container aktivieren, was beim Erstellen verschachtelter Layouts sehr hilfreich sein kann. Ist das Feld deaktiviert, wird Ihnen eine Vorschau angezeigt, die fast dem Endergebnis entspricht.

► **Outline-View**

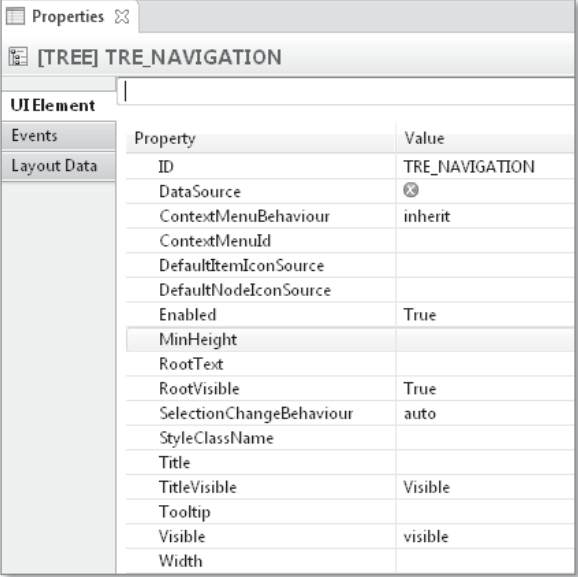
Die Hierarchie im View **OUTLINE** entspricht der Hierarchiedarstellung in der ABAP Workbench. Bei einem leeren Layout sehen Sie die Standardcontainer **CONTEXT_MENUES** und **ROOTELEMENTCONTAINER**. Neue Elemente fügen Sie über das Kontextmenü hinzu (**INSERT • ELEMENT**). Das Layout lässt sich viel intuitiver bearbeiten als in der ABAP Workbench. Neben der Möglichkeit, Elemente nach oben und unten zu verschieben, wurden auch das Kopieren, Ausschneiden und Einfügen von Elementen über das Menü oder Shortcuts ermöglicht. Außerdem können Sie mehrere Elemente markieren und gleichzeitig bearbeiten.

► **Properties-View**

Im View **PROPERTIES** werden die Eigenschaften der UI-Elemente angezeigt. Sie sind in Kategorien unterteilt, die auf jeweils eigenen Registerkarten untergebracht sind:

- **UI ELEMENT**: allgemeine Eigenschaften
- **LAYOUT**: Auf der Registerkarte **LAYOUT** stellen Sie den Layout-Manager ein und nehmen allgemeine Einstellungen zum Layout vor.
- **LAYOUT DATA**: Auf der Registerkarte **LAYOUT DATA** stellen Sie die elementspezifischen Eigenschaften zum Layout-Manager des übergeordneten Containers ein.
- **EVENTS**: Auf der Registerkarte **EVENTS** können Sie dem aktuell gewählten UI-Element Aktionen zuweisen.

Die auf der Registerkarte **UI ELEMENT** angezeigte Eigenschaft ist immer die Element-ID. Darauf folgen sämtliche Pflichtfelder des UI-Elements. Alle anderen Eigenschaften werden in alphabetischer Reihenfolge angezeigt (siehe Abbildung 4.19). Der View **PROPERTIES** bietet zudem die bekannten Filter- und Suchfunktionen über ein einzelnes Eingabefeld.



Property	Value
ID	TRE_NAVIGATION
DataSource	⊗
ContextMenuBehaviour	inherit
ContextMenuId	
DefaultItemIconSource	
DefaultNodeIconSource	
Enabled	True
MinHeight	
RootText	
RootVisible	True
SelectionChangeBehaviour	auto
StyleClassName	
Title	
TitleVisible	Visible
Tooltip	
Visible	visible
Width	

Abbildung 4.19 View »Properties« zu einem UI-Element

Ist im **OUTLINE-View** bzw. im **Layout-Editor** mehr als ein Element markiert, zeigt der **PROPERTIES-View** alle Eigenschaften an, die die selektierten Elemente gemeinsam haben. Bei gleichen Werten werden diese ebenfalls angezeigt, bei unterschiedlichen Werten bleibt das Feld **VALUE** leer. Sie können aber in beiden Fällen die Eigenschaftswerte aller markierten Elemente gleichzeitig bearbeiten.

Um Ihnen zu zeigen, wie Sie mit den Werkzeugen arbeiten, fügen wir als Erstes neue UI-Elemente ein.

1. Klicken Sie im **OUTLINE-View** mit der rechten Maustaste auf das Element **ROOTELEMENTCONTAINER**, und wählen Sie im Kontextmenü **INSERT • ELEMENT**.
2. Es öffnet sich der Dialog **NEW UI ELEMENT** (siehe Abbildung 4.20). Wählen Sie aus der Liste das UI-Element **TransparentContainer**.
3. Nennen Sie das Element »**TCN_NAVIGATION**«, und bestätigen Sie mit **OK**.

4. Wiederholen Sie den Vorgang, um einen weiteren Transparent Container einzufügen. Nennen Sie diesen »TCN_DETAIL«.
5. Die neu eingefügten Transparent Container tauchen nun im OUTLINE-View auf und können verwendet werden.

Der Dialog NEW UI ELEMENT in Abbildung 4.20 unterscheidet sich etwas vom Fenster UI-ELEMENT EINFÜGEN der ABAP Workbench. Sie haben zunächst die vollständige Liste aller verfügbaren UI-Elemente zur Auswahl und können die Liste über das Eingabefeld oberhalb der Liste auf einen Suchbegriff einschränken. Über die Auswahlliste CATEGORIES können Sie die Liste der UI-Elemente außerdem auf eine bestimmte Kategorie einschränken. Die Kategorien entsprechen denen, die Sie vielleicht schon aus der Palette des Layout-Editors in der ABAP Workbench kennen.

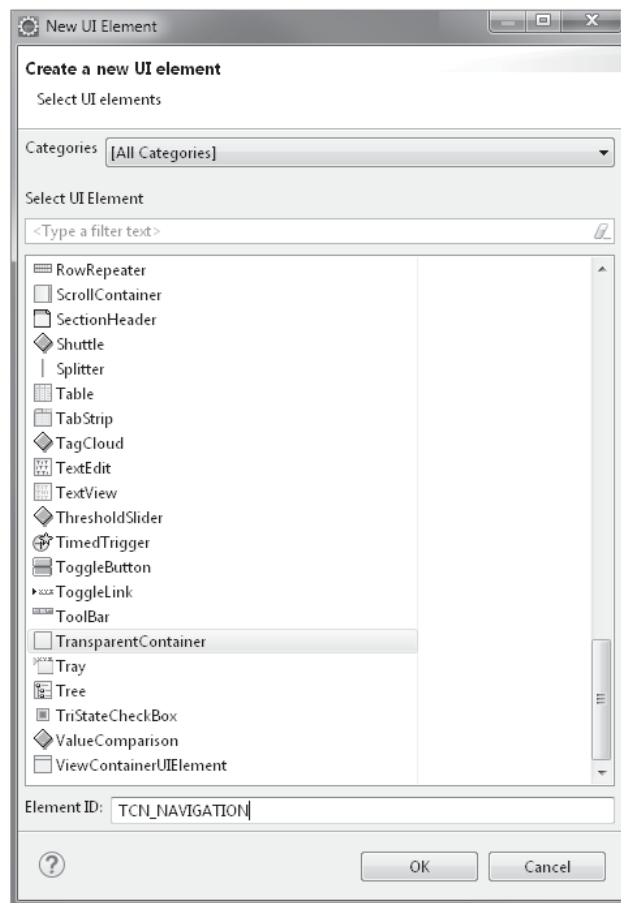


Abbildung 4.20 Dialog zur Anlage eines neuen UI-Elements

Einige UI-Elemente, wie das Input Field, werden übrigens direkt zusammen mit ihrem Label (Bezeichner) eingefügt. Wenn Sie lieber mit dem Layout-Editor arbeiten, ist das Vorgehen beim Einfügen neuer UI-Elemente analog.

Als Nächstes können wir die Eigenschaften unserer UI-Elemente, der Transparent Container, bearbeiten:

1. Klicken Sie im OUTLINE-View doppelt auf das Element ROOTELEMENTCONTAINER, um den PROPERTIES-View zu öffnen (Alternativ markieren Sie das Element und wählen $\text{Ctrl} + \text{Strg} + \text{Z}$ – diese Tastenkombination funktioniert auch im Layout-Editor.).
2. Wechseln Sie auf die Registerkarte LAYOUT, und wählen Sie für die Eigenschaft LAYOUT den Eintrag MatrixLayout.
3. Markieren Sie nun die angelegten Transparent Container TCN_NAVIGATION und TCN_DETAIL. Die Eigenschaften Ihrer Auswahl werden sofort im PROPERTIES-View angezeigt.
4. Stellen Sie nun gleichzeitig für diese beiden Container das LAYOUT auf MATRIX LAYOUT um. Die restlichen Einstellungen können Sie beibehalten.

Mit den beiden Transparent Containern allein können wir noch nicht viel anfangen, wir müssen auch eine Navigationsmöglichkeit anlegen. Dazu verwenden wir einen Navigationsbaum. Aufgrund des Aufbaus unseres Kontextes müssen wir dazu keinen großen Aufwand betreiben:

1. Fügen sie im Container TCN_NAVIGATION ein neues UI-Element vom Typ Tree an, und nennen Sie es »TRE_NAVIGATION«.
2. Im PROPERTIES-View wechseln Sie in den Bereich UI ELEMENT und markieren die Eigenschaft DataSource. Es erscheint ein Button mit der Aufschrift BIND, auf den Sie klicken. Wählen Sie im so aufgerufenen Menü den Eintrag TO CONTEXT.

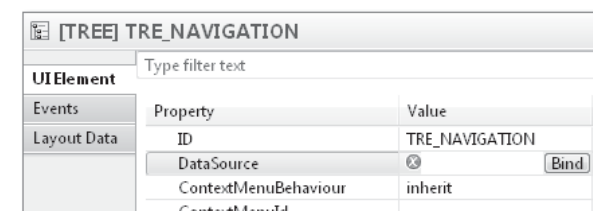


Abbildung 4.21 Kontext an das UI-Element anbinden

3. Daraufhin erscheint der Dialog BIND TO CONTEXT. Wählen Sie hier den Kontextknoten PROJECTS, für den nun ein kleines Ordnersymbol als Wert

der Eigenschaft angezeigt wird. Zudem erscheint der Button RESET, der es Ihnen ermöglicht, die Bindung wieder zu lösen.

- Als Letztes wollen wir dem Wurzelknoten des Baums noch einen nachvollziehbaren Namen geben. Wählen Sie dazu die Eigenschaft `RootText`, klicken Sie auf den Button BIND, und wählen Sie TO OTR im erscheinenden Menü.
- Im folgenden Dialog SEARCH ONLINE TEXT REPOSITORY OBJECT geben Sie als Paket das Standardpaket »SOTR_VOCABULARY_BASIC« an (siehe Abbildung 4.22). Sie können in diesem Feld auch die Code-Vervollständigung verwenden.
- Wählen Sie aus der Liste der OTR-Texte den Eintrag NAVIGATION aus, und bestätigen Sie mit OK. Der ausgewählte Text wurde damit an die Eigenschaft gebunden.

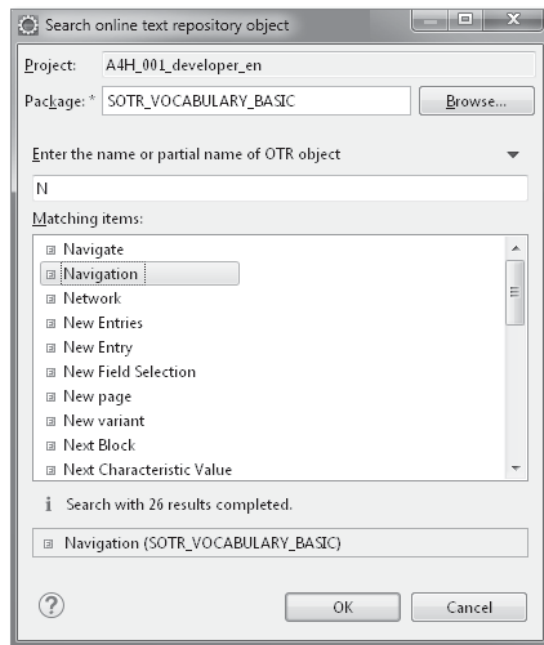


Abbildung 4.22 Dialog zur Anbindung eines Textes aus dem Online Text Repository

» OTR-Texte

Sie müssen OTR-Texte in den Eigenschaften nicht über den Dialog anbinden, sondern können auch »\$OTR:<Paket>/<Aliasname>« über die Tastatur eingeben. Eigene OTR-Texte können Sie über Transaktion SOTR_EDIT im SAP GUI pflegen. Eine direkte Eclipse-Integration ist derzeit noch nicht verfügbar.

Sie haben nun gesehen, wie Sie einzelne Werte an den Kontext binden können. Wenn Sie schon größere Anwendungen in Web Dynpro geschrieben haben, wissen Sie, wie zeitaufwendig es sein kann, viele UI-Elemente an den Kontext zu binden. Die Web-Dynpro-Tools in Eclipse bieten zu diesem Zweck einen *neuen Kontext-Editor* an, den wir uns im nächsten Beispiel ansehen werden. Dazu legen wir zu unserem Baum die Knoten für Projekte und Aufgaben an und binden diese in einem Rutsch an den Kontext:

- Klicken Sie im OUTLINE-View mit der rechten Maustaste auf das UI-Element `TRE_NAVIGATION`, und wählen Sie INSERT • NODE TYPE im Kontextmenü.
- Wählen Sie aus der Liste der verfügbaren Elemente `TreeNodeType`, nennen Sie das anzulegende Element »TNT_PROJECT«, und bestätigen Sie mit OK.
- Fügen Sie auf die gleiche Weise ein `TreeNodeType` mit dem Namen »TIT_ISSUE« ein.
- Klicken Sie nun mit der rechten Maustaste auf `TRE_NAVIGATION`, und wählen Sie BIND TO CONTEXT im Kontextmenü (`Ctrl + Strg + P`).
- Wählen Sie die Eigenschaft `DataSource` unterhalb des Elements `TNT_PROJECT`, und ziehen Sie diese per Drag & Drop auf den Knoten `PROJECTS` auf der rechten Seite. Sie haben nun die Eigenschaft an den Kontext gebunden, was Ihnen durch eine Verbindungslinie angezeigt wird.
- Alle Eigenschaften, die gebunden werden sollen, finden Sie in Tabelle 4.1. Schließlich sollte Ihr Binding wie in Abbildung 4.23 aussehen. Schließen Sie die Bearbeitung mit FINISH ab.

Layout	Kontext		
Element	Eigenschaft	Knoten	Attribut
TRE_NAVIGATION	DataSource	PROJECTS	–
TNT_PROJECT	DataSource	PROJECTS	–
TNT_PROJECT	Text	PROJECTS	Project_Name
TIT_ISSUE	DataSource	ISSUES	–
TIT_ISSUE	Text	ISSUES	Issue_Name
TIT_ISSUE	Tooltip	ISSUES	Issue_Description

Tabelle 4.1 Kontext-Binding für das UI-Element TRE_NAVIGATION

Auf diese Weise können Sie beliebige UI-Elemente mit dem Kontext verknüpfen. Dabei werden Ihnen im Editor alle UI-Elemente angezeigt, die dem Element zugeordnet sind, das Sie ausgewählt haben, z. B. ein transparenter

Container und alle Elemente, die sich in diesem Container befinden. Sie haben auch die Möglichkeit, die Ansicht im Dialog BIND TO CONTEXT nur auf die obligatorischen Eigenschaften einzuschränken. Klicken Sie dazu auf den Button SHOW ONLY MANDATORY UI ELEMENT PROPERTIES (🔗) oberhalb des Layout-Baums.

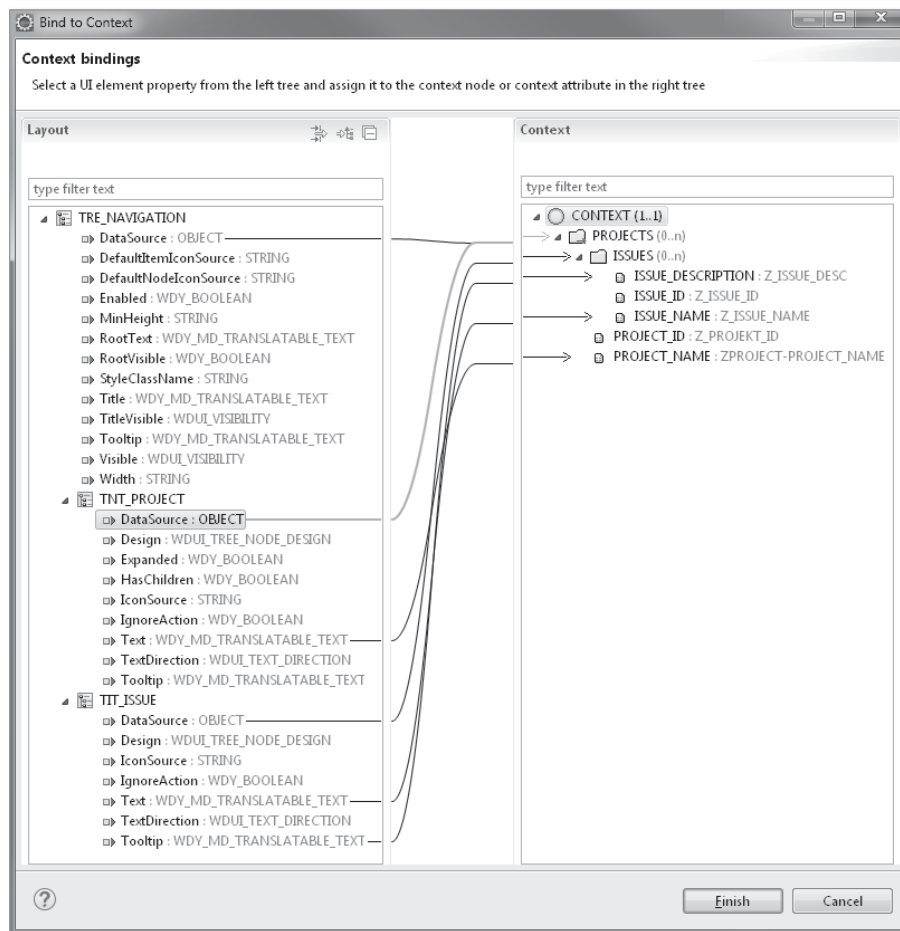


Abbildung 4.23 Dialog für das Kontext-Binding

Im nächsten Schritt bestimmen wir, wie die Detailanzeige aussehen soll, um ein ausgewähltes Projekt anzuzeigen. Diese soll rechts neben dem Navigationsbaum zu finden sein. Sie besteht aus einem einfachen Formular sowie einer Tabelle mit sämtlichen Issues zu einem Projekt.

Es gibt in den Web-Dynpro-Tools in Eclipse keine Möglichkeit, ein Formular aus dem Kontext zu generieren. Allerdings ist diese Methode auch in der

ABAP Workbench mit einigen Nacharbeiten verbunden, sodass die Zeiterparnis durch die Generierung dort nur minimal ist. Durch den neuen Binding-Editor und die Möglichkeit, Elemente zu kopieren, legen Sie Formulare daher in Eclipse mit etwas Übung eher schneller an. Gehen wir die Anlage an unserem Beispiel gemeinsam durch:

1. Fügen Sie unterhalb des UI-Elements TCN_DETAIL ein UI-Element `Caption` mit dem Namen »CPT_TITLE« ein.
2. Fügen Sie unterhalb des Elements TCN_DETAIL ein `InputField` ein, und nennen Sie es »INP_ID«. Eclipse fügt automatisch das passende Label dazu ein.
3. Markieren Sie das Label und das Input Field, und drücken Sie `[Strg] + [C]`, um beides zu kopieren.
4. Markieren Sie wieder das Element TCN_DETAIL, und drücken Sie dreimal `[Strg] + [V]`, um drei neue Label/InputField-Pärchen einzufügen.
5. Markieren Sie nun alle Label-Elemente, und wählen Sie im PROPERTIES-View unter LAYOUTDATA • LAYOUTDATA den Eintrag `MatrixHeadData`. Damit bewirken Sie, dass jedes Label in einer neuen Zeile angezeigt wird.
6. Als Nächstes benennen Sie die IDs der Elemente um. Markieren Sie dazu das Element, das Sie umbenennen möchten, und drücken Sie `[F2]`. Sie können es dann direkt im OUTLINE-View umbenennen. Einen Vorschlag für eine mögliche Benennung finden Sie in Abbildung 4.24.

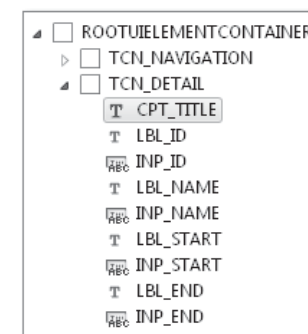


Abbildung 4.24 Das Detailformular

7. Es fehlen noch die Label-Zuordnungen für die Eingabefelder `LBL_NAME`, `LBL_START` sowie `LBL_END`, da wir diese lediglich kopiert hatten. Markieren Sie zuerst `LBL_NAME`, navigieren Sie im PROPERTIES-View zu UIELEMENT • LABELFOR, und wählen Sie das Label `INP_NAME` aus der Dropdown-Liste.

8. Gehen Sie bei den Eingabefeldern LBL_START und LBL_END auf die gleiche Weise vor.

Vielleicht haben Sie beim Anlegen des Detailformulars gedacht, dass es schöner wäre, diese Sicht in einen eigenen View auszulagern. Aber alles löschen und neu anlegen? Das ist nicht notwendig. Sie können alle UI-Elemente so, wie sie sind, in einen neuen View verlegen:

1. Klicken Sie im OUTLINE-View mit der rechten Maustaste auf den Transparent Container TCN_DETAIL, und wählen Sie COPY (Strg) + (C) aus dem Kontextmenü.
2. Klicken Sie im Project Explorer mit der rechten Maustaste auf die Web-Dynpro-Komponente ZWD_C_ISSUE_LIST, und wählen Sie NEW • VIEW im Kontextmenü.
3. Nennen Sie den neuen View »V_PROJECT_DETAILS«, und legen Sie diesen an.
4. Wechseln Sie auf die Registerkarte LAYOUT, markieren Sie den Container ROOTELEMENTCONTAINER, und drücken Sie (Strg) + (V). Die kopierte UI-Struktur wird eingefügt.
5. Wechseln Sie zur Registerkarte LAYOUT des Views V_MAIN, und löschen Sie das Element TCN_DETAIL. Fügen Sie stattdessen ein VIEWCONTAINERUIELEMENT ein, und nennen Sie es »VCN_DETAILS«.

Es wäre sogar noch besser, die Detailsicht für verschiedene Objekte in jeweils eigenen Komponenten zur Verfügung zu haben. Selbst dies können Sie über die Kopierfunktion erreichen. Sie können UI-Elemente, Methoden, Kontextknoten oder Attribute einfach zwischen Komponenten hin- und her kopieren.

Schließlich binden wir unsere Detailsicht an den Kontext:

1. Wechseln Sie dazu zur Registerkarte CONTEXT des Views V_PROJECT_DETAILS.
2. Ziehen Sie den Kontextknoten PROJECTS aus dem Komponenten-Controller in den Kontext des Views, um diesen zu mappen.
3. Wählen Sie alle Elemente aus (auch den Unterknoten PROJECT_ISSUES).
4. Wechseln Sie zur Registerkarte LAYOUT, und markieren Sie das Element TCN_DETAIL. Drücken Sie (Strg) + (P), um den Binding-Dialog zu öffnen. Stellen Sie ein Binding wie in Abbildung 4.25 her, und bestätigen Sie mit FINISH.

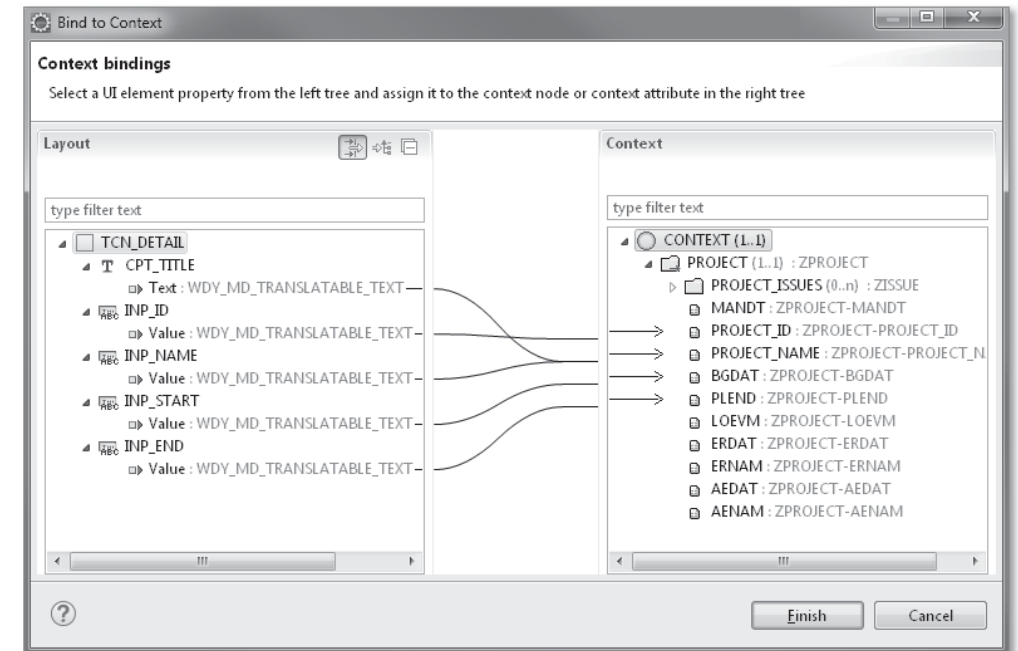


Abbildung 4.25 Kontext-Mapping für den View V_PROJECT_DETAILS

In unserer Dateiansicht benötigen wir noch eine Tabelle zur Anzeige der Aufgaben zu einem Projekt. Zur Anlage von Tabellen gibt es einen Assistenten, der Ihnen einiges an Arbeit abnehmen kann. Dieser funktioniert sowohl für das UI-Element Table als auch für das Element CTable.

1. Klicken Sie mit der rechten Maustaste auf den Knoten TCN_DETAILS, und wählen Sie INSERT • ELEMENT. Wählen Sie Table aus der Liste der UI-Elemente, und nennen Sie das neue Element »TBL_ISSUES«. Bestätigen Sie Ihre Angaben.
2. Beim Einfügen der Tabelle öffnet sich ein Assistent, der Ihnen den aktuellen Kontext zur Auswahl stellt. Markieren Sie hier den Knoten PROJECT_ISSUES, und heben Sie die Markierung des Feldes MANDT auf. Bestätigen Sie mit NEXT.
3. Im nächsten Schritt können Sie Änderungen an den Spalteneigenschaften vornehmen. Wir begnügen uns aber mit den Voreinstellungen und bestätigen mit FINISH.
4. Ändern Sie nun noch in den Eigenschaften Ihrer Tabelle den Wert für die Eigenschaft LayoutData auf MatrixHeadData und den Wert für die Eigenschaft ColSpan auf 2.

Sie haben nun eine Tabelle mit Issues eingebaut. Die Kontextbindung und die Eigenschaften der Spalten können Sie jederzeit über das Kontextmenü mit der Funktion EDIT TABLE ändern. Zudem können Sie auf die Tabelle und deren Spalten im OUTLINE-View zugreifen.

Nun müssen wir noch dafür sorgen, dass dem Anwender die Details auch angezeigt werden. Dazu müssen wir ein *Inbound*- und ein *Outbound*-Plug in unseren beiden Views definieren. Diese Plugs sollen bei der Auswahl eines Projektes ausgelöst werden. Wechseln Sie dazu zur Registerkarte PLUGS des Views V_MAIN, und klicken Sie auf ADD OUTBOUND PLUG. Im folgenden Dialog geben Sie »OP_PROJECT_DETAILS« als Namen für das neue Plug ein. Sie haben hier auch die Möglichkeit, über die Funktion ADD PARAMETER zusätzliche Übergabeparameter anzulegen. Dies ist in unserem Fall aber nicht notwendig. Legen Sie den Inbound-Plug für den DETAIL-View V_PROJECT_DETAILS über die Funktion ADD INBOUND PLUG an, und nennen Sie diesen Plug »IP_SHOW_PROJECT«.

Damit die Plugs ausgelöst und die korrekten Projektdetails geladen werden, legen wir eine neue Aktion für den Projektknoten an. Nach dem gleichen Prinzip können Sie bei Bedarf auch Ereignisbehandler für andere UI-Elemente anlegen:

1. Wechseln Sie zur Registerkarte LAYOUT des Views V_PROJECT_DETAILS. Klicken Sie im OUTLINE-View doppelt auf das UI-Element TNT_PROJECTS, und wechseln Sie im PROPERTIES-View auf die Registerkarte EVENT.
2. Markieren Sie die Eigenschaft onAction, und klicken Sie auf den Button CREATE.
3. Im folgenden Dialog nennen Sie die neue Aktion »DISPLAY_PROJECT« und vergeben eine Beschreibung (siehe Abbildung 4.26). Der Dialog zeigt Ihnen direkt an, wie der Ereignisbehandler zu Ihrer Aktion heißen wird. Sie können außerdem den Aktionstyp wählen. Wir belassen es bei der Einstellung STANDARD und bestätigen die Anlage mit FINISH.

Über den Button Go gelangen Sie in die Implementierung der Ereignisbehandlungsmethode. In dieser Methode sorgen Sie dafür, dass das gewählte Projekt geladen wird. Dazu rufen Sie die Methode SELECT_PROJECT im Komponenten-Controller auf und lösen anschließend das Plug OP_PROJECT_DETAILS aus. Es gibt keinen Assistenten, der die Methodenaufrufe oder den Code zum Auslösen eines Plugs erzeugt. Stattdessen können Sie die Code-Vervollständigung verwenden. Der Komponenten-Controller steht in jedem View über das Attribut WD_COMP_CONTROLLER zur Verfügung. Über die Tasten-

kombination `[Strg]` + Leertaste können alle verfügbaren Methoden aufgelistet und inklusive ihrer Signatur über die Tastenkombination `[⇧]` + `[↵]` eingefügt werden. Zum Auslösen eines Plugs steht zudem das Template `firePlug` zur Verfügung. Es fügt eine Code-Vorlage ein, die Sie nur noch um den Namen des Plugs ergänzen müssen. Diesen finden Sie ebenfalls mithilfe der Code-Vervollständigung.

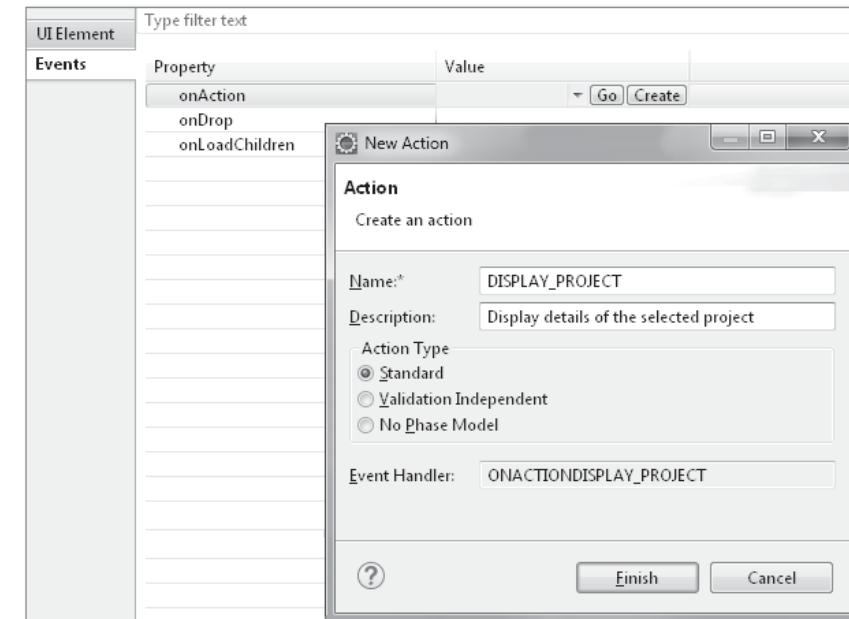


Abbildung 4.26 Dialog »New Action«

Bauen wir nun den Ereignisbehandler zusammen:

1. Mithilfe des Codes aus Listing 4.4 ermitteln Sie zunächst die ID des gewählten Projekts.

```
DATA:
  lr_project TYPE REF TO if_wd_context_element,
  lf_project_id TYPE z_projekt_id.

lr_project = wdevent->get_context_element(
  name = 'CONTEXT_ELEMENT' ).

lr_project->get_attribute(
  EXPORTING
    name = `PROJECT_ID`
```

```
IMPORTING
  value = lf_project_id ).
```

Listing 4.4 Projekt-ID ermitteln

- Als Nächstes rufen Sie die Selektionsmethode im Komponenten-Controller auf. Tippen Sie dazu »wd_comp_controller->« ein, gefolgt der Tastenkombination `[Strg]` + Leertaste. Wählen Sie aus der Liste die Methode `SELECT_PROJECT`, und fügen Sie den Code mit `[↵]` + `[↵]` ein.
- Vervollständigen Sie den Aufruf mit der Projekt-ID, sodass Sie die folgende Zeile erhalten:

```
wd_comp_controller->select_project( if_project_id = lf_project_id ).
```

- Fügen Sie das Template `firePlug` ein, und geben Sie »OP_PROJECTS_DETAILS« als Namen für das Plug ein. Am Ende sollte Ihr Quelltext wie in Listing 4.5 aussehen.

```
METHOD onactiondisplay_project.
  DATA:
    lr_project TYPE REF TO if_wd_context_element,
    lf_project_id TYPE z_projekt_id.

  lr_project = wdevent->get_context_element(
    name = 'CONTEXT_ELEMENT' ).
  lr_project->get_attribute(
    EXPORTING
      name = `PROJECT_ID`
    IMPORTING
      value = lf_project_id ).
  wd_comp_controller->select_project(
    if_project_id = lf_project_id ).
  WD_THIS->fire_op_projects_details_plg( ).
ENDMETHOD.
```

Listing 4.5 Implementierung der Methode `ONACTIONDISPLAY_PROJECT`

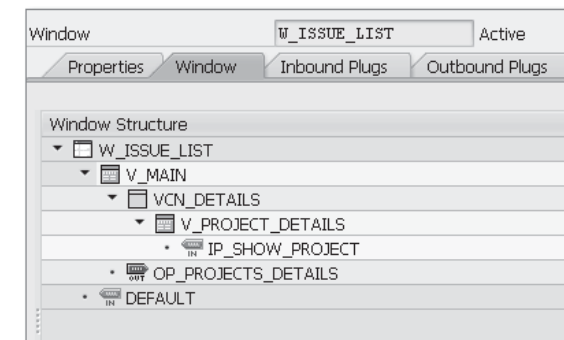
Wenn Sie diese Arbeitsabläufe erst einmal verinnerlicht haben, geht das Einfügen von Code schnell von der Hand. Der Vorteil im Vergleich zum Code-Wizard liegt vor allem darin, dass Sie nicht aus dem Fluss gerissen werden. Es ist nicht notwendig, die Maus in die Hand zu nehmen und eine Funktion in der Toolbar aufzurufen. Die Hilfen stehen Ihnen direkt während der Arbeit im Quelltext zur Verfügung.

4.2.2 Windows

Damit unser neuer View angezeigt werden kann, muss er im *Window-Controller* unserer Web-Dynpro-Komponente eingebunden werden. Der Editor für den Window-Controller ist einer der wenigen Editoren, die noch nicht in Eclipse umgesetzt wurden. Daher ist an dieser Stelle auch wenig Neues zu erklären:

- Klicken Sie im Project Explorer doppelt auf das Window `W_ISSUE_LIST`. Ein in Eclipse integriertes SAP-GUI-Fenster mit dem gewählten Window öffnet sich.
- Wechseln Sie mit `[Strg]` + `[F1]` in den Editiermodus, und expandieren Sie die `WINDOW STRUCTURE` vollständig.
- Klicken Sie mit der rechten Maustaste auf den View `VCN_DETAILS`, und wählen Sie `EMBED VIEW`, um den View einzubinden. Wählen Sie über die Werthilfe des Feldes `VIEW TO BE EMBEDDED` den Eintrag `V_PROJECT_DETAILS`.
- Klappen Sie auch die Views `VCN_DETAILS` und `V_PROJECT_DETAILS` auf.
- Ziehen Sie den Outbound-Plug `OP_PROJECT_DETAILS` auf den Inbound-Plug `IP_SHOW_PROJECT`, und aktivieren Sie Ihre Änderungen (`[Strg]` + `[F3]`).

Damit haben wir die Views in das Window eingebettet. Das Ergebnis sollte wie in Abbildung 4.27 aussehen.

**Abbildung 4.27** Struktur des Windows `W_ISSUE_LIST`

4.3 Web-Dynpro-Applikationen

Web-Dynpro-Komponenten lassen sich selbst nicht ausführen, es wird zwangsläufig eine Applikation benötigt, die dem Endanwender den Zugriff

auf das System erlaubt. Im Folgenden lesen Sie, was Sie tun müssen, um unsere Komponente ausführen zu können.

Eine Web-Dynpro-Anwendung ordnet der Komponente einen aktiven INTER-FACE-View (ein Window) zu, das zur Anzeige gebracht wird. Zudem wird ein ICF-Service (ICF = Internet Communication Framework) erstellt, über den die Applikation aufgerufen wird.

Das Erstellen einer Applikation ist einfach. Der Aufwand beschränkt sich mehr oder weniger darauf, sich einen Namen und eine Beschreibung einzufallen zu lassen:

1. Klicken Sie im Project Explorer mit der rechten Maustaste auf die Komponente ZWD_C_ISSUE_LIST, und wählen Sie NEW • APPLICATION.
2. Im folgenden Dialog geben Sie z. B. »ZWD_A_ISSUE_LIST« als Namen für die Applikation und »Issue List« als Beschreibung ein (siehe Abbildung 4.28). Fahren Sie mit einem Klick auf NEXT fort.

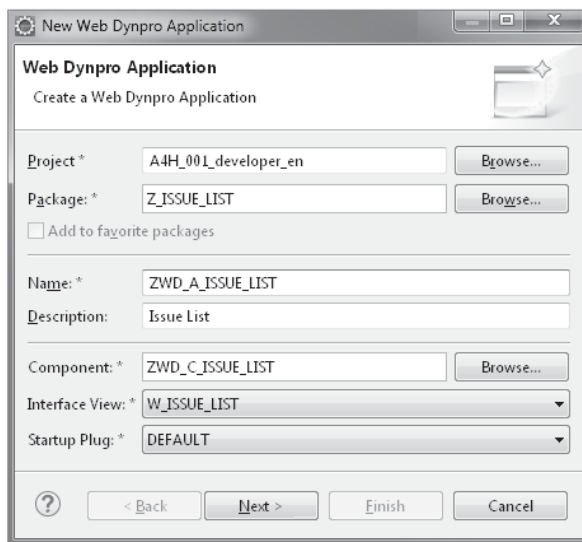


Abbildung 4.28 Neue Web-Dynpro-Applikation erstellen

3. Wählen Sie Ihren Transportauftrag für unsere Beispielanwendung aus, und legen Sie die Anwendung mit FINISH an.
4. Das SAP GUI startet und zeigt Ihre neue Anwendung an. Speichern Sie diese, und schließen Sie die SAP-GUI-Registerkarte.

Die neu angelegte Applikation taucht nun unterhalb der Komponente im Project Explorer auf. Sie können die Anwendung auch direkt über Eclipse

ausführen. Öffnen Sie dazu das Kontextmenü zu Ihrer Applikation im Project Explorer, und wählen Sie RUN AS... • WEB DYNPRO ABAP APPLICATION. Der in Eclipse integrierte Browser startet und zeigt Ihre Anwendung an. Unter Umständen werden Sie vorab noch nach Benutzernamen und Passwort gefragt.

4.4 Floorplan Manager

Der Floorplan Manager ist ein Framework zur Erstellung konfigurierbarer Web-Dynpro-Anwendungen. Über den Floorplan Manager können Sie verschiedene Bausteine in Form sogenannter *Building Blocks* zu einer Anwendung zusammensetzen. Der Floorplan Manager basiert zu einem Großteil auf Web-Dynpro-Konfigurationen.

In der Vergangenheit war es mitunter mühselig, eine neue Anwendung zu erstellen, da diese und eine Reihe von Konfigurationen zunächst angelegt werden mussten. Seit der Einführung des Application Creation Tools (ACT) kann diese Aufgabe automatisiert durchgeführt werden. Das ACT ist auch in Eclipse verfügbar.

Eine Floorplan-Manager-Anwendung legen Sie in Eclipse wie folgt an:

1. Markieren Sie im Project Explorer das Paket Z_ISSUE_LIST, und drücken Sie **Strg** + **N**, um ein neues Objekt zu erstellen. Wählen Sie FLOORPLAN MANAGER APPLICATION aus der Liste, und fahren Sie mit NEXT fort.
2. Aus der Liste der Assistenten wählen Sie WIZARD FOR CREATING EMPTY FPM APPLICATIONS und bestätigen die Auswahl mit FINISH (siehe Abbildung 4.29).

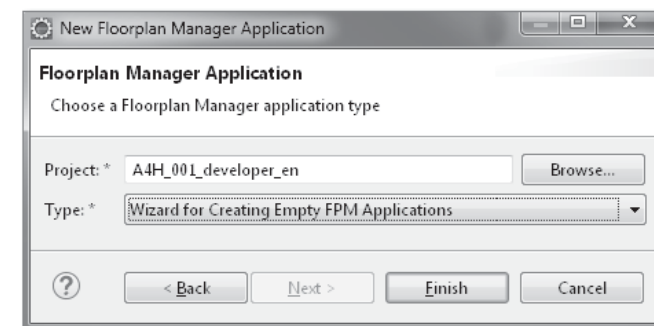


Abbildung 4.29 Neue Floorplan-Manager-Anwendung anlegen

3. Im integrierten Browser öffnet sich das ACT (siehe Abbildung 4.30). Geben Sie als Namen der Web-Dynpro-Applikation »ZWD_FPM_ISSUE_LIST« sowie eine passende Beschreibung ein.

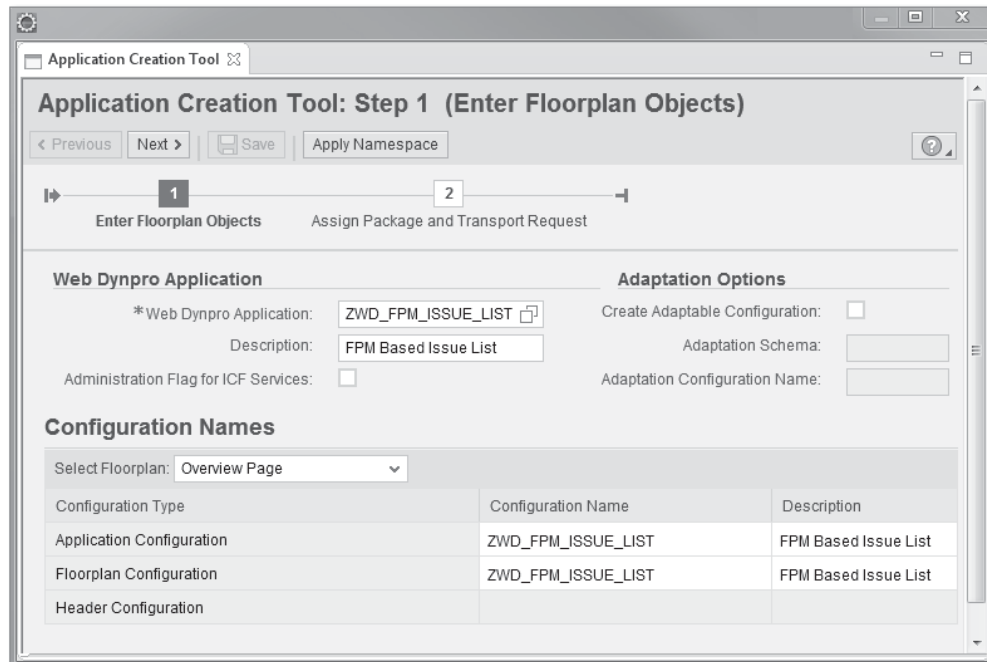


Abbildung 4.30 Application Creation Tool

4. Wählen Sie als Floorplan-Typ, d. h. als Grundriss, der der Anwendung zugrunde liegen soll, unter SELECT FLOORPLAN die Option OVERVIEW PAGE.
5. Das ACT schlägt automatisch Namen für die Konfigurationen vor. Sie können diese Namen einfach übernehmen und mit NEXT zum nächsten Schritt gehen.
6. Wählen Sie als PACKAGE Z_ISSUE_LIST, und tragen Sie Ihren Transportauftrag im Feld REQUEST/TASK ein.
7. Schließen Sie die Anlage mit SAVE ab.

Ihnen werden nun zwei Links angezeigt, einer, um in den Floorplan-Konfigurationseditor (*Flexible UI Designer*, FLUID) abzuspringen, und einer, um die Anwendung auszuführen. Wenn Sie den Project Explorer aktualisieren, werden Ihnen zudem die neu angelegte Anwendung und die zugehörigen Konfigurationen angezeigt (siehe Abbildung 4.31).

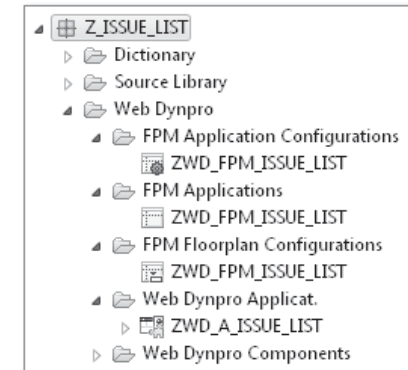


Abbildung 4.31 Floorplan-Manager-Objekte im Project Explorer

Die Floorplan-Manager-Tools sind alle webbasiert. Daher unterscheidet sich die Arbeit mit diesen Tools, ausgehend von Eclipse, nicht von der Arbeit, die von der SAP-GUI-Oberfläche ausgeht. Sie können unsere Aufgabenliste mithilfe dieser Tools nun als Floorplan-Manager-Anwendung zur Verfügung stellen. Die dazu notwendigen Konfigurationen sind nun angelegt und können in Eclipse aufgerufen werden.