Composite Applications – Einführung und Überblick

Integration von Daten und Anwendungen sind der Schlüssel bei dem Einsatz moderner Anwendungen. Wiederverwendbarkeit wird dabei genauso groß geschrieben, wie die Optimierung der Ressourcen sowohl in der Anwendungsentwicklung, aber auch in der Datenhaltung und nicht zuletzt bei dem Einsatz der Anwendungen und der Daten am Arbeitsplatz.

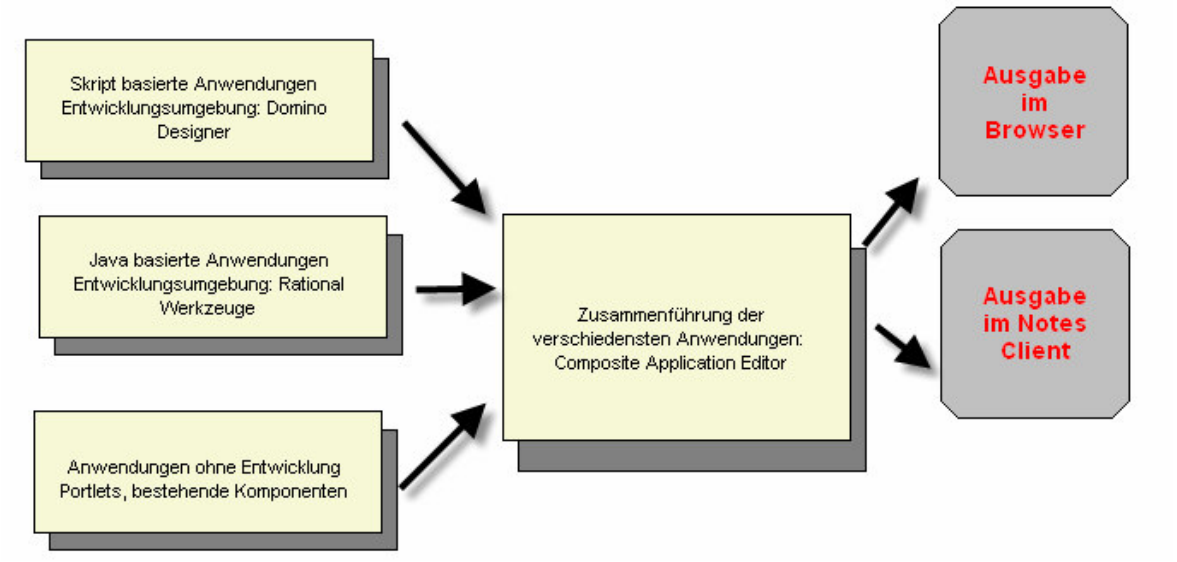
IBM hat Lotus in der letzten Jahren mehr und mehr dahingehend modifiziert und optimiert, dass das Spektrum der Einsatzmöglichkeiten sich stark über die klassischen Einsatzbereiche der bisherigen Lotus Familie erweitert hat. Das Zusammenstellen von neuen Anwendungen, basierend auf bestehenden Anwendungskomponenten ist dabei das Hauptthema, mit dem wir uns in dieser Ausgabe beschäftigen wollen. Das Verbinden von Anwendungen, man könnte auch sagen, das „Komponieren" von Funktionen und Anwendungen ist damit eine neue Aufgabe der Domino Anwendungsentwicklung. In diesem Zusammen hang werden Composite Anwendungen genannt – zu deutsch: Verbundanwendungen.

Doch was ist eine solche Verbund Anwendung eigentlich?

Bei Verbund Anwendungen handelt es sich um modulare Programmkomponenten, die zusammengesetzt eine Anwendung ergeben. Die einzelnen Komponenten können dabei aus den verschiedensten Anwendungen stammen, die mit den Verbundanweisungskomponenten zu einer neuen Anwendung zusammengefasst werden. Verbundanweisungen beinhalten quasi Links zu den einzelnen Anwendungskomponenten, die zur Ausführungszeit aufgerufen und ausgeführt werden. Domino Anwendungen lassen sich mit ein wenig Vorbereitung zu Modulen umwandeln, welche wir individuell in neue Anwendungen und Funktionen integrieren können. Auf diese Weise ist es beispielsweise möglich, Adressinformationen aus dem Notes-Adressbuch in eine Anwendung zu integrieren oder eine Ansicht aus einer Domino Anwendung in eine neue Webseite zu implementieren.

Das Kombinieren von Webservices ist ein Merkmal der Verbundanwendungen – die Wiederverwendbarkeit der Anwendungen soll es in Zukunft den Anwendungsentwicklern leichter machen, schnell und flexibel auf die Anforderungen der Anwender bzw. des Marktes reagieren zu können.

Die nachfolgende Abbildung zeigt schematisch die Verbundanwendungen:



Verbundanwendung Schema

Die zu einer Verbundanwendung erweiterte Notes-Anwendung verfügt neben den Notes-typischen Komponenten über spezielle Erweiterungen, welche neue Kommunikationswege

zwischen den Notes-Bereichen etablieren. Die Bestandteile einer Verbundanwendung – die so genannten Komponenten – sind Verweise auf Gestaltungselemente in einer Notes Anwendung. Beispiele für solche Verweise sind Ansichts- oder Dokumentenlinks.

Verbundanwendungen sind seit der Version 8 von Lotus Notes bzw. Domino verfügbar und lassen sich sowohl zentral auf dem Domino Server, als auch lokal auf dem Lotus Notes Client ausführen. Das besondere: Verbundanwendungen können auch auf einem WebSphere Portal Server ausgeführt werden. Der Zugriff auf diese Anwendung ist mittels Lotus Komponenten so realisiert, dass diese Anwendung mit Domino / Notes genutzt werden können. Allerdings ist diese Ausführung auf „vollwertige“ Lotus Notes Clients beschränkt. Der Einsatz mit Domino Webanwendungen ist nicht realisiert.

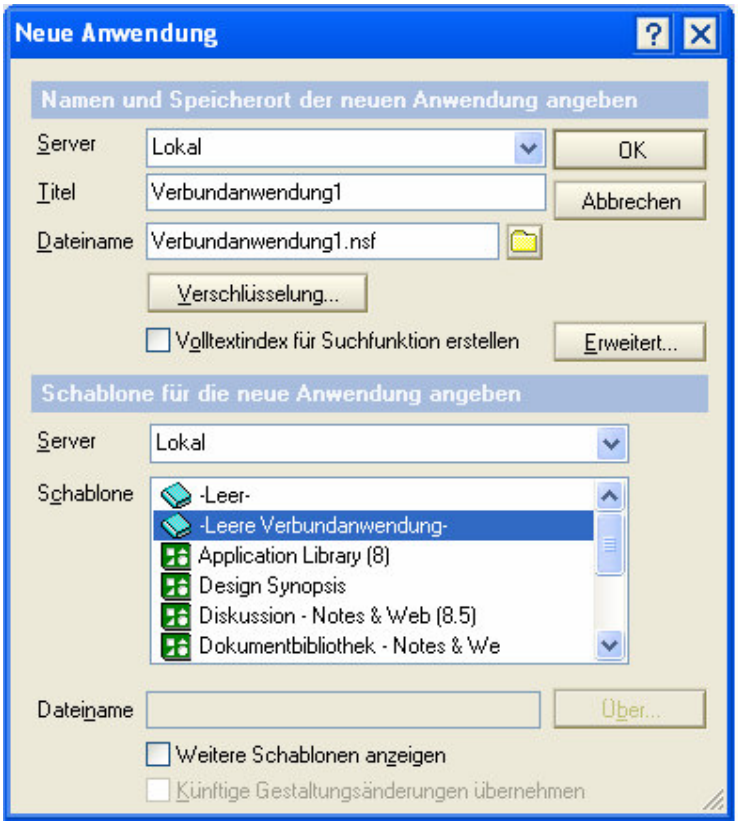
Das mit der Einführung der Verbundanwendung eine Domino Installation nicht automatisch zu einer SOA Anwendung mutiert, dürfte sicher klar sein. Dennoch bilden die Verbundanwendung die Basis für die Integration von Domino Anwendungen in eine SOA Umgebung. Damit stehen einem Unternehmen natürlich enorm viele Möglichkeiten der zentralen Nutzung von Daten und Anwendungen zur Verfügung. Allerdings: Solch ein Konstrukt muss wachsen und bedarf einer gewissen Erfahrung und Planung. Der Ansatz der SOA Integration sollte aber nicht abschreckend wirken, sondern ist in der Tat eine Brücke, die man schnell schlagen kann.

Sobald eine Verbundanwendung auf einem Portal Server ausgeführt wird, dann befindet sich diese Anwendung auch in dem Portal Server Katalog. Dabei handelt es sich um ein spezielles Verzeichnis für das Ablegen dieser Anwendungen.

Verbundanwendungen selbst werden über eine spezielle Erweiterung des Notes Clients – den Composite Application Editor“ verwaltet. Für die Basiserstellung der Verbundanwendungen ist zunächst kein Domino Designer erforderlich.

Die Verbundanwendungen werden aus Notes/Domino Sicht als Anwendungen im Notes Sinne erkannt und verwaltet. Damit stehen diese Anwendungen auch in der Auflistung zur Verfügung, wenn es um die Arbeit mit Notes Anwendung geht. Ein Beispiel dafür ist das Öffnen einer neuen Anwendung. Gewöhnlich wählt man dazu die Menüoption „Datei / Öffnen / Lotus Notes Anwendung“. Neben den klassischen NSF basierten Anwendungen finden wir in der Auflistung der verfügbaren Anwendungen auch die Verbundanwendungen.

Ähnlich verhält es sich mit der Neuanlage einer Verbundanwendung. Diese kann über die Funktion „Datei / Anwendung / Neu“ definiert werden.



Neue Verbundanwendung

Wie man anhand der vorhergehenden Abbildung sehen kann, befindet sich in den aufgelisteten Schablonen nun auch der neue Eintrag „Leere Verbundanwendung“.

Das diese neue Anwendung noch mit Leben gefüllt werden muss, ist selbstverständlich. Diese Aktionen werden wir jedoch an späterer Stelle kennenlernen.

Schauen wir uns die Hintergründe der Verbundanwendungen noch ein wenig genauer an. Der Entwicklungsansatz, der hinter den Verbundanwendungen steckt, ist nicht neu und durchaus mit anderen IT-Bereichen zu vergleichen. Nehmen wir zum Beispiel das ILE Konzept, dass auf dem System i bereits seit Jahren flexible und modulare Anwendungsentwicklung ermöglicht. Diesen Ansätzen ist eines gemein: Das beliebige Kombinieren von bestehenden Programmbereichen zu einer Anwendung. Im Falle von Lotus Notes sind diese zum Beispiel Web Services, die neu erstellt oder bereits existierend zur Realisierung einer neuen Anforderung eingesetzt werden können.

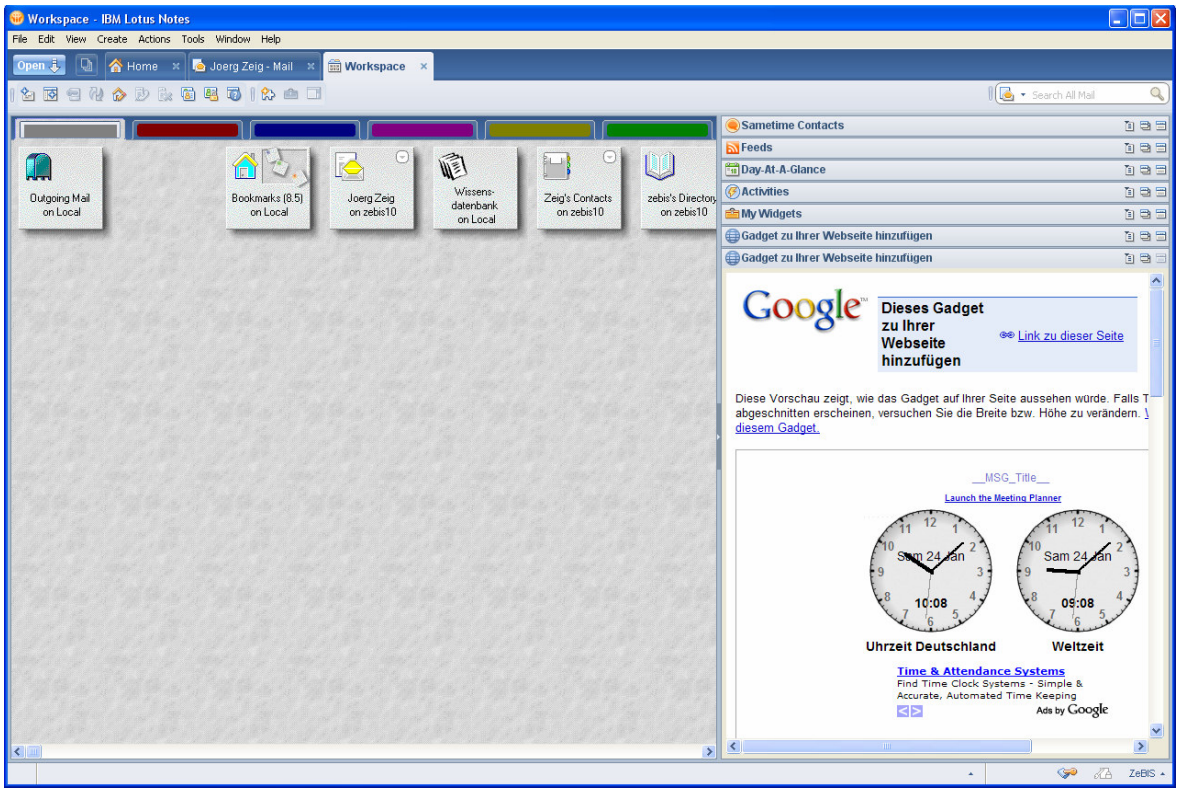
Webanwendungen bestehen heute in der Regel aus einzelnen Teilfunktionen, die zusammengefasst eine Anwendung darstellen. Portalanwendungen sind ein Beispiel für solche Kombinationen von Funktionen.

Der Begriff „Service orientiert“ hat einen gewissen Stellen- und auch Abschreckungswert erlangt. Verbund Anwendungen sind eine besondere Form des serviceorientierten Gedankens.

Das IBM Lotus Notes auf diese Form der Technologie ausgerichtet hat, kann als zentraler Schlüssel für die Zukunft von Notes angesehen werden. Mit entsprechender Kenntnis ist ein Notes-Anwendungsentwickler damit in der Lage, neben den Notes Funktionen auch Standards aus den Bereichen Web, Web 2.0 und beispielsweise .Net zu neuen Anwendungen zusammen zu fassen. Auf diese Weise lassen sich Optimierungen auf Basis von Notes Applikationen

realisieren, was der Notes-Umgebung sicher zuträglich sein kann. Der Begriff der Integration gewinnt damit für Notes eine neue Bedeutung.

Lotus Symphony, Widgets und Sidebar Funktionen sind Beispiele für die Integration von zusätzlichen Funktionen in eine Notes.



Sidebarinhalte

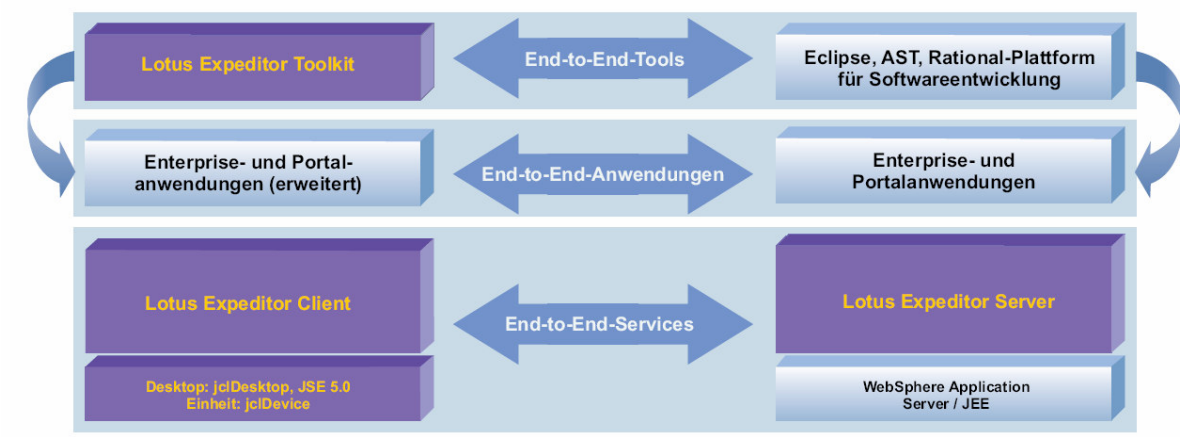
Natürlich können Komponenten aus den unterschiedlichsten Welten kommen. Das zeigen auch Lösungen, bei denen sich beispielsweise System i Bildschirme in Form einer 5250-Integration in eine Notes-Anwendung integrieren lassen. HATS oder auch WebFacing sind IBM Produkte, die in diesem Zusammenhang genutzt werden können. Das die Integration eines „Greenscreens“ in eine Verbund Anwendung sicher kein Lösungsansatz für externe Anwender darstellt, dürfte klar sein. Wenn es jedoch um die Integration von Anwendungen und Funktionen geht, dann bildet diese Option eine sinnvolle Ergänzung für System i Betreiber, welche noch nicht über eine grafische Oberfläche der Anwendungen verfügen.

Eclipse selbst liefert die Werkzeuge für die Verwendung der Webservices. IBM hat – wohl in weiser Voraussicht – in den letzten Jahren mehr und mehr Funktionen in einer Webservice Technologie erstellt. Sollten solche Technologien für Teilbereiche nicht zur Verfügung stehen, können diese unter Umständen relativ leicht umgesetzt werden.

Auch wenn es nicht nur Eclipse basierte Werkzeuge sind, die für das Erstellen und Verwalten von Verbund Anwendungen genutzt werden können, liegt der Fokus von IBM klar auf dem Einsatz der Eclipse basierten und für IBM Produkten erweiterten Tools wie zum Beispiel WebSphere- und Rational Produkte. Nicht unerwähnt bleiben darf an dieser Stelle auch der Domino Designer, der neben der originären Entwicklung von Notes Anwendungen auch die Funktionen für das Arbeiten mit Verbund Anwendungen beinhaltet. Das wohl am einfachsten anzuwendende Toolset steht in Form des Compossit Application Editors, der im Lieferumfang von Lotus Notes ab der Version 8 als zusätzliche Option zu enthalten ist.

Wer jetzt in dem Glauben ist, man müsse über einen umfassenden Programmierskill in den unterschiedlichen Bereichen verfügen, der sei beschwichtigt. Die Philosophie der IBM ist, dass Verbund Anwendungen von erfahrenen Benutzern mit entsprechenden Berechtigungen selbst erstellt werden können. Dabei sind keine speziellen Kenntnisse in den verschiedenen Programmiersprachen oder Anwendungen erforderlich. Natürlich muss eine ausreichende Kenntnis über die Werkzeuge und die bereits bestehenden Bausteine vorhanden sein, damit die Erstellung der Verbund Anwendungen möglich ist.

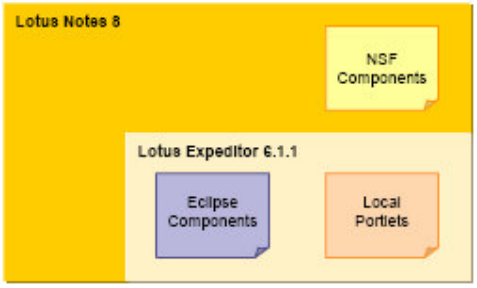
Die Kombination von Notes Funktionen mit Eclipse Bereichen und PlugIns ist letztlich das, was die Verbund Anwendungen ausmacht.



Lotus Expeditor (Quelle:IBM)

Im Lotus Notes Umfeld sprechen wir von NSF Komponenten, im Eclipse Umfeld stehen uns Eclipse Komponenten zum Einsatz zur Verfügung.

Beginnend mit der Version 8 hat IBM die Architektur der Notes Anwendungen so angepasst, dass diese dem folgenden Aufbau entsprechen:



Verbundanwendungen im Notes Umfeld Quelle:IBM

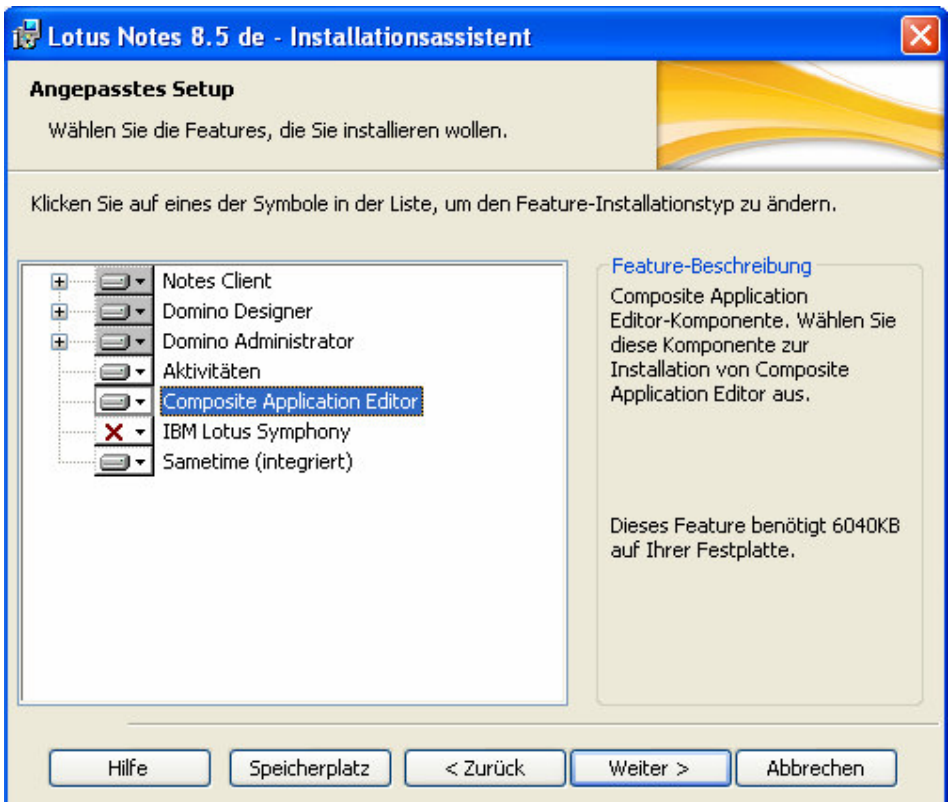
Verbunden mit dieser Umstrukturierung ist der Wandel von den Client basierten Anwendungen hin zu serverbasierten Client Funktionen. Der Lotus Expeditor bildet dabei eine der Grundlagen. Erweiterungen in Lotus Notes, die ab der Version 8 zur Verfügung stehen, spielen zudem Schlüsselrollen. NSF Komponenten sind weitere Neuerungen, die für das Öffnen von Notes bedeutsam sind. Bei einer NSF Komponente handelt es sich um eine Notes Anwendung, die in eine Verbund Anwendung eingebettet wurde.

Notes Designbestandteile, wie zum Beispiel Ansichten, Masken oder Dokumente lassen sich mit Hilfe des Composite Application Editors leicht für die Nutzung mit Verbund Anwendungen aufbereiten.

Dieses spezielle Tool ist in dem Lieferumfang des Lotus Notes Clients zwar enthalten, die Installation ist jedoch optional.

Um die Funktionen des Composite Application Editors nutzen zu können, müssen diese erst einmal installiert werden. Dazu wählen wir die entsprechende Option von der Installations-CD bzw. dem heruntergeladenen Installationsverzeichnis für den Lotus Notes Client aus.

Ein Beispiel für den zu installierenden Bereich finden Sie in der nachfolgenden Abbildung:



Composite Application Editor Installation

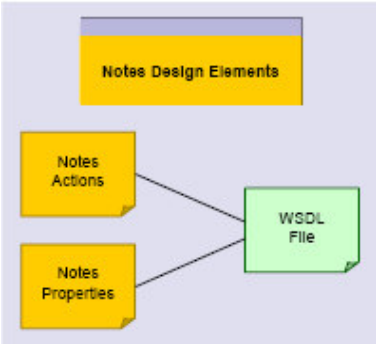
Details zu dem Umgang mit dem Composite Application Editor finden Sie an einer anderen Stelle.

Derzeit (Stand Frühjahr 2009) lassen sich die folgenden Notes Designelemente zusammen mit Verbund Anwendungen nutzen:

- Ansichten
- Masken
- Ordner
- Rahmen
- Seiten
- Navigatoren

Damit eine ursprüngliche Notes Komponente in der Lage ist, mit anderen Komponenten zusammen eingesetzt werden zu können, müssen zusätzliche Informationen definiert und genutzt werden. Mittels besonderer Eigenschaften „vertragen“ sich die Notes Komponenten dann auch mit der Nicht-Notes-Welt.

Eine NSF Komponente stellt eine Art Link zu dem eigentlichen Notes Element dar



Design Elemente Quelle: IBM

Damit die Verbund Anwendungen ideal miteinander verbunden und kombiniert werden können, stehen in Lotus Script neue APIs zur Verfügung, mittels deren die Arbeit mit Verbund Anwendungen definiert werden kann.

Nachfolgend eine Auflistung der in der Version 8 zur Verfügung gestellten Lotus Script Erweiterungen:

- NotesSession
  - NotesSession.GetPropertyBroker([Namespace])

    Returns NotesPropertyBroker. If no argument supplied, *Namespace* argument must be supplied to all Get/Set calls. If the *Namespace* argument is supplied, PropertyBroker is defaulted to a specific namespace. Get/Set calls can be made without the *Namespace* argument but if it is supplied, the argument overrides the default.
- NotesPropertyBroker
  - NotesPropertyBroker.InputPropertyContext (property)

    Returns an array of input NotesProperties. For current releases, only the first member of the array will be populated.
  - NotesPropertyBroker.Namespace (property)

    String. Provides the Read/Write default namespace of interest for PropertyBroker.
  - NotesPropertyBroker.HasProperty(Name,[Namespace])

    Boolean. Indicates whether a new property exists.
  - NotesPropertyBroker.GetProperty(Name,[Namespace])

    Returns NotesProperty.

- NotesPropertyBroker.GetPropertyValue(Name,[Namespace])

  Returns the value for a specified property.
- NotesPropertyBroker.SetPropertyValue(Name, Value, [Namespace])

  Returns NotesProperty. It creates a backend object and populates it with the value. This method throws an error if the property name is not defined by WSDL.
- NotesPropertyBroker.ClearProperty

  A new method that purges any new or modified property, not for input properties.
- NotesPropertyBroker.Publish

  A new method that publishes all new and modified properties.

► NotesProperty

- NotesProperty.Values (property)

  Read/Write for simple properties: Returns a Variant array of type String. Single value properties contain data in first array element. As input, accepts Variant or Array of type String. The Array must be homogeneous. At this time, an Array can contain only one String entry. Input properties cannot be set.
- NotesProperty.Name (property)

  Read only. Returns property name as string.
- NotesProperty.TypeName (property)

  Read only. Returns type name as string.
- NotesProperty.NameSpace (property)

  Read only. Returns namespace as string.
- NotesProperty.Title (property)

  Read only. Returns title as string.
- NotesProperty.isInput (property)

  Read only. Returns true if this is an input property, otherwise false. Returns Boolean.
- NotesProperty.Description (property)

  Read only. Returns description as string.
- NotesProperty.Publish( )

  A new method that publishes this property if modified.
- NotesProperty.Clear( )

Lotus Notes Komponenten verfügen, wenn diese auf Basis von NSF erstellt wurden, über spezielle XML Erweiterungen, welche in der Notes Anwendung gespeichert werden und bei der Verwendung von Verbund Anwendungen eine große Rolle spielen. Mit verschiedenen Werkzeugen sind wir in der Lage, bestehende klassische Notes Anwendungen so zu erweitern, dass diese mit XML Zusatzinformationen ausgestattet, in Verbund Anwendungen integriert werden können. Allerdings gibt es hier einige Beschränkungen, die wir im weiteren Verlauf noch kennen lernen werden.

Grundsätzlich können diese Anwendungen an einem beliebigen Ort abgelegt und für die neuen Anwendungen verfügbar gemacht werden. Sowohl die Speicherung innerhalb von Notes, aber auch die Verfügbarkeit der Anwendungen auf einem Portal Server sind Möglichkeiten, Notes Anwendungen für Composite Applications nutzbar zu machen.

Das besondere dabei ist, dass die einzelnen Bausteine einer Anwendung auf verschiedenen Servern verfügbar gemacht werden – wir sprechen hier vom „Deployen“. So kann eine Verbund Anwendung zum Beispiel aus den folgenden Bereichen bestehen:

- Notes Komponenten, die auf einem Domino Server deployed werden.
- Eclipse Komponenten, die auf einem Web Server deployed werden.

- Lokale Portlets

Einige der Standard Notes Anwendungen sind „von Hause aus“ als Composite Anwendungen ausgeliefert. Darunter fallen zum Beispiel die Mailfunktion, das Adressbuch oder der Lotus Notes Kalender.

Zusatzinformationen, die in speziellen WSDL Dateien abgelegt werden, beinhalten erweiterte Informationen, Eigenschaften und Aktionen, die für den Einsatz mit Composite Anwendungen benötigt werden.

Über eine spezielle MailComponent. WSDL Datei werden zusätzliche Informationen zu dem Mailbereich bereitgestellt. Diese umfassen unter anderem die folgenden Eigenschaften:

Property Name	Property Type	Assigned to
EmailAddressOutput	emailAddress	AvailabilityIcon (Inbox) AvailabilityIcon (All Documents)
CommonNameOutput	commonName	Sender (Inbox) Who (All Documents)
SubjectOtupt	string	Subject (Inbox) Subject (All Documents)
NotesURLOutput	notesURL	Mapped to NotesSelectedDocument

Neben den Eigenschaften stehen auch spezielle Aktionen zum Einsatz bereit:

Action Name	Notes
NewMemoUsingToField	This action takes any field as a parameter and places it in the SendTo field for a newly composed memo.
NewMemoUsingMailtoURL	This action takes a string in the format of a MailToURL, parses it, and places the contents in the appropriate fields of a newly composed memo.
NewMemuUsingEmailAddress	This action takes the provided e-mail address and places it in the SendTo field of a newly composed.

Ein weiteres Beispiel für eine Standard Notesanwendung, die bereits vorbereitet ist für Verbund Anwendungen ist der Notes Kalender.

Der Notes Kalender verfügt seit der Version 8 über die folgenden zusätzlichen Eigenschaften:

Property Name	Property Type
Current Date Parameter	DateType
Current Date Range Parameter	DataRangeType

Ähnlich wie auch bei dem Mailbereich, so stehen für den Kalender neue Aktionen zum Einsatz bereit:

Action Name	Notes
ChangeCalendarDate	Sets the date on the calendar.
ChangeNavigatorDateRange	Sets range of secondary visible dates in the navigation pane. These are dates that are active but not selected. For example, in a one work week view, Monday, Tuesday, Wednesday, Thursday, and Friday would be secondary visible dates.
ChangeNavigatorDate	Sets the selected date on the navigation pane.

Schauen wir uns nun einmal die ersten Schritte an, die für die Erstellung einer NSF Komponente notwendig sind.

Wie bereits zuvor erwähnt, bildet eine NSF Komponenten einen Verweis auf einen Notes Gestaltungsbereich.

Basierend auf den Erweiterungen, die IBM dem Lotus Notes Client seit geraumer Zeit verabreicht hat, können wir nahezu jede Standard Notes Anwendung so umsetzen, dass diese als quasi Java Anwendung genutzt werden kann.