

Mit den in diesen vier Abschnitten gewonnenen Kenntnissen und Erfahrungen wird es für Sie künftig kein Problem darstellen, weitere vorhandene Komponenten (zum Beispiel die bereits erwähnte Top-level- oder Detailnavigation) anzupassen oder neue, eigene Komponenten zu erstellen.

Light Framework Da wir ein Light Framework einrichten werden und unsere Anpassungen daher auch für dieses durchgeführt werden sollen, werden wir uns im Rahmen der Anpassung des Portalkopfes ausschließlich mit der Komponente des Light Frameworks befassen. Der Aufbau der Komponenten des Standard-Frameworks und somit auch deren Angleichung sind in der Regel erheblich komplexer als die des Light Frameworks (keine Tag Library, Navigation wird zum Teil erst im Browser per JavaScript erzeugt etc.) und würden im Rahmen dieses Buches zu weit führen. Zudem ist für ein extern zugängliches Portal in der Regel die Verwendung des Light Frameworks sinnvoller, da es schlanker ist und so dem Benutzer geringere Ladezeiten und damit eine bessere Performance bietet – hier haben wir uns daher dem Praxischarakter dieses Buches verschrieben.

10.2 Grundlagen

Bevor wir mit dem praktischen Teil dieses Kapitels beginnen, wollen wir kurz einige grundlegende Themen zur Entwicklung von Portalanwendungen behandeln. Wir wollen hier allerdings nicht zu sehr ins Detail gehen, sondern lediglich einen Überblick bieten, denn der Fokus dieses Buches ist schließlich die Praxis.

10.2.1 Was ist eine Portalanwendung?

Eine Portalanwendung ist vereinfacht ausgedrückt eine Sammlung von Komponenten und/oder Services, die normalerweise einem bestimmten Themenbereich zugeordnet sind. Dies können technische Themen sein, wie zum Beispiel grundlegende Bestandteile des Portals selbst, fachliche, wie die Darstellung von Berichten oder anderen Daten aus Backend-Systemen (SAP oder andere), oder auch inhaltliche, wie zum Beispiel das Knowledge Management Repository (KM) des Portals.

Komponenten Komponenten sind die Teile einer Portalanwendung, die die für einen Benutzer sichtbaren Bestandteile einer Anwendung im Portal

bereitstellen. Komponenten sind die Grundlage für iViews, die im Portal Content angelegt und konfiguriert werden können. Sie erzeugen den HTML-Code, der dem Benutzer im Browser angezeigt wird, und implementieren die Programmlogik, die zur Bedienung der Anwendung nötig ist.

Technisch gesehen, ist eine Komponente eine Java-Klasse, die die Klasse `AbstractPortalComponent` der Portal-API erweitert. In der Regel gehören noch weitere Bestandteile (HTML- oder JSP-Seiten, Stylesheets und JavaScript-Dateien, Grafiken etc.) zu einer Komponente.

Services sind Programmteile, die lediglich Funktionen zur Verfügung stellen, die von Komponenten oder wiederum anderen Services verwendet werden können. Der XML Transformation Service des Portals, der diverse Funktionen und Schnittstellen zur Umwandlung von XML-Daten bietet, ist ein Beispiel für solch einen Service. Portal-Services erlangen gerade im Rahmen von serviceorientierten Architekturen (SOA) besondere Bedeutung, da hierdurch die Programmlogik zur Bereitstellung von bestimmten Daten nicht in der Anwendung selbst implementiert (und damit fest an diese gebunden), sondern als Schnittstelle zur Verfügung gestellt wird, die dann von mehreren Anwendungen gleichermaßen verwendet werden kann.

Services

Aus technischer Sicht ist ein Service eine Java-Klasse, die das Interface `IService` der Portal-API implementiert. Services werden wir im Rahmen dieses Buches allerdings nicht behandeln.

10.2.2 SAP NetWeaver Developer Studio

Alle Bestandteile einer Portalanwendung können mit dem SAP NetWeaver Developer Studio (NWDS), der Entwicklungsumgebung für Portalanwendungen, die auf der weit verbreiteten Eclipse-Plattform basiert, entwickelt werden. Die entsprechende Perspective hierfür heißt Enterprise Portal Perspective. Die folgenden Aufgaben können Sie mit der Enterprise Portal Perspective ausführen:

- ▶ Erstellen von Portalanwendungen
- ▶ Hinzufügen von Komponenten und Services zu Portalanwendungen
- ▶ Bearbeiten des Quellcodes und Kompilieren von Portalanwendungen
- ▶ Hinzufügen von weiteren Ressourcen zu Portalanwendungen

Enterprise Portal Perspective

- ▶ Importieren und Exportieren einer Portalanwendung als PAR-Datei
- ▶ Deployen einer PAR-Datei in ein Portal

10.2.3 JavaServer Pages im Portal

Die Laufzeitumgebung des Portals ermöglicht es, JSP-Seiten (JavaServer Pages) innerhalb von Portalkomponenten zu verwenden, aber auch Portalkomponenten selbst direkt als JSP-Seiten (anstelle von Java-Klassen) zu entwickeln. Da das Portal jedoch seine eigene JSP-Implementierung besitzt, kann es vorkommen, dass unabhängig vom Portal erstellte JSP-Anwendungen nicht ohne Weiteres im Portal funktionieren. Wenn Sie also JSP-Anwendungen, die ursprünglich nicht für das Portal entwickelt wurden, zu Portalanwendungen konvertieren wollen, ist es möglich, dass Sie noch Anpassungen vornehmen müssen. Dies sei der Vollständigkeit halber erwähnt.

Java-Klassen aus
JSP-Seiten

Beim ersten Zugriff auf eine als JSP-Seite implementierte Komponente erzeugt das Portal aus dieser eine Java-Klasse, die die Klasse `AbstractPortalComponent` erweitert (siehe Abschnitt 10.2.1, »Was ist eine Portalanwendung?«). Diese wird anschließend zu einer `.class`-Datei kompiliert. Beim Zugriff auf die JSP-Seite ist es dann diese Klasse, die ausgeführt wird, und nicht die originale JSP-Seite selbst.

Durch diesen Prozess kann der erste Aufruf einer JSP-Seite nach deren Deployment mehr Zeit beanspruchen als die darauffolgenden, abhängig von der Komplexität der Seite. Daher ist es auch erforderlich, bei Änderungen an JSP-Seiten die Komponenten neu zu deployen, denn nur dann wird anschließend dieser Umwandlungsprozess für die JSP-Seite erneut ausgeführt. Ein Austauschen nur der JSP-Seite direkt im Dateisystem ist hier wirkungslos.

Entwicklungsmodus

Letzteres gilt nicht, wenn das Portal im Entwicklungsmodus betrieben wird, denn in diesem Fall prüft die Portal-Laufzeitumgebung bei jedem Zugriff auf eine JSP-Seite, ob diese seit dem letzten Kompilieren geändert wurde. Ist dies der Fall, wird der beschriebene Prozess für diese Seite erneut durchlaufen.

JSP-Seite als
Komponente

Eine JSP-Seite als Komponente zu definieren, ist zwar denkbar einfach, bietet sich aber nur für solche Komponenten an, die keine kom-

plexe Programmlogik beinhalten. Der JSP-Editor des SAP NetWeaver Developer Studios bietet nicht die gleichen komfortablen Entwicklungshilfen wie der Java Editor, sodass oft erst nach dem Deployment grundlegende Fehler auffallen, die im Java Editor bereits während der Entwicklung als solche erkannt und entsprechend markiert werden.

Zudem widerspricht die Entwicklung komplexer Anwendungen in Form von JSP-Seiten dem MVC-Prinzip (Model-View-Controller), da JSP-Seiten in der Regel nur für die Präsentationsschicht verwendet werden (View) und nicht die eigentliche Programmlogik (Controller) enthalten.

Um nun eine Komponente in Form einer JSP-Seite zu definieren, muss die jeweilige JSP-Datei erstellt und in der *portalapp.xml* als Komponente definiert werden. Wir werden dies in Abschnitt 10.4.1, »Erstellen eines neuen Projektes«, im Detail zeigen. Der übliche Ordner für JSP-Seiten, die als Komponenten verwendet werden, ist *PORTAL-INF/pagelet*.

Sie können JSP-Seiten jedoch auch als Ressourcen in anderen Komponenten verwenden, das heißt, nicht als eigenständige Komponenten. Dies bietet sich an, wenn Ihre Komponente viel Programmlogik enthält, die als Java-Code implementiert werden soll. Aufgrund der erwähnten Einschränkungen des JSP-Editors sollten Sie sich in einem solchen Fall darauf beschränken, JSP-Seiten lediglich für die formatierte Ausgabe von Daten zu verwenden und den Java-Code der Programmlogik in »echten« Java-Klassen implementieren. Auch diese Möglichkeit werden wir in Abschnitt 10.4.1, »Erstellen eines neuen Projektes«, näher erläutern. Der übliche Ordner für JSP-Seiten, die als Ressourcen eingebunden werden, ist *PORTAL-INF/jsp*.

JSP-Seite als
Ressource

10.2.4 Tag Librarys

Das Portal stellt Entwicklern zwei verschiedene Tag Librarys zur Verfügung, die die Entwicklung von Navigations-iViews und von iViews für den Portalkopf oder den Seitentitel erheblich vereinfachen. Außerdem wird der Code durch die Verwendung der Tag Librarys deutlich wartungsfreundlicher, da er besser lesbar und weniger fehleranfällig ist.

- Tag Librarys** Tag Librarys sind – wie der Name schon sagt – Bibliotheken mit Tags, die bestimmte Funktionen zur Verfügung stellen. Sie kommen ausschließlich in JSP-Seiten zum Einsatz. Der wesentliche Vorteil dieser Tag Librarys bzw. der damit zur Verfügung gestellten Tags ist, dass Sie im Code der JSP-Seite nicht ständig zwischen tagbasierter Notation für den zu erzeugenden HTML-Code einerseits und nativem Java-Code andererseits wechseln müssen. Denn das macht JSP-Seiten in der Regel ab einer gewissen Größe und Komplexität unübersichtlich und damit fehleranfällig und schwierig zu warten.
- Vorteile** Mithilfe der Tag Librarys steht eine Reihe von Java-Funktionen in Form von Tags zur Verfügung, die Sie genauso wie HTML-Tags im Quellcode verwenden können. Dadurch erhalten Sie eine durchgängig tagbasierte Notation, die die erwähnten Probleme stark reduziert oder sogar vollständig eliminiert. In Abschnitt 10.4 werden wir noch einmal detailliert die Verwendung der Tag Librarys beschreiben.

10.2.5 Mehrsprachigkeit

Das Thema Mehrsprachigkeit/Internationalisierung bei der Entwicklung von Portalkomponenten werden wir in diesem Kapitel nicht ausführlich besprechen, da es zu weit führen würde. Wir werden nur im Zuge der Anpassung der Login-Seite darauf eingehen, wie es bei dieser Komponente gelöst ist, damit Sie hier entsprechende Anpassungen durchführen können. Ausführliche Informationen hierzu finden Sie – wie auch zu allen anderen Themen – im SAP Help Portal unter <http://help.sap.com>. Das SAP-Heft *Globale Portalsysteme mit SAP* (<http://www.sap-hefte.de/1686>) beschäftigt sich ebenfalls mit diesem Thema und führt Sie im Rahmen eines Unternehmensszenarios praxisnah durch alle Aspekte von globalen Portalsystemen.

10.3 Anpassen der Login-Seite

Für unser Praxisbeispiel wollen wir auf der Login-Seite einen zusätzlichen Text anzeigen, der die Nutzungsbedingungen für externe Benutzer enthält. Diese Anforderung erfordert eine Anpassung der Komponente, denn ein zusätzliches Textfeld oder zusätzliche Links zu anderen Webseiten oder Dokumenten sind dort im Standard nicht vorgesehen. Im Gegensatz dazu können die Bilder im rechten Teil der Login-Seite jedoch recht einfach ohne Program-

mierung über Einträge in den folgenden UME-Parametern angepasst werden:

- ▶ `ume.logon.branding_image`
- ▶ `ume.logon.branding_text`

Hinweis

Die Anpassung der UME-Parameter ist – wie bereits in vorhergehenden Kapiteln beschrieben – im Gegensatz zum vorherigen Release des Portals nicht mehr im Portal selbst möglich, sondern muss über das *Config Tool* erfolgen.

UME-Parameter anpassen

Für Änderungen der Standardtexte wiederum müssen Dateien einer Bibliothek der Komponente angepasst werden, was ebenfalls eine Veränderung der Komponente darstellt.

Wir werden zunächst die PAR-Datei der Login-Seite aus dem Portal herunterladen, im SAP NetWeaver Developer Studio importieren und anschließend als eigene Komponente wieder in das Portal deployen. Wir werden in diesem Schritt zunächst nur eine kleine Anpassung an der Seite vornehmen, um den Prozess des Imports und Deployments durchgehend beschreiben zu können. Anschließend zeigen wir, wie Sie diese neue Login-Seite aktivieren, um sie verwenden zu können. Funktioniert dies, können wir uns auf die eigentliche Anpassung konzentrieren.

10.3.1 PAR-Dateien der Portalkomponenten

Um PAR-Dateien aus dem Portal herunterzuladen und zu deployen, benötigen Sie einen Portalbenutzer mit Administratorrechten. Es gibt auch die Möglichkeit, diese Dateien direkt aus dem Dateisystem des Servers zu kopieren, darauf möchten wir aber aus praktischen Gründen hier nicht weiter eingehen. In den meisten Fällen wird der Entwickler eher keinen solchen direkten Zugriff haben und ist daher auf die hier beschriebene Methode angewiesen.

Die Möglichkeit, PAR-Dateien aus dem Portal zu laden, finden Sie im Support-Bereich des Portals unter SYSTEMADMINISTRATION • SUPPORT. Dort folgen Sie auf der Seite den Links PORTAL RUNTIME • BROWSE DEPLOYMENT. Anschließend navigieren Sie durch einen Klick auf den Link ROOT im oberen Bereich zunächst in das Wurzelverzeichnis und dann durch Klicken auf BROWSE (in der Spalte ACTION) der jewei-

PAR-Dateien im Portal

ligen Ordner bis in den Ordner *ROOT/WEB-INF/deployment/temp* (siehe Abbildung 10.2).



Abbildung 10.2 Speicherort der PAR-Dateien im Portal

In diesem Ordner liegen die originalen PAR-Dateien der Portalkomponenten, die hier allerdings die zusätzliche Dateieindung *.bak* tragen (siehe Abbildung 10.3). Der Ordner heißt *temp*, da bei der Installation des Portals die Originalkomponenten zunächst hier hinein kopiert und dann in ihre Laufzeitverzeichnisse entpackt wurden. Es handelt sich nicht um ein temporäres Verzeichnis, wie Sie es zum Beispiel von Ihrem Betriebssystem kennen, dessen Inhalt regelmäßig automatisch gelöscht wird. Hier werden Sie auch Ihre eigenen PAR-Dateien wiederfinden, nachdem Sie sie deployt haben.

com.sap.ip.bi.export.xfa.par.bak	view , download	342576	Thu Jul 19 12:52:09 CEST 2007
com.sap.portal.wsrp.cspextension.par.bak	view , download	25744	Thu Jul 19 11:13:02 CEST 2007
com.sap.portal.runtime.logon.par.bak	view , download	290774	Thu Jul 19 11:12:50 CEST 2007
com.sap.portal.pcmbuilderservice.par.bak	view , download	62302	Thu Jul 19 11:12:50 CEST 2007

Abbildung 10.3 Liste der PAR-Dateien (Ausschnitt)

PAR-Datei der Login-Seite

Der Name der PAR-Datei, die die Login-Seite beinhaltet, ist *com.sap.portal.runtime.logon.par.bak*. Suchen Sie diese Datei in der Liste, und laden Sie die Datei nach dem im Folgenden beschriebenen Vorgehen herunter. Das Vorgehen zum Download einer PAR-Datei ist in den Browsern Internet Explorer und Mozilla Firefox unterschiedlich, da die Browser die Daten beim Download verschiedenartig behandeln. Wir beschreiben im Folgenden die Prozedur für beide Browser.

Download mit Internet Explorer

Wenn Sie den Internet Explorer benutzen, gehen Sie wie folgt vor, um die PAR-Datei herunterzuladen:

1. Markieren Sie zunächst den Namen der herunterzuladenden PAR-Datei in der Liste mit der Maus, und kopieren Sie ihn in die Zwischenablage (mit `Strg` + `C`) oder über das Kontextmenü).

Dieser Schritt ist sinnvoll, da im Internet Explorer beim Download nicht der Dateiname der gewählten Datei zur Verfügung gestellt

wird, sondern stets *com.sap.portal.support.browse.default*, und Sie den korrekten Dateinamen zum Speichern anderenfalls selbst eingetippen müssen.

2. Klicken Sie auf den Link VIEW in der Zeile der herunterzuladenden PAR-Datei (siehe Abbildung 10.3), und wählen Sie im sich öffnenden Dialog SPEICHERN.

View statt Download?

Bei Nutzung des Internet Explorers 6 liefert ein Klick auf DOWNLOAD erfahrungsgemäß häufig unbrauchbare Dateien. Wir empfehlen daher, stattdessen den Link VIEW zu verwenden.

3. Wählen Sie im folgenden Dialog zunächst einen Ordner aus, in dem Sie die Datei speichern möchten.
4. Fügen Sie nun den Dateinamen aus der Zwischenablage in das Eingabefeld für den Namen der Zielformatdatei ein, und entfernen Sie die Endung *.bak*, sodass die zu speichernde Datei anschließend nur noch die Endung *.par* besitzt (siehe Abbildung 10.4).

The image shows a file save dialog box with two input fields. The first field, labeled 'Dateiname:', contains the text 'com.sap.portal.runtime.logon.par'. The second field, labeled 'Dateityp:', contains the text 'Alle Dateien'. Both fields have a small downward-pointing arrow on the right side, indicating they are dropdown menus.

Abbildung 10.4 Speichern der PAR-Datei mit korrektem Namen (Beispiel)

5. Klicken Sie auf SPEICHERN.

Wenn Sie den Mozilla Firefox benutzen, gehen Sie wie folgt vor:

Download mit
Mozilla Firefox

1. Klicken Sie auf den Link DOWNLOAD in der Zeile der herunterzuladenden PAR-Datei. Der Standarddialog öffnet sich für den Datei-Download mit dem Dateinamen *com.sap.portal.support.browse.default* und dem Dateityp ZIP DATEI.
2. Klicken Sie auf ÖFFNEN. Die Datei wird mit Ihrem Standardprogramm für ZIP-Dateien geöffnet und enthält die zum Download ausgewählte Datei (in diesem Fall *com.sap.portal.runtime.logon.par.bak*).

Standardprogramm für ZIP-Dateien

Sollte der Download-Dialog nicht Ihr ZIP-Programm zum Öffnen zur Auswahl anbieten, müssen Sie dieses hier eventuell selbst auswählen, um die Datei zu öffnen.

3. Entpacken Sie diese Datei in einen Ordner Ihrer Wahl.
4. Benennen Sie die Datei um, indem Sie die Endung *.bak* entfernen.

Sie haben nun die PAR-Datei heruntergeladen und können sie im Folgenden im SAP NetWeaver Developer Studio importieren.

10.3.2 Import ins SAP NetWeaver Developer Studio

Starten Sie nun – falls noch nicht geschehen – das SAP NetWeaver Developer Studio, und wechseln Sie über OPEN PERSPECTIVE • ENTERPRISE PORTAL in die ENTERPRISE PORTAL PERSPECTIVE (siehe Abbildung 10.5).



Abbildung 10.5 Enterprise Portal Perspective

Schritte zum Import

1. Wählen Sie im Menü FILE • IMPORT.
2. Im folgenden Dialog wählen Sie PAR FILE aus der Liste aus und klicken auf NEXT.
3. Klicken Sie auf den Button rechts neben dem Feld ARCHIVE FILE NAME, und suchen Sie die soeben gespeicherte PAR-Datei für den Import aus (siehe Abbildung 10.6).

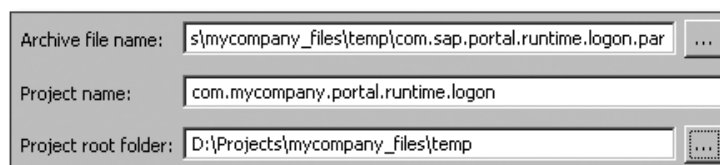


Abbildung 10.6 Einstellungen für das neue Projekt

4. Als PROJECT NAME geben Sie den Namen an, den die neue Portal-komponente erhalten soll. Dies wird auch der Name der neuen PAR-Datei werden, wenn Sie die Komponente deployen.

Selektieren Sie hier für die neue Komponente den Namen *com.mycompany.portal.runtime.logon* gemäß üblicher Namenskonvention.

5. Wählen Sie bei PROJECT ROOT FOLDER ein Zielverzeichnis für Ihre Komponente aus.
6. Klicken Sie auf FINISH.

Neues Projekt

Das SAP NetWeaver Developer Studio erstellt im gewählten Zielverzeichnis einen Ordner mit dem eingegebenen Projektnamen und entpackt dorthin den Inhalt der PAR-Datei zur Bearbeitung. Das neu erstellte Projekt wird nach dem Anlegen im PACKAGE EXPLORER automatisch geöffnet und ausgewählt.

In Abbildung 10.7 sehen Sie die Ordnerstruktur, die Sie nun auch selbst in Ihrem neuen Projekt vorfinden sollten. Den Ordner *PORTAL-INF* haben wir dort schon einmal geöffnet, da in diesem die Dateien liegen, die wir anpassen werden. Auf die Struktur bzw. den Aufbau einer Portalkomponente werden wir in Abschnitt 10.3.7, »Genereller Aufbau einer Portalkomponente«, noch genauer eingehen.

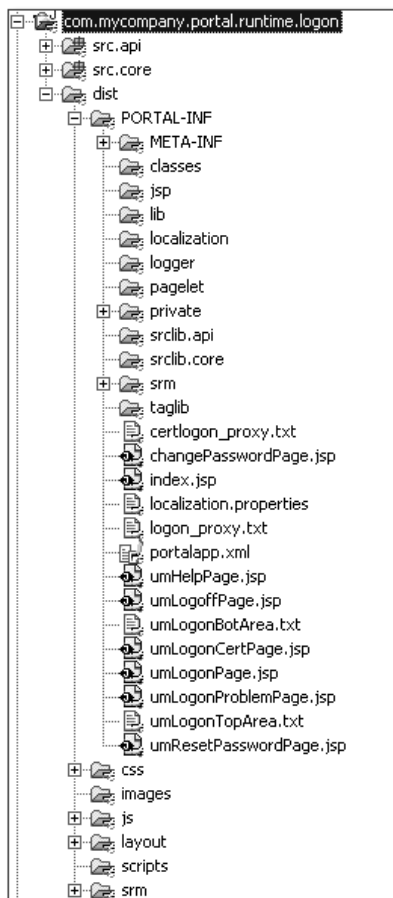


Abbildung 10.7 Ordnerstruktur des Projektes nach Import der PAR-Datei

Bei Ihnen werden zwischen den Ordnern *src.core* und *dist* noch weitere Elemente zu sehen sein, die wir hier allerdings für eine bessere Übersichtlichkeit ausgeblendet haben. Es handelt sich dabei um referenzierte Bibliotheken, die zum Kompilieren des Projektes erforderlich, jedoch selbst nicht Bestandteil des Projektes sind.

Anzeige filtern Sie können diese Elemente ebenfalls ausblenden, um eine übersichtlichere Anzeige im PACKAGE EXPLORER zu erhalten. Klicken Sie dazu auf das nach unten weisende Dreieck in der Icon-Leiste von PACKAGE EXPLORER, und wählen Sie im sich öffnenden Menü den Punkt FILTERS... (siehe Abbildung 10.8).

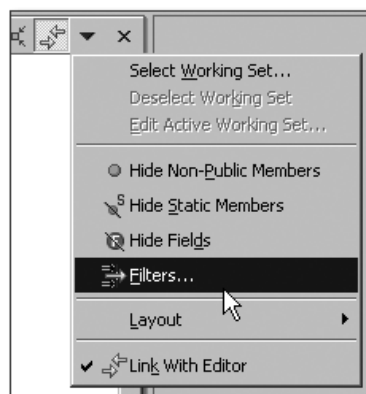


Abbildung 10.8 Anzeige im Package Explorer filtern

Im folgenden Dialog finden Sie in der Liste an letzter Stelle die Option REFERENCED LIBRARIES. Wählen Sie diese Option aus, und schließen Sie den Dialog durch einen Klick auf OK. Auf die übrigen Filtermöglichkeiten wollen wir an dieser Stelle nicht eingehen.

Sollten Sie eine andere Ordnerstruktur als in Abbildung 10.7 vorfinden, so überprüfen Sie, ob Sie wirklich die korrekte Datei aus dem Portal heruntergeladen und importiert haben.

Achtung: Fehlende Bibliotheken importieren

Wir haben nun zwar die PAR-Datei importiert, jedoch sind nicht alle Inhalte auch tatsächlich importiert worden. Beim Import gibt es allerdings keinen Hinweis auf diese Tatsache. Dies ist eine häufige Fehlerquelle bei der Anpassung von PAR-Dateien, daher ist es wichtig, dass Sie diese Problematik kennen und wissen, wie Sie damit umgehen müssen.

Im Ordner *dist/PORTAL-INF/lib* sollten eigentlich zwei Java-Bibliotheken (Dateiendung *.jar*) vorhanden sein, die uns die Importfunktion allerdings aus unbekanntem Gründen vorenthalten hat. Sie sollten daher nach jedem Import prüfen, ob dieser auch vollständig war.

Bei den fehlenden Dateien handelt es sich – im Gegensatz zu den soeben ausgeblendeten – um Bibliotheken, die tatsächlich Bestandteil des Projektes sein müssen, damit die Komponente nach dem Deployment im Portal auch funktioniert. Sie müssen nun diese fehlenden Dateien nachträglich manuell in den richtigen Ordner importieren. Würden Sie die Komponente ohne diese Dateien wieder deployen, würde beim Aufruf der Komponente im Portal ein Laufzeitfehler erzeugt.

Funktion der
fehlenden Dateien

Es gibt generell mehrere Wege, diese Dateien dem Projekt wieder hinzuzufügen. Wir haben den im Folgenden beschriebenen Weg gewählt, um Sie für künftige Importe schon einmal mit dem Importdialog vertraut zu machen.

Damit Sie die fehlenden Dateien importieren können, müssen Sie sie zunächst aus der originalen PAR-Datei entpacken. Dazu öffnen Sie die PAR-Datei mit WinZip und entpacken den Inhalt in einen Ordner Ihrer Wahl, der jedoch kein Unterordner des aktuellen Developer-Studio-Projektes sein sollte. Im SAP NetWeaver Developer Studio gehen Sie folgendermaßen vor, um die Dateien zu importieren:

Fehlende Dateien
finden

1. Klicken Sie mit der rechten Maustaste auf den Ordner *PORTAL-INF/lib*, und suchen Sie im Kontextmenü die Option **IMPORT** aus.
2. Im folgenden Dialog wählen Sie den Eintrag **FILE SYSTEM** und klicken auf **NEXT**. Sie befinden sich nun im Importdialog für Dateien aus dem Dateisystem.
3. Klicken Sie oben im Dialog auf den Button **BROWSE**, rechts von dem Feld für den Quellordner **FROM DIRECTORY**.
4. Wählen Sie den Ordner aus, in den Sie soeben die Inhalte der PAR-Datei entpackt haben, selektieren Sie darin den Unterordner *PORTAL-INF/lib* (siehe Abbildung 10.9), und bestätigen Sie dies durch einen Klick auf **OK**.

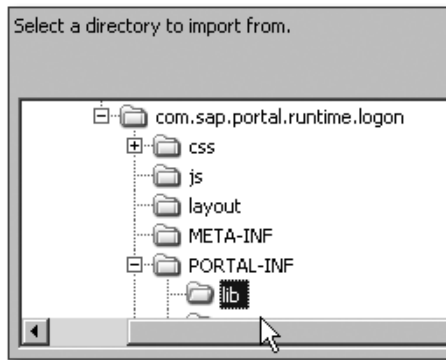


Abbildung 10.9 Ordner mit fehlenden Dateien

Auswahl der
Dateien für den
Import

Sie sehen nun im Importdialog auf der linken Seite den zum Import gewählten Quellordner und auf der rechten Seite die darin enthaltenen Dateien (siehe Abbildung 10.10). In unserem konkreten Fall sollten dies die beiden folgenden Dateien sein:

- ▶ *com.sap.portal.runtime.logon_api.jar*
- ▶ *umelogonbase.jar*

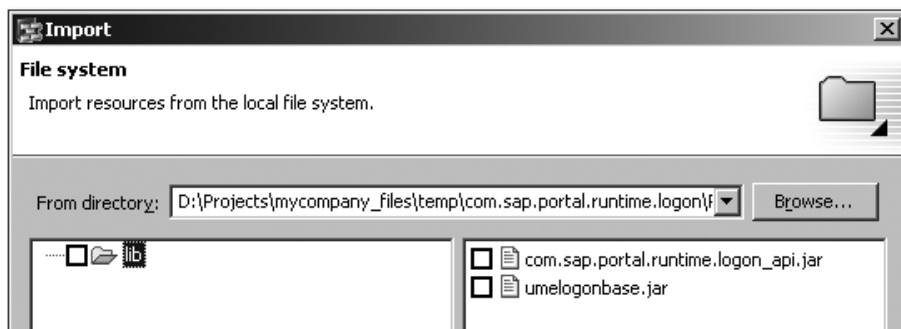


Abbildung 10.10 Zu importierende Dateien

5. Wählen Sie beide Dateien aus, indem Sie die vor den Dateien befindlichen Kästchen anklicken.

Importoptionen 6. Im unteren Teil des Dialogs (siehe Abbildung 10.11) sehen Sie nochmals den Ordner, in den die Dateien importiert werden. Da wir den Dialog durch Rechtsklick auf den entsprechenden Ordner geöffnet haben, ist dieser auch bereits dort ausgewählt. Bei den Optionen darunter sollte nur die Einstellung CREATE SELECTED FOLDERS ONLY ausgewählt sein, da wir lediglich einzelne Dateien aus einem Ordner und keine komplette Ordnerstruktur importieren wollen. Das Überschreiben eventuell bereits vorhandener Dateien ohne entsprechende Warnung (OVERWRITE EXISTING RESOURCES WITHOUT WARNING) sollten Sie nur dann aktivieren, wenn Sie

sicher sind, dass eventuell bereits vorhandene Dateien durch gleichnamige Dateien des Imports überschrieben werden dürfen.

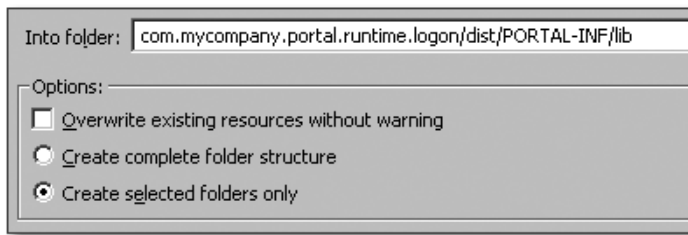


Abbildung 10.11 Importeinstellungen für die ausgewählten Dateien

7. Klicken Sie nun auf **FINISH**, und die beiden gewählten Dateien werden in das Projekt importiert und im Ordner *lib* angezeigt (siehe Abbildung 10.12). Jetzt erst ist das Projekt tatsächlich vollständig und kann weiterbearbeitet und anschließend deployt werden.



Abbildung 10.12 Beide Bibliotheken im Projektordner

Es gibt noch einen zweiten Ordner, der Bibliotheken enthalten kann, die beim Import einer PAR-Datei nicht mit importiert werden. Dies ist der Ordner *PORTAL-INF/private/lib*. Um das soeben beschriebene Vorgehen zu verinnerlichen, prüfen Sie auch diesen Ordner auf fehlende Bibliotheken. Hier noch einmal die Schritte in Kurzform:

Weiteren Ordner
überprüfen

1. Rechtsklick auf den Ordner *PORTAL-INF/private/lib* und **IMPORT** wählen
2. Auswählen der Option **FILESYSTEM** und Klick auf **NEXT**.
3. Klick auf **BROWSE** und Auswahl des Ordners der entpackten originalen PAR-Datei
4. Wie Sie sehen werden, existiert hier gar kein Ordner *PORTAL-INF/private/lib*, daher enthielt die originale PAR-Datei auch keine Bibliotheken in diesem Ordner.
5. Falls der Ordner Dateien enthalten hätte, hätten Sie diese durch Auswahl und anschließenden Klick auf **FINISH** ebenfalls importieren können.

Die Prüfung, ob die originale PAR-Datei Bibliotheken enthält, die Ihnen die Importfunktion vorenthalten hat, können Sie auch direkt

im Windows Explorer durchführen, indem Sie die beiden infrage kommenden Ordner der entpackten PAR-Datei auf eventuell vorhandene *jar*-Dateien überprüfen und anschließend nur dann die manuellen Importschritte durchführen, wenn Sie fündig geworden sind.

Zusammenfassung Fassen wir noch einmal kurz zusammen:

- ▶ Die Importfunktion des SAP NetWeaver Developer Studios berücksichtigt möglicherweise essenzielle Bibliotheken *nicht*.
- ▶ Sie müssen diese *manuell* in Ihr Projekt importieren, um ein funktionsfähiges Paket zu erhalten.
- ▶ Die betroffenen Dateien können in der originalen PAR-Datei in den beiden folgenden Ordnern gespeichert sein:
 - ▶ *PORTAL-INF/lib*
 - ▶ *PORTAL-INF/private/lib*

Beachten Sie dies immer bei der Entwicklung, anderenfalls werden Sie sich wundern, warum Ihre auf einer originalen PAR-Datei basierenden Komponenten nicht funktionieren.

10.3.3 Änderung des Titels der Login-Seite

Wenden wir uns nun den ersten Schritten auf dem Weg zu einer an unsere Wünsche angepassten Login-Seite zu. Wie bereits angekündigt, werden wir zunächst nur eine kleine Änderung an der Komponente vornehmen und die neue Login-Seite anschließend deployen. Diese Anpassung betrifft den Titel der Login-Seite, der im Standard in Deutsch lediglich »Willkommen« lautet (siehe Abbildung 10.13). Die auf der Login-Seite dargestellten Texte sind nicht fest in den JSP-Seiten hinterlegt, sondern liegen sprachabhängig in eigenen Dateien in der zuvor manuell importierten Datei *umelogonbase.jar* vor. Von dort werden die Texte abhängig von den aktuell gültigen Spracheinstellungen zur Laufzeit geladen.

Um die Texte ändern zu können, ist es zunächst erforderlich, die zu ändernde Sprachdatei aus dem Archiv zu entpacken. Anschließend können Sie dort die Textänderungen vornehmen und die geänderte Datei wieder zum Archiv hinzufügen.

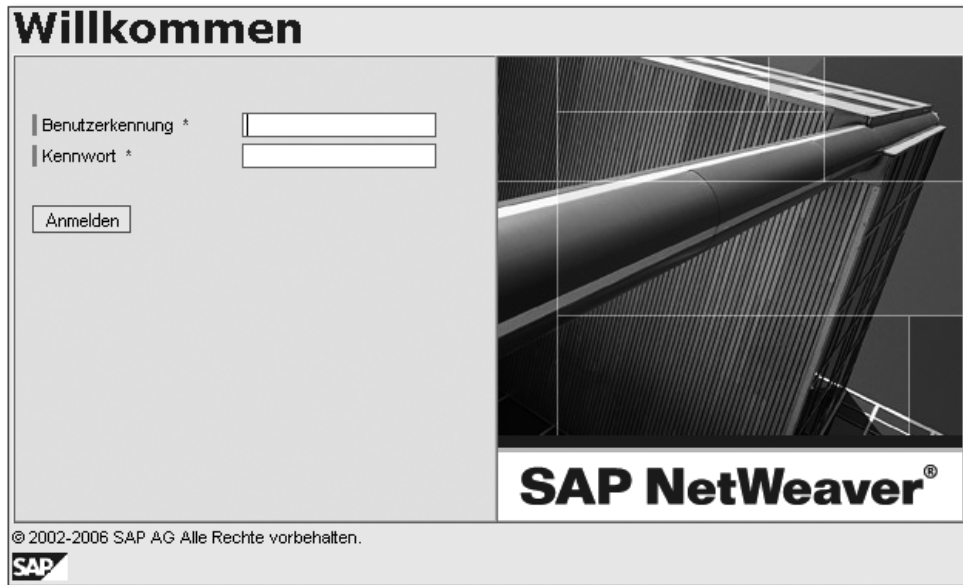


Abbildung 10.13 Originale Login-Seite

1. Navigieren Sie im Windows Explorer zu dem Ordner, in dem Ihr Projekt gespeichert ist. Wenn Sie diesen Ordner nicht mehr kennen, können Sie in den Projekteigenschaften (Rechtsklick auf das Projekt in PACKAGE EXPLORER • PROPERTIES) unter Info nachschauen. Sie finden den vollständigen Pfad rechts bei LOCATION (siehe Abbildung 10.14).

Sprachdatei
entpacken

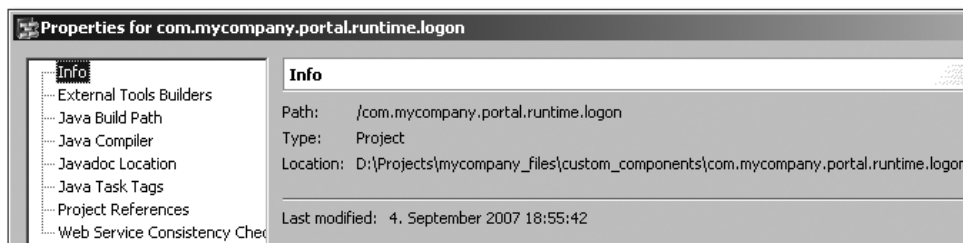


Abbildung 10.14 Projekteigenschaften mit Speicherort im Dateisystem

2. Öffnen Sie dort im Unterordner *PORTAL-INF/lib* mit WinZip die Datei *umelogonbase.jar*.
3. Entpacken Sie die Datei *logonLabels_de.properties* in einen Ordner Ihrer Wahl. Lassen Sie WinZip noch geöffnet.
4. Öffnen Sie mit dem Windows-Editor (DATEITYP: ALLE DATEIEN) die entpackte Datei (siehe Abbildung 10.15).

Willkommenstext
ändern



Abbildung 10.15 Öffnen der Datei logonLabels_de.properties

5. Finden Sie die Zeile, die mit `xtit_WELCOME=` beginnt.
6. Ergänzen Sie den Text nach dem Gleichheitszeichen: »Willkommen bei MyCompany«.
7. **SPEICHERN** Sie die Datei, und schließen Sie den Editor wieder.
8. Nun fügen Sie die Datei in WinZip wieder dem noch geöffneten Archiv hinzu.

Hinweis

Je nach verwendetem Programm kann es sein, dass sich die bereits im Archiv existierende Datei nicht einfach ersetzen lässt. In diesem Fall entfernen Sie die Originaldatei zunächst aus dem Archiv und fügen die soeben geänderte dann wieder hinzu. Achten Sie dabei auf das angezeigte Datum der Datei im Archiv, um festzustellen, ob die Datei auch wirklich ersetzt wurde.

9. Speichern Sie – falls nötig – das Archiv. In den meisten Packprogrammen ist dieser Schritt allerdings nicht erforderlich, da das Archiv automatisch mit der Änderung gesichert wird.

- Mehrsprachigkeit**
10. Wie Sie sicherlich anhand der Dateinamen bereits vermutet haben, liegen auch für alle anderen unterstützten Sprachen entsprechende Dateien in diesem Archiv vor. Wenn Sie den Titel für weitere Sprachen anpassen möchten, können Sie dies ebenfalls, wie beschrieben, tun. Die Dateien haben anstelle des `_de` für Deutsch im Namen das jeweilige Kürzel der Sprache, zum Beispiel `_en` für Englisch.

Wichtig: Sonderzeichen in den Sprachdateien!

Benutzen Sie in diesen Dateien ausschließlich ASCII-Zeichen und keine sprachspezifischen Sonderzeichen, wie zum Beispiel ü oder é. Wollen Sie diese Zeichen verwenden, so müssen Sie sie durch einen Platzhalter für Unicode-Zeichen ersetzen, der folgendes Format hat: `\u####`. Die Rautezeichen (#) ersetzen Sie durch den vierstelligen Unicode-Index des darzustellenden Zeichens in Hexadezimal-Schreibweise. Für ein »ü« ist dies beispielsweise `\u00FC`. Sie finden in den Dateien bereits mehrere solcher Beispiele.

Den Unicode-Index eines Zeichens können Sie mit der Windows-Zeichentabelle (START • PROGRAMME • ZUBEHÖR • SYSTEMPROGRAMME • ZEICHENTABELLE) eruieren. Den Unicode-Index zum ausgewählten Zeichen finden Sie dort in der Statuszeile (siehe Abbildung 10.16). Das Darstellungsformat dort ist `U+####`.



Abbildung 10.16 Unicode-Index in der Statuszeile der Zeichentabelle

10.3.4 Deployment in das Portal

Wie angekündigt, werden wir das Projekt nun vor der weiteren Bearbeitung in das Portal deployen, um sicherzustellen, dass die neue Komponente im Portal funktioniert. Am schnellsten geschieht dies über das Kontextmenü (Rechtsklick auf das Projekt im PACKAGE EXPLORER, dann QUICK PAR UPLOAD) oder den entsprechenden Button in der Symbolleiste des SAP NetWeaver Developer Studios CREATE/EXPORT PAR FILE (siehe Abbildung 10.17).



Abbildung 10.17 Exportoptionen: Symbolleiste – Kontextmenü

1. Benutzen Sie eine der beschriebenen Möglichkeiten, um mit dem Deployment zu beginnen.
2. Wählen Sie den Weg über den Button in der Symbolleiste, so erhalten Sie zunächst noch einmal einen Auswahldialog, in dem

Sie das zu erstellende Projekt selektieren können. In der Regel ist hier das aktuelle Projekt bereits ausgewählt. Durch einen Klick auf NEXT gelangen Sie zum Exportdialog. Wählen Sie den Weg über das Kontextmenü, so entfällt dieser Schritt.

3. Im Exportdialog, den Sie nun sehen, können Sie gegebenenfalls einen anderen lokalen Pfad für die zu erstellende PAR-Datei auswählen, wir belassen es jedoch beim Standardpfad, der dem Projektverzeichnis entspricht.

Deployment-Optionen

Sie haben beim Export die Möglichkeit, Java-Quelldateien mit zu exportieren (INCLUDE THE SOURCE CODE OF THE PORTAL APPLICATION), das heißt, nicht nur die während des Exportvorgangs erstellten kompilierten *class*-Dateien. Da es in dieser Komponente derzeit keine solchen Dateien gibt, ist diese Option im Moment ohne Bedeutung. Sollten Sie jedoch einmal Komponenten entwickeln, bei denen der Quellcode nicht zum vereinbarten Lieferumfang gehört, muss diese Option deaktiviert bleiben.

4. Wählen Sie die Option DEPLOY PAR aus, da wir das Projekt in das Portal deployen wollen. Daraufhin wird eine Tabelle mit möglichen Zielservers für das Deployment angezeigt (siehe Abbildung 10.18).

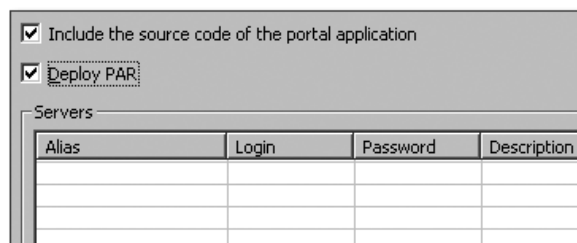


Abbildung 10.18 Exportoptionen mit noch leerer Servertabelle

Zielserver einrichten

Sofern Sie mit dem SAP NetWeaver Developer Studio noch keine Komponenten deployt haben, ist die Tabelle jetzt noch leer, und Sie müssen einen Server als Ziel für das Deployment konfigurieren:

1. Klicken Sie dazu auf CONFIGURE SERVER SETTINGS. Der Dialog öffnet sich zur Pflege der Zielsysteme für das Deployment.
2. Klicken Sie auf den Button ADD unter der Tabelle im Dialog, um nun einen Server hinzuzufügen. Ein Dialog mit fünf Eingabefeldern, die wir kurz erläutern, öffnet sich:

- ▶ **Alias**
ein kurzer Bezeichner für den Eintrag
 - ▶ **Host**
IP-Adresse oder Host-Name des Servers im Netzwerk
 - ▶ **Port**
HTTP-Port des Portals
 - ▶ **Login**
User-ID eines Benutzers mit Administratorrechten im Portal
(Das Passwort wird erst beim Deployment abgefragt.)
 - ▶ **Description**
eine kurze Beschreibung des Eintrags (optional)
3. Für unser Praxisbeispiel sind dies die folgenden Werte. Wenn Sie eine eigene/abweichende Portalinstallation verwenden, so müssen Sie die dafür korrekten Werte verwenden:
- ▶ ALIAS: Buying-Portal
 - ▶ HOST: *buying.mycompany.com*
 - ▶ PORT: 50000
 - ▶ LOGIN: Adm001
 - ▶ DESCRIPTION: Portal des Praxisbeispiels
4. Klicken Sie auf OK.
5. Klicken Sie über der Tabelle die Checkbox YES, I WANT TO REMEMBER PASSWORDS FOR DEPLOYMENT an (siehe Abbildung 10.19). Diese sorgt dafür, dass Sie nur beim ersten Deployment (bezogen auf den Programmstart) Ihr Passwort eingeben müssen und bei den folgenden nicht mehr.
6. Klicken Sie außerdem die Checkbox in der Spalte DEFAULT an. Damit ist dies ab sofort unser Standardzielsever für das Deployment, und wir müssen diesen bei späteren Deployments nicht mehr jedes Mal erneut als Ziel auswählen.

Einstellungen für
das Praxisbeispiel



Abbildung 10.19 Neuer Server in der Tabelle

7. Verlassen Sie den Dialog durch Klicken auf OK.
8. Nun sind alle Einstellungen für das Deployment getätigt. Klicken Sie auf FINISH, um den Vorgang zu starten. Unsere neue Komponente wird damit in das Portal deployt und steht für die Verwendung zur Verfügung.

10.3.5 Anpassen der Datei authschemes.xml

Nachdem wir unsere Login-Seite deployt haben, müssen wir das Portal noch so konfigurieren, dass diese auch anstelle der originalen verwendet wird. Konnten Sie die bisherigen Schritte noch ohne direkten Zugriff auf den Portalserver durchführen, benötigen Sie nun diesen direkten Zugriff auf den Server, da das Programm für die folgenden Einstellungen nur lokal auf dem Server ausgeführt werden kann.

Config Tool Bevor Sie die Änderungen durchführen können, sollten Sie allerdings zunächst alle Server Nodes und Dispatcher Nodes des J2EE-Engine-Clusters herunterfahren. Das benötigte Programm heißt *Config Tool* (siehe Abschnitt 4.5.1, »UME-Definitionen«, und Abschnitt 8.1.3, »Weitere UME-Parameter«), und Sie finden es in folgendem Verzeichnis: `<SAPJ2EEEngine_instance>\j2ee\configtool\configtool.bat`. Unter Linux heißt die auszuführende Datei entsprechend `configtool.sh`.

Download der
authschemes.xml

1. Starten Sie das Config Tool.
2. Die Frage, ob die Default-Einstellungen für die Datenbank verwendet werden sollen, beantworten Sie mit YES (siehe Abbildung 10.20).

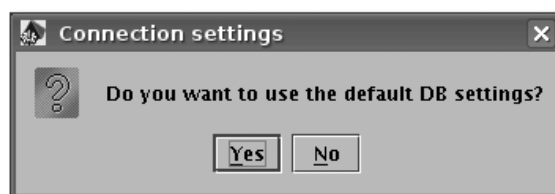


Abbildung 10.20 Default-Einstellungen

3. Wechseln Sie durch einen Klick auf das Icon mit dem Stift und den beiden farbigen Dokumenten (siehe Abbildung 10.21) in den EDITOR MODE.
4. In der Baumstruktur navigieren Sie zu folgendem Eintrag: CLUSTER_DATA • SERVER • PERSISTENT • COM.SAP.SECURITY.CORE.UME.SERVICE.

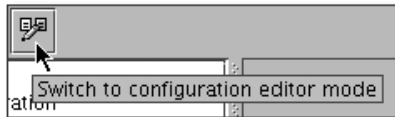


Abbildung 10.21 Wechsel in den Editor Mode

5. Wechseln Sie vom Ansichts- in den Bearbeitungsmodus, indem Sie auf das Icon mit dem Stift und der Brille (siehe Abbildung 10.22) in der Icon-Leiste klicken.



Abbildung 10.22 Wechsel in den Bearbeitungsmodus

6. Wählen Sie den Eintrag `AUTHSCHEMES.XML`, und öffnen Sie die Detailansicht durch einen Klick auf das Icon mit dem einzelnen Stift in der Icon-Liste (siehe Abbildung 10.23 und Abbildung 10.24).

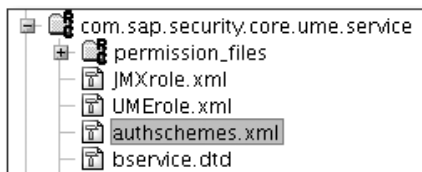


Abbildung 10.23 Auswahl der authschemes.xml

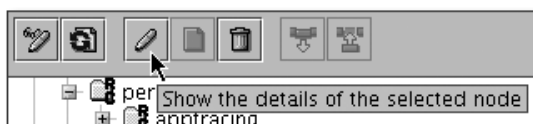


Abbildung 10.24 Öffnen der Detailansicht

7. Klicken Sie in dem sich öffnenden Dialog auf `DOWNLOAD`, und speichern Sie die Datei in einem Ordner Ihrer Wahl.

Sie haben nun eine lokale Kopie der Datei `authschemes.xml` gespeichert. In dieser Datei werden Sie die nötigen Änderungen vornehmen, damit unsere neue Login-Komponente verwendet wird. Dazu gehen Sie folgendermaßen vor:

1. Um die Datei anzupassen, können Sie zwar grundsätzlich einen beliebigen Texteditor verwenden, wir empfehlen Ihnen jedoch, einen XML-Editor zu benutzen.

Anpassung der
authschemes.xml

2. Öffnen Sie die soeben gespeicherte Datei *authschemes.xml* mit einem Editor.
3. Suchen Sie in der Datei alle Tags `<frontenttarget>`. Am besten verwenden Sie hierfür die Suchfunktion des Editors, um auch alle Tags zu finden.
4. Ersetzen Sie innerhalb dieser Tags alle »com.sap.portal.runtime.logon« durch »com.mycompany.portal.runtime.logon«. Eventuell weiteren Text innerhalb des Tags (auch solchen, der direkt im Anschluss an den Suchbegriff steht) belassen Sie unverändert, zum Beispiel:

► vorher:

```
<frontenttarget>
com.sap.portal.runtime.logon.header
</frontenttarget>
```

► nachher:

```
<frontenttarget>
com.mycompany.portal.runtime.logon.header
</frontenttarget>
```

5. Speichern Sie die Datei.

Original nicht überschreiben

Um nun nicht die originale *authschemes.xml* im Config Tool mit unserer geänderten zu überschreiben und diese damit als Sicherheit zu verlieren, legen wir im Folgenden in der Baumstruktur einen neuen Knoten mit dem Config Tool an, in dem wir die geänderte Datei ablegen werden. Dies hat zwei Vorteile:

- Wir behalten die Originalversion der Datei als Sicherheit.
- Durch ein Update (Support Package) könnte die originale *authschemes.xml* überschrieben werden. Unser neuer Eintrag bleibt dann aber erhalten.

Führen Sie die folgenden Schritte aus, um einen neuen Knoten im Baum zu erstellen:

Neuer Eintrag

1. Wählen Sie im Baum den Eintrag `COM.SAP.SECURITY.CORE.UME.SERVICE` aus.

- Erstellen Sie einen neuen Knoten nach dem ausgewählten Knoten, indem Sie auf das Icon mit dem leeren Blatt in der Icon-Leiste klicken (siehe Abbildung 10.25).

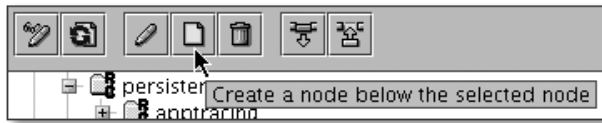


Abbildung 10.25 Neuen Knoten erstellen

- Im sich öffnenden Dialog wählen Sie in der Auswahlbox den Eintrag FILE-ENTRY (siehe Abbildung 10.26).
- Klicken Sie auf UPLOAD, und wählen Sie die soeben veränderte *authschemes.xml* aus.
- Standardmäßig wurde als neuer Name für den Knoten der Name der ausgewählten Datei eingetragen. Ändern Sie den Eintrag in *authschemes_custom.xml* um.

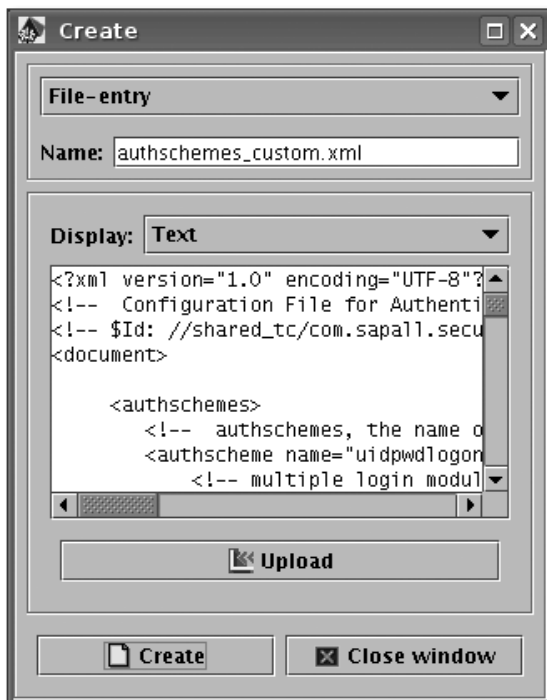


Abbildung 10.26 Eigenschaftsdialog des neuen Knotens

- Klicken Sie auf CREATE.
- Schließen Sie den Dialog durch einen Klick auf CLOSE WINDOW.

Der neue Knoten erscheint nun in der Baumstruktur. Damit dieser Eintrag anstelle des originalen verwendet wird, müssen Sie noch eine Einstellung vornehmen.

Neue Datei
bekannt geben

1. Navigieren Sie in der Baumstruktur zu folgendem Knoten: CLUSTER_ DATA • SERVER • CFG • SERVICES.
2. Doppelklicken Sie in der Baumstruktur auf den Eintrag PROPERTY-SHEET COM.SAP.SECURITY.CORE.UME.SERVICE.
3. Klicken Sie in dem sich öffnenden Dialog in der Liste auf den Eintrag LOGIN.AUTHSCHEMES.DEFINITION.FILE (siehe Abbildung 10.27).

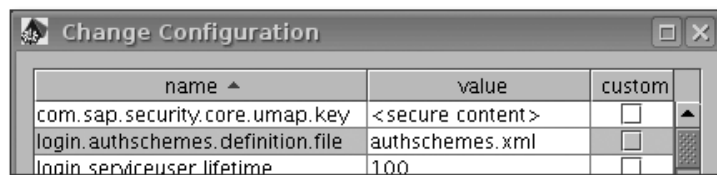


Abbildung 10.27 Zu ändernder Eintrag

4. Geben Sie den Namen der neuen Datei im Feld CUSTOM ein: »authschemes_custom.xml« (siehe Abbildung 10.28).

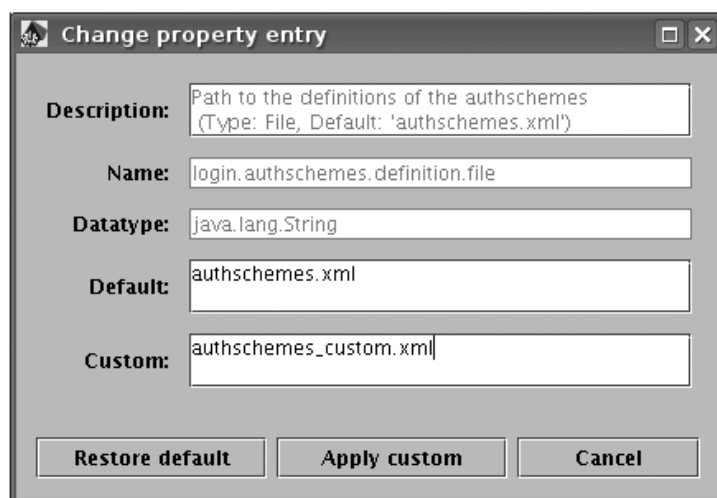


Abbildung 10.28 Neuer Name

5. Klicken Sie auf APPLY CUSTOM. In der Liste sehen Sie ein Häkchen in der Spalte CUSTOM, das anzeigt, dass der Wert dieser Einstellung kein Standardwert ist (siehe Abbildung 10.29).
6. Schließen Sie das Config Tool.

Server Nodes
neu starten

Nun starten Sie die Instanzen des J2EE-Clusters wieder, damit die Änderungen aktiv werden.

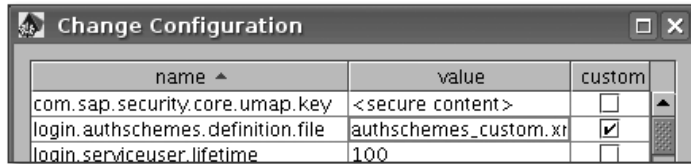


Abbildung 10.29 Eintrag geändert

10.3.6 Neue Login-Seite

Ist das Portal nach dem Neustart wieder verfügbar, sollte auf der Login-Seite Ihr geänderter Titel erscheinen (siehe Abbildung 10.30). Falls Sie anstelle der Login-Seite eine Fehlermeldung sehen, gehen Sie die zuvor beschriebenen Schritte noch einmal genau durch. Beachten Sie insbesondere den Abschnitt »Achtung: Fehlende Bibliotheken importieren« in Kapitel 10.3.2.



Abbildung 10.30 Neuer Titel der Login-Seite

Sollten Sie dem Fehler dennoch nicht auf die Spur kommen, können Sie in jedem Fall noch auf die originale Login-Seite zurückschalten, indem Sie die letzte Änderung im Config Tool rückgängig machen. Klicken Sie hierzu in dem Dialog aus Abbildung 10.28 einfach auf den Button RESTORE DEFAULT. Wenn Sie die Server Nodes erneut starten, haben Sie wieder die originale Login-Seite aktiviert.

Zurück zur
originalen
Login-Seite

Dadurch haben Sie die Möglichkeit, sich wieder anzumelden, wiederhergestellt und können in Ruhe nach dem Fehler suchen. Zukünftig genügt es dann, wieder die angepasste *authschemes_custom.xml* im Config Tool einzutragen und die Server Nodes neu zu starten, damit Ihre angepasste Login-Seite wieder aktiviert wird.

10.3.7 Genereller Aufbau einer Portalkomponente

Bevor wir uns nun weiter mit der Anpassung der Login-Seite beschäftigen, wollen wir zunächst einmal auf den generellen Aufbau – die Ordnerstruktur einer Portalkomponente – eingehen, den wir bisher vernachlässigt haben. Zunächst einmal müssen wir zwischen

dem Aufbau des Projektes im SAP NetWeaver Developer Studio und dem daraus resultierenden Aufbau der PAR-Datei unterscheiden.

Ordnerstruktur im
SAP NetWeaver
Developer Studio

Schauen wir uns zuerst den Aufbau im SAP NetWeaver Developer Studio an. Auf der obersten Ebene in unserem Projekt befinden sich drei Elemente (wenn Sie die Anzeige, wie in Abschnitt 10.3.2, »Import in SAP NetWeaver Developer Studio«, beschrieben, gefiltert haben), die wir erläutern wollen:

► **src.api**

Dieser Ordner dient der Speicherung von Java-Klassen (Sourcecode-Dateien, noch nicht kompiliert), die auch anderen Komponenten zur Verwendung zur Verfügung stehen sollen. Die hier implementierten Funktionen werden auch als API-Funktionen bezeichnet.

► **src.core**

Dieser Ordner dient der Speicherung von Java-Klassen (ebenfalls Sourcecode), die ausschließlich dieser Komponente zur Verfügung stehen.

Die Inhalte dieser beiden Ordner werden durch das SAP NetWeaver Developer Studio kompiliert und in entsprechende *jar*-Dateien gepackt, die dann der erstellten PAR-Datei hinzugefügt werden.

► **dist**

Der Inhalt dieses Ordners wird in seiner Originalstruktur in das Wurzelverzeichnis der erstellten PAR-Datei übernommen. Einige Inhalte (die sogenannten Webressourcen) werden auch direkt über eine URL abrufbar sein, andere Inhalte hingegen nicht. Auf den Inhalt dieses Ordners wollen wir im Folgenden näher eingehen.

Ordner *dist* Da die Inhalte des Ordners *dist* vollständig in die erstellte PAR-Datei übernommen werden, gelten die folgenden Beschreibungen sowohl für das Projekt im SAP NetWeaver Developer Studio als auch für den Aufbau der daraus erzeugten PAR-Datei.

Ordner
PORTAL-INF

Der wichtigste Bestandteil dieses Ordners ist der Unterordner *PORTAL-INF*. Dieser Ordner enthält einerseits die zentrale Konfigurationsdatei einer Portalanwendung *portalapp.xml*, andererseits sämtliche (kompilierten) Java-Klassen der Anwendung. Falls beim Deployment die Option gewählt wurde, auch den Quellcode mit einzubeziehen, so ist auch dieser hier zu finden. Die in Tabelle 10.1 gezeigten Unterordner haben fest definierte Bedeutungen.

Ordnername	Inhalt
<i>lib</i>	Enthält alle Java-Bibliotheken (JAR-Dateien).
<i>private/lib</i>	
<i>classes</i>	Enthält alle CLASS-Dateien sowie Ressourcendateien.
<i>private/classes</i>	

Tabelle 10.1 Spezielle Unterordner von PORTAL-INF

Die Inhalte der Ordner *lib* und *classes* auf oberer Ebene können auch von anderen Anwendungen referenziert werden. Hier werden bei Erzeugung der PAR-Datei vom SAP NetWeaver Developer Studio die kompilierten Inhalte des Ordners *src.api*, gepackt in Form eines JAR-Archivs, hineinkopiert. Die Inhalte der beiden Unterordner im Ordner *private* stehen ausschließlich der Anwendung selbst zur Verfügung und enthalten unter anderem die kompilierten Inhalte des Ordners *src.core*. Die Ressourcendateien sind in der Regel Dateien mit der Endung *.properties*, die als einzelne Dateien in den *classes*-Ordnern oder, in JAR-Archiven gepackt, in den *lib*-Ordnern abgelegt werden können.

Sichtbarkeit der Inhalte

Die Datei *portalapp.xml* beschreibt, welche Komponenten und Services eine Anwendung enthält. Sie beinhaltet sowohl Konfigurationseinstellungen für die gesamte Anwendung als auch für die einzelnen Komponenten bzw. Services. Letztere können dann im Portal zum Beispiel für jeden iView, der auf einer Komponente basiert, geändert werden. Sie haben diese Einstellungen bereits in Kapitel 3, »Portal Content«, kennengelernt. Es handelt sich um die verschiedenen Konfigurationsparameter eines iViews.

Datei *portalapp.xml*

Die ENTERPRISE PORTAL PERSPECTIVE bietet für die Bearbeitung der *portalapp.xml* einen speziellen Editor, der automatisch geöffnet wird, wenn Sie die Datei per Doppelklick öffnen. So müssen Sie die Datei in den meisten Fällen nicht direkt im XML-Code bearbeiten. Wir werden im Rahmen des Abschnitts 10.4 noch genauer auf die Bearbeitung dieser Datei und ihre XML-Struktur eingehen.

Editor für *portalapp.xml*

10.3.8 Bestandteile der Login-Komponente

Die Login-Komponente besteht aus mehreren JSP-Seiten, die dem Benutzer neben der bloßen Anmeldung noch zusätzliche Funktionen

zur Verfügung stellen. Diese Dateien sind in Tabelle 10.2 aufgelistet und liegen im Ordner *PORTAL-INF*. Die eigentliche Programmlogik für den Login-Prozess selbst liegt in der importierten Bibliothek *com.sap.portal.runtime.logon_api.jar* (siehe Abschnitt »Achtung: Fehlende Bibliotheken importieren« in Kapitel 10.3.2) in kompilierter Form vor. Deshalb kann die Login-Seite auch nicht funktionieren, wenn diese Bibliothek fehlt.

Wesentliche Dateien

Dateiname	Beschreibung
<i>umLogonPage.jsp</i>	Anmeldeseite
<i>umResetPasswordPage.jsp</i>	Seite, um Passwort zurückzusetzen, falls es vergessen wurde
<i>changePasswordPage.jsp</i>	Seite, um Passwort zu ändern, wenn das alte abgelaufen ist
<i>umHelpPage.jsp</i>	Hilfeseite zum Login, mit Auswahl der Art von Hilfe
<i>umLogonProblemPage.jsp</i>	Seite, um bei Problemen eine Nachricht an den Administrator zu senden
<i>umLogoffPage.jsp</i>	Bestätigungsseite nach Abmeldung

Tabelle 10.2 Dateien für die einzelnen Seiten der Login-Komponente

Alle Seiten, zu denen diese Dateien gehören, haben die gleiche Struktur. Sie bestehen aus einem inhaltlich funktionellen Teil, der in eben diesen JSP-Dateien implementiert ist, und aus einem Layout-Rahmen, der in zwei weiteren JSP-Dateien implementiert ist, die allerdings die Dateiendung *txt* besitzen. Diese beiden Dateien sind in allen JSP-Dateien aus Tabelle 10.2 per `include`-Direktive eingebunden und enthalten neben dem reinen HTML-Rahmen für die Seitenstruktur auch noch sichtbare Elemente, die in Tabelle 10.3 aufgelistet sind.

Dateiname	Sichtbare Elemente
<i>umLogonTopArea.txt</i>	Willkommenstext
<i>umLogonBotArea.txt</i>	Branding-Image (1 + 2), Copyright, SAP-Logo

Tabelle 10.3 Dateien des Layout-Rahmens

Aufbau der Login-Seiten

In Abbildung 10.31 ist der schematische Aufbau der Login-Seiten dargestellt. Dieser Aufbau ist, wie erwähnt, für alle in Tabelle 10.2 genannten Seiten gleich. Wenn Sie Änderungen an einer der beiden

Dateien aus Tabelle 10.3 vornehmen, wirken sich diese Änderungen auf alle Login-Seiten aus. Dabei müssen Sie außerdem beachten, dass durch die Art und Weise, wie durch diese Dateien der endgültige HTML-Code erzeugt wird, Änderungen an der generellen Seitenstruktur in der einen Datei auch Änderungen in der anderen Datei notwendig machen können.

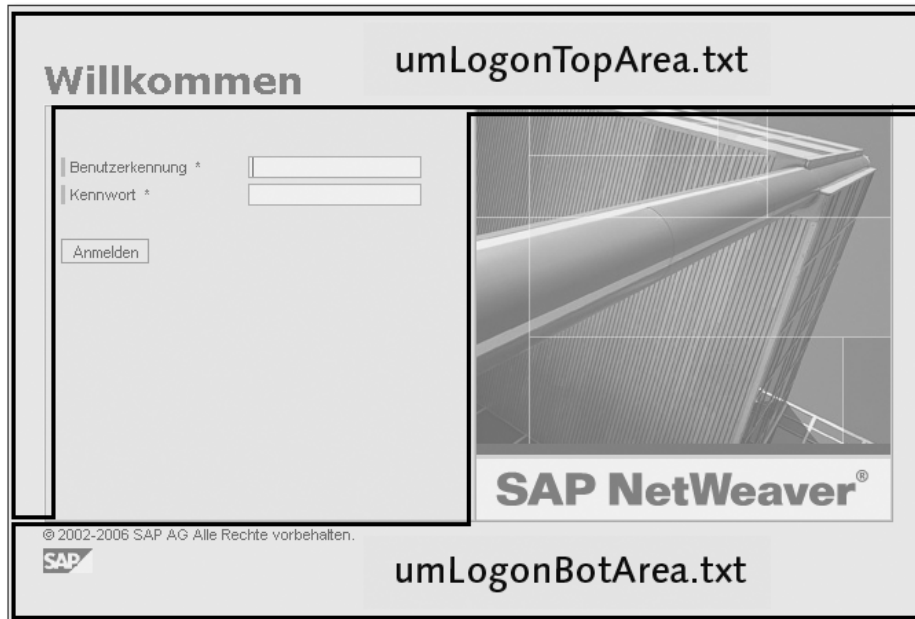


Abbildung 10.31 Schematischer Aufbau der Login-Seiten

10.3.9 Hinzufügen von Nutzungsbedingungen

Nun wollen wir uns der eigentlichen Anpassung der Login-Seite zuwenden. Wir wollen einen zusätzlichen Text, der einen Hinweis auf die Nutzungsbedingungen für das Portal enthält, in die Login-Seite einbinden. Zusätzlich soll eine Anmeldung nur möglich sein, wenn der Benutzer diese Nutzungsbedingungen auch akzeptiert. Dies werden wir durch eine zusätzliche Checkbox realisieren, die der Benutzer anklicken muss, um zu bestätigen, dass er die Nutzungsbedingungen akzeptiert, bevor eine Anmeldung möglich ist. Der Text und die Checkbox sollen vor dem Anmelde-Button auf der Login-Seite eingebunden werden.

Erstellen Sie zunächst einen neuen Ordner, in dem die Datei mit den Nutzungsbedingungen abgelegt werden soll.

1. Erstellen Sie im Ordner *dist* (siehe Abschnitt 10.3.7, »Genereller Aufbau einer Portalkomponente«) einen neuen Unterordner. Kli-

Neuen Ordner anlegen

cken Sie dazu mit der rechten Maustaste auf den Ordner *dist*, und wählen Sie NEW • OTHER.

2. Im folgenden Dialog wählen Sie links SIMPLE, dann rechts FOLDER und klicken auf NEXT.
3. Tragen Sie im Feld FOLDER NAME »html« ein.
4. Klicken Sie auf FINISH.

Nun erstellen Sie im SAP NetWeaver Developer Studio eine neue HTML-Datei, die die Nutzungsbedingungen enthält.

HTML-Datei
erstellen

1. Klicken Sie mit der rechten Maustaste auf den soeben erstellten Ordner, und wählen Sie NEW • OTHER.
2. Im dann geöffneten Dialog wählen Sie zunächst links den Eintrag SIMPLE und anschließend rechts den Eintrag FILE aus und klicken auf NEXT.
3. Im Feld FILE NAME geben Sie »nb.htm« ein.
4. Klicken Sie auf FINISH.

Die neue Datei wird automatisch geöffnet, allerdings im Vorschaumodus (PREVIEW). Wechseln Sie durch Klicken auf SOURCE unter dem leeren Vorschaufenster (siehe Abbildung 10.32) in die Quellcode-Ansicht.

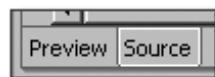


Abbildung 10.32 Wechsel zwischen Vorschau- und Quellcode-Ansicht

Beginnen wir nun mit dem Inhalt der Datei. Wir beschränken uns hier auf ein exemplarisches Grundgerüst, echte Inhalte können Sie später selbst einfügen.

Inhalt der Datei

1. Geben Sie den folgenden HTML-Code im Editor ein:

```
<html>
  <head>
    <title>MyCompany.com - Nutzungsbedingungen</title>
  </head>
  <body>
    <h1>ALLGEMEINE NUTZUNGSBEDINGUNGEN</h1>
    <p>Mit Ihrer Anmeldung akzeptieren Sie folgende
      Punkte:</p>
```

```

<ol>
  <li>Punkt 1</li>
  <li>Punkt 2</li>
  <li>Punkt 3</li>
</ol>
</body>
</html>

```

Listing 10.1 Inhalt der neuen HTML-Datei nb.htm

2. Wichtig für die Bearbeitung ist, dass im SAP NetWeaver Developer Studio Dateien standardmäßig mit dem Zeichensatz UTF-8 bearbeitet werden.
3. Damit ein Browser die Datei zweifelsfrei korrekt darstellen kann, sollten Sie im `<head>` ein `<meta>`-Tag einfügen, das eine Angabe zum verwendeten Zeichensatz enthält. Da wir außerdem sicherstellen wollen, dass ein Benutzer immer die aktuelle Version der Datei erhält und nicht etwa eine ältere, die noch in seinem Browser-Cache gespeichert ist, benötigen wir außerdem ein entsprechendes `<meta>`-Tag, das den Browser veranlasst, diese Datei bei jedem Aufruf erneut vom Server abzurufen:

Zeichensatz und
Cache-Verhalten

```

<meta http-equiv="content-type"
      content="text/html; charset=UTF-8">
<meta http-equiv="expires" content="0">

```

Listing 10.2 Zusätzliche Meta-Tags im `<head>`

4. Fügen Sie diese Zeilen im `<head>` der Datei *nb.htm* ein, und speichern Sie die Datei.

Nun wollen wir der Seite noch ein wenig Layout gönnen, wofür wir eine CSS-Datei anlegen, die wir anschließend in der HTML-Datei einbinden.

Layout für
die Nutzungs-
bedingungen

1. Erstellen Sie im Ordner *css* des Projektes eine neue Datei *nb.css*. Wie Sie neue Dateien anlegen, wissen Sie mittlerweile.

Anmerkung

Wir könnten den CSS-Code auch direkt in der HTML-Datei eingeben, wählen hier aber bewusst nicht diesen Weg, denn so können Sie diese CSS-Datei später gegebenenfalls auch noch für andere Dateien verwenden (zum Beispiel andere Sprachversionen der *nb.htm*).

Auch diese CSS-Datei wird direkt nach dem Erstellen geöffnet, allerdings in diesem Fall direkt in einem Texteditor ohne Vorschau, denn für CSS-Dateien ist im SAP NetWeaver Developer Studio kein eigener Editor vorgesehen.

2. Geben Sie den folgenden CSS-Code ein, der als Schriftart für das gesamte Dokument Verdana und die Schriftgrößen für Überschrift, Absätze und Listenpunkte definiert:

```
body { font-family: Verdana; }
h1 { font-size: 110%; }
p, li { font-size: 90%; }
```

CSS-Datei
einbinden

3. Wechseln Sie wieder zu der Datei *nb.htm*, indem Sie auf den entsprechenden Karteireiter über dem Editor klicken.
4. Fügen Sie die folgende Zeile nach den `<meta>`-Tags ein, um auf die neue CSS-Datei zu verweisen:

```
<link rel="stylesheet" href="../css/nb.css">
```

5. Speichern Sie die Datei.
6. Wechseln Sie durch Klicken auf PREVIEW unter dem Editor wieder in den Vorschaumodus. Das Ergebnis sollte nun aussehen wie in Abbildung 10.33.

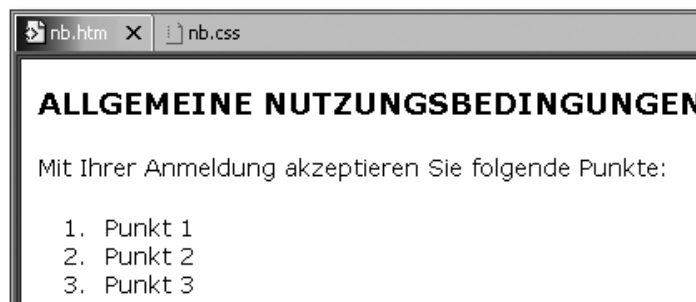


Abbildung 10.33 Neue Vorschau der Datei *nb.htm* mit eingebundenem Stylesheet

Anpassung der
umLogonPage.jsp

Um den Text in die Login-Seite einzubinden, müssen Sie Änderungen an der Datei *umLogonPage.jsp* vornehmen.

1. Öffnen Sie die Datei *umLogonPage.jsp* im Ordner *PORTAL-INF* durch einen Doppelklick oder über den Kontextmenübefehl OPEN, und wechseln Sie in die Quellcode-Ansicht.
2. Suchen Sie im Quellcode folgende Zeile:

```
<!-- logon button -->
```

3. Fügen Sie nun vor diese Zeile den kompletten Code aus Listing 10.3 ein, und speichern Sie die Datei.

```
<!-- disclaimer -->
<tr>
  <td colspan="2" class="urLblStd">
    Mit meiner Anmeldung akzeptiere ich die
    <a href="<%=webpath%>html/nb.htm" target="_blank">
      Nutzungsbedingungen
    </a>
    <input type="checkbox" name="accept_nb" />
    <span class="urLblReq">&nbsp;*</span>
  </td>
</tr>
<tr>
  <td colspan="2" height="20">&nbsp; </td>
</tr>
```

Listing 10.3 Neuer Codeabschnitt in umLogonPage.jsp

In der Pfadangabe des `<a>`-Tags sehen Sie, dass dem Pfad `<%=web-path%>` vorangestellt ist. Die Variable `webpath` steht Ihnen in allen JSP-Seiten der Login-Komponente zur Verfügung, da sie in der JSP-Datei `logon_proxy.txt` definiert ist, die zusätzlich zu den in Tabelle 10.3 genannten Dateien ebenfalls in jeder Seite per `include`-Direktive eingebunden ist. Diese Datei erzeugt allerdings selbst keine sichtbaren Elemente.

Variable `webpath`

In dieser Variablen befindet sich zur Laufzeit der Pfad zu Ihrer Komponente im Portal, sodass die Kombination dann den vollständigen gültigen URL-Pfad ergibt, unter dem die Datei vom Browser abgerufen werden kann. Sie fragen sich jetzt vielleicht, warum wir hier mit dieser Variablen arbeiten und nicht einfach wie beim Einbinden der CSS-Datei mit relativen Pfaden. Der Unterschied bei der Referenzierung ist folgender:

- ▶ Im Falle der Dateien `nb.htm` und `nb.css` handelt es sich um Dateien, die zur Laufzeit tatsächlich direkt über das Web abrufbar sein werden und deren vollständiger Pfad sich dann lediglich durch ihre beiden Ordner unterscheidet, deren relative Pfade uns jetzt schon durch die Ordnerstruktur im Projekt bekannt sind.

Unterschiede der Referenzierung

- Im Falle der *umLogonPage.jsp* handelt es sich jedoch zur Laufzeit nicht mehr um die von uns bearbeitete JSP-Datei, die ausgeführt wird, sondern um eine Java-Klasse, die dann auch über einen anderen (uns im Moment unbekanntem) Pfad über das Web abgerufen wird. Daher ist hier eine andere Art der Referenzierung von externen Dateien erforderlich, auf die erst zur Laufzeit zugegriffen wird.

Deployment Deployen Sie nun die Login-Komponente erneut, um den aktuellen Stand live im Portal zu überprüfen. Das Ergebnis sollte Abbildung 10.34 entsprechen.



Abbildung 10.34 Eingebundene Nutzungsbedingungen

Prüfung der Checkbox

Nun fehlt noch die Prüfung der Checkbox, damit ein Login nur dann möglich ist, wenn der Benutzer die Checkbox auch ausgewählt hat. Hierfür benötigen wir eine JavaScript-Funktion, die beim Absenden des Login-Formulars ausgeführt wird und gegebenenfalls das Absenden verhindert.

1. Suchen Sie in der *umLogonPage.jsp* die folgenden Zeilen:

```
</script>
<% if( !inPortal ) { %>
</head>
```

2. Fügen Sie vor dem schließenden `</script>`-Tag den folgenden JavaScript-Code ein:

```
function check_accept_nb(form) {
    var accepted = form.elements['accept_nb'].checked;
    if (!accepted) {
        alert('Sie müssen die Nutzungsbedingungen akzeptieren!');
    }
    return accepted;
}
```

Die Funktion erwartet als Parameter ein Formularobjekt und liefert als Ergebnis `true`, falls die Checkbox `accept_nb` des Formulars ausgewählt ist. Anderenfalls wird eine `Messagebox` mit einer Meldung angezeigt, und die Funktion liefert `false`. Alles, was jetzt noch getan werden muss, ist, dem `submit`-Ereignis des Formulars diese Funktion zuzuweisen.

HTML-Formularprüfung bei submit

Ein HTML-Formular, das abgesendet wird, erzeugt ein Ereignis `submit`, bevor die Daten des Formulars tatsächlich gesendet werden. An dieses Ereignis kann nun eine Prüffunktion gekoppelt werden, die `true` oder `false` zurückliefern muss. Im Falle von `true` werden die Daten gesendet, anderenfalls nicht.

3. Suchen Sie die Zeile, die mit `<FORM name="logonForm"` beginnt, und gehen Sie an das Ende der Zeile. Dort wird dem `<form>`-Tag unter einer bestimmten Bedingung ein `onSubmit`-Attribut mit einem Funktionsaufruf hinzugefügt.
4. Ändern Sie das Ende der Zeile, das im Original so aussieht:

```
?onSubmit=\"javascript:addTenantPrefix();\" \"%>>
```

Die Änderung sollte folgendermaßen aussehen (alles in einer Zeile, die Ergänzungen sind hervorgehoben):

```
?onSubmit=\"javascript:addTenantPrefix();
return check_accept_nb(this);\" :
onSubmit=\"return check_accept_nb(this);\" \"%>>
```

Durch diese Änderung wird in jedem Fall ein `onSubmit`-Attribut erzeugt, das mindestens unsere Prüffunktion beinhaltet. Als Parame-

Prüffunktion
einbauen

ter benutzen wir `this`, das auf das den Aufruf beinhaltende Tag weist, in diesem Fall das Formular. Damit ist sichergestellt, dass die Funktion das Formular bzw. die darin enthaltene Checkbox `accept_nb` referenzieren und damit auch prüfen kann.

Erneutes Deployment

Deployen Sie die Login-Komponente ein weiteres Mal, und prüfen Sie die Funktion im Browser. Hat alles funktioniert, sollte nun eine Anmeldung nur noch möglich sein, wenn die Checkbox ausgewählt ist. Im anderen Fall sollte die in der Prüffunktion hinterlegte Meldung erscheinen, die Sie selbstverständlich noch nach Ihren Wünschen anpassen können.

Kein JavaScript

Einen Punkt haben wir mit dieser Lösung jedoch noch nicht berücksichtigt: die Möglichkeit, dass ein Anwender JavaScript deaktiviert hat. In diesem Fall kann er sich auch ohne Auswahl der Checkbox anmelden (wenngleich der Rest des Portals dann nicht korrekt funktionieren wird). Um auch diesen Punkt zu beheben, wollen wir die Anzeige des Anmeldeformulars unterdrücken, falls JavaScript im Browser deaktiviert ist. Stattdessen werden wir eine entsprechende Meldung für den Benutzer anzeigen.

Hierzu werden wir das Anmeldeformular initial auf unsichtbar einstellen, eine Meldung hinzufügen, die initial sichtbar ist, sowie ein JavaScript einfügen, das die Sichtbarkeit dieser beiden Elemente umkehrt. Ist JavaScript aktiviert, sieht der Benutzer das Anmeldeformular, anderenfalls die Meldung.

1. Erweitern Sie das `FORM`-Tag um den im Folgenden hervorgehobenen Teil (wir haben hier nur den Anfangsteil des Tags dargestellt, die restlichen Attribute des Tags lassen Sie unverändert):

```
<FORM name="logonForm" style="display:none"
...

```

2. Suchen Sie im Quellcode das schließende `FORM`-Tag `</form>`.
3. Fügen Sie nach dem Tag den Code aus Listing 10.4 ein, und speichern Sie die Datei.

```
<div id="no_javascript" class="urMsgBarErr">
  <table border="0" cellpadding="0" cellspacing="0">
    <tbody>
      <tr>
        <td class="urTxtStd"><span class="urMsgBarImgError">
```

```

        &nbsp;</span></td>
    <td><span class="urTxtStd" tabindex="0">
        Sie haben JavaScript deaktiviert.</span></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td><span class="urTxtStd" tabindex="0">
        Die Nutzung des Portals ist ohne JavaScript nicht
        möglich.</span></td>
</tr>
</tbody>
</table>
</div>
<script language="JavaScript">
    document.getElementById("no_javascript")
        .style.display = "none";
    document.forms.logonForm
        .style.display = "block";
</script>

```

Listing 10.4 Hinweis und Skript

4. Deployen Sie die Komponente.

Sie sollten nun, wenn Sie die Login-Seite des Portals ohne JavaScript aufrufen, den entsprechenden Hinweis sehen (siehe Abbildung 10.35). Wenn Sie JavaScript aktiviert haben, sollte alles wie bisher funktionieren.

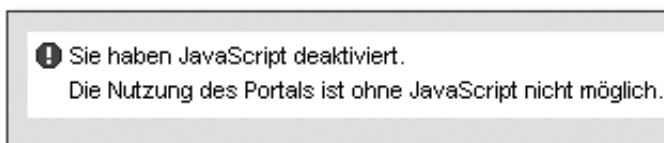


Abbildung 10.35 Neuer Hinweis bei deaktiviertem JavaScript

JavaScript im Browser aktivieren/deaktivieren

Im Internet Explorer finden Sie die Einstellung zum Aktivieren/Deaktivieren von JavaScript im Menü EXTRAS • INTERNETOPTIONEN unter dem Karteireiter SICHERHEIT. Hier werden verschiedene Sicherheitszonen für Webseiten verwaltet. Rechts in der Statusleiste des Browsers sehen Sie den Namen der gültigen Sicherheitszone, wenn Sie die Login-Seite des Portals aufrufen. Wählen Sie diese Sicherheitszone hier aus, und klicken Sie auf den Button

STUFE ANPASSEN. Die Einstellung für die Ausführung von JavaScript können Sie nun unter `SCRIPTING • ACTIVE SCRIPTING` aktivieren oder deaktivieren. Verlassen Sie die Einstellungsdialoge jeweils durch Klicken auf OK.

Im Mozilla Firefox finden Sie die Einstellung im Menü `EXTRAS • EINSTELLUNGEN` unter der Kategorie `INHALT`. Dort können Sie über die Checkbox `JAVASCRIPT AKTIVIEREN` die Ausführung von JavaScript auf Webseiten aktivieren oder deaktivieren.

Damit haben wir die Anpassung der Login-Seite im Rahmen dieses Kapitels vollendet.

10.4 Erstellen einer Sitemap

Um es Benutzern zu erleichtern, bestimmte Inhalte des Portals schneller zu erreichen, wollen wir eine Sitemap entwickeln, die jedem Benutzer zur Verfügung gestellt werden kann. Diese Sitemap soll für jeden Benutzer selbstverständlich nur diejenigen Elemente beinhalten, die er aufgrund seiner zugewiesenen Portalrollen auch tatsächlich sehen darf. Die Sitemap wird grundsätzlich wie ein Navigations-iView funktionieren, der den vollständigen Navigationsbaum des aktuellen Benutzers wiedergibt.

In Abbildung 10.36 sehen Sie, wie die vollständige Sitemap für einen Administratorbenutzer aussehen wird. Wir verwenden hier bewusst einen Administratorbenutzer für die Entwicklung und die Screenshots, damit Sie eine bessere Vergleichsbasis Ihrer Ergebnisse mit den Screenshots haben.

Für die Entwicklung bietet es sich an, auf die vom Portal zur Verfügung gestellte Navigation Tag Library zurückzugreifen. Diese bietet verschiedene Funktionen, die wir für die Erstellung einer Sitemap benötigen, und wir müssen nur wenig Java-Code selbst schreiben.

Um die Tag Library nutzen zu können, müssen wir unsere Komponente entweder komplett als JSP-Seite entwickeln oder eine JSP-Seite erstellen, die wir als Ressource in einer ebenfalls zu erstellenden Java-Klasse verwenden, die die Klasse `AbstractPortalComponent` erweitert. Da die Sitemap allerdings keine komplexe Programmlogik enthalten wird, und die zweite Möglichkeit uns daher keine Vorteile bietet, sondern in diesem Fall nur unnötige Mehrarbeit bedeutet, wählen wir die erste Möglichkeit.

Index

A

ABAP 370, 373, 375
Access denied 70, 123
ACL (Access Control List) 147, 155
Active Directory 125, 134, 228
Administration 33
Administrator 130
 Rolle 130
Alias 122, 341
Anmeldemethode 122, 157
Anonymous 106
Anonymous User 105
 Gruppe 106
Anpassung 246, 254, 271
Ansatz
 deskriptiver 349
 visuell-generischer 349
Ansichtsmodus 271
Antwort-E-Mail-Adresse 225
Anwendung 350, 366, 368, 376, 379,
 381, 391, 392, 394
Anwendungsnamensraum 92, 93
Anwendungsparameter 92, 94, 95
Anzeigeregeln 178
API 358, 359
Application Name 101
Architektur 348
ASCII 82, 267
Attribut 394, 396, 398
Aufruf 375
Auftragsbestätigung 345
Auktion 59, 60
Ausführberechtigung 123
Ausschneiden 65
Authenticated Users 104
Authentifizierung 143, 147
Automatismus 347, 348, 361

B

Bandbreite
 niedrige 177
BAPI (Business Application Program-
 ming Interface) 348, 376

Basisbetreuung 31
Bean-Anlage-Wizard 356
Bearbeitungsmodus 271
Bedingung
 komplexe 182
Beispielaufruf 373, 375
Beispielcode 387, 403
Beispielmodell 350, 351
Benutzer 67, 127, 168
 Guest 106
Benutzeradministration 19, 125
Benutzerantrag 393, 400, 401, 403
Benutzer-ID 147
Benutzerverwaltung 121, 228, 350
 Eindeutigkeit 230
 gemeinsame 70
Berechtigung 70, 103, 122, 194, 239
 iView 106
Berechtigungsdialog 103
Berechtigungskonzept 19
Berechtigungsprüfung 70
BEx Analyser 96
Bibliothek 277, 318, 340, 360, 401,
 402
 *com.sap.portal.run-
 time.logon_api.jar* 262
 fehlende 260
 SapSecuLib 164
 umelogonbase.jar 262, 264
BigVendor 370, 375
Bildschirmfeld 389
Branding
 Image 255, 278
 Text 255
Browser
 Fehlersymbol 147
 Titel 173
BSP (Business Server Page) 89
 Alias 90
 Alias-iView 92
 iView 89, 91
Business Layer 346
Button 386
buying 117

C

Checkbox 279
 Prüfung 284
 Client Proxy 372
 com.sap.portal.appintegrator.sap 102
 Company 132
 Company-Konzept 105, 134, 138
 STPD_ 134, 138
 Config Tool 139, 225, 227, 255, 270
 Content
 exportieren 234
 Content-Administration 57, 61
 Content-Bereich 59
 Content-Element 61
 anlegen 63
 Context 376, 381ff., 390, 394ff.
 Element 383
 Knoten 395, 398
 Variable 389
 Controller 379, 381ff., 394
 Cookie 344
 Cross-Domain-Cookie 148, 150
 MYSAPSSO2 147
 CSS (Cascading Stylesheets) 246, 281, 310, 316
 Datei 170
 Sprachkenntnis 172

D

Data Binding 399
 Data Source Configuration 230
 DataSource 396, 398, 403, 405
 Datei
 authschemes.xml 270
 authschemes_custom.xml 273
 com.sap.portal.navigation.lightdetailednavigationtree.par 341
 com.sap.portal.navigation.lighttoplevel.par 341
 com.sap.portal.navigation.masthead.par 317, 341
 com.sap.portal.navigation.pagetoolbar.par 341
 com.sap.portal.navigation.toolarea.par 341

com.sap.portal.runtime.logon.par 341
 com.sap.portal.support.browse.default 257
 logon_proxy.txt 283
 logonLabels_de.properties 265
 portalapp.xml 248, 276, 277, 291, 293, 319, 320, 325
 sapcrypto.dll 165
 umLogonPage.jsp 282
 verify.pse 164, 165
 web.xml 181
 Dateisystem 252, 255, 261
 Datenbank 346
 Datenhaltung 346
 Datentransport 381
 delegated_user_admin 104, 105
 Deltalink 67, 73
 Deploy 365, 392
 Deploy Tool 47
 Deployment 267, 331
 Option 268
 Deploy-Service 50
 Design 62, 180
 Designkomponente 245
 Desktop 62, 168, 178, 182
 Innenbereich 174, 175
 Directory 350
 Direktive 278, 283
 Domäne 148
 Download 234
 Download-Manager 27
 Drei-Schichten-Architektur 345, 346, 348
 DTO (Data Transfer Object) 353, 357
 Dummy-Load 138

E

Eclipse 251
 Eigenentwicklung 246, 249, 349
 Eigenschaftskategorie 83
 Darstellung – Größe 75, 84
 Drag & Relate 86, 96
 Erweitert 87
 Hilfe 88
 Information 84

Internet Transaction Server 99, 120
Konnektor 118
Laden (Cache) 87
Monitoring 88
Navigation 83
Tray 85
Verfügbarkeit 87
Web Application Server 91, 94, 119
Web Dynpro 100
Web Page Composer 214
Webseitenbereich 89
 Einfügen 65
 Einkauf 59, 60, 71
 Einkäufer 343, 345, 346
 Einkaufsportal 15
 Einkaufsprozess 16
 Einstellung 378
 Einstiegsnavigation 174
 Einstiegspunkt 83
 EJB (Enterprise JavaBean) 351, 352, 355, 356, 357, 364, 365
 Wizard 357
 E-Mail 129, 223, 343, 390
 Benachrichtigungs-E-Mail 223
 Enterprise Application Project 360, 365
 Enterprise Portal Perspective 251
 Enterprise Service Proxy 372
 Entity Bean 356
 Entpacken 261, 264
 Entwicklertool 150
 Entwicklung 344, 348
 Entwicklungsmodus 252
 Entwicklungspaket 371
 Entwicklungsszenario 344
 Entwicklungsumgebung 348, 371
 EPA-Datei 234
 Erleichterung 361
 Erscheinungsbild 168, 174
 Erstellung 376
 EvaluateTicketLoginModul 161
 Event 398, 400
 Export 134
 Exportdatei 135, 136
 Exportdialog 268
 Extensible Markup Language → XML
 External Facing Portal 20, 66, 167
 Extranet 16

F

Farbgestaltung 94, 168, 169
 Farbumrechnung 171
 Fehlermeldung 275
 fehlende Ausführungsberechtigung 105
 Fehlerquelle 260, 357, 358
 Feinnavigation 59, 60, 72, 73, 175, 188, 198, 203
 Feld
 Benutzer 105, 141
 Rollenzuordnung 105, 141
 Filtern 260
 Firefox → Mozilla Firefox
 Firmengrenze 347
 Firmenlogo 59, 170, 192
 Firmenmotiv
 zentrieren 172
 Formular 345, 393, 396, 397
 Formularprüfung 285
 Formularwert 344
 Framework 62, 168, 174, 178, 180, 334
 Standard 174
 Fremdanwendung 144, 165
 Fremdsystem 163
 Full_Page 75, 84
 Funktionalität 358, 361

G

Generator 348, 349, 351, 356, 364, 372
 Geschäftslogik 346ff., 357, 361, 364, 370
 Geschäftsprozess 345, 348
 Get-Parameter 82
 Größenbeschränkung 372
 Gruppe 125, 127, 132, 134, 140, 168, 180
 Guest_User_Company 139
 Guest 89, 106
 Guided Procedures 35

H

Herunterladen 255, 256, 342
 Hierarchie 304
 Hilfe 359
 Hilfe-Link 175
 How-to Guide 162, 181
 HTML (Hypertext Markup Language) 246, 328, 345
 HTTP (Hypertext Transfer Protocol)
 Authentifizierung 145
 Destination 372

I

ID 64
 ändern 65
 Präfix 64
 IDE (Integrated Development Environment) 349
 Identity Management 37
 iFrame 59, 174
 Import 60, 135, 237, 258, 318
 Dialog 261
 Importbereich 59
 Importstruktur
 Benutzer 137
 Gruppe 137
 Rolle 138
 InputField 386
 Installation 25
 Interactive Form 393ff., 398
 Internet Explorer 256, 287
 IP-Adresse 345
 IPortalComponentRequest 310
 IPortalComponentResponse 310
 ITS (Internet Transaction Server) 98, 120
 Direktaufruf 99
 Web GUI 98
 IUser 359, 361, 389
 iView 57, 62, 67, 76, 277, 316, 336
 Anordnung 75
 erstellen 302
 generic 102
 masthead 175
 Navigations-iView 288
 neu 326

SAP-Transaktion 96
Sprache 89
Vorlage 77, 79

J

J2EE-Engine 31
 J2EE-Prozess 37
 J2EE-Wizard 356
 JARM (Java Application Response Time Measurement) 34
 Java 246, 349, 354
 Bibliothek 261
 java.popups 84
 JavaScript 246
 aktivieren 287
 deaktivieren 286, 287
 Java-Servlet 76
 JSP (JavaServer Page) 246, 252
 Import 312
 Seite 277, 288, 329

K

Kardinalität 384, 394
 Karteireiter-Navigation 59, 187
 Kennwortregel 222
 Keytool 164
 Klasse 354, 357, 359, 363
 AbstractPortalComponent 251, 252, 288
 Knoten 384, 394, 395
 Komponente 250, 293, 294, 319, 394
 masthead 317
 Konfiguration 351
 Konfigurationseinstellung 293, 294
 Konfigurationsparameter 277
 Konnektor 98, 118
 externer 188
 Kontextmenü 364, 384, 397, 398
 Kopfbereich 59
 Kopieren 65, 68
 Kundennamensraum 92, 93

L

Layout 73, 281, 310
 LDAP (Lightweight Directory Access Protocol) 125, 221, 228, 350
 Anbindung 228
 Gruppe 134
 Lieferant 59, 60, 345, 358, 361, 362, 370, 375, 376, 393
 Light Framework 73, 177, 249, 313
 Linktext 329
 Login
 accept_sso2_ticket 154
 create_sso2_ticket 155
 password_change_for_SSO 154
 Login-Seite 254
 Aufbau 278
 Logo 167, 225, 226
 Löschen 65

M

Mailserver 344
 Main Rule 65, 168, 178, 180, 238, 313, 315
 DANN-Bedingung 182
 WENN-Bedingung 180
 Mapped Password 82
 Mapped User 82
 Mapping 381
 Master Page 398
 Mehrsprachigkeit 254, 266
 Menüaufbau 67, 69, 130
 Menübaum 70, 198
 gemeinsamer 112
 Merge-ID 111
 Merge 84
 ID 111
 Parameter 111
 Priorität 112
 Message-driven Bean 356
 Metamodell 376
 Methode
 onActionssubmitPDF 400
 wdDoInit 391
 Model 377
 Model Proxy 380
 Modell-Context 384

Modelldaten 394
 Modellimport 378
 Modellklasse 378
 Modellkontext 381, 382, 383
 Modellstellvertreter 377
 Modul 351
 Monitoring 32
 Motiv 168, 178, 180
 Motiveditor 168, 245
 Motivvorlage 62
 Mozilla Firefox 150, 257, 288
 Web Developer 150
 MVC (Model-View-Controller) 253
 MyCompany 393, 404

N

Name 64, 84
 Namensraum 147
 Navigation 130, 175
 hierarchische 60
 rollenbasierte 57
 Navigationsbaum 288, 304
 Navigationsbereich
 unsichtbar 84
 Navigationsknoten 304
 Navigationsstruktur 58
 Aufbau 58
 Neuanlage 79
 System 117
 Nutzungsbedingung 249, 279

O

Objekt 353, 404
 Ordner 57, 62, 63ff., 279
 classes 277
 dist 276
 jsp 253
 lib 261, 277
 pagelet 253
 PORTAL-INF 259, 276
 private/classes 277
 private/lib 263, 277
 src.api 276
 src.core 276
 Struktur 276

themes 170
virtueller 66
 Originalkomponente 246

P

Package Explorer 259
 Page 62
 Paket 353, 371ff., 379
 Anlage 353
 Schnittstelle 371
 Palette 398
 Parameter 28, 285, 309, 316, 344
 auslesen 325
 category 321
 Post 144
 PAR-Datei 79, 80, 238, 248, 255, 258, 261, 276, 294, 340
 com.sap.nw.wpc.runtime 212
 com.sap.portal.runtime.logon.par 256
 importieren 81
 Liste 341
 PCD (Portal Content Directory) 333, 336
 PDF (Portable Document Format) 393, 394, 396, 400ff.
 Persistence Layer 347, 348
 Persistenz 356
 Personalisierung 73, 88, 146, 180, 293
 Benutzerkennwort 102
 Plausibilitätsprüfung 329
 Port 94, 98, 99, 100, 226, 373
 Portal Archive 248
 Portal Content 57, 61
 Portal Desktop 313
 Portalanwendung 250, 290, 339
 Portalfavorit 175
 Portalgrundkonfiguration 59
 Portalkomponente 80, 275, 339
 definieren 291
 Portalkopf 174, 249, 313, 316, 331, 334
 Aufbau 316
 Portalnavigation 19
 Portalrolle 67
 Portalserver 367

Portal-Service 340
 Portalzertifikat 154
 Post-Methode 82
 Post-Parameter 82
 Präfix 372
 Presentation Layer 346ff.
 Priorität 84
 Programmierung 360
 Projekt 365
 neu 289
 Projekteigenschaft 265
 Projektordner 65, 78
 Property 404
 Protokoll 347
 Proxy 372, 373, 377, 378, 380
 Klasse 373
 PSE-Datei 165
 Public-Key-Technologie 154

Q

Quellcode 295
 Ansicht 280
 Query-String BEx 94
 Quicklink 182, 183, 240, 334
 anonymous 107

R

Rahmenbedingung 18
 Recht 105
 Referenz 359, 360, 366, 367
 Referenzierung 283
 Regelwerk 62
 Registrierung 348
 Registry 294
 Reiseplanung 59, 60, 71
 Rekursion 305, 309
 Relaxing 149
 Remote-GUI 45
 Remote-Portal 159
 Report 373, 375
 Ressource 253, 290
 RFC (Remote Function Call) 344ff., 351, 372, 384
 Rolle 57, 62, 66, 67, 125, 127, 130, 168, 180, 187, 203, 206

- delegated_user_admin* 132, 134, 138ff.
 - Endbenutzerrolle* 132
 - eu_role* 132
 - Portalrolle* 125, 130
 - Rollenzuordnung* 141
 - SAP R/3* 130
 - super_admin_role* 132
 - user_admin_role* 132
 - RPC (Remote Procedure Call) 344
 - Runtime Error 146
- S**
-
- Sammlung
 - Stammregel* 180
 - SAP ERP 348
 - SAP GUI 96, 98, 116, 346
 - SAP Help Portal 154, 165, 166, 230
 - SAP Interactive Forms by Adobe 393
 - SAP Logon Ticket 146, 148, 157, 163
 - SAP Logon-Cookie 144
 - SAP NetWeaver 348
 - SAP NetWeaver Administrator 31
 - SAP NetWeaver Developer Studio 29, 251, 348, 349, 351, 359, 364, 365, 367, 396, 401, 402
 - SAP NetWeaver Portal 349, 350
 - Konfiguration* 33
 - Seitenaufbau* 174
 - SAP R/3 373
 - SAP Service Marketplace 163
 - SAP-Benutzerverwaltung 228
 - sapbw 117
 - SapGenPse 164
 - SAPLOGONTICKET 122
 - sapr3 117
 - SAP-System 116
 - SAP_R3_Dedicated* 117
 - SAP-Zielsystem* 154
 - SAP-Zertifikat 153
 - Schnittstelle 345ff., 369, 384
 - Parameter* 347
 - SAI_TOOLS* 371
 - SDM (Software Deployment Manager) 45
 - SDN (SAP Developer Network) 52, 149, 181
 - Security Store 160
 - Seite 57, 62, 67, 73, 84
 - Layout* 335
 - Seitengeometrie* 73
 - Selbstregistrierung 138, 139, 225
 - Self-Service 16
 - Serializable 354
 - Serverprogramm 344
 - Serversoftware 345
 - Serverumgebung 344
 - Service 250, 293, 357, 365
 - Session 403, 404
 - Session Management Alert 152
 - SessionBean 359
 - SharingReference 296
 - Sicherheitseinstellung 222
 - Sicherheitszone 217
 - Sichtbarkeit 277
 - Single Sign-on 17, 94, 117, 143, 149, 154, 157, 163
 - Fehlermeldung* 156
 - Sitemap 17, 249, 288, 294
 - Überschrift* 301
 - verlinken* 316, 333
 - Smallvendor 393
 - SMTP (Simple Mail Transfer Protocol)
 - Server* 223
 - SOA (Service-Oriented Architecture) 348
 - SOAP 347
 - Softwareentwicklung 344
 - Softwareprojekt 347
 - Sonderzeichen 267
 - Sortierpriorität 84
 - Sourcecode 357, 360
 - Spaltenanzahl 73
 - Sprache 89
 - Spracheinstellung 264
 - Stateful Session Bean 356
 - Stateless Session Bean 356
 - Stellvertreter 372, 377
 - String 82
 - Suchergebnis
 - Anzahl* 140
 - super_admin_role* 104
 - System 62, 92, 93, 94, 98ff., 116
 - Systemadministration 221
 - Systemalias 117, 122, 239
 - System-ID 147
 - Systemlandschaft 116, 144

T

Tag 253
 Meta 281
 Navigation Tag Library 300
 property 294
 Tag Library 253, 296
 Descriptor 296, 298
 einbinden 300
 Navigation 288, 300
 Technologieplattform 344
 TextView 386
 Titel 264
 Transaktion 70, 98, 99, 371
 LPCONFIG 373
 RZ11 154
 SE80 371
 SICF 99
 STRUST 157
 STRUSTSSO2 155
 SU01 229
 Transport 123, 233
 Transportmechanismus 20
 Transportpaket 234
 Trennzeichen 330
 Trusted System 143, 158
 trusteddn1 161
 trustediss1 161
 trustedsys1 161
 Trust-Konfiguration 160

U

Übersetzung 107
 Arbeitsvorratsübersetzung 109
 Übersetzungspaket 107
 Übersetzungsvorrat 107
 Übung
 praktische 69
 UDDI (Universal Description, Discovery and Integration) 348
 UME (User Management Engine)
 221, 350ff., 359, 360, 363ff., 370,
 372, 375, 377, 389, 391
 Bibliothek 366, 367
 Definition 139
 Parameter 225, 255
 ume.tpd.companies 138
 UME-API 359, 366

Unable to lookup System 122
 Unicode 82, 267
 Unterprogramm 344
 URL (Uniform Resource Locator)
 Alias 148
 Destination 372
 Domäne 163
 Encoding 230
 iView 76, 79, 81, 99, 144
 kurze 182, 304
 Parameter 81
 User Mapping 82, 144
 UserDTO 353ff., 357, 361
 UTF-8 281

V

Variable
 webpath 283
 &VARIABLE_SCREEN=X 96
 Verbindung 381, 382
 vertrauenswürdige 155
 Vererbung 67, 69, 103, 161
 Verwendungserklärung 371
 Verzeichnisdienst 348
 Verzweigung 71
 View 381
 Visual Administrator 39, 160, 225
 Voreinstellung 324
 Vorschau 280, 282, 304, 334
 Vorstand 59

W

Web Dynpro 100, 349, 375ff., 388,
 389, 393
 Java-Anwendung 80
 Webadministrations-Oberfläche 30
 Webanwendung 346
 Webbrowser 344ff.
 Formularfeld 345
 Webcontent 16, 189
 Webfilter 163
 Webkomponente 346
 Webprojekt 365
 Webserver 344, 345

Webservice 347, 348, 351, 356, 364,
 365, 367, 369, 371, 372, 375, 377,
 389, 390
Proxy 372, 373
 Willkommenseite 183
 Windows-GUI 98
 Wizard 358, 365, 397
 Workset 57, 62, 66, 67, 68, 71, 187
Zuordnung 72
 Workspace 29
 WPC (Web Page Composer) 17, 187
Artikel 193
Berechtigung 194, 217
Cross-Site Content 190, 192
externer Konnektor 210
Layout 200
Navigation 203
Navigation publizieren 206
Navigation verlinken 206
navigation.wpc 206
Publizieren 201
Rollenzuordnung 188
Site Content 192
Sites 190
UME-Actions 217
Webpage 198
Webpage einbinden 212
wpc_editor_role 188, 189

WSDL (Web Services Description Lan-
 guage) 347, 370, 372, 378

X

XML (Extensible Markup Language)
 246, 358

Z

Zeichensatz 281
 Zeichentabelle 267
 Zertifikat 147, 153, 158
Ablauf 155
übertragen 159
verify.der 147, 154, 155, 159, 164,
 165
vertrauenswürdiges 159
Zertifikatsspeicher 147, 153, 154,
 159, 164
 Zielsystem 268
 Zielsystem 70, 143, 147
Kennwortwechsel 146
 Zielverzeichnis 259, 289
 Zugriffsberechtigung 66, 336