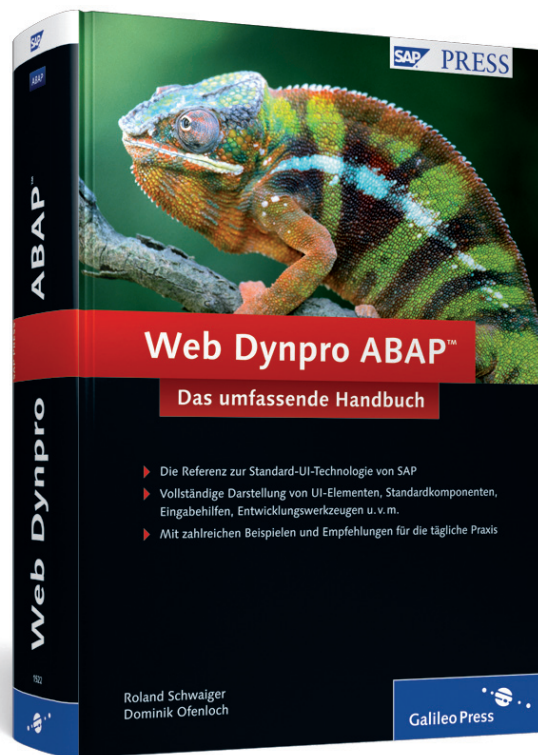


Roland Schwaiger, Dominik Ofenloch

Web Dynpro ABAP™ – Das umfassende Handbuch




Galileo Press

Bonn • Boston

Service-Klassen und Interfaces

In diesem Dokument werden die Service-Klassen und Interfaces für Web Dynpro vorgestellt. Im Unterschied zu den Web-Dynpro-Programmierschnittstellen sind diese ein Angebot für den Anwendungsentwickler zur Erleichterung der täglichen Arbeit.

Klasse CL_WD_CONTEXT_SERVICES

Die Klasse CL_WD_CONTEXT_SERVICES ermöglicht es Entwicklern, Änderungen an einem bestimmten Context-Knoten aufzuzeichnen, um diese zu einem späteren Zeitpunkt auswerten zu können. Im Unterschied zum Context-Change-Log, zeichnet diese Klasse sowohl Benutzereingaben als auch programmatisch durchgeführte Änderungen auf.

Die Verwendung der Klasse lässt sich in zwei Schritten beschreiben:

1. Zuerst wird eine Instanz der Klasse CL_WD_CONTEXT_SERVICES erzeugt, die sich auf die Änderungen eines bestimmten Context-Knotens durch die Methode subscribe_to_node_changes() registriert. Listing 1 gibt Ihnen hierzu ein Beispiel.
2. Anschließend kann die Änderungsliste des Knotens durch die Methode get_context_change_list() abgeholt und zurückgesetzt werden. Dies kann zu einem beliebigen Zeitpunkt erfolgen.

Bei Bedarf ist es darüber hinaus auch möglich, eine Registrierung auf Änderungen durch die Methode deactivate_subscriptions() wieder zu löschen.

```
DATA:
  ls_sub  TYPE cl_wd_context_services=>subscription_element,
  lt_sub  TYPE cl_wd_context_services=>subscription_list,
  lo_view TYPE REF TO if_wd_view_controller.
* Erzeuge eine Instanz von CL_WD_CONTEXT_SERVICES
CREATE OBJECT wd_this->go_ctx_service.
* Registriere alle Änderungen von Context-Knoten T005T
lo_view      = wd_this->wd_get_api( ).
ls_sub-controller ?= lo_api_view.
ls_sub-node_name  = wd_this->wdctx_t005t.
APPEND ls_sub TO lt_sub.
wd_this->go_ctx_service->subscribe_to_node_changes( subscriptions = lt_sub ).
```

Listing 1 Registrierung der Context-Service-Klasse auf Knoten T005T

Tabelle 1 listet die Methoden der Klasse CL_WD_CONTEXT_SERVICES zur Überwachung von Context-Knoten auf.

Methode	Beschreibung
activate_subscriptions()	Aktiviert wieder alle Knotenregistrierungen nach Verwendung von deactivate_subscriptions().
deactivate_subscriptions()	Deaktiviert alle Knotenregistrierungen. Änderungen werden nicht mehr aufgezeichnet.
get_change_list()	Liefert eine Liste CHANGED_ITEMS vom Klassentyp CHANGED_LIST der aufgezeichneten Änderungen zurück. Über den optionalen Parameter RESET_LIST kann die Zurücksetzung der Liste verhindert werden.
reset_change_list()	Setzt die Änderungsliste zurück.

Tabelle 1 Instanzmethoden von CL_WD_CONTEXT_SERVICES

Methode	Beschreibung
subscribe_to_node_changes()	Registriert eine Liste von Context-Knoten, SUBSCRIPTIONS vom Klassentyp SUBSCRIPTION_LIST, zur Überwachung.
unsubscribe_all()	Deregistriert alle registrierten Context-Knoten der Überwachung.
unsubscribe_from_node_changes()	Streicht eine Liste von Context-Knoten, SUBSCRIPTIONS vom Klassentyp SUBSCRIPTION_LIST, aus der Überwachung.

Tabelle 1 Instanzmethoden von CL_WD_CONTEXT_SERVICES (Forts.)

Neben den Instanzmethoden zur Überwachung von Context-Knoten bietet die Klasse CL_WD_CONTEXT_SERVICES noch weitere statische Hilfsmethoden, die in Tabelle 2 aufgelistet sind. Sie dienen hauptsächlich dazu, dynamische Binding-Pfade für dynamisches Mapping in Views anzulegen.

Methode	Beschreibung
get_element_path_for_ctrl()	Liefert den Pfad PATH eines Elementes ELEMENT in einem Controller CONTROLLER als STRING.
get_mapped_node_for_controller()	Liefert den ersten gefundenen gemappten Knoten NEW_NODE aus dem Controller CONTROLLER eines Originalknotens NODE zurück. Falls kein Knoten gefunden wurde, ist NEW_NODE initial.
get_node_metapath_for_ctrl()	Liefert den ersten gefundenen Metapfad PATH zu einem Controller CONTROLLER über einen gemappten Knoten NODE.
get_node_path_for_controller()	Liefert den ersten gefundenen Pfad PATH zu einem Controller CONTROLLER über einen gemappten Knoten NODE.
get_original_node()	Liefert den Originalknoten OUT_NODE eines gemappten Knotens IN_NODE.

Tabelle 2 Statische Methoden von CL_WD_CONTEXT_SERVICES

Klasse CL_WD_DYNAMIC_TOOL

Die Klasse CL_WD_DYNAMIC_TOOL bietet Methoden zur Erleichterung der alltäglichen Arbeit und zur Verwendung in der dynamischen Programmierung. So können Sie zur Laufzeit dynamisch Formulare und Tabellen generieren oder das System überprüfen lassen, ob alle als Pflichtfeld markierten Context-Attribute/UI-Elemente mit Eingaben gefüllt wurden.

Tabelle 3 listet die Methoden der Klasse CL_WD_DYNAMIC_TOOL zur Erweiterung und Manipulation der UI-Elemente-Hierarchie auf.

Methode	Beschreibung
create_form_from_node()	Generiert aus den Attributen eines Knotens ein Formular. Hierzu müssen ein UI-Element-Container UI_PARENT und ein Knoten NODE mitgegeben werden (siehe auch create_table_from_node()).
create_layout_data()	Erzeugt für ein UI-Element UI_ELEMENT ein dazu passendes Layoutobjekt, abhängig vom Typ des Eltern-Containers PARENT_OBJECT. Das UI-Element wird direkt in das Layoutobjekt integriert.
create_table_from_node()	Generiert aus den Attributen eines Knotens eine Tabelle. Hierzu müssen ein UI-Element-Container UI_PARENT, ein Knoten NODE und eine Tabellen-ID TABLE_ID mitgegeben werden. (siehe auch create_form_from_node()).

Tabelle 3 Methoden zur Änderung der UI-Elemente-Hierarchie

Methode	Beschreibung
set_visible_from_element()	Setzt die Sichtbarkeit eines UI-Elementes UIELEMENT_ID abhängig davon, ob im Context des Views VIEW ein als Pfad PATH spezifiziertes Context-Element existiert oder nicht. Optional kann ein zweites UI-Element ALTERNATIVE_UIELEMENT_ID angegeben werden, das stattdessen angezeigt wird.

Tabelle 3 Methoden zur Änderung der UI-Elemente-Hierarchie (Forts.)

Tabelle 4 listet die Methoden von CL_WD_DYNAMIC_TOOL zur Verwendung im Zusammenhang mit Context-Knoten auf. Vor allem die beiden Methoden check_mandatory_attr_on_view() und check_mandatory_attributes() haben eine große Praxisrelevanz und werden deshalb häufig eingesetzt.

Methode	Beschreibung
check_mandatory_attr_on_view()	Überprüft für die UI-Elemente und Context-Attribute eines Views, ob alle als Pflichtfeld markierten und sichtbaren Context-Attribute gefüllt sind. Falls dies nicht der Fall ist, werden für die noch fehlenden Elemente und Eingaben Fehlermeldungen ausgegeben. Als Eingangsparameter ist der View-Controller VIEW_CONTROLLER nötig. Über den Parameter DISPLAY_MESSAGES kann die Meldungsausgabe deaktiviert werden; in diesem Fall kann man die Fehler manuell über die Exporting-Tabelle MESSAGES behandeln.
check_mandatory_attributes()	Verhält sich wie check_mandatory_attr_on_view(). Über ATTRIBUTE_LIST können jedoch gezielt die zu prüfenden Context-Attribute mitgegeben werden.
create_range_table()	Erzeugt dynamisch aus dem mitgegebenen Typ I_TYPENAME eine Range-Tabelle RT_RANGE_TABLE. Diese kann anschließend in Select-Options verwendet werden.
is_structure_or_table()	Liefert zurück, ob ein DDIC-Objekt NAME eine Struktur oder eine Tabelle ist.
set_tree_lead_selection()	Setzt die Lead-Selection innerhalb eines rekursiven Knotenbaums. Falls bereits ein anderes Element innerhalb der Hierarchie ausgewählt ist, wird dieses deselektiert. Diese Methode sollte vor allem vor dem ersten Rendern eines Tree-Elementes aufgerufen werden, um sicherzustellen, dass nur ein Context-Element ausgewählt ist. Über I_WD_TREE muss der Baum mitgegeben werden. Der Parameter I_NEW_LEAD_SELECTION definiert die neue Lead-Selection im Baum.

Tabelle 4 Methoden zur Änderung und Überprüfung des Contexts

Tabelle 5 listet die Methoden der Klasse CL_WD_DYNAMIC_TOOL zur Verwendung im Zusammenspiel mit dem TreeByNestingTableColumn-Element auf. Dieses UI-Element ermöglicht die Anzeige einer Baumstruktur in einer Web-Dynpro-Tabelle. Im Unterschied zum UI-Element Tree ermöglicht die hierarchische Tabelle Benutzereingaben, Scrolling und die Durchsuchbarkeit der Daten.

Methode	Beschreibung
clear_table_tree_selection()	Löscht alle Selektionen innerhalb und unterhalb des Knotens DATA_SOURCE. Standardmäßig wird dabei die Lead-Selection mit gelöscht.
get_table_tree_lead_selection()	Ermittelt die Lead-Selection LEAD_SELECTION in der beim Knoten DATA_SOURCE beginnenden Tabellenwurzel.

Tabelle 5 TreeByNestingTableColumn-Methoden

Methode	Beschreibung
<code>get_table_tree_selection()</code>	Verhält sich wie <code>get_table_tree_lead_selection()</code> ; zusätzlich sind jedoch auch normale Selektionen in der Ergebnismenge <code>SELECTED_ELEMENTS</code> mit einbegriffen.
<code>make_table_element_visible()</code>	Öffnet die Baumstruktur der Tabelle <code>WD_TABLE</code> bis hin zu einem bestimmten Context-Element <code>CONTEXT_ELEMENT</code> .
<code>set_table_tree_lead_selection()</code>	Setzt die Lead-Selection innerhalb der Tabellenhierarchie <code>DATA_SOURCE</code> auf das über <code>LEAD_SELECTION</code> spezifizierte Element.

Tabelle 5 TreeByNestingTableColumn-Methoden (Forts.)

Klasse CL_WD_RUNTIME_SERVICES

Die Klasse `CL_WD_RUNTIME_SERVICES` ermöglicht es über die Methode `attach_file_to_response()`, Dateien ohne Verwendung des UI-Elementes `FileDownload` direkt an den Anzeige-Client zu schicken. Hierzu müssen die Parameter `I_FILENAME`, `I_CONTENT` und `I_MIME_TYPE` mitgegeben werden.

Die Methode `get_url_parameter()` erlaubt das Auslesen von URL-Parametern. Füllen Sie hierzu bei Verwendung den Parameter `NAME`.

Klasse CL_WD_UTILITIES

Tabelle 6 listet die Methoden der Klasse `CL_WD_UTILITIES` auf.

Methode	Beschreibung
<code>construct_wd_url()</code>	Berechnet die URL für eine bestimmte Web-Dynpro-Anwendung <code>APPLICATION_NAME</code> .
<code>get_otr_text_by_alias()</code>	Liefert den OTR-Text (Online Text Repository) für einen bestimmten OTR-Alias <code>ALIAS</code> in der Sprache <code>LANGUAGE</code> zurück.
<code>set_width_4ddlbykey()</code>	Setzt die ideale Breite für ein <code>DropDownByKey</code> -Element <code>DDLBY</code> anhand seiner Listeneinträge.

Tabelle 6 Methoden von Klasse `CL_WD_UTILITIES`

Interface IF_WD_SELECT_OPTIONS

Das Interface `IF_WD_SELECT_OPTIONS` dient zur Konfiguration der Selektionsoptionen-Component `WDR_SELECT_OPTIONS`. Die ausführliche Behandlung dieser Component im Zusammenspiel mit dem Interface `IF_WD_SELECT_OPTIONS` finden Sie in in »Web Dynpro ABAP– Das umfassende Handbuch« (ISBN 978-3-8362-1522-0) Abschnitt 8.3.2, »Web-Dynpro-Component `WDR_SELECT_OPTIONS`«.

Klasse CL_WD_WEB_ICON

Eine Vielzahl von UI-Elementen, wie beispielsweise der `Button`, unterstützt die Einbindung sogenannter Icons. Hierbei handelt es sich um nichts anderes als kleine in die Anwendung einbindbare Symbole, wie zum Beispiel eine Diskette (Speichern). Die Klasse `CL_WD_WEB_ICON` ermöglicht die dynamische Verwendung dieser Icons zur Laufzeit.

Tabelle 7 listet die Methoden von Klasse `CL_WB_WEB_ICON` auf.

Methode	Beschreibung
<code>get_collections()</code>	Liefert eine Übersicht aller vorhandenen Icon-Sammlungen als Tabelle zurück. Zurzeit existieren Icon, IconLage und Pictogram.
<code>get_web_icon_url()</code>	Gibt die URL eines Icons <code>WEB_ICON_ALIAS</code> passend für eine Anwendung <code>WD_APPLICATION</code> zurück.
<code>get_web_icons()</code>	Liefert eine Liste aller Icons einer Sammlung <code>COLLECTION_NAME</code> zurück. Wird der Parameter leer gelassen, werden die Icons aller Sammlungen zurückgegeben.

Tabelle 7 Methoden von Klasse `CL_WD_WEB_ICON`

Klasse `CL_WEB_DYNPRO`

Die Klasse `CL_WEB_DYNPRO` gehört zu den kleinsten Klassen des Web-Dynpro-Frameworks. Über das statische Klassenattribut `IS_ACTIVE` können beliebige Aufrufer abfragen, ob sie innerhalb des Web-Dynpro-Umfeldes läuft, das heißt:

```
IF cl_web_dynpro=>is_active
  EQ abap_true.
  ...
ELSE.
  ...
ENDIF.
```

Interface `IF_WD_ACFUPDOWNLOAD_SWFS`

Das Interface `IF_WD_ACFUPDOWNLOAD_SWFS` stellt die Programmierschnittstelle für die Öffnung eines Datei-Upload-Dialogs des UI-Elementes `AcfUpDownload` dar. Das Interface stellt zum Öffnen eines Upload-Dialogfensters die Methode `choose_file()` bereit.

Die Verwendung des Interface für den Upload einer Datei gliedert sich in die folgenden Schritte:

1. Fügen Sie das UI-Element `AcfUpDownload` in die UI-Elemente-Hierarchie Ihres Views ein.
2. Speichern Sie in der Methode `wddommodifyview()` eine Instanz des UI-Elementes `AcfUpDownload` im lokalen View-Controller. Führen Sie hierbei einen Cast auf den Typ `IF_WD_ACFUPDOWNLOAD_SWFS` durch. Listing 2 zeigt Ihnen ein Beispiel.

```
METHOD wddommodifyview.
  IF first_time = 'X'.
    wd_this->acf_method_handler
      ?= view->get_element( '<ACF-UI-Element-Name>' ).
  ENDIF.
ENDMETHOD.
```

Listing 2 UI-Element »AcfUpDownload« im lokalen Controller sichern

3. Rufen Sie anschließend die Methode `choose_file()` des Interface auf. Die Methode verfügt über die in Tabelle 8 aufgelisteten Parameter.

Parameter	Beschreibung
<code>WINDOW_TITLE</code>	Titel für das Dateiauswahlfenster. Wird als Titel <code>@acf_null</code> eingesetzt, wenn der für die aktuelle Sprache eingestellte Standardtitel verwendet wird.
<code>START_DIRECTORY (optional)</code>	Startverzeichnis für den Dateiauswahl-Dialog

Tabelle 8 Parameter der Methode »choose_file()«

Parameter	Beschreibung
DEFAULT_FILE (optional)	standardmäßig voreingestellte Datei, die geöffnet werden soll
FILTER (optional)	Filterung von Dateien bestimmter Dateitypen. Durch die Eingabe von *.csv, *.txt werden nur CSV- und TXT-Dateien angezeigt.
FILTER_DESCRIPTION (optional)	Beschreibung der Filterwerte, zum Beispiel »Nur CSV- und TXT-Dateien auswählbar«
CONTEXT_ATTRIBUTE	Struktur vom Typ WDR_EXT_ATTRIBUTE_POINTER. Die Struktur zeigt auf das Datei-Ziel-Context-Attribut.
DIRECTORY_ONLY (optional)	Hat dieser Parameter den Wert X, können im Dateiauswahl-Dialog nur Verzeichnisse ausgewählt werden.

Tabelle 8 Parameter der Methode »choose_file()« (Forts.)