

LÖSUNGEN

Microsoft Dynamics™ NAV

What's New:
Entwicklung von Lösungen für
Microsoft Dynamics™ NAV "6.0"

Whitepaper

November 2007

<http://www.microsoft.com/germany/dynamics/nav/default.aspx>

Inhaltsverzeichnis

Einleitung	3
Architekturänderungen in Microsoft Dynamics NAV "6.0"	4
RoleTailored Client	5
Microsoft Dynamics NAV Server.....	6
Unterstützung von Webservices	7
Entwicklungsverbesserungen in Microsoft Dynamics NAV "6.0"	9
Objektvergleich zwischen den Architekturen	9
Page Designer	11
Berichte	14
Webservices.....	15
Redesign der Funktionalitäten in Microsoft Dynamics NAV "6.0"	17
Nicht unterstützte Funktionen, Trigger und Eigenschaften.....	17
Umwandlung eines Forms in eine Page mithilfe des <i>Form Transformation Tools</i>	17
Verwendung von XMLports und Dataports	19
Dateispeicherorte für den Upload/Download	20
Automation Objects.....	20
Display-Properties in Forms und Pages	22
Stylesheets.....	22
Lang andauernde Prozesse.....	22
Unterstützung mehrerer Clients in Microsoft Dynamics NAV "6.0"	23
Codes zur Unterstützung mehrerer Clients	23
Berichte zur Unterstützung mehrerer Clients.....	24
Schritt-für-Schritt-Anleitung: Erstellen und Ändern von Pages	25
Schritt-für-Schritt-Anleitung: Erstellen eines Reports	31
Anhang	36
Formulare.....	36
Berichte	36
XMLports und Dataports	37

Einleitung

Dieses Whitepaper beschreibt die in Microsoft Dynamics™ NAV "6.0" geplanten Funktionsänderungen und richtet sich an Microsoft Dynamics NAV-Entwickler und -Vertriebsmitarbeiter, die mit bisherigen Microsoft Dynamics NAV gearbeitet haben.

Mit Microsoft Dynamics NAV "6.0" sind Sie in der Lage, Ihre bestehenden Microsoft Dynamics NAV-Investitionen zu steigern und die folgenden Vorteile zu nutzen:

- **Anpassbare, rollenbasierte Benutzeroberfläche** – Microsoft Dynamics NAV "6.0" bietet mit dem neuen Clientmodell, dem "RoleTailored Client", eine überaus intuitive Benutzeroberfläche, die maßgeschneidert an die unterschiedlichen Aufgaben oder Benutzerrollen innerhalb Ihrer Organisation angepasst werden kann. Für jede Rolle steht ein Rollencenter (Role Center) mit rollenspezifischen Daten zur Verfügung, das individuell angepasst werden kann und den Benutzer dabei unterstützt, die ihm zugeteilten Aufgaben schnell und einfach von einer zentralen Stelle aus auszuführen. Sie können z. B. das für die Verkaufsauftragsbearbeitung vorgesehene Role Center anpassen, in dem eine interaktive Übersicht von Kunden, regelmäßig durchzuführenden Aktivitäten und Vertriebsinformationen angezeigt und der direkte Zugriff auf die Microsoft Office Outlook®-E-Mails, -Aufgaben und -Kalender eines Benutzers dieser Rolle ermöglicht wird. Für Partner, Administratoren und Superuser besteht die Möglichkeit, die Benutzeroberfläche so zu modifizieren, dass spezifische Rollen abgebildet werden können. Auch können die jeweiligen Benutzer die Oberfläche jederzeit an die individuellen Bedürfnisse anpassen.
- **Dieselben Daten, unterschiedliche Clients** – Microsoft Dynamics NAV "6.0" beinhaltet das neue Page-Objekt (= Seite), deren Verwendung auf ähnliche Weise erfolgt wie bei Formularen. Jedoch kann eine Page im Gegensatz zu einem Formular auf mehreren Clients angezeigt werden, einschließlich dem RoleTailored Client, einem Webbrowser für die Anzeige von Microsoft SharePoint®-Diensten, oder andere benutzerdefinierte Clients. Auf diese Weise können die Informationen einer Website direkt auf einem Desktopclient angezeigt werden, ohne dabei Codeänderungen vornehmen zu müssen.
- **Verbessertes Berichtsdesign und optimierte Funktionalität** – Das verbesserte Berichtsdesign in Microsoft Dynamics NAV "6.0" ermöglicht Ihnen das Erstellen interaktiver Berichte, in die Diagramme, Tabellen und andere bislang in Microsoft Dynamics NAV nicht verfügbare Datenpräsentationselemente eingebunden werden können. Darüber hinaus lassen sich die Berichte wahlweise im PDF- oder XLS-Dateiformat abspeichern.
- **Geschäftsverbindungen mithilfe von Webservices** – Webservices ist ein weit verbreiteter Standard für die Integration von Anwendungen, welcher in Microsoft Dynamics NAV "6.0" unterstützt wird. Das Implementieren von Webservices ermöglicht Ihnen den externen Zugriff auf die Daten und Geschäftslogik in Microsoft Dynamics NAV auf standardisierte und sichere Weise. Zudem bieten Webservices die Möglichkeit, vom Microsoft Dynamics NAV-System aus eine Verbindung zu anderen Systemen innerhalb eines Unternehmens herzustellen.

In diesem Whitepaper werden schwerpunktmäßig die folgenden Aspekte beschrieben:

- Architekturänderungen beim Umstieg von einer 2-Tier- zu einer 3-Tier-Architektur, einschließlich des neuen RoleTailored Client und dem Microsoft Dynamics NAV Server mit Unterstützung von Webservices.
- Entwicklungsverbesserungen in Microsoft Dynamics NAV "6.0": Einführung des Page Designers, Änderungen am Object Designer sowie verbessertes Berichtswesen mithilfe von SQL Server® Reporting Services
- Redesign von Funktionalitäten früherer Versionen für den effizienten Einsatz in Microsoft Dynamics NAV "6.0".
- Schritt-für-Schritt-Anleitung für die Vorführung neuer Entwicklungsfeatures sowie zum Erstellen und Ändern von Page- und Report-Objekten
- Anhang mit einer Übersicht der in der 3-Tier-Architektur geänderten Elemente.

Architekturänderungen in Microsoft Dynamics NAV "6.0"

Frühere Microsoft Dynamics NAV-Versionen basieren auf einer 2-Tier-Architektur, bestehend aus einem Client und einem Server, wobei der Client für die Anzeige der Darstellungsebene von Formularen zuständig ist und die Geschäftslogik im C/AL-Code ausgeführt wird.

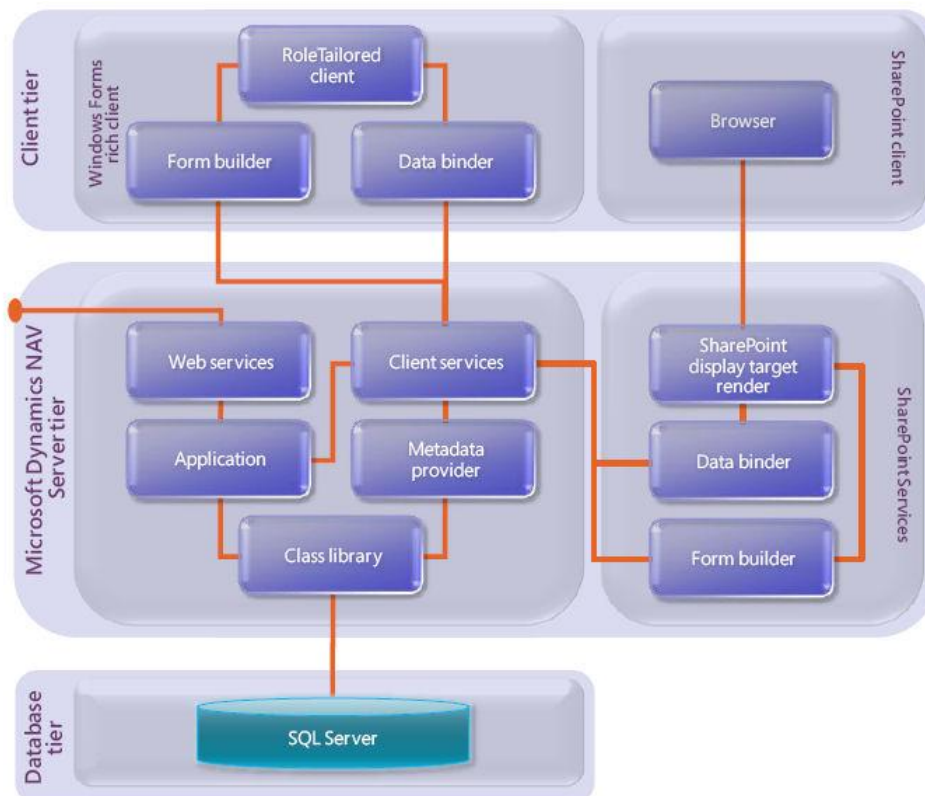
Die Microsoft Dynamics NAV "6.0"-Architektur beinhaltet ein 2-Tier-Modell, in dem der Classic Client ein Frontend zur Datenebene bereitstellt, bei dem es sich entweder um den Datenbankserver des Microsoft Dynamics NAV Classic Client oder um den SQL Server handelt.

Zur Unterstützung eines 3-Tier-Modells wurde die Architektur folgendermaßen erweitert:

- **RoleTailored Client:**
Eine Clientbenutzeroberfläche für die Bereitstellung des Frontends. Die Benutzer können die Informationen über einen webbasierten Client abrufen, der auf Microsoft SharePoint Services ausgeführt wird.
- **Microsoft Dynamics NAV Server:**
Ein Middle-Tier-Server für die Bereitstellung der Geschäftslogikebene
- **SQL Server:**
Ein Datenserver für die Bereitstellung der Datenebene

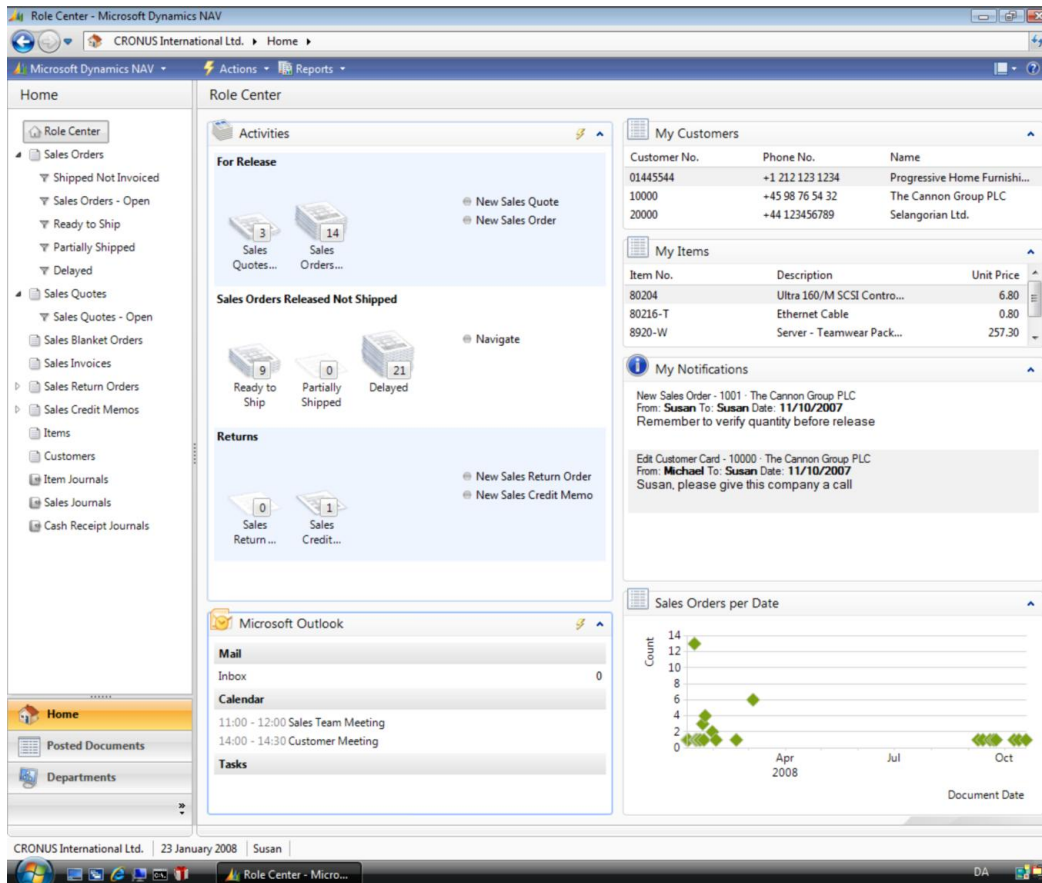
Wenn Sie den Datenbankserver für den Microsoft Dynamics NAV Classic Client verwenden, ist für die Migration zur 3-Tier-Architektur (und somit zum RoleTailored Client und dem Microsoft Dynamics NAV Server) ein Umstieg Ihres Systems auf die SQL Server Option erforderlich.

In der folgenden Abbildung sehen Sie eine Darstellung der 3-Tier-Architektur in Microsoft Dynamics NAV "6.0":



RoleTailored Client

Der RoleTailored Client, ein auf Windows-Forms basierender Client, wurde als Frontend zur Unterstützung der neuen Architektur eingeführt. Ein wesentliches Merkmal des RoleTailored Client ist die neue Benutzeroberfläche mit interaktiven, rollenbasierten und aufgabenorientierten Funktionen, die auf Basis umfangreicher Studien über Kundenunternehmen und deren Arbeitsvorgänge, Geschäftsziele und der Funktionsweise einzelner Abteilungen entwickelt wurde.



Zur Unterstützung dieser Benutzeroberfläche erfolgt die Datenanzeige im Microsoft Dynamics NAV RoleTailored Client modellgesteuert und auf Basis von Metadaten. Dabei wird die Geschäftslogik nicht mehr auf dem Client ausgeführt. Stattdessen werden zur Ausführung von C/AL-Code Anfragen an den Middle-Tier gesendet.

In Microsoft Dynamics NAV-Vorgängerversionen sowie im Microsoft Dynamics NAV "6.0" Classic Client werden die Daten mithilfe von Formularen angezeigt, z. B. über Kartenformulare (Card Forms) oder Listenformulare (List Forms). Dabei lassen sich die Forms in der WYSIWYG-Umgebung mit präzisen Layoutinformationen entwerfen.

Im neuen RoleTailored Client werden die Daten mithilfe des Page Designers entworfen und in Form von *Pages* angezeigt. Dabei werden die Elemente einer Page in einer bestimmten Reihenfolge aufgelistet. Über die Eigenschaften eines Elements lassen sich die Darstellungsfeatures festlegen, ohne spezifische Angaben zur Größe und Position eines Elements. Der Vorteil hierbei ist, dass ohne diese Layoutbeschränkungen die Pages über die verschiedenen Clients auf unterschiedliche Weise abgerufen und dargestellt werden können.

Pages stellen eine flexible Grundlage zum Erstellen verschiedener Typen von Anzeigeobjekten bereit. Während im Classic Client nur Karten- und Listenformulare unterstützt werden, sind im RoleTailored Client neue und ähnliche Page-Typen verfügbar, die das Entwickeln von Pages mit spezifischem Layout ermöglichen. Ein weiteres Feature von Pages sind die neuen Control-Typen, die zusätzliche Darstellungsformen von Systemdaten sowie die Erstellung von Shortcuts zu den Systemfeatures ermöglichen. Ebenso beinhalten Pages eine geringere Anzahl an Triggern, und zwar aus den folgenden zwei Gründen:

- Der RoleTailored Client beinhaltet optimierte Controls, die auf ihm ausgeführt werden. Aufgrund dieser neuen Controls wurde ein Teil des zuvor benötigten Codes ausgelassen.
- Pages wurden entwickelt, um die Performance zu steigern. Durch die Einführung von Pages wurde der Bedarf an Formulartriggern minimiert.

Weitere Informationen zu Pages finden Sie im Abschnitt Entwicklungsverbesserungen in Microsoft Dynamics NAV "6.0" unter [Page Designer](#).

Der RoleTailored Client bietet eine SDI-Umgebung (Single Document Interface), die der in Outlook verwendeten Umgebung ähnelt. Die Übersichtsanzeige von E-Mails, Aufgaben oder Kontakten erfolgt in der Navigationsebene, wobei die einzelnen E-Mails zum Lesen und Schreiben in einem separaten Fenster angezeigt werden.

Über die Navigationsebene des RoleTailored Client werden Übersichten von Arbeitsbereichen, so genannte *List Places*, angezeigt. Hierzu zählen z. B. die Auflistung von Debitoren oder Verkaufsaufträgen. Standardmäßig werden die Übersichten in ein und demselben Fenster angezeigt, von dem aus Sie die Möglichkeit haben, eine Übersichts- oder Kartenseite (List oder Card Page) in einem separaten Fenster aufzurufen. Neben den List Places werden in der Navigationsebene auch die *Role Center Places* (Rollencenter) und *Department Places* (Abteilungen) angezeigt.

Zum Bearbeiten und Anzeigen einer Entität wird die jeweilige Entität, z. B. ein bestimmter Debitor oder ein Verkaufsauftrag, in einem separaten Fenster geöffnet. Diese Fenster werden als Aufgabenseiten (*Task Pages*) bezeichnet, die ebenso Unterkategorien mit verschiedenen Layouts beinhalten, z. B. Kartenfenster, Belege und Buch.-Blätter.

Der RoleTailored besteht zudem aus einer kleinen Kategorie für Dialogfelder, die sich im Layout von Task Pages unterscheiden und keine Entitäten darstellen. Diese Dialogfelder dienen zur Aufforderung des Benutzers für die Eingabe von Parametern, die für die weitere Bearbeitung benötigt werden.

Microsoft Dynamics NAV Server

Der neue Microsoft Dynamics NAV Server ist eine .NET-basierte und ausschließlich unter SQL Server ausführbare Windows-Dienstanwendung. Der Server verwendet Windows Communication Framework als Kommunikationsprotokoll für die RoleTailored Clients, Internet Information Services (IIS) für die Ausführung von Webservices sowie Microsoft SharePoint-Clients, die die Verbindung mit ihm herstellen. Der Microsoft Dynamics NAV Server kann mehrere Clientanfragen parallel ausführen und über den Webservicezugriff zu authentifizierten Clients weitere Clients bedienen.

Ein wichtiger Unterschied zwischen der 2-Tier-Architektur des Classic Client und der 3-Tier-Architektur des RoleTailored Client besteht darin, dass in der 3-Tier-Umgebung die Geschäftslogik auf dem Microsoft Dynamics NAV Server ausgeführt wird und nicht auf dem Client. Ein einfaches Beispiel hierfür ist die FILE.CREATE-Funktion. In früheren Microsoft Dynamics NAV-Versionen und im Classic Client wurden bei Ausführung von Code wie diesem die Dateien auf dem Client erstellt. Beim RoleTailored Client unter dem Microsoft Dynamics NAV Server erfolgt die Erstellung der Dateien auf dem Dienst selbst. Wird die Datei einem Benutzer zur Verfügung gestellt, so sind die neuen C/AL-Dateifunktionen erforderlich, um den Download der Datei auf den Clientcomputer des Benutzers zu ermöglichen.

Der Microsoft Dynamics NAV Server stellt eine zusätzliche Sicherheitsebene zwischen den Clients und der Datenbank bereit. Auf Basis der Authentifizierungsfeatures von Windows Communications Framework wird hier eine weitere Ebene für die Benutzerauthentifizierung genutzt, um sicherzustellen, dass die Geschäftslogik in einem vom Benutzer instanziierten Prozess ausgeführt wird, der die Anforderung ausgegeben hat. Das bedeutet, dass die Autorisierung und das Protokollieren von Benutzeranfragen weiterhin pro Benutzer durchgeführt wird. Auf diese Weise wird sichergestellt, dass die Windows-Authentifizierung und alle gewährten Microsoft Dynamics NAV-Rollen und -Berechtigungen korrekt sind und weiterhin ein Auditing auf Geschäftslogikebene erfolgt.

Die neue Architektur bietet höhere Skalierbarkeit, indem mehrere Computer mit dem Microsoft Dynamics NAV Server installiert werden können, der den Zugriff auf dieselbe Datenbank ermöglicht und Sie dabei unterstützt, durch das Dedizieren unterschiedlicher Dienste für bestimmte Anwendungsbereiche oder verschiedene Clients eine optimale Systemperformance zu erzielen.

Unterstützung von Webservices

Webservices ist ein Standard für die Kommunikation zwischen unabhängigen Softwaresystemen mithilfe von Standardinternetprotokollen in einer Architektur, die auf die dynamische Programm-zu-Programm-Interaktion ausgerichtet ist. In der Webservices-Architektur können viele Arten von verteilten Systemen implementiert werden. Hierzu zählen u. a. synchrone und asynchrone Nachrichtensysteme, verteilte Computercluster, mobile Netzwerksysteme, Grid-Systeme oder Peer-to-Peer-Umgebungen. Das breite Anforderungsspektrum für die Programm-zu-Programm-Interaktion impliziert, dass die von Webservices genutzten Protokolle weitaus flexibler sind als Webprotokolle. Jedoch beruhen Webservices wie das Internet auf einer kleinen Anzahl spezifischer Protokolle, z. B. SOA (Simple Object Access Protocol).

Webservices dienen speziell zur Vereinfachung hochdynamischer Datenaustauschvorgänge, die die Basis für Geschäftstransaktionen bilden. Mithilfe standardisierter Integrationstechnologien wie Webservices können Sie durch die Unterteilung größerer Datenblöcke mithilfe proprietärer Integrationsoptionen die Effizienz Ihres Business steigern. Diese proprietären Integrationstechnologien stellen sicher, dass keine Informationen in die verschiedenen Systeme hinein- bzw. aus den Systemen hinausgelassen werden.

Für die Bereitstellung einer stabilen Entwicklungs- und Betriebsumgebung beinhalten Webservices *Metadaten*, die von allen Computern gelesen werden können. Webservice-Metadaten dienen vielerlei Zwecke: Metadaten werden zur Beschreibung der Nachrichtenaustauschformate, die von einem Webservice unterstützt werden, sowie der gültigen Nachrichtenaustauschmuster eines Dienstes. Auch werden Metadaten zur Beschreibung der Anforderungen und Optionen eines Dienstes verwendet. Web Services Description Language (WSDL), eine XML-basierte Sprache zum Definieren von Webservices, wird für die Abbildung der Austauschformate und Nachrichtenaustauschmuster genutzt.

Die Einführung des Microsoft Dynamics NAV Servers in Microsoft Dynamics NAV ermöglicht die Unterstützung von Webservices und somit die einfache Integration von Microsoft Dynamics NAV in andere Systeme. Die Webserviceintegration in Microsoft Dynamics NAV wird insbesondere durch webserviceorientierte Codeunits und Pages vereinfacht. Durch die ordnungsgemäße Authentifizierung und Autorisierung können externe Systeme Page-Daten lesen und schreiben sowie über die Definition in den allgemeinen Webserviceprotokollen Codeunits aufrufen. Die in Microsoft Dynamics NAV unterstützten Webservices ermöglichen so dem Kunden, die Vorteile einer serviceorientierten Architektur (SOA) vollständig zu nutzen.

Microsoft Dynamics NAV-Webservices bieten einen unmittelbaren Nutzen für Kunden und Partner, die die Geschäftslogik oder eine Standardoberfläche benötigen, die den externen Zugriff auf Microsoft Dynamics NAV-Daten bietet. Dabei können Sie auf Basis der gängigsten Entwicklungsumgebungen, wie z. B. Microsoft Visual Studio® 2005, Anwendungen erstellen, die Webservices unterstützen. Da Webservices zudem auf XML basiert, haben Sie die Möglichkeit, Webservices für unterschiedliche Plattformen und in mehreren Programmiersprachen zu erstellen.

In Microsoft Dynamics NAV veröffentlichte Webservices

In Microsoft Dynamics NAV können eine Reihe einfacher Webservicetypen veröffentlicht werden. Dabei entspricht die Komplexität dieser Webservices den Anforderungen der Webserviceoberfläche. Alle Webservices unterstützen die Ausführung von C/AL-Code und Validation Triggern.

- Die grundlegenden Webservicetypen wurden auf Basis des Page-Objekts entwickelt. Microsoft Dynamics NAV erstellt einen Standardwebservice mit einer festen Anzahl an Methoden. Dies entspricht im Allgemeinen dem Datenzugriff, wie z. B. beim Abrufen und Festlegen einzelner Werte oder beim Abrufen und Aktualisieren von Wertelisten.
- Ein weiterer Webservicetyp beinhaltet die Berücksichtigung von Codeunits und deren Funktionen im Webservice.
- Der dritte Webservicetyp ermöglicht die Weitergabe komplexer Datentypen mithilfe eines XMLport-Objekts als Parameter in einer Codeunit-Funktion.

Durch die Integration von Webservices in Microsoft Dynamics NAV können Sie das komplexe und manuelle Einrichten von Webservices, wie z. B. die Verwaltung der WSDL-Beschreibungen, einfach umgehen. Für die Veröffentlichung eines beliebigen Microsoft Dynamics NAV-Webservices (Page oder Codeunit) muss lediglich der jeweilige Dienst in die Webservicetabelle eingetragen werden.

Sämtliche Pages und Codeunits, die Sie in Microsoft Dynamics in die Webservicetabelle übernehmen, werden umgehend veröffentlicht und stehen sofort für Webserviceanfragen im Netzwerk zur Verfügung. Die in Microsoft Dynamics NAV integrierten Systeme, die diese Webservices abfragen, benötigen nur den Netzwerknamen (oder die Adresse) des Computers, auf dem der Microsoft Dynamics NAV Server ausgeführt wird, sowie die Namen der einzelnen Pages und Codeunits.

Beispiel: Ein Computer, auf dem der Microsoft Dynamics NAV Server ausgeführt wird, hat den Namen **NAV_Server1**. Demnach wäre der Webservice mit der Bezeichnung **MyCustomer** unter folgender URL verfügbar:

`http://NAV_Server1/NavisionServer/CRONUS_International_Ltd/Page/MyCustomer.navws`

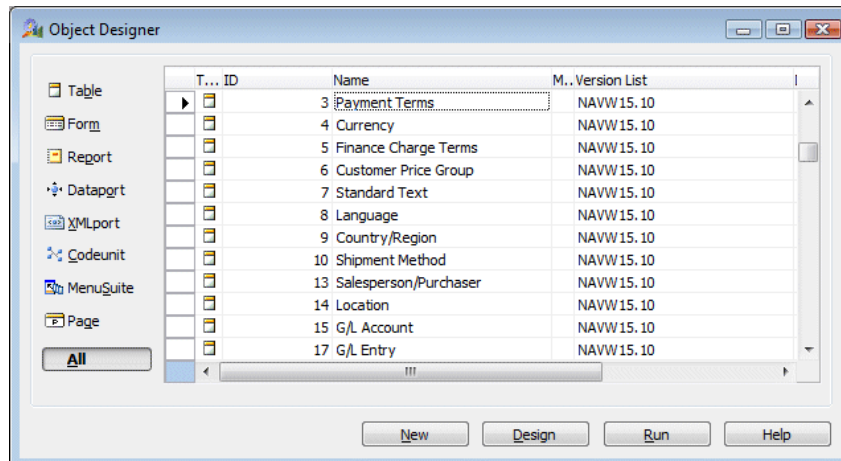
Dabei ist unbedingt zu beachten, dass Microsoft Dynamics NAV-Webserviceanfragen auf dieselbe Weise bearbeitet werden wie die Anfragen eines Endbenutzers. Dabei werden Zugriffsrechte und Autorisierungen, Überprüfungen von Dateneingaben, der Aufruf von Geschäftslogik sowie die Steuerung gleichzeitiger Datenzugriffe wie die Anfragen eines Microsoft Dynamics NAV-Clients bearbeitet. Somit wird sichergestellt, dass die Integrität der Microsoft Dynamics NAV-Daten durch die Ausführung eines Dienstes nicht gefährdet wird. Das bedeutet auch, dass für die Datenüberprüfung oder für den Aufruf von Geschäftslogik keine Codereplizierung erforderlich ist, wenn Sie Systeme für die Verwendung von Microsoft Dynamics NAV-Webservices entwickeln.

Vermeiden Sie beim Schreiben von Webservicecode sämtliche Dialogfelder und Meldungsfenster, die von einem Endbenutzer zu bestätigen sind. Der für Webservices verwendete Code unterstützt nicht die Interaktion mit dem Client, von dem aus der Code aufgerufen wurde. Daher kann der Webservice nicht auf Dialogfelder oder andere Clientinteraktionsanforderungen reagieren. Die Ausführung solcher Codes verursacht einen Ausnahmefall, der verworfen werden muss. Die Ausnahme lässt sich zwar bearbeiten, jedoch ist die vollständige Ausführung der Webservice-Aufgabe nicht möglich.

Weitere Informationen über das Entwickeln von Webservices finden Sie im folgenden Abschnitt [Entwicklungsverbesserungen in Microsoft Dynamics NAV "6.0"](#).

Entwicklungsverbesserungen in Microsoft Dynamics NAV "6.0"

In Microsoft Dynamics NAV "6.0" steht weiterhin der Object Designer für die Entwicklung von Lösungen zur Verfügung. Dabei können die neuen Features der Entwicklungsumgebung, wie z. B. der Page Designer, direkt vom Object Designer aus im Classic Client aufgerufen werden.



Objektvergleich zwischen den Architekturen

In der folgenden Tabelle sehen Sie einen Vergleich der Objekte zwischen dem Classic und RoleTailored Client.

Objekt	Classic Client	RoleTailored Client
Codeunit	Wie in früheren Microsoft Dynamics NAV-Versionen	Entwurf und Ausführung wie im Classic Client. Einführung neuer Funktionen für die Abbildung von Dateien und Daten zwischen dem RoleTailored Client und dem Microsoft Dynamics NAV Server. Ausführung von Code auf dem Microsoft Dynamics NAV Server. Für alle Anwendungen, die den Zugriff auf COM-Objekte über die Clients benötigen, ist ein Redesign erforderlich.
Dataport	Wie in früheren Microsoft Dynamics NAV-Versionen	Nicht unterstützt
Form	Wie in früheren Microsoft Dynamics NAV-Version. Der für Formulare geschriebene Code wird weiterhin ausgeführt. Ausführung von Code und Triggern auf dem Client.	Nicht unterstützt. Ersetzt durch Pages. C/AL-Codereferenzen auf Forms werden in denselben Methodenaufruf eines Page-Objekts mit derselben ID übersetzt.

MenuSuite	Wie in früheren Microsoft Dynamics NAV-Versionen	<p>Navigation erfolgt auf Page-Basis, und nicht auf MenuSuite-Basis. Für die Navigation zu einer anderen Page müssen in einer Page Aktionen hinzugefügt werden, mit denen die neue Page geöffnet wird. Das Actions-Menü ähnelt den Menüschaftflächen, die Verknüpfungen zu neuen Forms herstellen.</p> <p>Es besteht auch die Möglichkeit, über den Departments-Bereich auf das herkömmliche MenuSuite zuzugreifen. Dabei wird zunächst eine Übersicht (und keine Karte) angezeigt, damit der Benutzer einen bestimmten Eintrag schneller finden kann.</p>
Page	Nicht unterstützt	<p>Neuer Objekttyp mit Eigenschaften und Methoden, die den Form-Objekttyp ersetzen.</p> <p>Ausführung von Code und Triggern auf dem Microsoft Dynamics NAV Server.</p>
Report	<p>Wie in früheren Microsoft Dynamics NAV-Versionen.</p> <p>Reports verfügen über drei neue Properties, die ignoriert werden.</p>	<p>Report Datasets, Sections und Request Forms oder Pages werden mithilfe des C/SIDE Report Designers entworfen.</p> <p>Das Berichtslayout erfolgt über einen Report Definition Language-Editor, z. B. Visual Studio, wobei die Layoutinformationen in den neuen Eigenschaften des Report-Objekts gespeichert werden.</p>
Table	<p>Wie in früheren Microsoft Dynamics NAV-Versionen.</p> <p>Einführung der neuen ExtendedDataType-Eigenschaft hat keinen Einfluss auf das Tabellenverhalten.</p> <p>Ausführung von Code und Triggern auf dem Client.</p>	<p>Entwurf und Ausführung wie im Classic Client unter Verwendung derselben Trigger und Eigenschaften.</p> <p>Die neue ExtendedDataType-Eigenschaft ermöglicht das Hinzufügen von Metadaten in Feldern, um ein anderes Rendering zu ermöglichen.</p> <p>Ausführung von Code und Triggern auf dem Microsoft Dynamics NAV Server.</p>
Webservice	Nicht unterstützt	<p>Neuer Objekttyp zum Erstellen und Wiederverwenden einer Page bzw. eines Page-Objekts zusammen mit einem Codeunit-Objekt.</p> <p>Verwendung von XMLports zum Versenden komplexer Daten via Webservice, indem das XMLport in einen Codeunit-Parameter umgewandelt wird, in der diese Codeunit in einem Webservice enthalten ist.</p> <p>Veröffentlichung durch Hinzufügen öffentlicher Referenzen in die Webservicetabelle.</p>
XMLport	<p>Wie in früheren Microsoft Dynamics NAV-Versionen.</p> <p>Funktion für die Verwendung von XMLports in Form von Dataports nicht verfügbar.</p>	<p>Entwurf und Ausführung wie im Classic Client.</p> <p>XMLports wurden auf Dataports (mit derselben Funktionalität) erweitert, um den Import und Export strukturierter Dateien zu ermöglichen.</p>

Page Designer

Im RoleTailored Client werden die Daten in Form von Pages angezeigt, die mithilfe des Page Designers entworfen werden. In Microsoft Dynamics NAV "6.0" sind die folgenden Page-Arten enthalten:

Pages in der Navigationsebene

- *Role Center Page* – Diese Page-Art wird zum Erstellen der Startseite verwendet, die dem Benutzer beim Anmelden in der Anwendung angezeigt wird. Für jeden Benutzer gibt es eine eigene Role Center-Definition, wobei mehrere Benutzer ein und demselben Benutzerprofil zugewiesen werden können. Diese Page-Art besteht aus einer Anzahl von FactBoxes. Dies sind kleinere Fenster mit Informationen zum aktuell geöffneten Hauptfenster. Dabei kann der Benutzer je nach Bedarf einzelne FactBoxes auf seiner Role Center-Page ein- oder ausblenden.
- *List Page* – Diese Page-Art entspricht dem Übersichtsformular im Classic Client. Darüber hinaus beinhaltet eine List Page oben eine *Action Pane* mit häufig verwendeten Befehlen, einen Filterbereich in der oberen Page-Section sowie einen FactBox-Bereich auf der rechten Seite.
- *Department Page* – Dies ist eine systemgenerierte Page-Art. Das System erstellt auf Basis der Anwendungsinhalte, die Sie über die MenuSuite definieren, die Department-Pages.

Pages in einzelnen Fenstern

- *Task Page* – Diese Page-Art beinhaltet alle Aufgaben, die Bestandteil eines Geschäftsprozesses sind, die ein Benutzer auszuführen hat. Auf der *Task Page* befindet sich die Action Pane, in der die auf Basis eines bestimmten Kontextes am häufigsten verwendeten Befehle enthalten sind.
 - *Karten* – Pages mit einer oder mehreren FastTabs oder Abschnitten einer Page, deren Anzeige erweitert oder minimiert werden kann. Karten beinhalten Detail- und Statistikinformationen, z. B. die **Debitoren- oder Artikelkarte**.
 - *Belege* – Belege mit einer oder mehreren FastTabs für den Belegkopf und einer FastTab für die Belegzeilen (Beispiel: Verkaufsaufträge und Gutschriften sind Task Pages für Belege, in denen FastTabs verwendet werden.)
 - *Buch.-Blätter/Vorschlagsblätter* – Übersichten, in denen entwurfswise Zeilen erstellt werden, über die dann, häufig mit spezifischen Detailinformationen, Daten ermittelt/berechnet werden, z. B. **VK-Preisvorschlag** oder **Artikel Buch.-Blatt**
 - *List Task Pages* – Automatische Wiederverwendung der Page-Definition für ein List Place, das zum Rendern der Page als Popup-Liste im Anzeige-, Editier- und Auswahlmodus verwendet wird. Dabei wird ein Wert in das Fenster übernommen, von dem aus der Abruf erfolgt. Die *List Task Page* besteht lediglich aus einer Übersicht. FastTabs mit Feldern sind hier nicht vorhanden.
 - *List Plus* – Pages, die eine FastTab mit einer Übersicht und zusätzlichen FastTabs mit Feldern oder weiteren Übersichten beinhalten. Diese Page stellt keine Entität dar, wird jedoch für die benutzerdefinierte Anzeige oder für die Bearbeitung von Informationen aus mehreren Entitäten, z. B. **Vorschlags Buch.-Blätter** oder **Statistikformulare**, verwendet.
 - *Matrizen* – Eine Page mit einem Grid, in dem jede Zelle (keine Zeile) eine Entität oder eine Summe von Entitäten darstellt.
 - *Assistenten* – Eine Gruppe zugehöriger Pages, die den Benutzer bei der Durchführung einer Aufgabe unterstützen.

- *Dialogfelder* – Mit dieser Part-Art werden Pages angezeigt, die den Benutzer dazu auffordern, bestimmte Parameter für die weitere Bearbeitung anzugeben. Diese Pages stellen keine Entität dar.
- *Bestätigungdialogfelder* – Pages, die die kurze Bestätigung, z. B. durch *Ja* oder *Nein*, für die weitere Bearbeitung erfordern. In der Regel beinhaltet diese Page einen Detailbereich mit weiteren Informationen zu den Auswahloptionen.
- *Anforderungsfenster (Request Forms)* – Pages mit einer oder mehreren FastTabs, jede mit einem Bereich zum Filtern einer spezifischen Art oder Entität. Diese Pages, die automatisch auf Basis der in Berichten/Stapelverarbeitungen festgelegten Definitionen generiert werden, beinhalten ebenfalls eine FastTab im oberen Bereich, die Optionen und Präferenzen für die Bearbeitung bereithält.

In anderen Page-Arten integrierte Component Page-Arten

- *FactBoxes* – Diese Page-Art stellt ein herkömmliches Formular in Kleinformat dar, in dem ergänzende Informationen enthalten sind. Eine FactBox übernimmt als Eingabe einen von der Hostpage gesendeten Schlüsselwert für den Abruf und die Anzeige von Informationen, die mit dem Schlüsselwert verbunden sind. Beispielsweise übernimmt die FactBox *Customer Statistics* (Debitorenstatistik) in einem Verkaufsauftrag eine Debitorennummer als Eingabe für die Anzeige der Debitorensalden, ausstehender Beträge und anderer Daten.
- *Role Center Parts* – Diese Page-Art ähnelt einer FactBox. Der einzige Unterschied liegt darin, dass ein *Role Center Part* keine Eingaben für Abfragen, sondern immer dieselbe Abfrage verwendet und eine konstante Anzahl an Einheiten in einer Übersicht anzeigt. Eine Top 10-Liste von Artikeln ist z. B. ein Role Center Part. Die in der Top 10-Liste enthaltenen Artikel können von Zeit zu Zeit variieren; die Anzahl der bestverkauften Artikel (10) bleibt jedoch immer gleich.

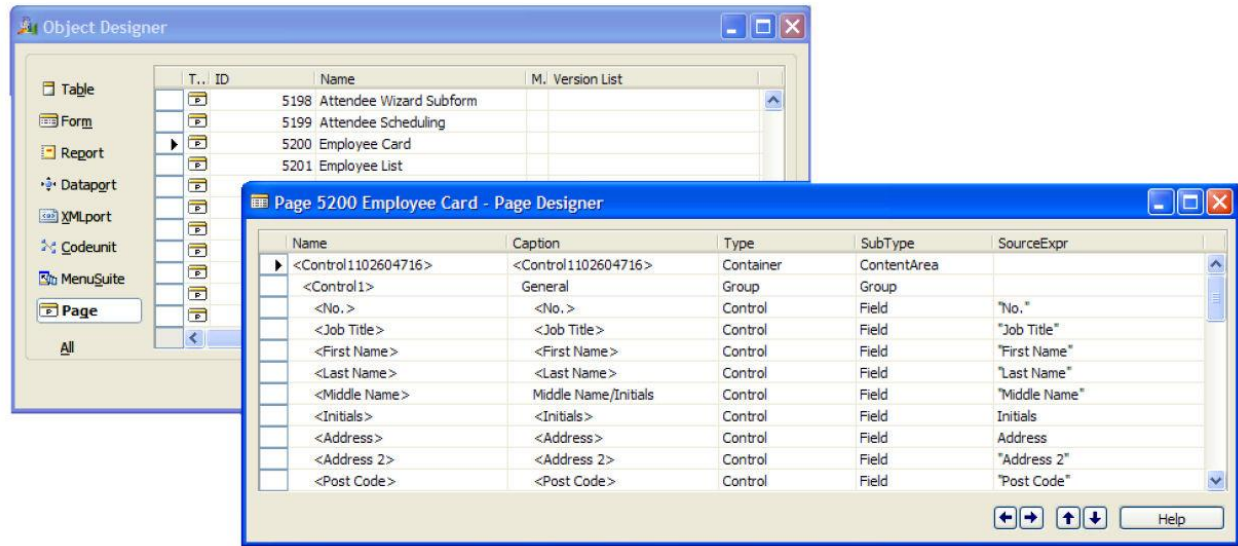
Entwerfen von Pages mit dem Page Designer

Die neuen Page-Layouts ermöglichen Ihnen das Erstellen von Pages, die dem Benutzer automatisch die die am häufigsten verwendeten Daten und Aktionen zur Verfügung stellen. Diese effektive, überaus einfach gestaltete Benutzeroberfläche ermöglicht selbst einem Neuanwender, die Page schnell zu erlernen und zu bedienen.

Beim Entwerfen einer Page sollte als Erstes geklärt werden, welche Benutzer dieses Fenster nutzen und welche Aufgaben damit ausgeführt werden. Erstellen Sie hierzu eine Liste der Felder, Befehle und Links, die für Aufgabe dieses Fensters benötigt werden, und wählen Sie die Page-Art aus, die die Angaben dieser Liste optimal erfüllt. Die Quelltablette dieser Page sollte die Tabelle sein, die die in der Page angezeigten Primärdaten enthält.

Danach muss festgelegt werden, ob ergänzende Informationen in Form von FactBoxes benötigt werden. Hierzu können bei Bedarf bereits bestehende FactBoxes verwendet werden.

Nachdem Sie das Page-Design festgelegt haben, können Sie sie mithilfe des Page Designers erstellen. Hierbei müssen Sie eine Hierarchie von Page-Elementen angeben. Die Anzeige der Daten kann dabei durch entsprechende Anpassung der Eigenschaften eines jeden Pages-Elements definiert werden.



Es ist kein statisches Layout oder eine Bearbeitung auf Pixelebene erforderlich, da Sie lediglich den anzuzeigenden Inhalt und die Darstellungsweise festlegen müssen. Die zentrale Aufgabe beim Page-Design liegt darin, die Page-Inhalte in Beziehung zu den verschiedenen Page-Arten in Microsoft Dynamics NAV zu setzen.

Anschließend kann die Benutzeroberfläche mithilfe folgender Designfeatures optimiert und vereinfacht werden:

- Verwenden Sie in der Action Pane häufig verwendete Befehle. Erstellen Sie in der Action Pane Gruppen von Befehlen, über die verschiedene Aufgaben oder der Status der Befehle vorgeschlagen wird. Legen Sie für jede Gruppe eine logische Reihenfolge fest. Verwenden Sie große Symbole für Standardbefehle und kleine für optionale und selten verwendete Befehle.
- Legen Sie in jeder FastTab die **Importance**-Eigenschaft auf *Additional* für alle Felder fest, die der Benutzer bei Ausführung einer bestimmten Aktion weniger als drei Mal verwendet. Für wichtige Felder (1 bis 4) legen Sie die Eigenschaft auf *Promoted* fest, um die Werte dieser Felder direkt über die FastTab-Zusammenfassung anzuzeigen, ohne dabei die FastTab zu öffnen.
- Legen Sie standardmäßig die Anzeige von 1 bis 3 FactBoxes fest (in der Regel 1 bis 2 FactBoxes oder Notizen).

Anschließend übernehmen Sie die neue Page zur Berücksichtigung im Departments-Place in die MenuSuite. Handelt es sich bei der erstellten Page um ein List Place, dann sollten Sie beachten, ob Sie für spezifische Benutzerprofile im rollenorientierten Navigationsbereich einen Link zur Page bereitstellen. Handelt es sich bei der neuen Page um eine Task Page, dann muss überlegt werden, von welchen List Places aus die Links hergestellt und hingefügt werden können.

Der Endbenutzer hat die Möglichkeit, die Anzeige einer Page seinen Bedürfnissen entsprechend anzupassen. Er kann z. B. in List Pages Spalten ein- oder ausblenden und die anzuzeigenden FactBoxes festlegen. Ebenso kann im jeweiligen Role Center oder auf einer bestimmten Page die Reihenfolge, in der die FactBoxes angezeigt werden, geändert bzw. nicht relevante FactBoxes können einfach ausgeblendet werden.

Über die Personalisierungsfeatures ist der Zugriff auf sämtliche Systembestandteile, für die ein Benutzer keine Zugriffsberechtigungen hat, nicht möglich. Jedoch unterstützen diese Features das Ausblenden von nicht gewünschten Page-Elementen oder das Einblenden jener Elemente, die standardmäßig nicht zu sehen ist.

Beim Entwerfen einer Page sollten Sie eine Liste mit allen Page-Elementen bereitstellen. Beachten Sie dabei, dass von einem Benutzer viel häufiger bestehende Elemente entfernt als neue hinzugefügt werden.

Eine detaillierte Beschreibung zum Erstellen und Ändern von Pages in Microsoft Dynamics NAV "6.0" finden Sie im Abschnitt [Schritt-für-Schritt-Anleitung: Erstellen und Ändern von Pages](#).

Berichte

Die im RoleTailored Client enthaltenen Berichte basieren auf dem SQL Server Reporting Services-Format und stellen Verbesserungen der bisher in C/SIDE verfügbaren Berichterstellung dar. Zu den neuen Berichtsfeatures zählen:

- Grafische und visuelle Verbesserungen, z. B. Bilder, Diagramme, Kurven und benutzerdefinierte Darstellungsobjekte.
- Interaktive Features zum:
 - Erweitern bzw. Verkleinern zum Ein- bzw. Ausblenden von Detailinformationen.
 - Dynamischen Ändern der Sortierreihenfolge einzelner oder mehrerer Spalten.
- Umfangreiche Berechnungen.
- Export nach Microsoft Office Excel® sowie PDF-Funktionalität.

Die vom RoleTailored Client benötigten Zusatzinformationen werden im RD-Format (Report Definition Language) sowie in zusätzlichen Eigenschaften im Report-Objekt gespeichert. RDL ist ein SQL Server Reporting-Standardformat für Berichte. Dabei können eine Reihe von Berichtseditoren verwendet werden, wie z. B. Microsoft Visual Web Developer™ Express Edition, welches kostenfrei zum Download zur Verfügung steht.

Zum Erstellen eines Reports im RoleTailored Client werden die Datenelemente des Reports definiert, der anschließend in Visual Studio in ein SQL Server Reporting Services-Projekt exportiert wird. SQL Server Reporting Services wird verwendet, um das Layout des Berichts zu entwerfen und sämtliche Nachbearbeitungen, wie z. B. Gruppierungen und Summierungen, vorzunehmen. Die Übernahme bestehender Reports aus Vorgängerversionen von Microsoft Dynamics NAV bzw. vom Classic Client in den RoleTailored Client und das Entwerfen von neuen Reports erfolgt auf die gleiche Art und Weise.

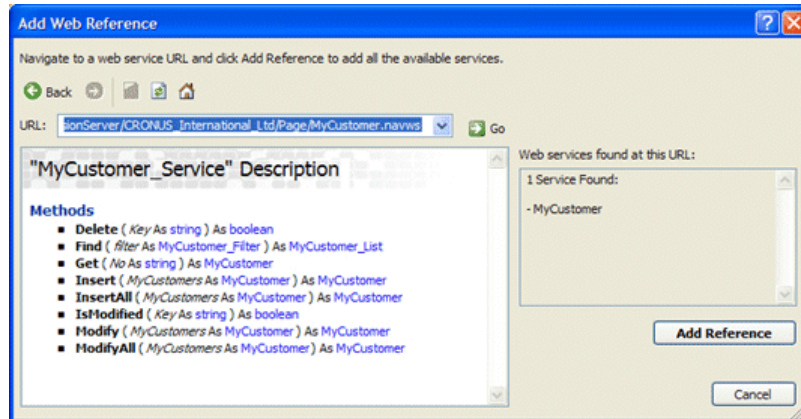
Ein Teil bestehender Report-Eigenschaften werden in der neuen Architektur nicht unterstützt. Im Allgemeinen handelt es sich hierbei um Eigenschaften für den Berichtlayout, für Gruppierungen, Summierungen und Druckerspezifikationen. Beim Öffnen von Visual Studio werden diese Eigenschaften ignoriert und müssen daher vom Layoutdesigner aus zugewiesen werden. Die Zuweisung muss nur einmal vorgenommen werden. Sämtliche Änderungen, die Sie an den Reports in Visual Studio vornehmen, werden beim Speichern des Reports automatisch übernommen.

Eine detaillierte Beschreibung zum Erstellen von Reports in Microsoft Dynamics NAV "6.0" finden Sie im Abschnitt [Schritt-für-Schritt-Anleitung: Erstellen eines Reports](#).

Webservices

Page-basierte Webservices

Beim Veröffentlichen einer Page als Webservice stehen acht verschiedene Methoden zur Verfügung, mit denen sich allgemeine Datensatzvorgänge, wie z. B. Einfügen, Ändern und Suchen, verwalten lassen.

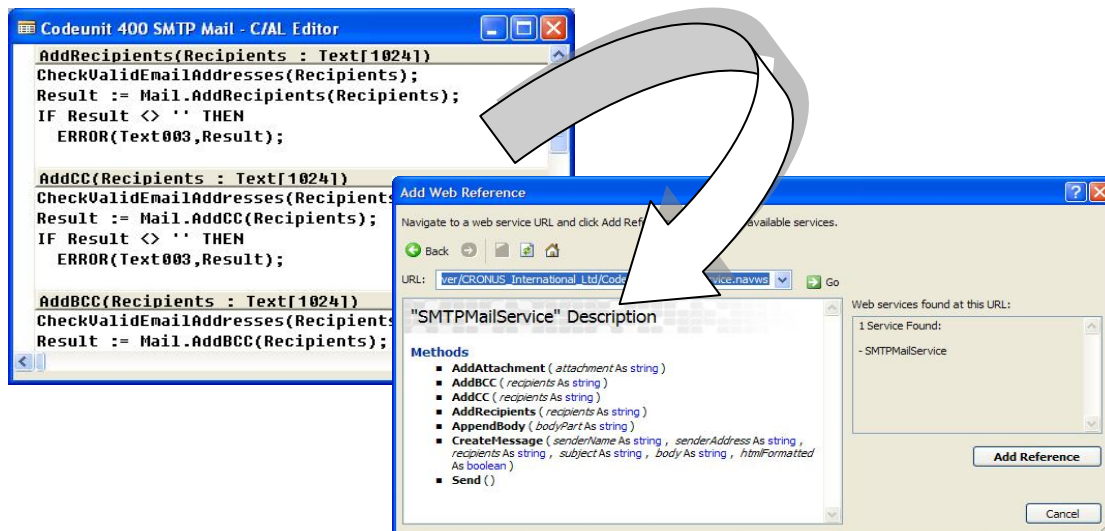


Für die Erstellung eines einfachen Webservices auf Basis einer Page in Microsoft Dynamics NAV "6.0" müssen Sie ein Page-Objekt erstellen und dieses in die **Web Service**-Tabelle übernehmen. In der folgenden Abbildung sehen Sie den Page-basierten Webservice **MyCustomer**.

Object Type	Service Name	Object ID	Published
Codeunit	SMTPMailService	400	✓
Page	MyCustomer	21	✓

Codeunit-basierte Webservices

Beim Veröffentlichen einer Codeunit als Webservice stehen alle in der Codeunit definierten Methoden für die Integration in die Codeunit zur Verfügung.

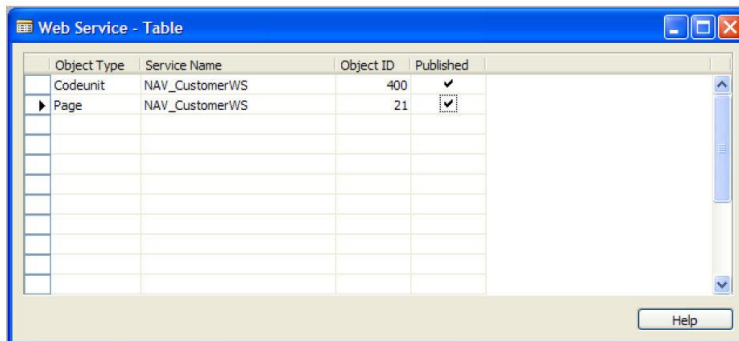


Für die Erstellung eines einfachen Webservices auf Basis einer Codeunit in Microsoft Dynamics NAV "6.0" müssen Sie ein Codeunit-Objekt erstellen und dieses in die **Web Service**-Tabelle übernehmen.

Webservices auf Basis von Pages und Codeunits

Bei einem Page-Objekt haben Sie die Möglichkeit, durch die Zuordnung von Codeunits weitere Methoden für den Webservice zu übernehmen. Beim Veröffentlichen einer Page und einer Codeunit als Webservice stehen acht Methoden auf der Page sowie zusätzliche Methoden zur Verfügung.

Für die Erstellung eines Webservices auf Basis einer Page und einer Codeunit müssen Sie ein Page-Objekt sowie eine Codeunit erstellen und diese in die **Web Service**-Tabelle übernehmen. Anschließend nehmen Sie die Zuordnung beider Objekte vor. Geben Sie hierbei für beide Objekte denselben Dienstnamen an.



Object Type	Service Name	Object ID	Published
Codeunit	NAV_CustomerWS	400	<input checked="" type="checkbox"/>
Page	NAV_CustomerWS	21	<input checked="" type="checkbox"/>

Webservices auf Basis komplexer Codeunits

Ebenso können Webservices mit komplexen Parametern entwickelt werden. Komplexe Parameter können im Gegensatz zu einfachen Parametern wie *Integer* oder *Boolean* eine eigene Struktur haben. Ein Verkaufsauftrag stellt z. B. ein komplexes Objekt dar, da es aus Kopf- und Zeileninformationen besteht, wobei jede Zeile unterschiedliche Detailangaben enthalten kann.

Die Übernahme komplexer Informationen in einen Webservice erfolgt mithilfe eines XMLports als Parameter für Codeunit-Funktionen. Dabei ist es ratsam, nicht alle Felder eines Page-Objekts zu verwenden. Stattdessen könnten Sie einen Webservice veröffentlichen, der den Zugriff auf eine kleinere Gruppe von Feldern auf der Page ermöglichen. Hierzu müssen Sie ein XMLport erstellen, in dem das benötigte Feld und das XMLport als Parameter für die Funktion einer Codeunit enthalten sind. Anschließend wird die Codeunit mit einer Page veröffentlicht; und über die Webserviceoberfläche kann dann auf eine Funktion mithilfe des XMLports in Form eines Parameters zugegriffen werden. Der Abruf der C/AL-Codewerte ist weiterhin möglich, da beim Aufruf der Validate Trigger auf die Geschäftslogik zugegriffen wird.

Redesign der Funktionalitäten in Microsoft Dynamics NAV "6.0"

Zur Unterstützung der 3-Tier-Architektur mit dem RoleTailored Client und dem Microsoft Dynamics NAV Server musste für einen bestimmten Anteil an Funktionalitäten in Microsoft Dynamics NAV "6.0" ein Redesign durchgeführt werden.

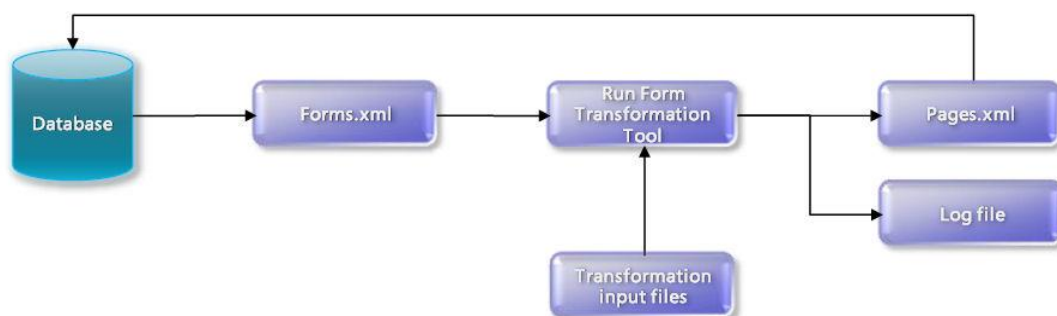
Nicht unterstützte Funktionen, Trigger und Eigenschaften

Funktionen, Trigger oder Eigenschaften, die in der Servicearchitektur nicht unterstützt werden, erzeugen zwar keine Fehler, führen aber zu keinem Ergebnis. Mit dem *Form Transformation Tool* können diese nicht unterstützten Trigger und Eigenschaften entfernt werden. Sämtliche Löschvorgänge werden dabei in einer LOG-Datei protokolliert.

Umwandlung eines Forms in eine Page mithilfe des *Form Transformation Tools*

Um ein Form in eine Page zu überführen, wird in Microsoft Dynamics NAV "6.0" das *Form Transformation Tool*, ein Art Mappingtool, zur Verfügung gestellt, mit dem Sie auf Basis der in Ihrer derzeitigen Anwendung enthaltenen Forms Pages generieren können. Das Tool basiert auf strukturierten Regeln, die speziell für das Umwandeln von Forms in Pages entwickelt wurden. Dabei werden die Pages mithilfe von Controls und Properties generiert, die bestimmtem neuen Verhalten im RoleTailored Client entsprechen, wobei ein Großteil der neu erstellten Page-Objekte aus derselben Funktionalität wie die in herkömmlichen Formularen besteht.

Das Tool erstellt für jedes Formular Ihrer Anwendung ein neues Page-Objekt, indem die Controls und Properties auf jedem Formular analysiert und eine Page mit übereinstimmenden Controls und Properties erstellt wird. Während des Umwandlungsvorgangs werden die bestehenden Formulare weder geändert noch gelöscht.



Als Erstes wird jedes Formular in eine der in der 3-Tier-Architektur verfügbaren Page-Arten überführt. Für jedes umzuwandelnde Formular muss festgelegt werden, ob das Formular ein Kartenfenster, eine Übersicht, ein Beleg, eine FactBox usw. darstellen soll.

ID	Name	FormType	PageType	CardFormID	Ignore	FormReplacedBy	DestinationName	Comment
1	Company Information	Card-Editable	Card					
4	Payment Terms	List-Editable	List					
5	Currencies	List-Editable	List	495				
6	Finance Charge Terms	List-Editable	List	494				
7	Customer Price Groups	List-Editable	List					
8	Standard Text Codes	List-Editable	List					
9	Languages	List-Editable	List					
10	Countries/Regions	List-Editable	List					
11	Shipment Methods	List-Editable	List					
14	Salespeople/Purchasers	List-Editable	List	5116				
15	Location List	List-NonEditable	List	5703				
16	Chart of Accounts	Chart of Accounts	List	17				
17	G/L Account Card	Card-Editable	Card					
18	G/L Account List	Chart of Accounts	List	17				
20	General Ledger Entries	List-NonEditable	List					
21	Customer Card	Card-Editable	Card					
22	Customer List	List-NonEditable	List	21				
23	Cust. Invoice Discounts	List-Editable	List					
25	Customer Ledger Entries	List-Editable	List					
26	Vendor Card	Card-Editable	Card					
27	Vendor List	List-NonEditable	List	26				
28	Vend. Invoice Discounts	List-Editable	List					
29	Vendor Ledger Entries	List-Editable	List					

Anschließend wird das Tool ausgeführt, in dem genau festgelegt werden kann, welche Forms berücksichtigt und welche Forms ignoriert werden sollen.

Dabei behält das Tool beim Erstellen einer Page aus einem Form jede Art von Code. Besteht ein Form aus einem Trigger, der Code beinhaltet, und es sind keine entsprechenden Trigger auf der Page vorhanden, wird der Code entweder in einen ähnlichen Trigger der Page übernommen oder in die LOG-Datei übertragen. Anhand der hier protokollierten Informationen können Sie für die nicht umgewandelten Forms entscheiden, ob diese weiter benötigt werden, und falls ja, ist ein Redesign erforderlich, um diese Forms erfolgreich in Pages umzuwandeln. Alternativ hierzu könnten Sie alle bzw. einen Teil dieser Forms ignorieren und neue Pages mit derselben Funktionalität mithilfe des Page Designers erstellen.

Ob ein Form mit dem Tool erfolgreich überführt werden kann, hängt davon ab, welchen Umfang und welche Art von Anpassungen Sie an einer Form vorgenommen haben. Entsprechen die Anpassungen dem Design der Standardanwendung, dann gestaltet sich der Umwandlungsprozess einfacher als bei komplexeren Anpassungen.

Auch wurde ein bestimmter Anteil an Forms in Microsoft Dynamics NAV "6.0" bereits so angepasst, dass mit dem Tool eine Page mit derselben Funktionalität eines Formulars automatisch erstellt werden kann. Allerdings kann die erneute Anpassung bestimmter Formulare erforderlich sein, um die erfolgreiche Umwandlung sicherzustellen.

In den folgenden Abschnitten finden Sie eine Beschreibung einiger spezifischer Formularänderungen in Microsoft Dynamics NAV "6.0".

Statistikformulare

Statistikformulare mit mehr als zwei Spalten müssen angepasst werden, um sie fehlerfrei in eine Page umzuwandeln. Hierzu müssen die Spalten in einer Matrix gruppiert und die Zellen in der Page gefüllt werden. Ein Platzhalter-Control muss hinzugefügt werden, um die Zellen zu leeren und die Leerzeichen entsprechend zu füllen.

Des Weiteren kann die Umgestaltung einiger Statistikformulare in Berichte nützlich sein, da Statistikformulare im Allgemeinen für die Datenanalyse verwendet werden.

Formulare mit Information Frames oder Filter Frames

Formulare mit Information Frames, wie z. B. das Fenster **Artikel Zu-/Abschl.-Zuw.** oder das **Zlg.-Eing. Buch.-Blatt**, müssen so wie Statistikformulare angepasst werden. Für die erfolgreiche Umwandlung in eine Page muss der Info Frame-Bereich in ein neues Frame gesetzt werden.

Werden diese Änderungen nicht vorgenommen, wird das Standardlayout des RoleTailored Client mit zwei Spalten verwendet. In einigen Fällen kann es jedoch erforderlich sein, dass Sie die Felder neu anordnen, um eine benutzerfreundliche Reihenfolge der Felder im RoleTailored Client aufzustellen.

Matrixformulare

Im RoleTailored Client werden keine Matrix-Controls unterstützt. Sie können ein Matrix-Control eines Forms auf ein Grid-Control einer Page zuordnen, jedoch sind hier einige Einschränkungen zu beachten:

- Eine neue Anforderungsseite (Request Page) muss erstellt werden, um die anzuwendenden Filter und Auswahloptionen zu definieren, sowie eine Schaltfläche, unter der das Ergebnis im Grid angezeigt wird.
- Das Grid kann nur gelesen werden. Grids und Matrixformulare, die im RoleTailored Client für die Dateneingabe verwendet werden, können nicht erstellt werden.
- Das Grid unterstützt eine begrenzte Anzahl an Spalten. Der Standardwert kann bei Bedarf geändert werden, wenn ein größeres Grid benötigt wird, ist für diese Page aber immer statisch.

Buch.-Blätter und Vorschlags Buch.-Blätter mit Stapelverarbeitungen

Der Zugriff auf Buch.-Blätter und Vorschlags Buch.-Blätter erfolgt nun über ein List Place, in dem die Stapelverarbeitungen aufgeführt werden. Hierzu müssen Sie im Stapelverarbeitungsformular den Befehl zum Bearbeiten des Buch.-Blattes hinzufügen.

Die Navigationsbereiche in den jeweiligen Role Centern sollten ebenfalls auf diese List Places referieren.

Typspezifisches Filtern auf Formularen

Im RoleTailored Client wird das Filtern im Department Place nicht unterstützt (auch bekannt als die MenuSuite). Im Classic Client können Sie z. B. einen Filter auf die Übersicht von Verkaufsangeboten setzen. Anschließend wird von der MenuSuite aus nur die Übersicht von Verkaufsangeboten angezeigt. Da diese Funktionalität im RoleTailored Client nicht vorhanden ist, müssen Sie für jeden Filterwert eine neue Page erstellen. In diesem Beispiel würde dann die Verkaufsübersicht aus sechs neuen Übersichten bestehen, die die verschiedenen Belegarten wiedergeben.

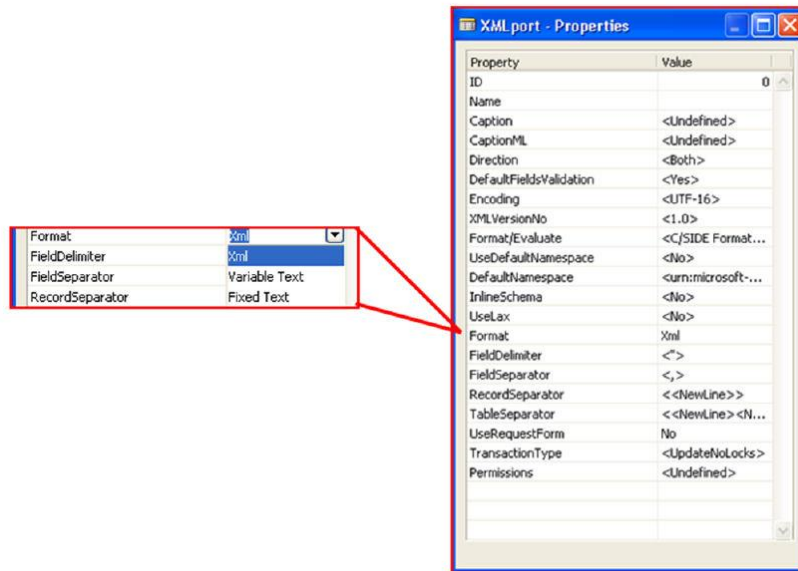
Verwendung von XMLports und Dataports

In Vorgängerversionen von Microsoft Dynamics NAV unter dem Classic Client können Sie mithilfe von XMLports und Dataports XML-Dokumente und Textdateien erstellen (kommaseparierte, zeichenbegrenzte Dateien mit fester Breite).

Im RoleTailored Client unter dem Microsoft Dynamics NAV Server wurden beide Objekttypen im XMLport-Objekttyp zusammengeführt, das dieselbe Funktionalität beinhaltet.

XMLports lassen sich mithilfe des C/SIDE XMLport-Designers für beide Architekturen entwickeln. Für den Microsoft Dynamics NAV Server können Sie mithilfe zusätzlicher Eigenschaften den Zugriff auf zusätzliche Funktionalität erhalten. Request Pages sowie die Import-/Exporttextfunktionalitäten wurden ebenfalls in XMLports übernommen. Diese Eigenschaften können im Classic Client über den XMLport-Designer aufgerufen werden. Jedoch wird das Erstellen und Ausführen eines XMLports mit diesen Eigenschaften im Classic Client nicht unterstützt.

In der folgenden Abbildung sehen Sie die im XMLport Designer hinzugefügten Elemente:



Der Import und Export von Textdateien erfolgt in XMLports ebenfalls auf unterschiedliche Weise:

- XMLports verfügen über keinen **OnBeforeExportRecord**-Trigger.
- XMLports unterstützen durch die CaptionML sowie durch Spracheigenschaften (Language Properties) mehrere Sprachen.
- XMLports unterstützen UTF-8- und UTF-16-Formate für XML.

Dateispeicherorte für den Upload/Download

Beim Erstellen von Dateien für die 3-Tier-Architektur ist die Codeaktualisierung erforderlich, so dass die Dateien am richtigen Speicherort erstellt und gespeichert werden. Das Herunterladen, Bearbeiten und Hochladen von Dateien muss im Middle-Tier erfolgen, also in der Ebene, in der der Microsoft Dynamics NAV Server ausgeführt wird.

Diese Vorgehensweise gilt ebenso für die Bearbeitung von Berichten, da ein Bericht nach dem Ausführen in der Regel als Datei gespeichert oder in einer Datei gedruckt wird.

Die folgenden neuen Funktionen wurden eingeführt, um den Transfer und die Verwaltung dieser Dateien zu vereinfachen:

- Die Funktionen FILE.UPLOAD und FILE.UPLOADINTOSTREAM zum Senden einer Datei vom RoleTailored Client an den Middle-Tier
- Die Funktionen FILE.DOWNLOAD und FILE.DOWNLOADINFROMSTREAM zum Senden einer Datei vom Middle-Tier an den RoleTailored Client

Automation Objects

Die Verwendung von Automation Objects erfolgt über die Ausführung von Code im Microsoft Dynamics NAV Server. Daher muss sichergestellt werden, dass die Dateien auf dem Computer mit Microsoft Dynamics NAV Server erstellt und der Code an den richtigen Stellen ausgeführt wird. In den folgenden Szenarien wird gezeigt, wie die Automation Objects zur Unterstützung im Microsoft Dynamics NAV Server angepasst werden müssen.

Hinweis: Sie müssen lediglich COM-Objekte auf dem Computer installieren, auf dem der Microsoft Dynamics NAV Server ausgeführt wird, und nicht auf jedem einzelnen Client.

Verwendung einer COM-Komponente zum Erstellen einer Excel-Datei für einen Benutzer

Beim Ausführen von C/AL-Code auf dem Microsoft Dynamics NAV Server können Sie die CREATE-Funktion verwenden, um das Automation Object zu erstellen und dann über die SAVE-Funktion auf dem Computer zu speichern. Wenn der RoleTailored Client den Zugriff auf die Datei benötigt, so kann über die FILE.DOWNLOAD-Funktion die Datei von dem Computer aus, auf dem der Microsoft Dynamics NAV Server ausgeführt wird, an den Client gesendet werden.

Um die ordnungsgemäße Ausführung sicherzustellen, muss die COM DLL auf dem Computer, auf dem Microsoft Dynamics NAV Server installiert ist, gespeichert und registriert sein.

Verwendung einer COM-Komponente für die externe Bearbeitung (z. B. Aufruf eines Webservices)

Beim Ausführen von C/AL-Code auf dem Microsoft Dynamics NAV Server können Sie die CREATE-Funktion verwenden, um ein Automation Object zu erstellen, und anschließend eine Methode aufrufen. Beim Ausführen der Methode werden sämtliche Rückgabewerte wieder an den Code zurückgegeben.

Um die ordnungsgemäße Ausführung sicherzustellen, muss die COM DLL auf dem Computer, auf dem Microsoft Dynamics NAV Server installiert ist, gespeichert und registriert sein.

Verwendung einer COM-Komponente für den Abruf einer MSMQ-Nachricht

Beim Ausführen von C/AL-Code auf dem Microsoft Dynamics NAV Server können Sie die CREATE-Funktion zum Erstellen eines Automation Objects sowie die READ-Funktion für den Abruf aus der Message Queue verwenden. Im RoleTailored Client wird dann über den Microsoft Dynamics NAV Server das Ergebnis der READ-Funktion bearbeitet.

Hinweis: Wenn die Queue auf einem lokalen Computer gespeichert ist (z. B. ./myqueue), so wird der Code aus der Message Queue mit der Bezeichnung *myqueue* von dem Computer abgerufen, auf dem der Microsoft Dynamics NAV Server installiert ist, und nicht aus einer Message Queue des Clients.

Verwendung eines Visual OCX über das eigene Fenster

Ein OCX mit einem eigenen Fenster kann nicht zum Ausführen spezifischer Clientaufgaben verwendet werden.

In der 3-Tier-Architektur kann der Client keinen Code ausführen oder mit anderen Komponenten interagieren, die ebenfalls auf dem Client oder vom Endbenutzer ausgeführt werden. Um das Verhalten des bestehenden Clients zu duplizieren, muss das Design der anderen Anwendungen, die auf dem Client ausgeführt werden, als webservicebasierte Lösungen erstellt werden. Die Clientanwendung kann dann den Webservice aufrufen, der die vom System benötigten Informationen liefert. Sie haben auch die Möglichkeit, Code zu erstellen, um auf einer Page Aktionen einzubinden, die nach dem Aufruf des Webservices die Datenbank abrufen.

Interaktion zwischen Client und Hardware

Im RoleTailored Client wird die Interaktion mit lokaler Hardware, wie z. B. Barcode-Scannern und elektronischen Messgeräte, nicht unterstützt.

In der 3-Tier-Architektur wird die Geschäftslogik statt auf dem Client auf dem Microsoft Dynamics NAV Server ausgeführt. Für alle bestehenden Komponenten, für deren Ausführung auf dem Client Geschäftslogik erforderlich ist, muss ein Redesign durchgeführt werden. Dies könnte z. B. die Umgestaltung in Webservices sein, um Daten an den Microsoft Dynamics NAV Server zu übertragen oder das Erstellen einer Clientkomponente, die mithilfe von C/Front.NET Werte direkt in die SQL Server-Datenbank schreibt.

Display-Properties in Forms und Pages

In der 3-Tier-Architektur werden Eigenschaften für die Definition von Schriftarten, wie z. B. **FontSize** und **FontItalic**, nicht unterstützt. Diese Eigenschaften werden im RoleTailored Client ignoriert, da das Verhalten der UI über den Client gesteuert wird.

Die Eigenschaften **Visible**, **Editable** und **Enabled** wurden ebenfalls geändert. Im Classic Client können Sie für Forms für die Eigenschaft den Wert *Boolean* festlegen, wohingegen im RoleTailored Client für Pages der Wert in einer Eigenschaft des Controls definiert wird. Es besteht auch die Möglichkeit, den Wert eines Ausdrucks festzulegen. Beispielsweise kann der Wert auf *TRUE* gesetzt werden, wenn ein bestimmter Wert in einer Variable vorhanden ist oder ein anderes Control über einen bestimmten Wert verfügt.

Stylesheets

Für die in beiden Clienttypen verwendeten Stylesheets (Layoutdateien) sind verschiedene Vorlagen erforderlich. Für den RoleTailored Client müssen Sie die bestehenden Stylesheets entsprechend umgestalten. Jedoch werden sie weiterhin in derselben Tabelle in Microsoft Dynamics NAV gespeichert.

Beim Speichern eines Stylesheets müssen Sie in einem zusätzlichen Feld angeben, dass ein bestimmtes Stylesheet für eine Form (im Classic Client) oder für eine Page (im RoleTailored Client) gilt.

Lang andauernde Prozesse

Der Microsoft Dynamics NAV Server bietet nicht die Unterstützung lang andauernder Prozesse. Ein Beispiel hierfür ist eine Lösung, die auf dem Microsoft Dynamics NAV Application Server ausgeführt wird und externe COM-Ereignisse bearbeitet.

Um eine ähnliche Lösung zu erstellen, müssen Sie anstelle eines abfragegesteuerten Ansatzes einen ereignisgesteuerten Ansatz anwenden, z. B. die Implementierung einer Lösung, in der ein Webservice als Ereignistrigger verwendet wird. So kann die Interoperabilität und der zuverlässige Ablauf sichergestellt werden.

Unterstützung mehrerer Clients in Microsoft Dynamics NAV "6.0"

Der Microsoft Dynamics NAV Classic Client und der Microsoft Dynamics NAV Application Server können auf derselben SQL Server-Datenbank ausgeführt werden, die vom Microsoft Dynamics NAV Server verwendet wird. Das bedeutet, dass sowohl RoleTailored als auch Classic Clients in einem System genutzt werden können, um den verschiedenen Benutzern eines Unternehmens den Zugriff auf dieselbe Anwendung und dieselben Daten zu ermöglichen.

Der größte Vorteil bei der Bereitstellung unterschiedlicher Clients innerhalb eines einzigen Systems besteht darin, dass bestimmte Lösungen, bei denen der Umstieg auf die 3-Tier-Architektur noch nicht vorgenommen wurde, weiterhin eingesetzt werden können. Das gleiche gilt für alle Integrationslösungen, die auf dem Microsoft Dynamics NAV Application Server ausgeführt werden.

Bei der Bereitstellung verschiedener Clientversionen gibt es keine Beschränkungen in Microsoft Dynamics NAV. Die Hauptschwierigkeit bei der Unterstützung mehrerer Clients besteht in der erhöhten Komplexität für die Entwicklung der Lösung. In der 2-Tier-Architektur des Classic Client wird der gesamte Code in der lokalen Umgebung ausgeführt. In der 3-Tier-Architektur des RoleTailored Client hingegen wird der gesamte Code auf dem Computer unter dem Microsoft Dynamics NAV Server ausgeführt. Bei Entscheidungen hinsichtlich des Designs muss die Parität im Codeausführungsverhalten zwischen den beiden Clients sichergestellt werden. Treten Differenzen auf, führen die verschiedenen Clients zwar dieselbe Aufgabe aus, führen jedoch zu unterschiedlichen Ergebnissen, was eine Dateninkonsistenz zur Folge haben kann.

Codes zur Unterstützung mehrerer Clients

Der größte Anteil an Code weist in beiden Clienttypen das gleiche Verhalten auf. Jedoch gibt es bei der Ausführung von Forms und Pages folgenden Unterschied im Verhalten:

Zum Beispiel erstellen Sie folgenden C/AL-Code für ein Form:

```
FORM.RUN(<form id>,<optional parameters>);
```

Dieser Code wird zur Laufzeit wie folgt übersetzt:

- Wenn das Form auf dem Computer mit dem Classic Client ausgeführt wird, dann wird das Formularobjekt mit dieser ID zugrunde gelegt.
- Wenn das Form auf dem Computer mit dem Microsoft Dynamics NAV Server ausgeführt wird, dann wird das Page-Objekt mit dieser ID zugrunde gelegt.

Wenn Sie den Code für mehrere Clients erstellen, gelten folgende Bedingungen:

- Forms und Pages müssen synchronisiert sein. Wenn über die Codeunit das *Customer Form* ausgerufen wird, dann muss die *Customer Page-ID* mit der *Customer Form-ID* übereinstimmen. Andernfalls würde die Codeunit nicht funktionieren.
- Forms und Pages müssen über dieselben Methoden mit derselben Anzahl an Parametern verfügen. Andernfalls müssen Sie zusätzlichen Code schreiben, um Forms und Pages abzubilden.

Zum Erstellen von Code für beide Clienttypen wurde die neue Systemeigenschaft ISSERVICETIER eingeführt. Bei Ausführung des Codes über den Dienst wird der Wert TRUE zurückgegeben (andernfalls FALSE). Über die Systemeigenschaft ISSERVICETIER können Sie den Code festlegen, der explizit vorgibt, ob die Ausführung auf dem Dienst erfolgen soll.

Hinweis: Diese Eigenschaft ähnelt der Systemeigenschaft GUIALLOWED, die zum Erstellen von Code für den Microsoft Dynamics NAV Application Server eingeführt wurde.

Wenn z. B. das Formularobjekt die WRITE-Methode mit zwei Integer-Parametern enthält, dann muss auch das entsprechende Page-Objekt dieselbe Methode mit denselben Parametern beinhalten. Haben die Methoden nicht dieselbe Signatur, dann schlägt jede Art von Code fehl, der sich außerhalb des Objekts befindet, mit dem die Page oder das Form aufgerufen wird.

Im folgenden Beispiel verfügt das Page-Objekt über einen zusätzlichen Textparameter. Der Code kann wie folgt geschrieben werden, um den auszuführenden Client festzulegen:

```
IF (ISSERVICETIER) THEN
    FORM.WRITE(int,int,text)
ELSE
    FORM.WRITE(int,int)
ENDIF
```

Hierbei werden folgende Komplikationen beim Erstellen und Verwalten von Code hervorgerufen:

- Der Code kann nicht nur schwer gelesen, verstanden und verwaltet werden. Dabei kann die Synchronisierung des Codes an mehreren Stellen erforderlich sein. Berichte und Code, die Automation Objects und Dateien verwenden, sind in den beiden Plattformen unterschiedlich. Beispielsweise funktionieren die Stylesheets für die Office-Integration im Classic Client anders als im RoleTailored Client. Das Schreiben und Verwalten von Code für Stylesheets erfordert die Arbeit in zwei Formaten und das Verständnis darüber, wie bestimmte Stylesheets auf die verschiedenen Clients anzuwenden sind.
- Es müssen mehrere Tests durchgeführt werden, da der Code mit mehreren Clients getestet werden muss, um die ordnungsgemäße Ausführung des Codes sicherstellen.
- Beide Clients verwenden das MenuSuite-Objekt für die Navigation. Allerdings wurde das Design unterschiedlich implementiert. Im Classic Client öffnet ein Benutzer eine Übersicht über ein zuvor geöffnetes Kartenfenster. Im RoleTailored Client wird dem Benutzer gleich eine Übersicht angezeigt, aus der die gewünschte Page (Card) geöffnet werden kann. Um beide Designtypen in Microsoft Dynamics NAV zu unterstützen, ist ein hoher Arbeitsaufwand erforderlich.
- Beide Clients verfügen über unterschiedlich implementierte Features für die Erfassung von Notizen und Kommentaren zu einem Beleg oder Kontakt. Im Classic Client werden Kommentare verwendet, im RoleTailored Client hingegen Notizen, wobei beide Features dieselbe Funktionalität darstellen. Jedoch kann ein Benutzer nicht die Eingabedaten eines anderen Benutzers sehen, der auf einem anderen Clienttyp arbeitet.

Berichte zur Unterstützung mehrerer Clients

Sie haben die Möglichkeit, ein einzelnes Report-Objekt zu erstellen, das optimal für die Ausführung auf beiden Clienttypen ausgerichtet ist. Hierbei ist zu beachten, dass das Layout der Reports unterschiedlich definiert wird. Beim Entwerfen eines Berichts für den Classic Client wird das Layout mithilfe des Section Designers entworfen. Beim RoleTailored Client erfolgt dies über einen RDL-Editor, wie z. B. dem Visual Studio Report Editor.

Ein Teil der Eigenschaften der herkömmlichen Classic Client-Berichte werden in der neuen Architektur unterstützt. Dies sind in der Regel Eigenschaften für das Reportlayout sowie für Summierungen, Gruppierungen und Druckerspezifikationen. Beim Öffnen von Visual Studio werden diese Eigenschaften ignoriert, so dass sie vom Layoutdesigner aus zugewiesen werden müssen. Dies muss nur ein einziges Mal vorgenommen werden. Sämtliche Änderungen, die Sie über Visual Studio für die Berichte vornehmen, werden beim Speichern des jeweiligen Berichts automatisch übernommen.

Dabei sollten sämtliche spezielle Codeangaben eines Berichts aus den Section Triggern in ein Datenelement-Section des Reports oder in eine Codeunit übernommen werden.

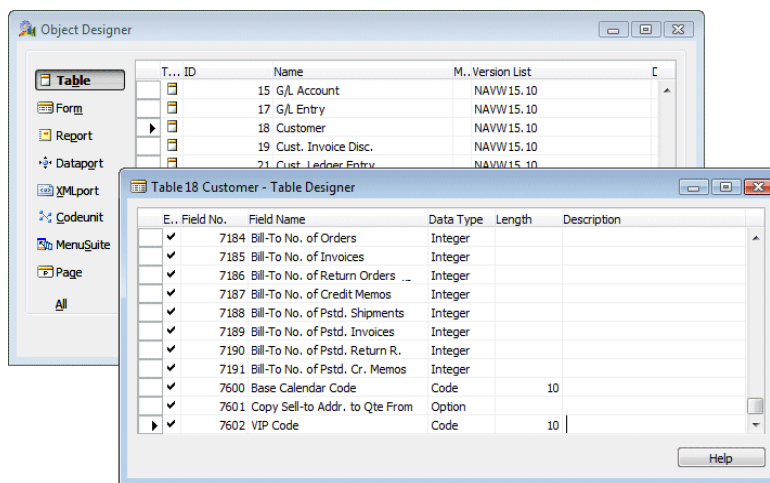
Schritt-für-Schritt-Anleitung: Erstellen und Ändern von Pages

In diesem Abschnitt wird das Erstellen und Ändern von Pages im RoleTailored Client vorgeführt. Zunächst wird in der **Customer**-Tabelle ein neues Feld erstellt. Danach werden für die Anzeige des neuen Feldes das **Customer List**-Place und die **Customer Card**-Page entsprechend angepasst. Anschließend wird eine FactBox erstellt, in der Detailinformationen über den im **Customer List**-Place ausgewählten Debitor angezeigt werden.

Hinzufügen eines neuen Feldes in der Debitorentabelle

Sie beschließen, eine neue Tabelle mit der Bezeichnung **VIP** für die wichtigsten VIP-Statusoptionen (Bronze, Silber, Gold und Diamant) für Debitoren zu erstellen.

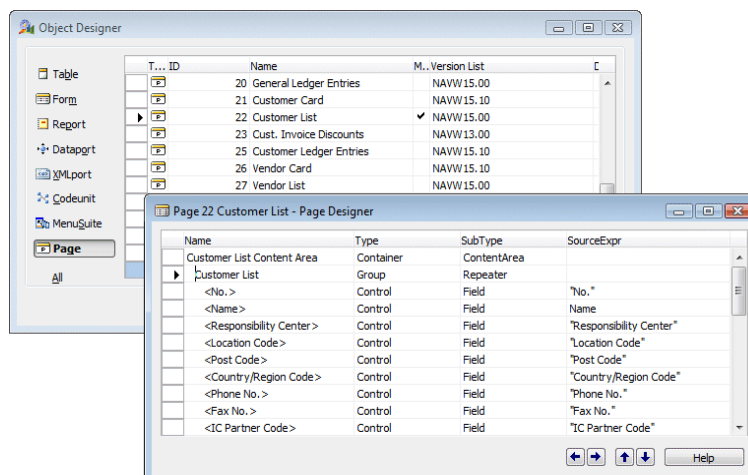
1. Um in der **Customer**-Tabelle das Feld **VIP-Code** hinzuzufügen, welches den VIP-Status eines Debtors angeben soll, öffnen Sie den Classic Client, klicken Sie auf den Tabellendesigner, und geben Sie das neue Feld **VIP-Code** ein.
2. Verknüpfen Sie das Feld **VIP-Code** mit der **VIP-Code**-Tabelle.



Ändern der Customer List-Page

Nun sollen die Informationen über das Feld **VIP-Code** in die **Customer List**-Page übernommen werden.

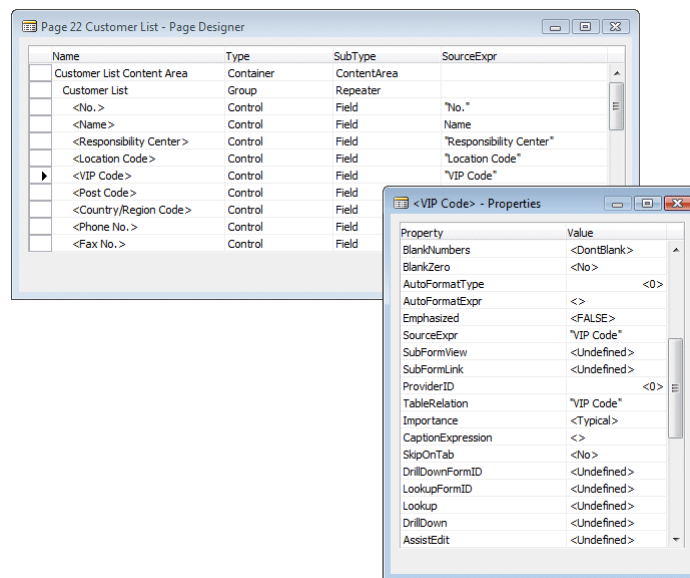
1. Öffnen Sie im Object Designer über den Page Designer die **Customer List**-Page.



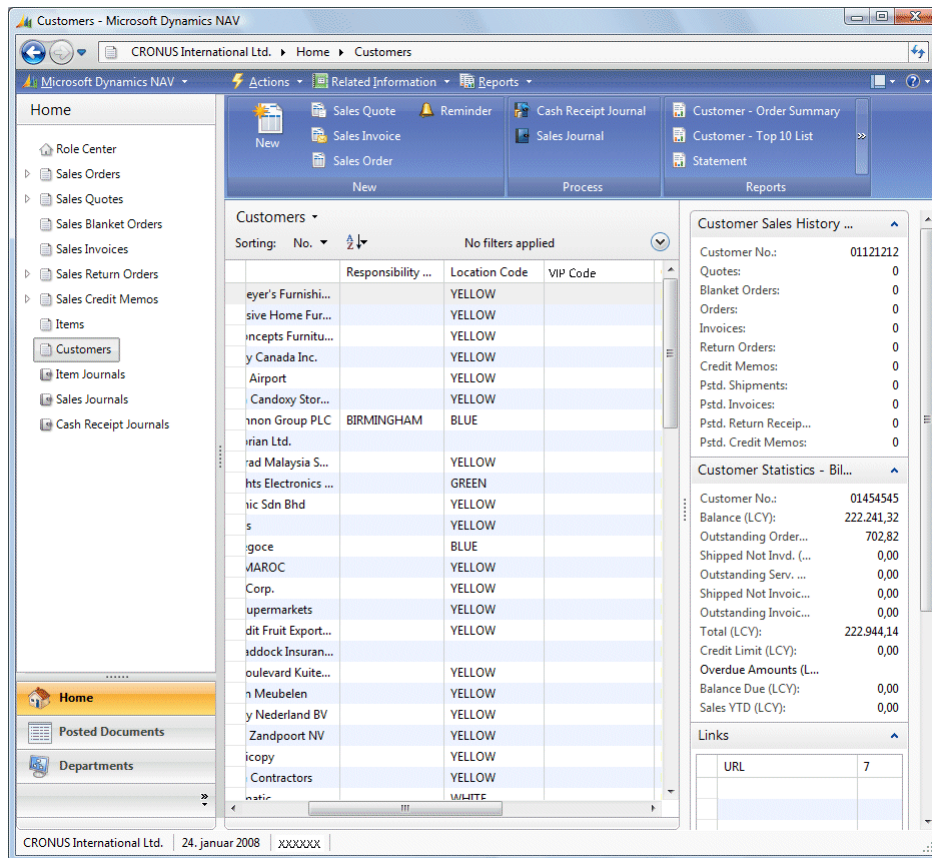
Pages verfügen über eine hierarchische Struktur, die es Ihnen ermöglicht, die Inhalte einer Page in Gruppen, in so genannte *Container*, zu organisieren. Das erste Element bzw. die erste Zeile im Page Designer definiert die primäre Gruppe der Page. In diesem Beispiel wird sie als **Customer List Content Area** bezeichnet. Dieses Element hat die Subtype **ContentArea**, was bedeutet, dass die Inhalte dieser Gruppe beim Aufruf über den RoleTailored Client im Inhaltsbereich angezeigt werden. InfoParts und HomeParts sind weitere wichtige Subtypes, die später in diesem Abschnitt verwendet werden.

Die hierarchische Struktur der Page wird durch die Einrückung der im Page Designer angezeigten Zeilen festgelegt. Die **Customer List**-Gruppe verfügt über eingerückte Elemente, die eine separate Gruppe innerhalb des **Customer List Content Area**-Containers darstellen.

In diesem Beispiel sind in der **Customer List**-Gruppe die Felder enthalten, die als Spaltenüberschriften in der **Customer List**-Page erscheinen.



2. Erstellen Sie im Page Designer in der **Customer List**-Gruppe die Zeile **VIP-Code**. Hierbei wird in die **Customer List**-Page eine neue Spalte eingefügt. Setzen Sie die **VIP-Code**-Zeile in die Stelle, in der die Spalte in der Übersicht angezeigt werden soll.
3. Um die Eigenschaften für das neue Page-Element festzulegen, öffnen Sie das **Properties**-Fenster für die **VIP-Code**-Zeile.
4. Legen **SourceExpr** für den **VIP-Code** fest. Das **VIP-Code**-Feld befindet sich in der **Customer**-Tabelle, die zugleich die Quelltable des **Customer List**-Place ist.
5. Speichern und kompilieren Sie die geänderte Page. Beim Kompilieren der Page wird das System aktualisiert. Ein RoleTailored Client, der nun eine Verbindung mit der Datenbank herstellt, kann das aktualisierte Page-Objekt lesen.
6. Öffnen Sie im RoleTailored Client das Customer List-Place, um die neue **VIP-Code**-Spalte anzuzeigen.

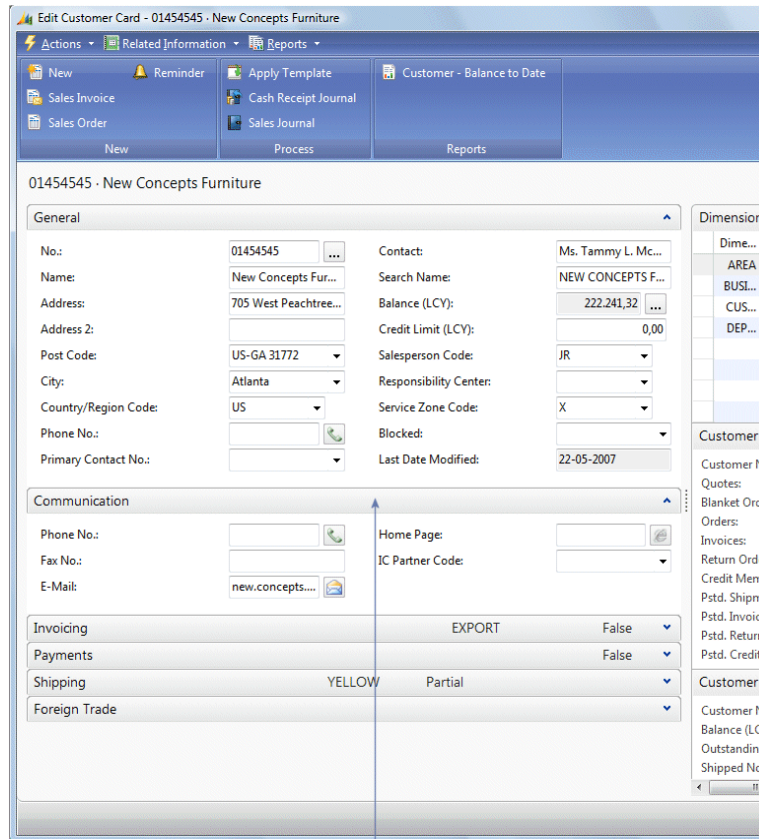


Ändern der Customer Card-Page

List Places können nur gelesen werden. Um Endbenutzern das Anzeigen und Aktualisieren des **VIP-Code**-Feldes zu ermöglichen, muss auch die **Customer Card**-Page geändert werden. Die **Customer Card**-Page wird auf eine andere Weise geändert, da eine Karte auf einer anderen Page-Art basiert. Kartenfenster beinhalten anstelle von Spalten und Zeilen FastTabs und Eingabefelder.

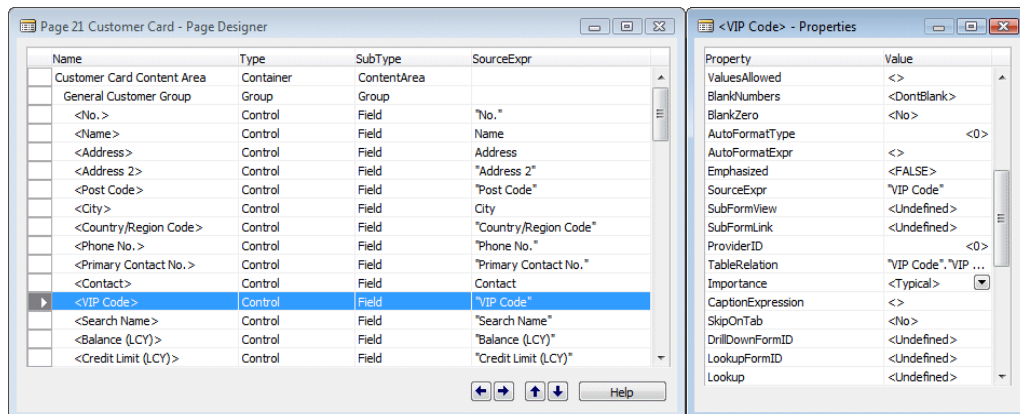
1. Öffnen Sie im Object Designer über den Page Designer die **Customer Card**-Page.
2. Setzen Sie den Mauszeiger im unteren Bereich des Page Designers in eine leere Zeile, und öffnen Sie das **Properties**-Fenster, um die Art dieser Page anzuzeigen.

Über die **PageType**-Eigenschaft wird die Art der Page bestimmt und wie die Gruppen von Page-Elementen im RoleTailored Client angezeigt werden. Ist die **PageType**-Eigenschaft **Card**, so wird jede Gruppe dieser Page als FastTab angezeigt. Ist die **PageType**-Eigenschaft **List**, so wie in der vorigen Beschreibung, dann sind in jeder Gruppe die in einer Übersicht angezeigten Spalten enthalten.

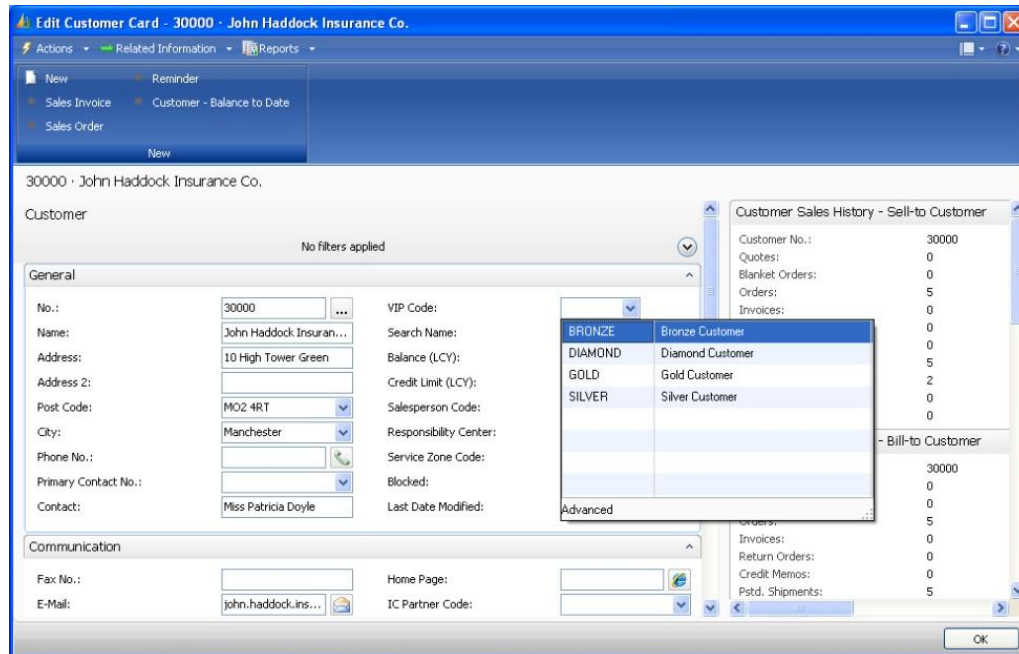


Fast Tabs

3. Erstellen Sie im Page Designer in der **General Customer Group** ein neues Feld, indem Sie in der entsprechenden Position eine Zeile einfügen.
4. Öffnen Sie das **Properties**-Fenster, und legen Sie in der **Customer**-Tabelle das **VIP-Code**-Feld als **SourceExpr** für das neue **VIP-Code**-Control in der **Customer Card**-Page fest. Die **Customer**-Tabelle ist ebenfalls die Quelltable für die **Customer Card**-Page.



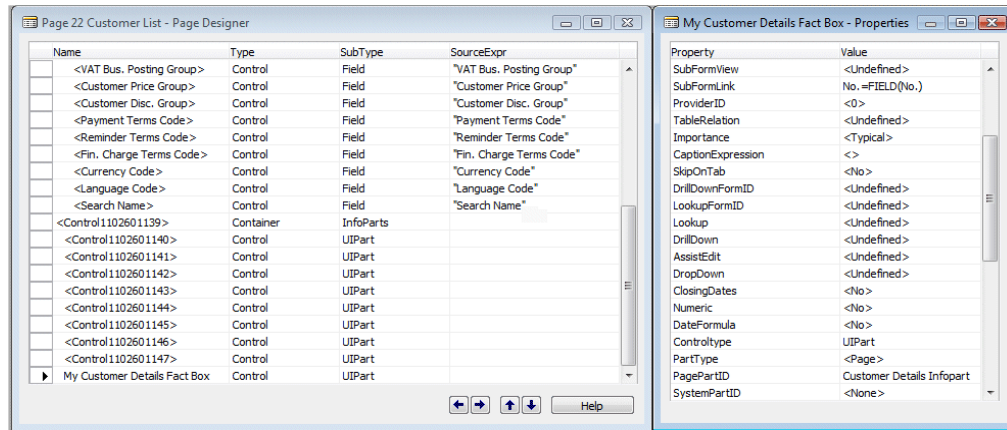
- Legen Sie die **TableRelation**-Eigenschaft auf "VIP-Code" fest, um den Aufruf und die Anzeige der festgelegten Auswahlmöglichkeiten für den RoleTailored zu ermöglichen.
- Speichern und kompilieren Sie die geänderte Page. Beim Kompilieren der Page wird das System aktualisiert. Ein RoleTailored Client, der nun eine Verbindung mit der Datenbank herstellt, kann das aktualisierte Page-Objekt lesen.
- Öffnen Sie den RoleTailored Client, um die geänderte **Customer Card**-Page anzuzeigen.



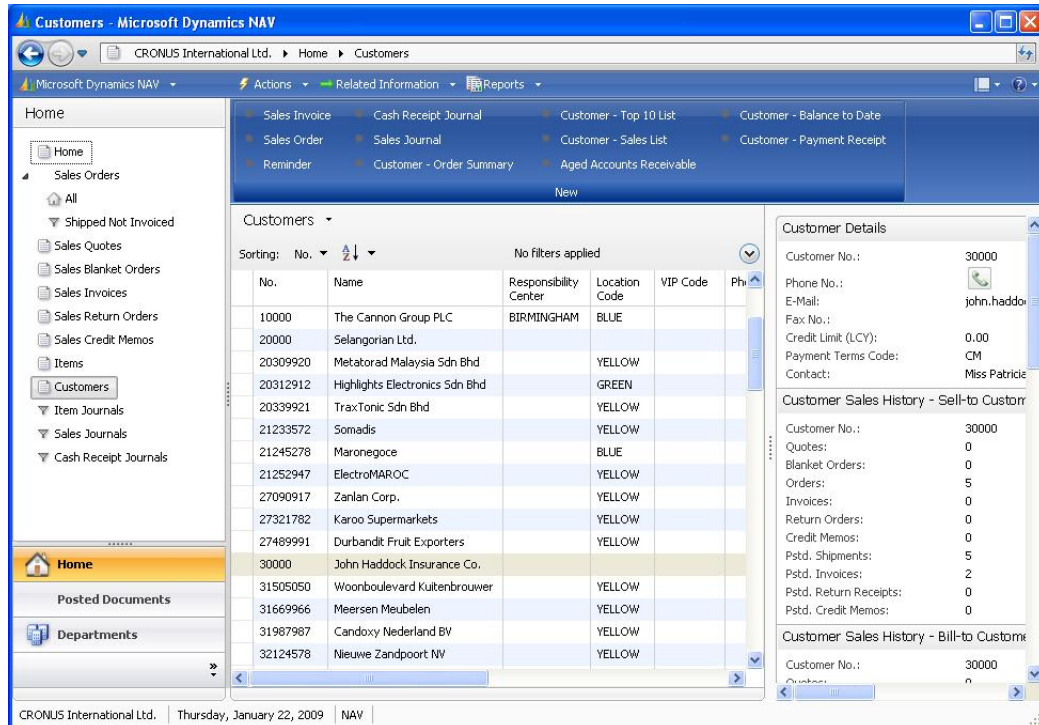
Hinzufügen einer neuen FactBox in der Customer List Page

Zum Schluss dieser Vorführung wird eine neue FactBox erstellt und in die **Customer List**-Page übernommen. In der FactBox sollen Details über den in der Übersicht ausgewählten Debitor angezeigt werden. So hat der Benutzer sofort alle Detaildaten im Überblick, und er muss nicht erst die **Customer Card**-Page öffnen, um Basisinformationen, wie z. B. Telefonnummern oder E-Mail-Adressen, anzuzeigen.

- Öffnen Sie im Object Designer die **Customer List**-Page.
- Fügen Sie eine neue Zeile mit der Beschreibung **My Customer Details FactBox** ein.
- Öffnen Sie das **Properties**-Fenster für die neue Zeile, und stellen Sie sicher, dass der **PagePartID**-Eigenschaftswert auf Page 9084, also auf die **Customer Details FactBox**, festgelegt ist. Diese FactBox ist bereits standardmäßig im RoleTailored Client enthalten.
- Verwenden Sie im **My Customer Details Fact Box – Properties**-Fenster die **SubFormLink**-Eigenschaft, um die Beziehung zwischen der Page und der Quelle der FactBox zu definieren. FactBoxes sind, genauso wie Forms und Subforms untereinander verknüpft sind, an die Hauptinhalte einer Page gebunden.



- Speichern und kompilieren Sie die geänderte Page. Beim Kompilieren der Page wird das System aktualisiert. Ein RoleTailored Client, der nun eine Verbindung mit der Datenbank herstellt, kann das aktualisierte Page-Objekt lesen.
- Öffnen Sie im RoleTailored Client die **Customer List**-Page, um die Änderungen einzusehen.

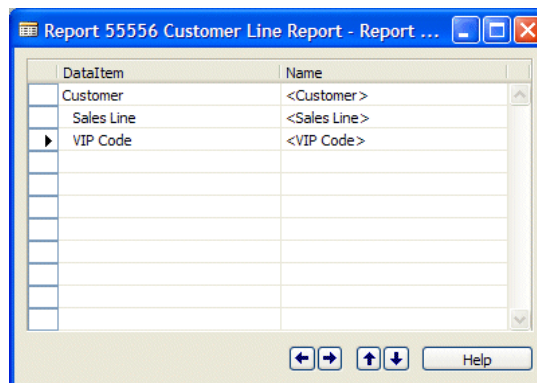


Schritt-für-Schritt-Anleitung: Erstellen eines Reports

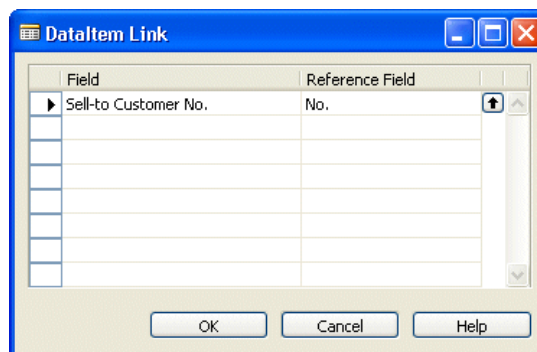
In diesem Abschnitt wird ein Bericht für die **Customer**-Tabelle erstellt. Für jeden im Berichtsfilter angegebenen Debitor soll im Bericht die gesamte **Sales Line**-Tabelle überprüft und die gefundenen Datensätze (Debitoren und Verkaufsaufträge) nach dem Lieferdatum sortiert aufgeführt werden.

Definieren des Datenmodells

1. Wählen Sie im Report Designer die **Customer**-Tabelle als erstes Datenelement, die **Sales Line**-Tabelle als zweites Datenelement und die **VIP-Code**-Tabelle als drittes Datenelement. Rücken Sie die **Sales Line**- und **VIP-Code**-Datenelemente wie folgt ein:



2. Öffnen Sie das Properties-Fenster für das **Sales Line**-Datenelement. Legen Sie die **DataltemLinkReference**-Eigenschaft auf den Namen des übergeordneten Datenelements (**Customer**) fest, auf das das eingerückte Datenelement (**Sales Line**) referieren muss. In der Regel ist dies die Standardeinstellung.
3. Legen Sie die **DataltemLink**-Eigenschaft fest, um auf das Feld **Sell-to Customer No.** der **Sales Line**-Tabelle zu referieren. Wählen Sie im **Reference Field** aus der **Customer**-Tabelle das Feld **No.**



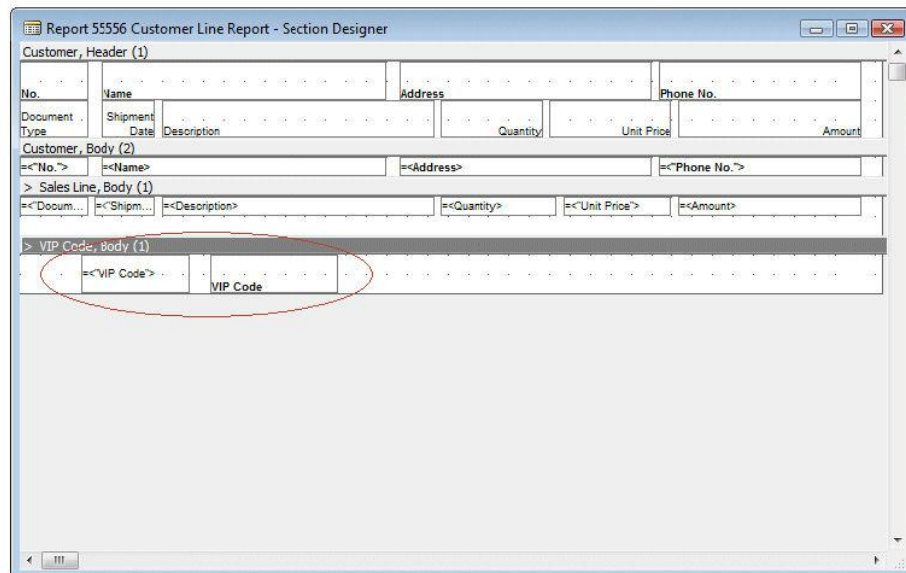
4. Legen Sie im **Properties**-Fenster für das **VIP-Code**-Datenelement die **DataltemLink**-Eigenschaft fest, um auf das **VIP Code**-Feld zu referieren. Wählen Sie im **Reference Field** aus der **Customer**-Tabelle das Feld **No.**
5. Legen Sie im **Properties**-Fenster des **Customer**-Datenelements die **PrintOnlyIfDetail**-Eigenschaft auf **Yes** fest. Somit wird sichergestellt, dass die **Customer**-Body Sections nur gedruckt werden, wenn Daten aus der **Sales Line**-Tabelle vorhanden sind.

Das Datenmodell funktioniert dann wie folgt:

- Der Bericht überprüft das **Customer**-Datenelement.
- Für jeden Datensatz des **Customer**-Datenelements werden im **Sales Line**-Datenelement Datensätze ausgewählt, wenn der Inhalt des Feldes **Sell-to Customer No.** mit dem Feld **No.** im **Customer**-Datenelement übereinstimmt.
- Wenn für ein **Customer**-Datenelement keine **Sales Lines**-Datensätze vorhanden sind, wird kein Druckvorgang gestartet (auch nicht für die Informationen aus dem **Customer**-Datenelement).
- Der Bericht überprüft die VIP Code-Tabelle und druckt den VIP-Code des jeweiligen Debtors aus.

Auswahl der im Berichts anzuzeigenden Felder

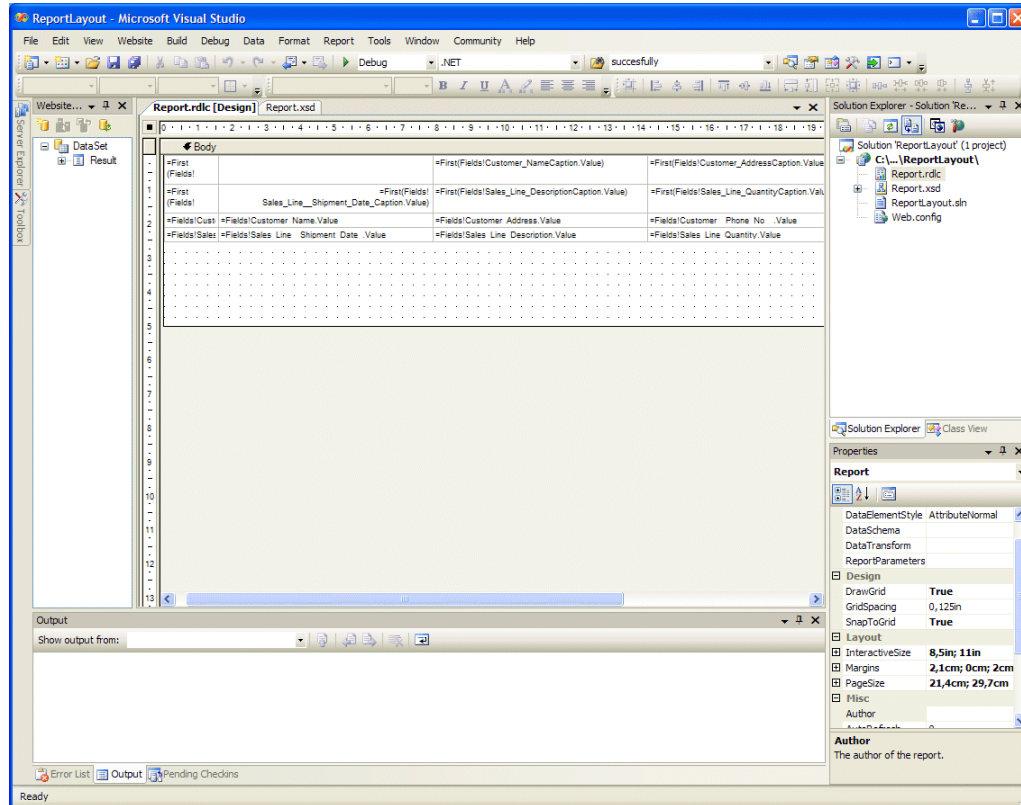
1. Wählen Sie im Report Designer das **Customer**-Datenelement aus.
2. Öffnen Sie im Section Designer das **Field Menu**-Fenster. Fügen Sie hier die Felder **No.**, **Name**, **Address** und **Phone No.** hinzu.
3. Wählen Sie das **Sales Line**-Datenelement aus, und übernehmen Sie die Felder **Document No.**, **Shipment Date**, **Description**, **Quantity**, **Unit Price** und **Amount** in den Report.
4. Wählen Sie das **VIP Code**-Datenelement aus, und übernehmen Sie das **VIP Code**-Feld in den Bericht. Dies erfolgt über den Section Designer.



Beim Erstellen von Sections über den Report Designer werden alle in diese Sections übernommenen Felder im neuen Reportlayout angezeigt. Ebenso werden alle im Primärschlüssel der Tabellen enthaltenen Felder, die als Datenelemente im Report verwendet werden, im neuen Reportlayout angezeigt.

Berichtsentwurf in Visual Studio

- Klicken Sie auf **Tools, Transform Layout**. Das Layout des Reports wird umgewandelt und im Report Designer in Visual Studio geöffnet.



Note: Das Dataset zeigt alle Microsoft Dynamics NAV-Felder an, die Sie im vorigen Report Designer angegeben haben. Sie können jetzt mithilfe von Visual Studio das Layout des Reports entwerfen.

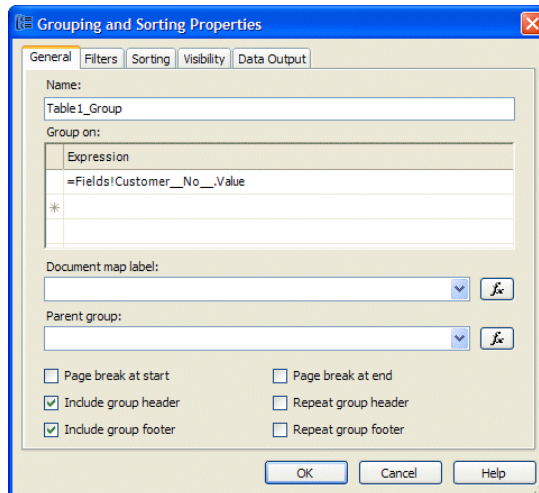
Enwerfen Sie das Reportlayout in Visual Studio. Legen Sie vor dem Layouten sämtliche Gruppierungen und Summierungen an, die Sie im Report berücksichtigen möchten.

Hinzufügen und Sortieren einer Gruppe sowie Sortieren von Detailangaben

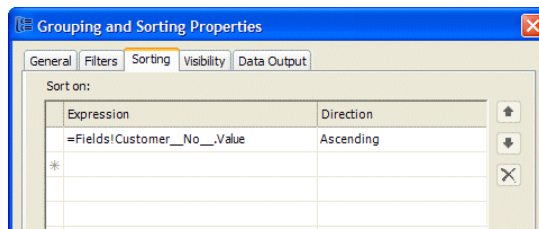
1. Klicken Sie in Visual Studio auf ein beliebiges Feld im Report, um die Editoren der Spalten und Zeilen zu aktivieren.
2. Klicken Sie mit der rechten Maustaste auf ein Feld einer beliebigen Zeile. Klicken Sie auf **Insert Group**, und geben Sie anschließend folgenden Ausdruck ein:

=Fields!Customer_No_.Value

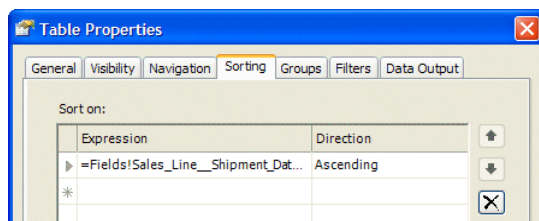
Mit diesem Ausdruck legen Sie die Gruppierung der Einträge nach der Debitorennummer fest.



3. Wählen Sie auf der Registerkarte **Sorting** den Feldwert aus, nach dem Sie die Gruppe sortieren möchten.



4. Öffnen Sie das **Properties**-Fenster.
5. Wählen Sie auf der Registerkarte **Sorting** den Feldwert aus, nach dem Sie die detaillierten Dateneinträge sortieren möchten.



Sie haben nun angegeben, dass im Report die Verkaufszeile für jeden Debitor nach dem Lieferdatum sortiert werden soll.

Hinzufügen einer Zwischensumme

Sie haben in Visual Studio ebenfalls die Möglichkeit, Summierungsfunktionen in den Report zu übernehmen.

1. Fügen Sie unterhalb der **Sales Line**-Zeile eine Zeile ein.
2. Geben Sie in der letzten Zelle dieser neuen Zeile folgenden Ausdruck ein:

=Sum(Fields!Sales_Line_Amount.Value)

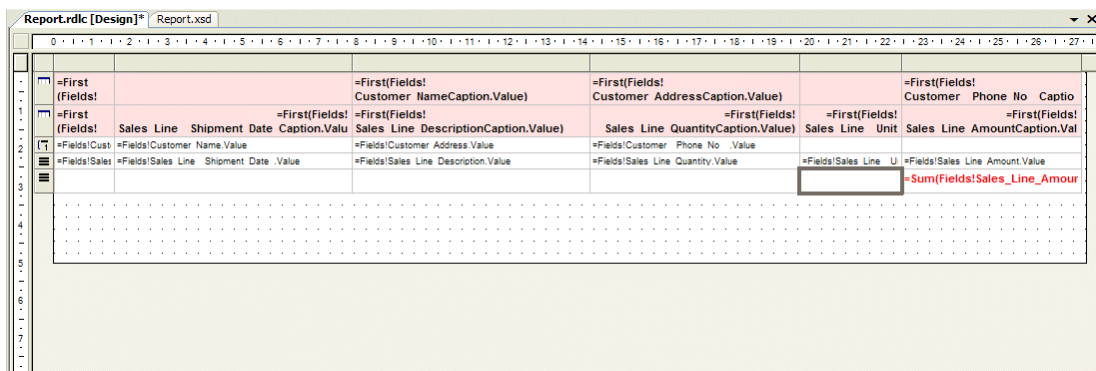
Mit diesem Ausdruck wird der Gesamtbetrag eines Debitorenverkaufsauftrags berechnet.

Entwurf des Berichtslayouts

Sie haben die Möglichkeit, die Anzeige für viele der im Bericht enthaltenen Felder, einschließlich Datums- und Währungsfelder, einfach zu formatieren.

1. Wählen Sie im **Properties**-Fenster die Registerkarte **Format**.
2. Klicken Sie auf die Schaltfläche **More (...)**. Wählen Sie im Fenster **Choose Format** das Format aus, in dem Datumsangaben erscheinen sollen.
3. Wählen Sie die Überschriftenzeilen aus, um die Tabellenüberschriften zu formatieren und sie vom restlichen Teil des Reports hervorzuheben.

Sie können nun die Schriftart, -farbe, -größe usw. sowie die Hintergrundfarbe der Zellen ändern. Nachdem Sie Ihre Angaben vervollständigt haben, sieht der Report wie folgt aus.



=First(Fields! Customer NameCaption.Value)		=First(Fields! Customer AddressCaption.Value)		=First(Fields! Customer Phone No Caption.Value)	
=First(Fields! Sales Line Shipment Date Caption.Value)	=First(Fields! Sales Line DescriptionCaption.Value)	=First(Fields! Sales Line QuantityCaption.Value)	=First(Fields! Sales Line Unit)	=First(Fields! Sales Line AmountCaption.Value)	
=Fields!Cust	=Fields!Customer Name Value	=Fields!Customer Address.Value	=Fields!Customer Phone No Value	=Fields!Sales Line U	=Fields!Sales Line Amount Value
=Fields!Sales	=Fields!Sales Line Shipment Date Value	=Fields!Sales Line Description.Value	=Fields!Sales Line Quantity.Value		=Sum(Fields!Sales_Line_Amount

4. Speichern und schließen Sie das Projekt nach Beendigung des Reportlayouts.
5. Wenn Sie diesen Report anschließend im Classic Client über den Report Designer öffnen, erscheint eine Meldung, in der Sie darüber informiert werden, dass die Datei .rdlc für diesen Bericht geändert wurde, und ob Sie die Änderungen übernehmen möchten. Bestätigen Sie mit **Ja**, um die Änderungen in der Datenbank zu speichern.
6. Kompilieren Sie den Report.

Die Datenbank beinhaltet jetzt zwei Versionen dieses Berichts:

- Eine C/SIDE-Version für die Anzeige im Classic Client
- Eine SQL Server Reporting Service-Version für die Anzeige im RoleTailored Client

Wichtig: Wenn Sie weitere Änderungen am Report vornehmen möchten, müssen Sie den Report im C/SIDE Report Designer öffnen und auf **View, Layout** klicken. Wenn Sie auf **Tools, Transform Layout** klicken, müssen Sie die ursprüngliche C/SIDE-Version des Reports wieder umwandeln. Alle anderen bereits vorgenommenen Änderungen gehen dabei verloren.

Zudem besteht zwischen dem C/SIDE Report Designer und Visual Studio keine nahtlose Integration. Die in einer Umgebung vorgenommenen Änderungen können bei unterschiedlicher Ausführung in der jeweils anderen Umgebung nicht angezeigt werden. Daher müssen alle Änderungen einer Umgebung gespeichert und in die jeweils andere Umgebung importiert oder exportiert werden.

Anhang

Formulare

Die folgenden nicht unterstützten Eigenschaften und Funktionen werden beim Erstellen einer Page, die auf Basis eines Formulars mithilfe des *Form Transformation Tools* generiert wird, entfernt. Alle beim Umwandlungsvorgang gelöschten Codetexte und Eigenschaften werden in der Ausgabedatei *Transaction.log* protokolliert.

Form-Eigenschaften

Werden noch hinzugefügt

Report-Eigenschaften

Werden noch hinzugefügt

Berichte

Die folgenden Eigenschaften und Funktionen werden in den neuen RoleTailored Client-Berichten nicht unterstützt:

Report-Eigenschaften

- BottomMargin
- DeviceFontName
- HorzGrid
- LeftMargin
- Orientation
- PaperSize
- PaperSourceFirstPage
- PaperSourceOtherPages
- RightMargin
- ShowPrintStatus
- TopMargin
- VertGrid

Datenelementeigenschaften

- GroupTotalFields
- NewPagePerGroup
- NewPagePerRecord
- TotalFields

Report-Funktionen

- NewPagePerRecord
- ObjectID
- URL

CurrReport-Funktionen

- CreateTotals
- NewPage
- NewPagePerRecord
- ObjectID
- PageNo
- PaperSource
- SaveAsXML
- ShowOutput
- TotalsCausedBy
- URL

Report-Trigger/-Funktionen

- Section-Trigger
- Layoutspezifische Funktionen und Summierungsfunktionen
- Nicht unterstützte Funktionen mit Verweisen aus anderen Objekten (Codeunits, Tabellen usw.)

XMLports und Dataports

Zur Unterstützung der Dataport-Funktionalität im RoleTailored Client wurden die folgenden neuen XMLport-Eigenschaften eingeführt:

Neue XMLport-Eigenschaften

- FieldDelimiter
- FieldSeparator
- Format
- RecordSeparator
- TableSeparator
- UseRequestForm
- Width

Nicht über Dataports übertragbare XMLport-Funktionen

- OnBeforeExportRecord

Microsoft Dynamics ist eine Produktlinie integrierter, anpassbarer Businessmanagementlösungen, die es Ihnen und Ihren Mitarbeitern ermöglicht, fundierte Geschäftsentscheidungen zu treffen. Microsoft Dynamics arbeitet wie Ihnen bekannte Softwareprodukte von Microsoft und ist auch mit diesen integriert. Es enthält umfassende Funktionen, die Sie dabei unterstützen, Ihre Geschäftsprozesse in den Bereichen Finanzmanagement, Customer Relationship Management und Supply Chain Management zu automatisieren und zu optimieren und den Erfolg Ihres Business voranzutreiben.

Microsoft Deutschland GmbH
Gasstraße 6a
22761 Hamburg

Deutschland

Tel.: +49 40 897 22 0

Fax: +49 40 897 22 66 99

Weltweit: (+1) (701) 281-6400

Dieses Dokument dient nur zu Informationszwecken.

Die Microsoft Corporation lehnt jede Haftung für Fehler und Unterlassungen in diesem Dokument ab.

© 2007 Microsoft Corporation. Alle Rechte vorbehalten. Microsoft, Microsoft Dynamics und das Microsoft Dynamics-Logo sind entweder eingetragene Markenzeichen oder Markenzeichen der Microsoft Corporation in den USA und/oder anderen Ländern. Alle anderen in diesem Dokument genannten Firmennamen und Produktnamen sind möglicherweise Markenzeichen der jeweiligen Rechtsinhaber.