



CMOne – Das integrierte Change Management System für i5, iSeries und AS/400



Kurzbeschreibung 5250 und WDSC / RDi

Warum CMOne als Softwareverwaltungs- und Verteilungssystem?.....	2
Der Workflow in CMOne	4
Workflow Beispiel	5
(einfach)	5
Workflow Beispiel	6
(mit Expansionsbereich sowie *Special-Form)	6
Ebenen / Umgebungen.....	7
Anwendungsdefinition	8
Initialisierung der Anwendung.....	9
Anwendungsentwicklung	10
PTF`s	13
Beispiel:	13
Übersicht der CMOne -Funktionalitäten	14

Warum **CMOne** als Softwareverwaltungs- und Verteilungssystem?

Die heutige traditionelle Anwendungsentwicklung wird hauptsächlich dadurch geprägt, dass der Entwickler eine Vielzahl von zeitintensiven und störanfälligen manuellen Arbeiten zu erledigen hat, die eigentlich automatisiert werden können. Hierzu gehören aufwendige Tätigkeiten wie z.B.:

- das manuelle Erstellen von Testumgebungen und Einrichten von Bibliotheksumgebungen.
- das Ermitteln von zu ändernden Objekten, die zu einer Aufgabenstellung gehören, sowie die manuelle Auflistung in Schriftform.
- das Zusammenstellen von abhängigen Objekten, die ebenfalls angepasst werden müssen.
- das Zusammenstellen von abhängigen Objekten, die wegen Aktualisierung rekompiliert werden müssen.
- das Auffinden des zum Objekt gehörigen Quellcodes.
- das manuelle Abändern von Erstellungsbefehlen bei sämtlichen Objekttypen.
- das manuelle Zusammenstellen von geänderten Objekten in Form von Sicherungsvorgängen.
- die manuelle Verteilung der geänderten Objekte auf die Zielsysteme.
- die zeitintensive Installation von geänderten Objekten auf dem Zielsystem sowie das Wiederherstellen von Datenbankdateien durch Zurückkopieren aus vorher gesicherten Dateien.

Was leider in den meisten Fällen aus Zeitgründen vom Entwickler nicht gemacht wird oder oftmals vergessen wird, sind wichtige Aufgaben der modernen Anwendungsentwicklung wie:

- die Trennung von Produktions- und Entwicklungsumgebung.
- die Dokumentation der Änderung in einheitlicher Schriftform.
- die Zusammenstellung von geänderten Objekten auf Basis eines Projektes bzw. einer Aufgabe, um eine Transparenz gegenüber anderen Entwicklern zu ermöglichen.
- die Archivierung von Objekten und Quellen auf den jeweiligen Systemen, aus denen notfalls eine Wiederherstellung der Änderung gemacht werden kann.
- die Bereitstellung von geänderten Objekten zur QS-Kontrolle.
- die Sicherstellung der Synchronität zwischen Objekt und Quelle.

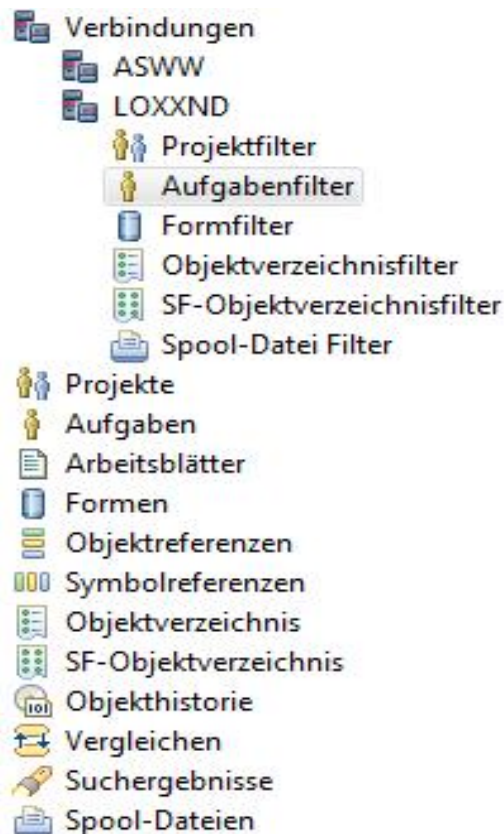
Durch die Vielzahl manueller Schritte werden zwangsläufig immer wieder potentielle Fehlerquellen aufgetan. Man kann noch so ein Perfektionist sein, von Fehlersituationen durch nicht standardisierte Vorgänge im Entwicklungsbereich kann sich niemand freisprechen. Auf die Dauer werden sich in der Anwendung immer wieder Fehler einschleichen, die mit der Zeit eskalieren.

Wenn Sie einmal analysieren, welchen Zeitaufwand Sie als Entwickler für die reine Entwicklung benötigen, werden Sie feststellen, dass diese Zeit ein Bruchteil der gesamten Implementierung ist. Ein Großteil der Zeit wird leider für das oben beschriebene manuelle Handling benötigt. Bei einer Erweiterung einer zentralen Datenbankdatei ist das Zeitverhältnis zwischen Entwicklung und Installation oftmals 30 zu 70. Stellen Sie sich einmal vor, Sie könnten nur 40% Ihrer Entwicklungstätigkeit durch **CMOne** einsparen. Der Idealfall einer modernen Entwicklung sollte so sein, dass sich der Programmierer tatsächlich nur mit der reinen Anwendungsentwicklung beschäftigt.

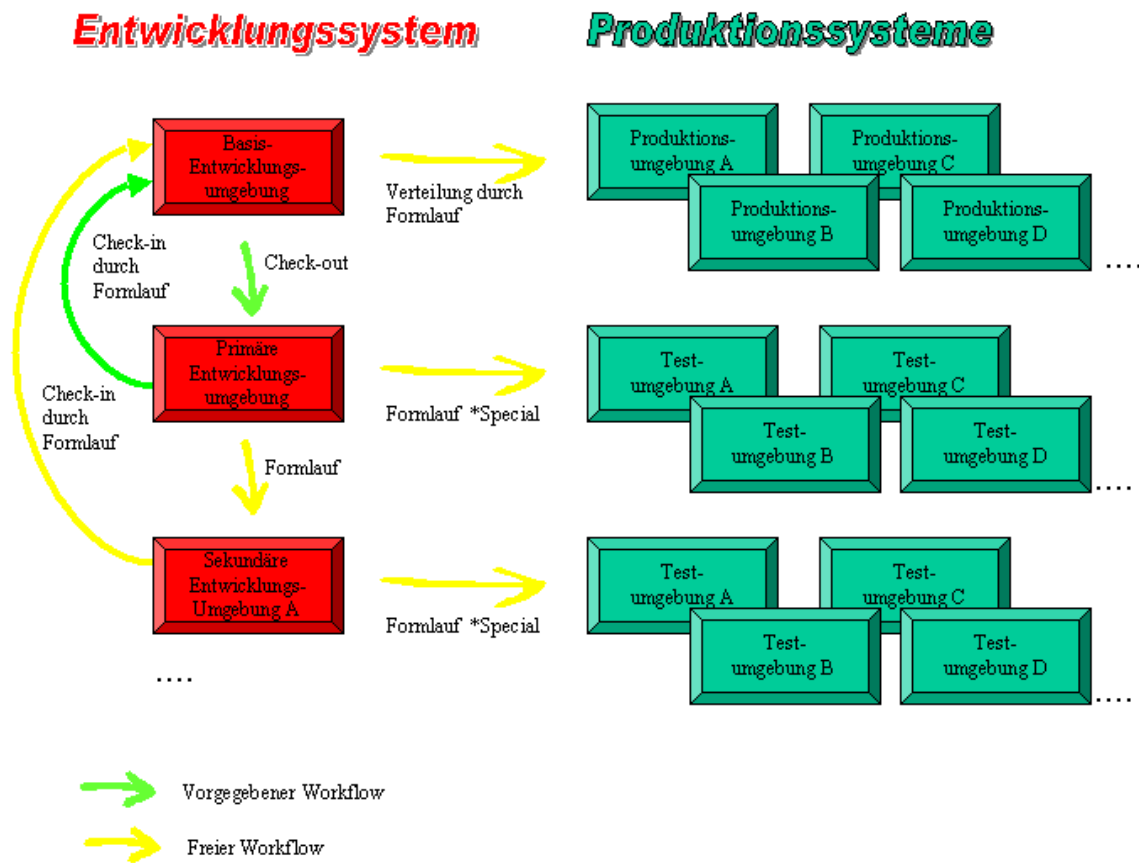
Es gibt auf dem Markt eine Vielzahl von Tools, die Teilbereiche der Anwendungsentwicklung abdecken. Was jedoch wirklich gebraucht wird, ist ein integriertes Change Management System wie **CMOne**, mit dem die gesamte Palette von Funktionalitäten einer moderne Anwendungsentwicklung abgedeckt wird. Dieses Change Management System sollte darüber hinaus auch so offen sein, dass die Integration von Produkten anderer Anbieter ermöglicht wird.

Die Task Force IT-Consulting GmbH hat mit dem neu entwickelten Produkt **CMOne** ein integriertes Change Management System geschaffen, welches dem Entwickler erlaubt, sich um sein Kerngeschäft, der Anwendungsentwicklung zu kümmern. Der gesamte anfallende Overhead wird von **CMOne** in automatisierten und individuell zu definierenden Prozessen übernommen. **CMOne** Next Generation steht als Plugin für eine moderne Entwicklungsplattform unter Eclipse-Technik für WDSO und RDi bereit.

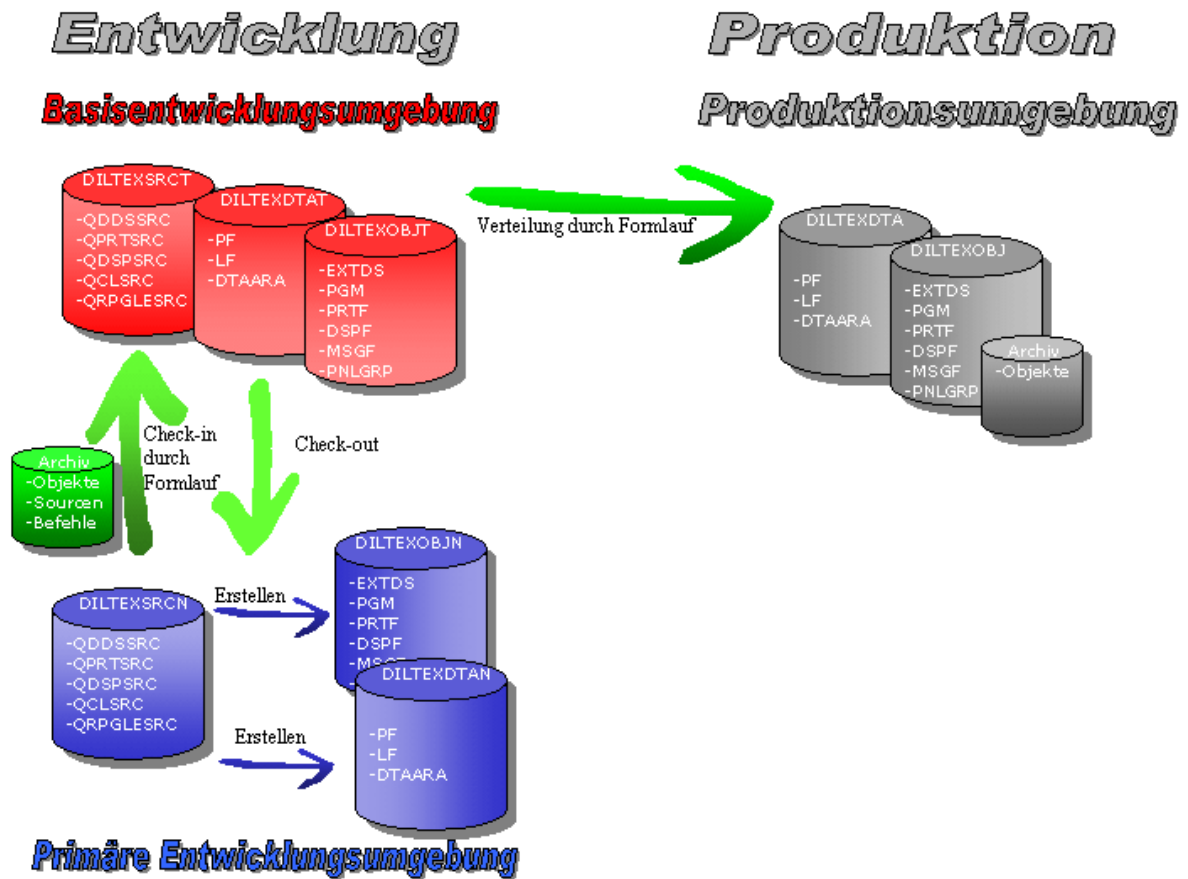
CMOne ist für Unternehmen mit eigener Softwareentwicklung als auch für reine Software-Dienstleister konzipiert. Nachfolgend werden die grundlegenden Strukturen von **CMOne** beschrieben, die zu einem groben Überblick des Change Management Systems führen sollen.



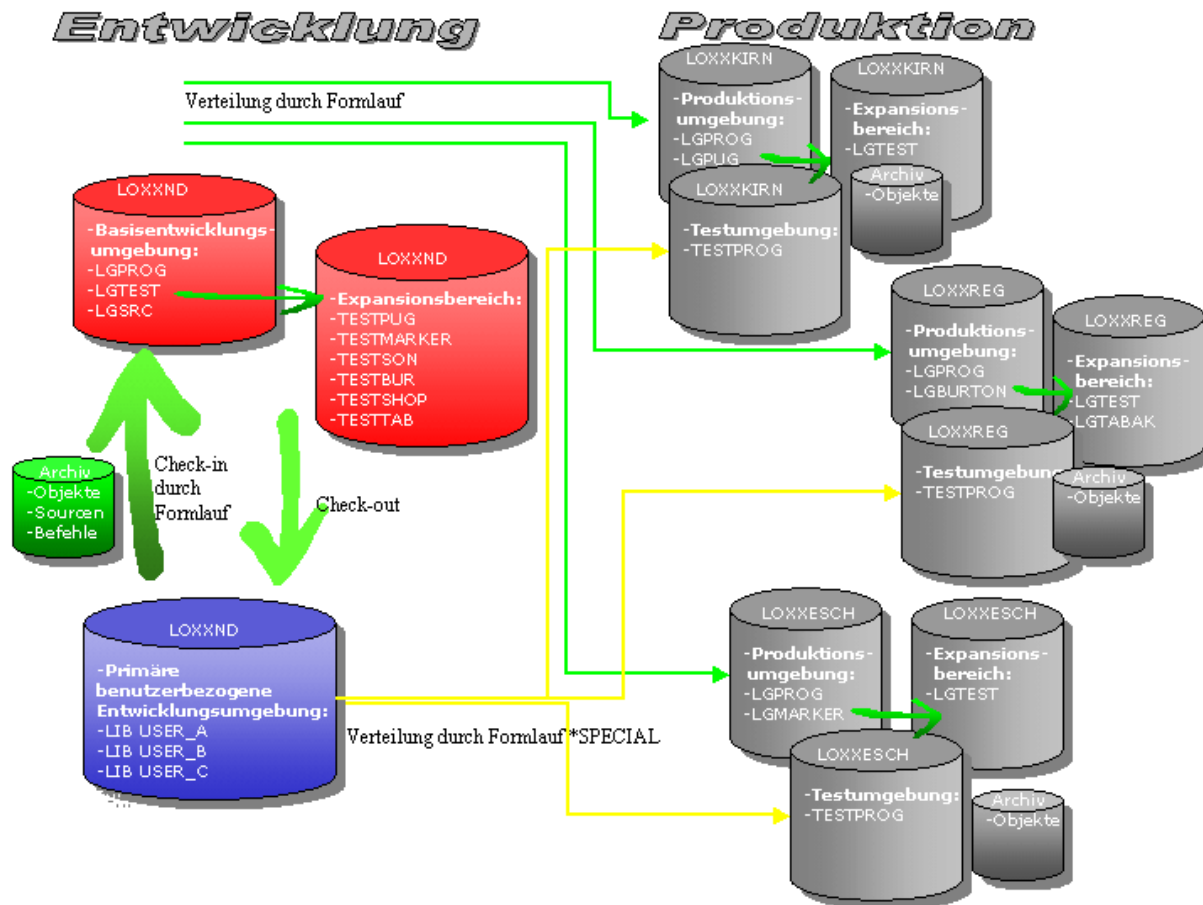
Der Workflow in CMOne



Workflow Beispiel (einfach)



Workflow Beispiel (mit Expansionsbereich sowie *Special-Form)



Ebenen / Umgebungen

CMOne ist hierarchisch aufgebaut, um jede individuelle Anwendung des Kunden flexibel abbilden zu können. Die Definition der Stammdaten erfolgt in der 5250 Green-Screen Umgebung und kann grundsätzlich auf verschiedenen Ebenen erfolgen:

- Globalen Ebene => hier definierte Daten gelten für alle Anwendungen.
- Anwendungsebene => hier definierte Daten gelten nur für die jeweilige Anwendung.
- Umgebungsebene => hier definierte Daten gelten für die jeweilige Umgebung, wie Entwicklungsumgebung und Produktionsumgebung innerhalb einer Anwendung.

Die Entwicklungsumgebung ist in folgende Bereiche gegliedert:

- Basisentwicklungsumgebung => hier definierte Daten beschreiben den Bereich der Anwendung, der die Quellen und Objekte enthält, die für eine spätere Modifikation herangezogen werden können.
- Primäre Entwicklungsumgebung => hier definierte Daten beschreiben den Bereich der Anwendung, der die Quellen und Objekte enthält, die aktuell modifiziert werden. Dies ist die eigentliche Testumgebung, in die der Entwickler seine Änderungen erstellt.
- Die primäre Entwicklungsumgebung kann in vier variable Bereiche gegliedert werden:
 - Anwendungsbezogene primäre Entwicklungsumgebung => hier wird beschrieben, wie die primäre Entwicklungsumgebung bezogen auf die Anwendung auszusehen hat.
 - Projektbezogene primäre Entwicklungsumgebung => hier wird bei Bedarf beschrieben, wie ein Teil der primären Entwicklungsumgebung bezogen auf ein definiertes Projekt auszusehen hat.
 - Aufgabenbezogene primäre Entwicklungsumgebung => hier wird im Bedarfsfall beschrieben, wie ein Teil der primären Entwicklungsumgebung bezogen auf eine definierte Aufgabe innerhalb eines Projektes auszusehen hat.
 - Benutzerbezogene primäre Entwicklungsumgebung => hier kann im Bedarfsfall beschrieben werden, wie ein Teil der primären Entwicklungsumgebung bezogen auf ein definiertes Benutzerprofil auszusehen hat.
- Sekundäre Entwicklungsumgebung => hier könnte beispielsweise ein Bereich definiert werden, in dem die Quellen und Objekte für einen letzten Abschlusstest durch die Qualitätssicherung abgelegt werden, bevor die Quellen und Objekte zu den Produktionsumgebungen verteilt werden.

Bei der Produktionsumgebung handelt es sich um den Bereich, in dem die Anwendung produktiv arbeitet. Jede eingesetzte Produktionsumgebung ist hier zu definieren. Eine Produktionsumgebung kann sich entweder auf dem Entwicklungsrechner oder auf verschiedenen Remote-Rechnern befinden.

Anwendungsdefinition

In **CMOne** ist für jede eingesetzte Anwendung eine individuelle Anwendungsdefinition zu erstellen und erfolgt in der 5250 Green-Screen Umgebung. Der Name der Anwendung setzt sich zusammen aus den Feldern Anwendung und wahlweise, je nach Bedarf Version, Release und Modifikationsstufe. Damit kann man die Anwendung in der Weise definieren, wie man es von IBM kennt.

Eine der wichtigsten Aufgaben bei der Konfiguration der Anwendung ist die einmalige Definition, mit welchen Quellen- und Objektarten in einer Anwendung gearbeitet wird und in welchen Quellendateien und Bibliotheken die Quellen und Objekte gespeichert werden. Dazu werden in der Anwendung Gruppen und Objekttypen mit deren Vorkommen erstellt, auf deren Basis die Namen der Quellendateien und der Bibliotheken hinterlegt werden können.

Auf der Anwendungsebene erfolgt die Zuordnung der Gruppen zur Anwendungsdefinition. Typischerweise sind Anwendungen gruppiert nach Gruppen für Quellen, Daten, Sprache und einer Gruppe für allgemeine Objekte.

Ebenfalls auf der Anwendungsebene werden der Anwendung die Objekttypen zugeordnet, die in der Anwendung verwendet werden. Dabei wird jedem einzelnen Objekttyp eine Quellen- und/oder eine Objektgruppe zugeordnet, je nachdem ob der Objekttyp aus einer Quelle und/oder einem Objekt besteht. Die Quellen- und Objektgruppen müssen vorher wie oben beschrieben definiert worden sein.



```

17.10.05      Anwendungsdefinitionen (Pflegen)      CM0003M002

Rt./Aw./Vr./Rl./Md.: -----

Auswahl eingeben und Eingabetaste drücken.
2=Ändern 3=Kopieren 4=Löschen 5=Anzeigen 9=Bibl. ändern 10=Sperrn ...

Ausw.  Rlt.  Anw.  Ver.  Rel.  Mod.  Beschreibung      Sperre
-----
  *BAS  ACS      ACS  Basisanwendung      *NO
  *BAS  CML      CML  (Kundenspezifische Anpassungen) *NO
  *BAS  DT       Dilok/400 LVS      *NO
  *BAS  EDI      EDI  (Edifact-Anwendung) *NO
  *BAS  TKS      TKS  (Kundenvariante TKS) *NO

Befehl
====>
F3=Verlassen F4=Bedienerführung F5=Aktualisieren F24=Weitere Tasten
  
```

Den Gruppen bzw. den Objekttypen wiederum werden Vorkommen zugeordnet, die wie der Name schon sagt angeben, wo eine Gruppe bzw. ein Objekttyp innerhalb einer Anwendung vorkommt.

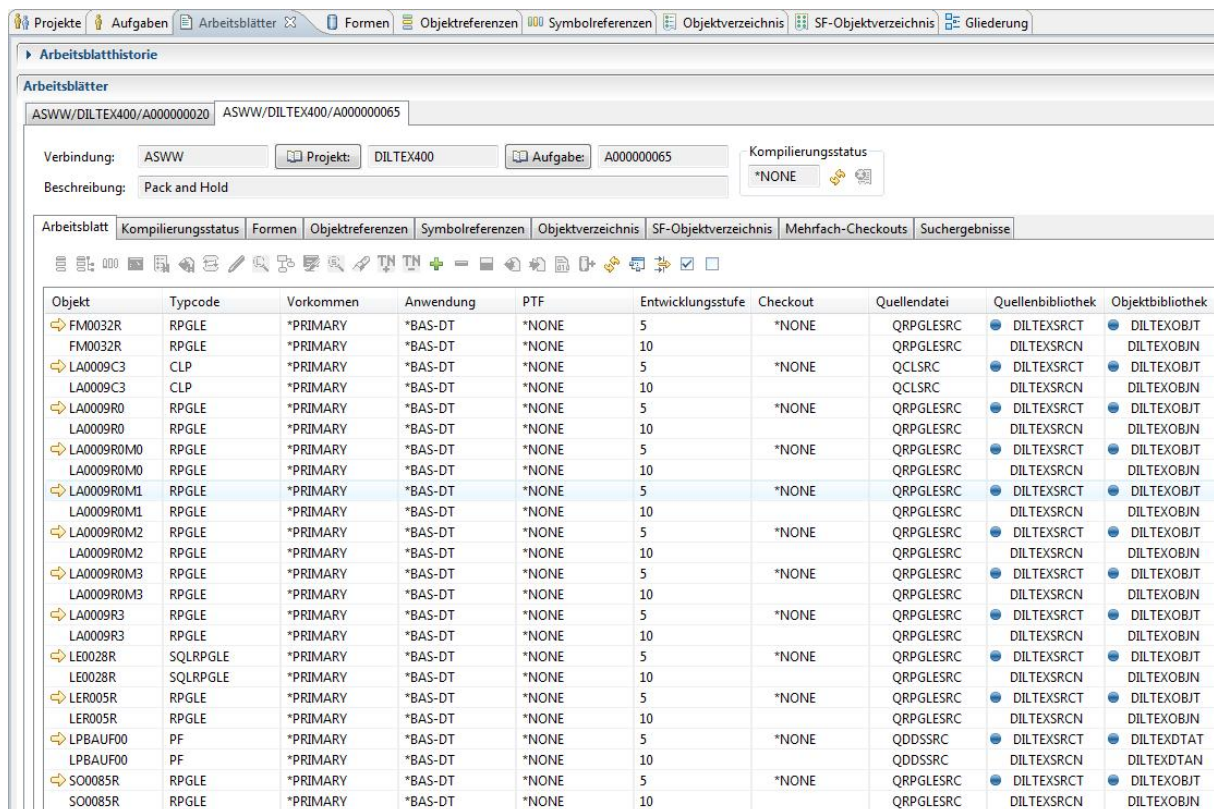
Im Allgemeinen ist einer Gruppe bzw. einem Objekttyp nur ein Vorkommen zugeordnet. Allerdings kommt es bei Sprachobjekten, wie z.B. bei einer Nachrichtendatei vor, dass dasselbe Objekt in unterschiedlichen Bibliotheken vorhanden ist. Beispielsweise in einer Bibliothek, welche Nachrichtendateien mit den deutschen Texten enthält und in einer weiteren Bibliothek, welche Nachrichtendateien mit den englischen Texten enthält. In einem solchen Fall wäre ein Vorkommen für die deutschen Sprachobjekte und ein Vorkommen für die englischen Sprachobjekte zu erstellen.

Auf Umgebungsebene können zu Gruppe/Vorkommen und Objekttyp/Vorkommen Quellendateien und Bibliotheken hinterlegt werden. Durch die Zuordnung der Objekttypen auf Anwendungsebene zu einer Quellen- und Objektgruppe ist **CMOne** in der Lage, die Bibliotheken zu ermitteln, in denen sich die Quellen und Objekte zu den Objekttypen in der jeweiligen Umgebung befinden.

Anwendungsentwicklung

In **CMOne** können Projekte und Aufgaben, die Anwendungen zugeordnet sind, definiert werden. Den Aufgaben wiederum sind Arbeitsblätter zugeordnet. Das Arbeitsblatt stellt den zentralen Arbeitsbereich für die Tätigkeiten eines Entwicklers dar. Alle Objekte, die ein Entwickler für eine bestimmte Aufgabe benötigt, werden auf diesem Arbeitsblatt zugeordnet. So ergibt sich eine umfassende Transparenz, welche Objekte zu einer bestimmten Aufgabe zu modifizieren sind. Ebenfalls werden die Bereiche von Basisumgebung sowie Entwicklungsumgebung getrennt dargestellt. Die Anwendungsentwicklung ist unter 5250 Green-Screen als auch unter WDSC bzw. RDi möglich (**CMOne** Next Generation).

Muss der Entwickler ein Objekt modifizieren, so hat er das Objekt dem Arbeitsblatt hinzuzufügen. Dabei gibt er nur den Namen des Objektes ein, und **CMOne** sucht im Objektverzeichnis nach Objekten mit dem vorgegebenen Namen. Findet **CMOne** nur einen einzigen Eintrag, so wird der Objekttyp und das Vorkommen automatisch übernommen. Findet **CMOne** mehrere Einträge, so werden dem Entwickler die im Objektverzeichnis zum Objekt existierenden Objekttypen mit deren Vorkommen zur Auswahl angeboten. Wird kein Eintrag gefunden (was bei neu zu erstellenden Objekten der Fall ist), so hat der Entwickler zusätzlich den Objekttyp sowie das Vorkommen direkt vorzugeben.

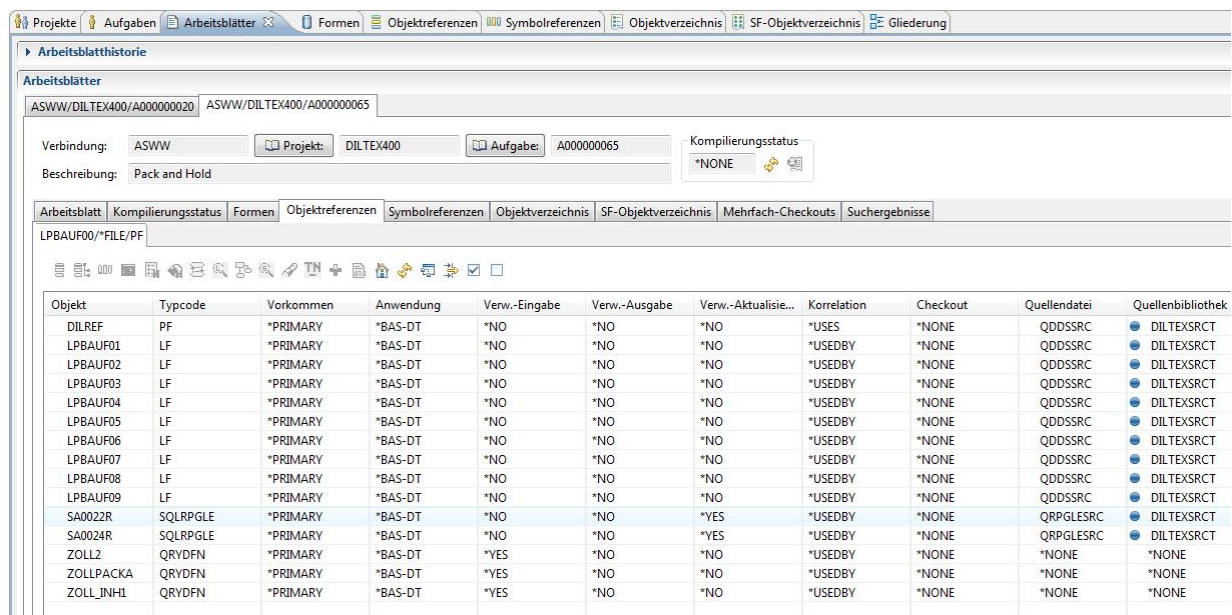


Objekt	Typcode	Vorkommen	Anwendung	PTF	Entwicklungsstufe	Checkout	Quellendatei	Quellenbibliothek	Objektbibliothek
FM0032R	RPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
FM0032R	RPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN
LA0009C3	CLP	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QCLSRC	DILTEXSRCT	DILTEOBJT
LA0009C3	CLP	*PRIMARY	*BAS-DT	*NONE	10		QCLSRC	DILTEXRCN	DILTEOBJN
LA0009R0	RPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
LA0009R0	RPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN
LA0009R0M0	RPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
LA0009R0M0	RPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN
LA0009R0M1	RPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
LA0009R0M1	RPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN
LA0009R0M2	RPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
LA0009R0M2	RPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN
LA0009R0M3	RPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
LA0009R0M3	RPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN
LA0009R3	RPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
LA0009R3	RPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN
LE0028R	SQLRPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
LE0028R	SQLRPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN
LER005R	RPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
LER005R	RPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN
LPBAUF00	PF	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QDDSSRC	DILTEXSRCT	DILTEXDTAT
LPBAUF00	PF	*PRIMARY	*BAS-DT	*NONE	10		QDDSSRC	DILTEXRCN	DILTEXDTAN
SO0085R	RPGLE	*PRIMARY	*BAS-DT	*NONE	5	*NONE	QRPGLSRC	DILTEXSRCT	DILTEOBJT
SO0085R	RPGLE	*PRIMARY	*BAS-DT	*NONE	10		QRPGLSRC	DILTEXRCN	DILTEOBJN

Nachdem das Objekt auf dem Arbeitsblatt hinzugefügt wurde, muss es ausgecheckt werden. Dabei werden Quelle und/oder Objekt von der Basisentwicklungsumgebung in die primäre Entwicklungsumgebung kopiert. Vorher wird jedoch geprüft, ob nicht ein anderer Entwickler das Objekt bereits ausgecheckt hat. In diesem Fall wird dann eine entsprechende Fehlermeldung generiert.

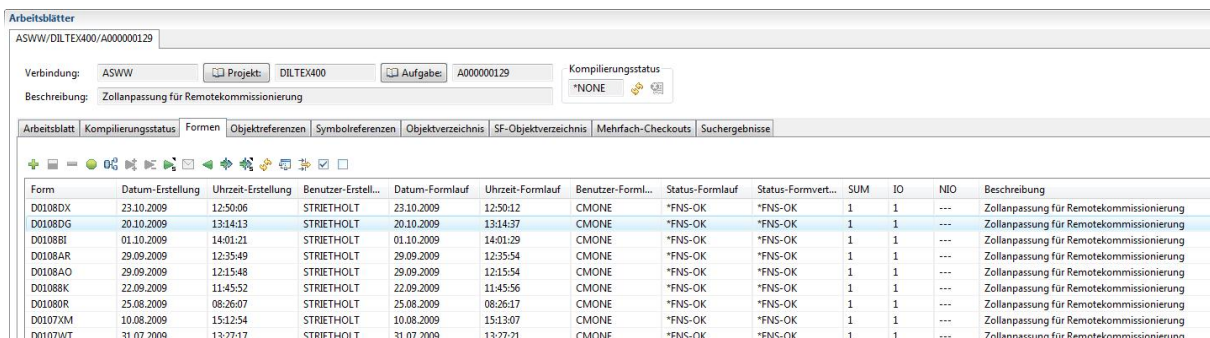
Nachdem das Objekt ausgecheckt ist, kann es mit SEU, SDA, RLU, CODE/400 oder sonstigen Editoren wie LPEX oder Screen Designer bearbeitet werden. Sind dem Objekt Vorlauf-, Nachlauf- oder ein vom Standarderstellungsbefehl abweichender Erstellungsbefehl hinzuzufügen, so kann das auf dem Arbeitsblatt getan werden. Diese Befehle werden gespeichert, damit diese bei der nächsten Modifizierung des Objektes dann wieder herangezogen werden können. Nachdem die Änderung an der Quelle vorgenommen wurde, wird das Objekt in die primäre Entwicklungsumgebung kompiliert.

Vom Arbeitsblatt besteht die Möglichkeit, sich die Objektreferenzen zu einem Objekt anzeigen zu lassen. Wurde eine physische Datei geändert, so müssen auch die dazugehörigen logischen Dateien und sämtliche Module, Programme und Serviceprogramme in die Testumgebung rekompiliert werden. Ruft man die Objektreferenzen auf, so gibt es eine Option, mit der die abhängigen Objekte in die primäre Entwicklungsumgebung rekompiliert werden. Von hier aus ist es auch möglich, ein abhängiges Objekt direkt dem Arbeitsblatt hinzuzufügen, damit Änderungen an der Quelle vorgenommen werden können.



Objekt	Typcode	Vorkommen	Anwendung	Verw.-Eingabe	Verw.-Ausgabe	Verw.-Aktualisie...	Korrelation	Checkout	Quellendatei	Quellenbibliothek
DILREF	PF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USES	*NONE	QDDSSRC	DILTEXSRCT
LPBAUF01	LF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USEDDBY	*NONE	QDDSSRC	DILTEXSRCT
LPBAUF02	LF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USEDDBY	*NONE	QDDSSRC	DILTEXSRCT
LPBAUF03	LF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USEDDBY	*NONE	QDDSSRC	DILTEXSRCT
LPBAUF04	LF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USEDDBY	*NONE	QDDSSRC	DILTEXSRCT
LPBAUF05	LF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USEDDBY	*NONE	QDDSSRC	DILTEXSRCT
LPBAUF06	LF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USEDDBY	*NONE	QDDSSRC	DILTEXSRCT
LPBAUF07	LF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USEDDBY	*NONE	QDDSSRC	DILTEXSRCT
LPBAUF08	LF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USEDDBY	*NONE	QDDSSRC	DILTEXSRCT
LPBAUF09	LF	*PRIMARY	*BAS-DT	*NO	*NO	*NO	*USEDDBY	*NONE	QDDSSRC	DILTEXSRCT
SA0022R	SQLRPGLE	*PRIMARY	*BAS-DT	*NO	*NO	*YES	*USEDDBY	*NONE	QRPGLESRC	DILTEXSRCT
SA0024R	SQLRPGLE	*PRIMARY	*BAS-DT	*NO	*NO	*YES	*USEDDBY	*NONE	QRPGLESRC	DILTEXSRCT
ZOLL2	QRVDFN	*PRIMARY	*BAS-DT	*YES	*NO	*NO	*USEDDBY	*NONE	*NONE	*NONE
ZOLLPACKA	QRVDFN	*PRIMARY	*BAS-DT	*YES	*NO	*NO	*USEDDBY	*NONE	*NONE	*NONE
ZOLL_INHL	QRVDFN	*PRIMARY	*BAS-DT	*YES	*NO	*NO	*USEDDBY	*NONE	*NONE	*NONE

Sind alle Objekte auf dem Arbeitsblatt modifiziert, so kann eine Form erstellt werden. Eine Form übernimmt die Installation von modifizierten Quellen und Objekten in einer bestimmten Umgebung. In unserem Beispiel werden die Quellen und Objekte, welche sich in der primären Entwicklungsumgebung befinden, in die Basisumgebung installiert. Zur Erstellung der Form markiert der Entwickler alle Positionen auf dem Arbeitsblatt mit einer bestimmten Option. Dabei wird die Form erstellt und alle markierten Positionen werden der Form hinzugefügt.

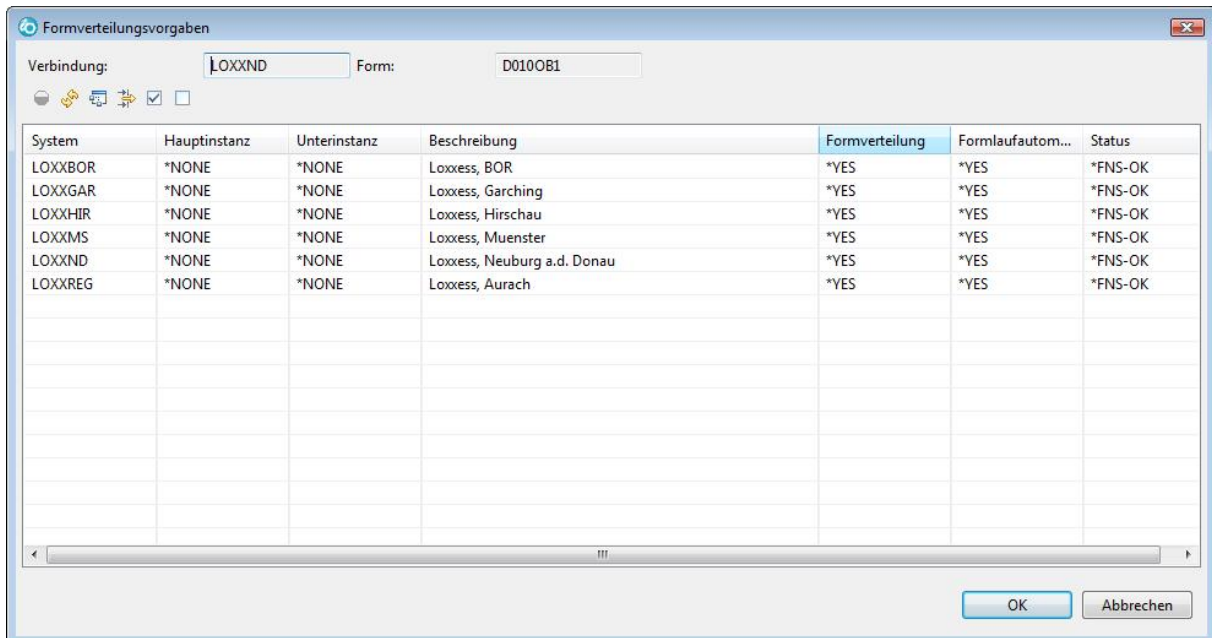


Form	Datum-Erstellung	Uhrzeit-Erstellung	Benutzer-Erstell...	Datum-Formlauf	Uhrzeit-Formlauf	Benutzer-Forml...	Status-Formlauf	Status-Formvert...	SUM	IO	NIO	Beschreibung
D0108DX	23.10.2009	12:50:06	STRIETHOLT	23.10.2009	12:50:12	CMONE	*FNS-OK	*FNS-OK	1	1	---	Zollanpassung für Remotekommissionierung
D0108DG	20.10.2009	13:14:13	STRIETHOLT	20.10.2009	13:14:37	CMONE	*FNS-OK	*FNS-OK	1	1	---	Zollanpassung für Remotekommissionierung
D0108BI	01.10.2009	14:01:21	STRIETHOLT	01.10.2009	14:01:29	CMONE	*FNS-OK	*FNS-OK	1	1	---	Zollanpassung für Remotekommissionierung
D0108AR	29.09.2009	12:35:49	STRIETHOLT	29.09.2009	12:35:54	CMONE	*FNS-OK	*FNS-OK	1	1	---	Zollanpassung für Remotekommissionierung
D0108AO	29.09.2009	12:15:48	STRIETHOLT	29.09.2009	12:15:54	CMONE	*FNS-OK	*FNS-OK	1	1	---	Zollanpassung für Remotekommissionierung
D01088K	22.09.2009	11:45:52	STRIETHOLT	22.09.2009	11:45:56	CMONE	*FNS-OK	*FNS-OK	1	1	---	Zollanpassung für Remotekommissionierung
D01080R	25.08.2009	08:26:07	STRIETHOLT	25.08.2009	08:26:17	CMONE	*FNS-OK	*FNS-OK	1	1	---	Zollanpassung für Remotekommissionierung
D0107XM	10.08.2009	15:12:54	STRIETHOLT	10.08.2009	15:13:07	CMONE	*FNS-OK	*FNS-OK	1	1	---	Zollanpassung für Remotekommissionierung
D0107WT	31.07.2009	13:27:17	STRIETHOLT	31.07.2009	13:27:21	CMONE	*FNS-OK	*FNS-OK	1	1	---	Zollanpassung für Remotekommissionierung

Anschließend wird über das Cross-Referenz-Verzeichnis ermittelt, ob es zu den der Form hinzugefügten Positionen abhängige Objekte (z.B. PF -> LF -> Modul -> Programm) gibt, bei denen eine Rekompilierung erforderlich ist. Diese abhängigen Objekte werden dem Entwickler zur Auswahl angeboten, diese Objekte ebenfalls der Form hinzuzufügen.

Enthält die Form endgültig alle Objekte, die für eine erfolgreiche Installation benötigt werden, so kann der Formlauf erfolgen. Die in der Zielumgebung zu ersetzenden Objekte werden durch den Formlauf in temporär erstellte Bibliotheken gestellt, die nach einer in der Anwendungsdefinition vorgegebenen Zeit gelöscht werden. Die Quellen und Objekte, welche sich in der Ausgangsumgebung befinden, werden in der Zielumgebung installiert. D.h., sie werden je nach Vorgaben in der Anwendungsdefinition verschoben oder neu kompiliert. Die Ausgangsumgebung wird bereinigt. Ersetzte Quellen und Objekte werden je nach Vorgabe in der Anwendungsdefinition archiviert. Objektverzeichnis, Objektreferenzen werden durch **CMOne** neu aufgebaut.

Nach erfolgreicher Installation der Form in der Entwicklungsumgebung kann die Form je nach Vorgabe in der Anwendungsdefinition in die Produktionsumgebungen verteilt werden. Die Produktionsumgebungen sind definierte Umgebungen, in denen die Anwendung produktiv arbeitet. Vom jeweiligen Status des Formlaufes in den Produktionsumgebungen werden Rückmeldungen an das Entwicklungssystem gesendet, wie z.B. *FNS-OK = Form erfolgreich installiert oder *FNS-ERR = Form nicht erfolgreich installiert, so dass der Entwickler jederzeit den Fortschritt des Formlaufs auf den Produktionsumgebungen ermitteln kann.



System	Hauptinstanz	Unterinstanz	Beschreibung	Formverteilung	Formlaufautom...	Status
LOXXBOR	*NONE	*NONE	Loxess, BOR	*YES	*YES	*FNS-OK
LOXXGAR	*NONE	*NONE	Loxess, Garching	*YES	*YES	*FNS-OK
LOXXHIR	*NONE	*NONE	Loxess, Hirschau	*YES	*YES	*FNS-OK
LOXXMS	*NONE	*NONE	Loxess, Muenster	*YES	*YES	*FNS-OK
LOXXND	*NONE	*NONE	Loxess, Neuburg a.d. Donau	*YES	*YES	*FNS-OK
LOXXREG	*NONE	*NONE	Loxess, Aurach	*YES	*YES	*FNS-OK

Die hier unter Punkt Anwendungsentwicklung dargestellten Szenarien stellen nur einen kleinen Ausschnitt sämtlicher Funktionalitäten von **CMOne** dar. Im Anhang befindet sich eine Übersicht sämtlicher Funktionalitäten.

PTF`s

Mit **CMOne** kann man analog der bekannten Technik von IBM PTF`s erstellen und verwalten. Sie können mit Installation eines PTF's bestehende bereits aktive PTF`s ersetzen lassen. Sie können notwendige PTF`s angeben, die mit einem bestimmten PTF installiert werden müssen, um die Funktionsfähigkeit des Systems zu gewährleisten. Ebenso ist ein Begleitschreiben aus der PTF-Dokumentation zu erstellen. Mit **CMOne** ist es möglich, Einzel-PTF`s zu einem kumulativen PTF zusammenfassen.

Sobald eine Version einer Anwendung zur Auslieferung freigegeben ist, muss die Anwendung gesperrt werden. Offenbaren sich nach der Freigabe noch Fehler in der Anwendung, so sind diese durch PTF`s zu korrigieren. PTF`s werden direkt bezogen zu einer Anwendung erstellt.

Nachfolgend wird in einer kurzen Beschreibung der Ablauf bei der Verwendung von PTF`s mit **CMOne** dargestellt. Wichtig ist hier zu erwähnen, dass **CMOne** Ihnen alle Prüfungen abnimmt, um die Integrität Ihrer Anwendung jederzeit zu gewährleisten. Es wird in dem Beispiel von der Grundlage ausgegangen, mit Einzel-PTF`s zu arbeiten. Dies bedeutet, dass zu jeder Fehlersituation in der Anwendung diese sofort behoben und auch sofort freigegeben wird.

Beispiel:

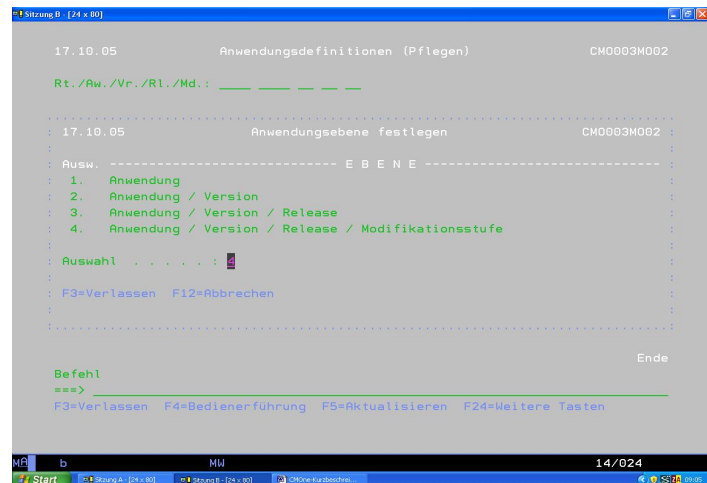
In einer Anwendung sind die Programme PGM1, PGM2, PGM3, PGM4 definiert. Nach Freigabe und Installation der Anwendung stellt sich eine Fehlersituation heraus, bei der die Programme PGM2 und PGM3 modifiziert werden müssen. Der Entwickler erstellt ein PTF mit der Bezeichnung PTF1. Er checkt PGM2 und PGM3 aus (**CMOne** entnimmt beide Programme aus der Basisentwicklungsumgebung der Anwendung), modifiziert die Programme und stellt diese in PTF1 bereit.

Nach gewisser Zeit erfordert eine weitere Fehlersituation, dass PGM3 und PGM4 modifiziert werden müssen. Der Entwickler erstellt ein PTF mit der Bezeichnung PTF2. Er weiß ohne **CMOne** nicht, dass sich PGM3 bereits in PTF1 befindet. Er versucht PGM3 zu PTF2 auszuchecken und erhält die Information, dass das Programm nicht zu PTF2 ausgecheckt werden kann, da es sich in PTF1 befindet. Der Entwickler muss nun PTF1 durch PTF2 ersetzen. Hierbei werden die Objekte von PTF1 nach PTF2 kopiert.

Nachdem der Entwickler PTF1 durch PTF2 ersetzt hat kann er PGM3 und PGM4 auschecken (**CMOne** entnimmt PGM3 aus der Basisentwicklungsumgebung zu PTF2 und PGM4 aus der Basisentwicklungsumgebung der Anwendung). Er modifiziert die Programme und stellt diese in PTF2 zur Installation bereit.

Übersicht der CMOne-Funktionalitäten

- Benennung einer Anwendung nach
 - Anwendung
 - Version
 - Release
 - Modifikationsstufe
- Anwendungsdefinition auf
 - Globaler Ebene
 - Anwendungsbezogener Ebene
 - Umgebungsbezogener Ebene



- Die umgebungsbezogene Ebene gliedert sich in
 - Entwicklungsumgebungen
 - Produktionsumgebungen
- Die Entwicklungsumgebung gliedert sich in
 - Basisentwicklungsumgebung
 - Primäre Entwicklungsumgebung
 - Sekundäre Entwicklungsumgebungen
- Die primäre Entwicklungsumgebung gliedert sich in
 - Anwendungsbezogene primäre Entwicklungsumgebung
 - Projektbezogene primäre Entwicklungsumgebung
 - Aufgabenbezogene primäre Entwicklungsumgebung
 - Benutzerbezogene primäre Entwicklungsumgebung

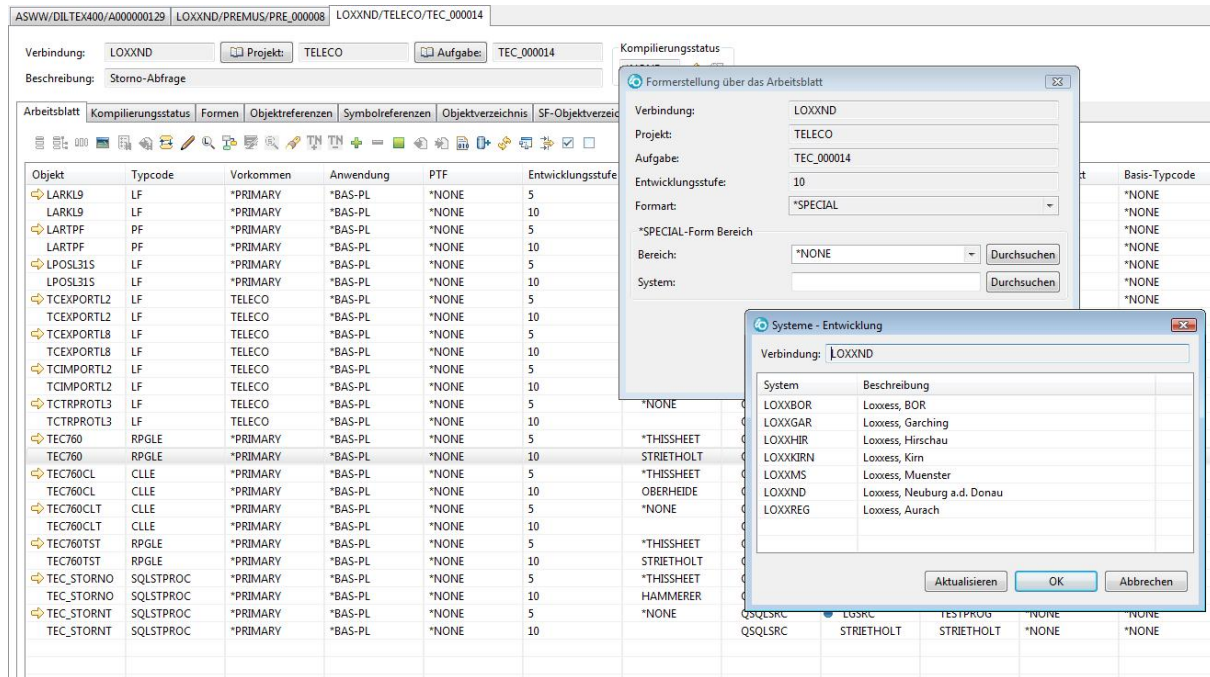
- Definition der Gruppen nach denen die Anwendung gruppiert ist auf Anwendungsebene wie z.B. Gruppe für Quellen = QUELLE, Gruppe für Sprachobjekte = LANGUAGE, Gruppe für Datenobjekte = DATA und Gruppe für sonstige Objekte = OBJECT (werden automatisch in die Umgebungsebenen übertragen).



- Definition der in einer Anwendung verwendeten Objekttypen auf Anwendungsebene wie z.B. RPGLE für RPG IV und PF für physische Datei (werden automatisch in die Umgebungsebenen übertragen).

- Definition von Vorkommen zu Gruppen und Objekttypen auf Anwendungsebene z.B. das Vorkommen DEU für ‚Deutsch‘ und ENG für ‚Englisch‘ zur Gruppe LANGUAGE oder zum Objekttyp MSGF (werden automatisch in die Umgebungsebenen übertragen).
- Zuordnung von Gruppe/Vorkommen zu Objekttyp/Vorkommen auf Anwendungsebene. Dadurch werden Quellendatei, Quellenbibliothek und Objektbibliothek von Gruppe/Vorkommen vererbt an Objekttyp/Vorkommen.
- Generelle Vorgabe von Quellendatei, Quellenbibliothek und Objektbibliothek auf Umgebungsebene zu Gruppe/Vorkommen und wenn nötig, dann auch zu Objekttyp/Vorkommen.
- Bibliothekslisten
 - auf globaler Ebene
 - auf anwendungsbezogener Ebene
 - auf umgebungsbezogener Ebene
 - Feste Hinterlegung der Bibliothek in der Bibliotheksliste oder Bezugnahme auf in der Umgebungsebene hinterlegte Bibliotheken
 - werden bei der Kompilierung von Objekten verwendet
- Ersatzvariablen
 - auf globaler Ebene
 - auf anwendungsbezogener Ebene
 - auf umgebungsbezogener Ebene
 - werden bei der Kompilierung von Objekten verwendet
- Allgemeine Erstellungsbefehle zu Objekttypen
 - auf globaler Ebene
 - auf anwendungsbezogener Ebene
 - werden bei der Kompilierung von Objekten verwendet
- Cross-Referenz-Tabellen
 - für Stand-Alone Anwendungen (Die Anwendung referenziert keine Objekte aus anderen Anwendungen)
 - für interagierende Anwendungen (Die Anwendung referenziert Objekte aus anderen Anwendungen)
- PTF-Verwaltung
 - Einzel-PTF`s
 - Kumulative PTF`s
 - PTF-Dokumentation
 - Notwendige PTF`s
 - Ersetzte PTF`s

- o Die primäre und die sekundäre Entwicklungsumgebung kann als Ausgangspunkt für die Verteilung einer *SPECIAL-Form zu einer Testumgebung (*SPECIAL-Form Umgebung) auf einem fernen System verwendet werden.

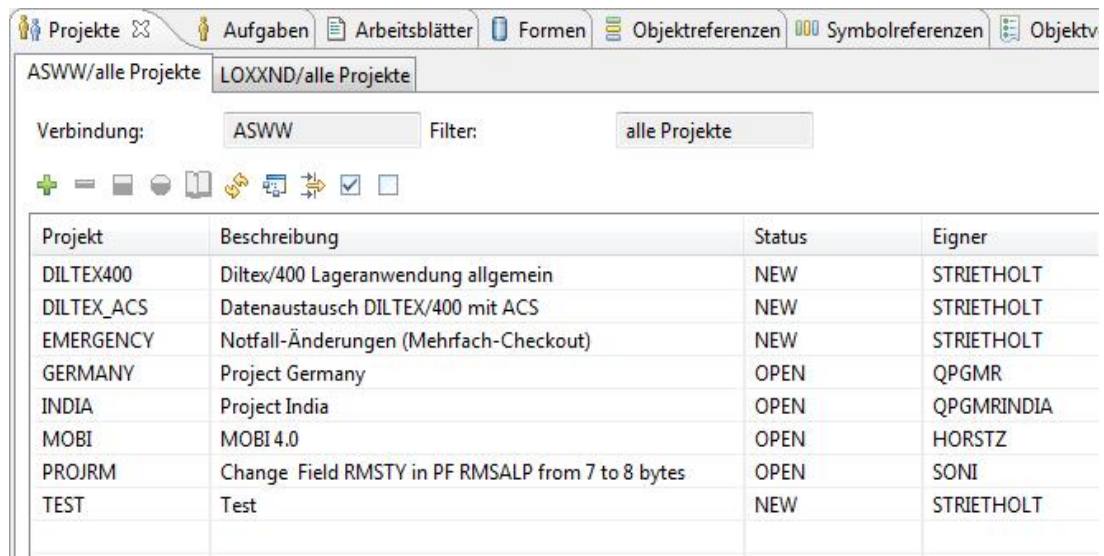


The screenshot shows the SAP Change Management System interface. At the top, there are tabs for different projects: ASWW/DILTEX400/A00000129, LOXXND/PREMIUS/PRE_000008, and LOXXND/TELECO/TEC_000014. Below this, there are fields for 'Verbindung: LOXXND', 'Projekt: TELECO', and 'Aufgabe: TEC_000014'. The main area contains a table with columns: Objekt, Typcode, Vorkommen, Anwendung, PTF, and Entwicklungsstufe. The table lists various objects like LARKL9, LARTPF, LPOSL315, etc. Two dialog boxes are open: 'Formerstellung über das Arbeitsblatt' and 'Systeme - Entwicklung'. The 'Formerstellung' dialog shows 'Verbindung: LOXXND', 'Projekt: TELECO', 'Aufgabe: TEC_000014', and 'Entwicklungsstufe: 10'. The 'Systeme - Entwicklung' dialog shows a list of systems with columns 'System' and 'Beschreibung'.

Objekt	Typcode	Vorkommen	Anwendung	PTF	Entwicklungsstufe
LARKL9	LF	*PRIMARY	*BAS-PL	*NONE	5
LARKL9	LF	*PRIMARY	*BAS-PL	*NONE	10
LARTPF	PF	*PRIMARY	*BAS-PL	*NONE	5
LARTPF	PF	*PRIMARY	*BAS-PL	*NONE	10
LPOSL315	LF	*PRIMARY	*BAS-PL	*NONE	5
LPOSL315	LF	*PRIMARY	*BAS-PL	*NONE	10
TCEXPORL2	LF	TELECO	*BAS-PL	*NONE	5
TCEXPORL2	LF	TELECO	*BAS-PL	*NONE	10
TCEXPORL8	LF	TELECO	*BAS-PL	*NONE	5
TCEXPORL8	LF	TELECO	*BAS-PL	*NONE	10
TCIMPORL2	LF	TELECO	*BAS-PL	*NONE	5
TCIMPORL2	LF	TELECO	*BAS-PL	*NONE	10
TCIMPORL3	LF	TELECO	*BAS-PL	*NONE	5
TCIMPORL3	LF	TELECO	*BAS-PL	*NONE	10
TCTRPROL3	LF	TELECO	*BAS-PL	*NONE	5
TCTRPROL3	LF	TELECO	*BAS-PL	*NONE	10
TEC760	RPGLE	*PRIMARY	*BAS-PL	*NONE	5
TEC760	RPGLE	*PRIMARY	*BAS-PL	*NONE	10
TEC760CL	CLLE	*PRIMARY	*BAS-PL	*NONE	5
TEC760CL	CLLE	*PRIMARY	*BAS-PL	*NONE	10
TEC760CLT	CLLE	*PRIMARY	*BAS-PL	*NONE	5
TEC760CLT	CLLE	*PRIMARY	*BAS-PL	*NONE	10
TEC760TST	RPGLE	*PRIMARY	*BAS-PL	*NONE	5
TEC760TST	RPGLE	*PRIMARY	*BAS-PL	*NONE	10
TEC_STORNO	SQLSTPROC	*PRIMARY	*BAS-PL	*NONE	5
TEC_STORNO	SQLSTPROC	*PRIMARY	*BAS-PL	*NONE	10
TEC_STORNT	SQLSTPROC	*PRIMARY	*BAS-PL	*NONE	5
TEC_STORNT	SQLSTPROC	*PRIMARY	*BAS-PL	*NONE	10

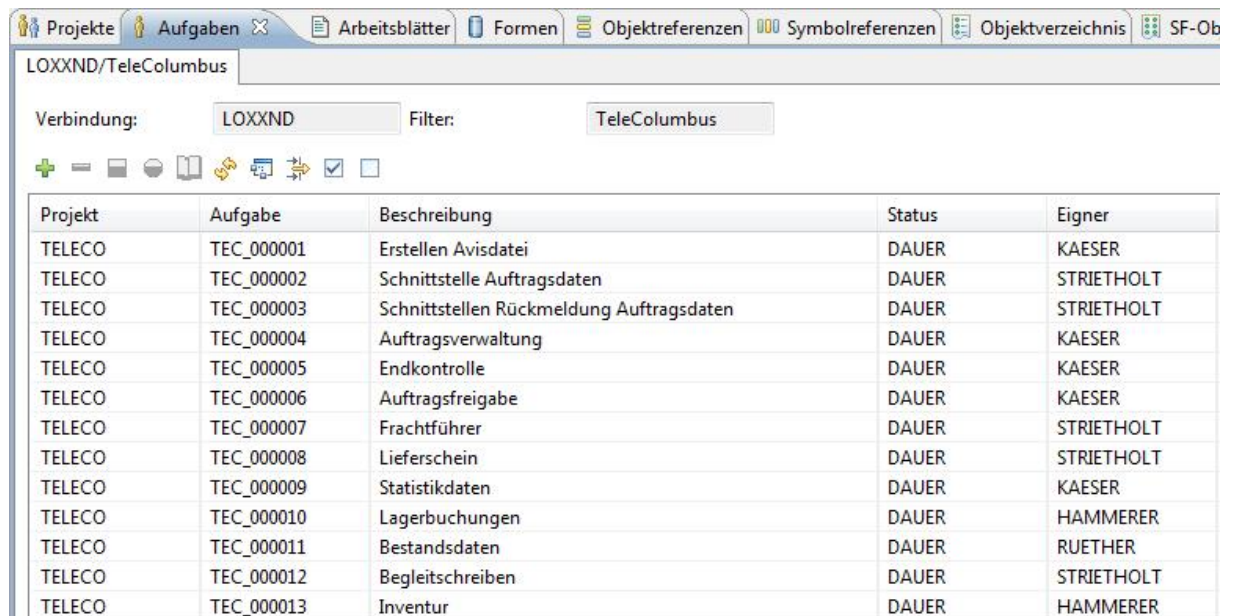
- In der Basisentwicklungsumgebung und in der Produktionsumgebung kann ein Expansionsbereich angegeben werden, in den Objekte bei Formlauf dupliziert werden. Das ist besonders interessant, wenn jemand eine nicht mandantenfähige Anwendung hat und die Mandantenfähigkeit über eine Datenbibliothek je Mandant erreicht.
- Versionsmanagement wird voll unterstützt. Das bedeutet, dass einer höheren Version eine Abhängigkeit hinzugefügt werden kann, die auf die Vorgängerversion verweist. Zusätzliche Funktionen können dabei verwendet werden.

- Definition von Projekten mit
 - Allgemeinen Daten
 - Projektdokumentation
 - Primäre Entwicklungsumgebung je Anwendung



Projekt	Beschreibung	Status	Eigner
DILTEX400	DilTEX/400 Lageranwendung allgemein	NEW	STRIETHOLT
DILTEX_ACS	Datenaustausch DILTEX/400 mit ACS	NEW	STRIETHOLT
EMERGENCY	Notfall-Änderungen (Mehrfach-Checkout)	NEW	STRIETHOLT
GERMANY	Project Germany	OPEN	QPGMR
INDIA	Project India	OPEN	QPGMRINDIA
MOBI	MOBI 4.0	OPEN	HORSTZ
PROJRM	Change Field RMSTY in PF RMSALP from 7 to 8 bytes	OPEN	SONI
TEST	Test	NEW	STRIETHOLT

- Definition von Aufgaben mit
 - Allgemeinen Daten
 - Aufgabendokumentation
 - Primäre Entwicklungsumgebung je Anwendung



Projekt	Aufgabe	Beschreibung	Status	Eigner
TELECO	TEC_000001	Erstellen Avisdatei	DAUER	KAESER
TELECO	TEC_000002	Schnittstelle Auftragsdaten	DAUER	STRIETHOLT
TELECO	TEC_000003	Schnittstellen Rückmeldung Auftragsdaten	DAUER	STRIETHOLT
TELECO	TEC_000004	Auftragsverwaltung	DAUER	KAESER
TELECO	TEC_000005	Endkontrolle	DAUER	KAESER
TELECO	TEC_000006	Auftragsfreigabe	DAUER	KAESER
TELECO	TEC_000007	Frachtführer	DAUER	STRIETHOLT
TELECO	TEC_000008	Lieferschein	DAUER	STRIETHOLT
TELECO	TEC_000009	Statistikdaten	DAUER	KAESER
TELECO	TEC_000010	Lagerbuchungen	DAUER	HAMMERER
TELECO	TEC_000011	Bestandsdaten	DAUER	RUETHER
TELECO	TEC_000012	Begleitschreiben	DAUER	STRIETHOLT
TELECO	TEC_000013	Inventur	DAUER	HAMMERER

- Arbeitsblatt

Arbeitsblatthistorie

Arbeitsblätter

ASWW/DILTEX400/A000000129 LOXXND/PREMUS/PRE_000008 LOXXND/TELECO/TEC_000014

Verbindung: LOXXND Projekt: PREMUS Aufgabe: PRE_000008 Kompilierungsstatus: *FNS-OK

Beschreibung: IDOC-Verarbeitung Auftragsdaten

Arbeitsblatt Kompilierungsstatus Formen Objektreferenzen Symbolreferenzen Objektverzeichnis SF-Objektverzeichnis Mehrfach-Checkouts Suchergebnisse

Objekt	Typcode	Vorkommen	Anwendung	PTF	Entwicklungsstufe	Checkout	Quellendatei	Quellenbibliothek	Objektbibliothek
↔ E2BPEXC00	PF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ E2BPEXC03	PF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ E2HDR0000	PF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ E2HDRORGOO	PF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ E2ITEM0010	PF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ E2ITEMORGO	PF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ E2ITEMRFO	PF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ E2SHPOBDLV	PF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ LPOSL3E	LF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ LPOSL5E	LF	*PRIMARY	*BAS-PL	*NONE	5		QPFSRC	LGSRC	LGTEST
↔ PRE720	RPGLE	*PRIMARY	*BAS-PL	*NONE	5		QRPGLSRC	LGSRC	TESTPROG
↔ PRE720CL	CLLE	*PRIMARY	*BAS-PL	*NONE	5		QCLSRC	LGSRC	TESTPROG
↔ PRE720PR	PRTF	*PRIMARY	*BAS-PL	*NONE	5		QDDSSRC	LGSRC	TESTPROG

- Arbeitsblatteintrag auschecken
- Arbeitsblatteintrag einchecken
- Befehle zu Arbeitsblatteintrag editieren

DILTEXSRC/QRPGLESRC(S00009R) : Textvergleich

DILTEXSRC/QRPGLESRC(S00009R)	DTARCHEV/000013295(S00009R)
<pre> ELSE XX_EMNR = AXEMNR; ENDIF; ELSE; XX_ABAR = 'FV'; XX_EMNR = *BLANK; XX_EANR = *BLANK; ENDIF; // Verzögerung der Artikelnr: grundsätzlich 7-stellig (ohne Lockennung), damit auf Article // ohne verdrichtet wird XX_ARNR = %SUBST (ZPARNR:1:7); // Zolaugangsdatei aktualisieren DOU MOE VEZUG; CHAIN(E) REY2AU03 ZOLAUSFI; IF NERROR; /END-FREE C CALL 'MOD160R' C FARM EXCPGM /FREE ENDIF; </pre>	<pre> ELSE; XX_EMNR = AXEMNR; ENDIF; ELSE; XX_ABAR = 'FV'; XX_EMNR = *BLANK; XX_EANR = *BLANK; ENDIF; // Verzögerung der Artikelnr: grundsätzlich 7-stellig (ohne Lockennung), damit auf Article // ohne verdrichtet wird XX_ARNR = %SUBST (ZPARNR:1:7); // Zolaugangsdatei aktualisieren DOU MOE VEZUG; CHAIN(E) REY2AU03 ZOLAUSFI; IF NERROR; /END-FREE C CALL 'MOD160R' C FARM EXCPGM /FREE ENDIF; </pre>

- Befehle zu Arbeitsblatteintrag ausführen (kompilieren)
- Quellenvergleich
- CheckOut Historie
- Objekthistorie
- Objektreferenzen
- Symbolreferenzen
- Konflikte
- Sitzungsstandardwerte
- SEU anzeigen/editieren
- SDA anzeigen/editieren
- RLU anzeigen/editieren
- Arbeitsblatteintrag einer Form hinzufügen
- Arbeitsblatthistorie
- Benutzerdefinierte Optionen

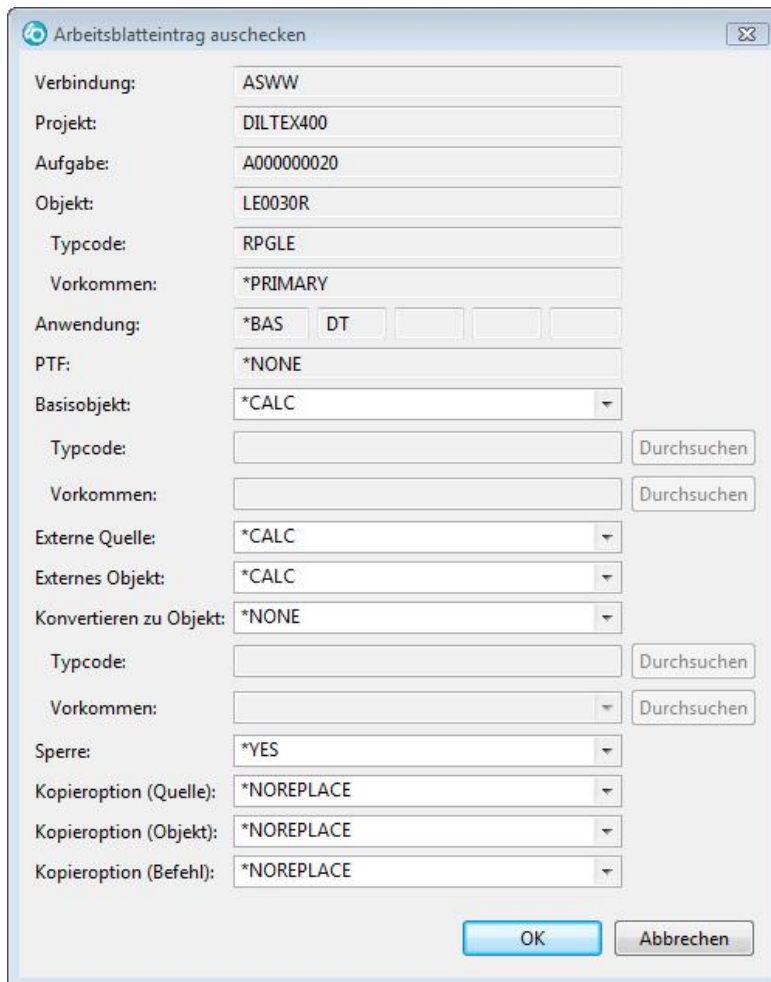
- o LPEX-Editor anzeigen/editieren

```

LA0002R.RPGLE X
Zeile 736   Spalte 10   Ersetzen   Durchsuchen
.....+.....1.....+.....2.....+.....3.....+.....4.....+.....5.....+.....6.....+.....7.....+.....8
068000      C           Z-ADD      Z1           I1
068001      /FREE
068100      ENDIF;
068200      // Positionszeile Artikel drucken
068201      /END-FREE
068300      C           MOVEL      EAARNR      $$ARNR
068400      C           MOVEL      EALONR      $$LONR
068500      // Ermitteln Grössen-ID Lot
068501      /FREE
068600      Chain KEYEE LOSTAMF1;
068700      If Not %Found;
068701          $$LONR = *BLANK;
068800          Clear *All LOSTAMF1;
068900      Endif;
069000      // Ist Grössen-ID Lot gefüllt, belegen Feld ##LONR
069100      If LOGID <> *Blank;
069200          $$LONR = LOGID;
069300      Endif;
069400
069500      // Chargennummer ausgeben
069600      $$CHARGE = EACHAR;
069700
069800      // Bei Lerros/TK's wird aus der Lotkomponente die erste Farbe der Lotkom
069900      // Der Grössenschlüssel wird ebenfalls aufgelistet.
070000      /*           IF           PKDNR = '100031'
070100      XXLOGRO= *BLANK;
070200      KZ_LO_FAR ==*OFF;
  
```

- o Screen-Designer

- Konvertierungen von einem Objekttyp zu einem anderen wie z.B. RPG III nach RPG IV werden voll unterstützt.
- CheckOut sowie CheckIn-Funktion mit Überprüfung auf Mehrfachcheckout von gleichen Objekten auf unterschiedlichen Arbeitslisten.

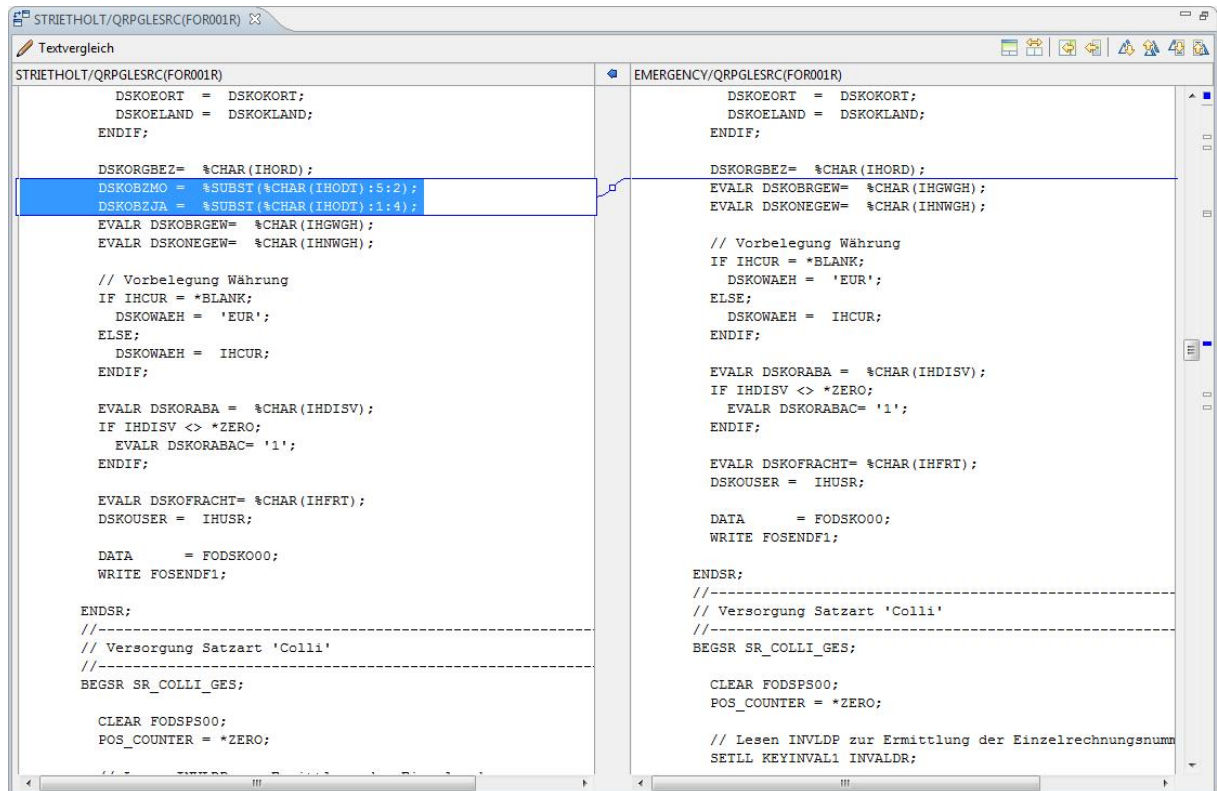


Arbeitsblatteintrag auschecken

Verbindung:	ASWW	
Projekt:	DILTEX400	
Aufgabe:	A000000020	
Objekt:	LE0030R	
Typcode:	RPGLE	
Vorkommen:	*PRIMARY	
Anwendung:	*BAS DT	
PTF:	*NONE	
Basisobjekt:	*CALC	
Typcode:		Durchsuchen
Vorkommen:		Durchsuchen
Externe Quelle:	*CALC	
Externes Objekt:	*CALC	
Konvertieren zu Objekt:	*NONE	
Typcode:		Durchsuchen
Vorkommen:		Durchsuchen
Sperre:	*YES	
Kopieroption (Quelle):	*NOREPLACE	
Kopieroption (Objekt):	*NOREPLACE	
Kopieroption (Befehl):	*NOREPLACE	

OK Abbrechen

- Volle Unterstützung von Mehrfach-CheckOuts gleicher Objekte zur gleichen Zeit. **CMOne** versorgt die Entwickler mit den notwendigen Informationen, welche Änderungen bei einem Emergency-Checkout nachgezogen werden müssen, damit die vollständige Integrität gewährleistet wird.



```

STRIETHOLT/QRPGLESRC(FOR001R)
DSKOEORT = DSKOKORT;
DSKOEOLAND = DSKOKLAND;
ENDIF;

DSKORGBEZ= %CHAR(IHORD);
DSKOBZMO = %SUBST(%CHAR(IHODI):5:2);
DSKOBZJA = %SUBST(%CHAR(IHODI):1:4);
EVALR DSKOBRGEW= %CHAR(IHGWGH);
EVALR DSKONEGEW= %CHAR(IHNBWGH);

// Vorbelegung Währung
IF IHCUR = *BLANK;
DSKOWAEH = 'EUR';
ELSE;
DSKOWAEH = IHCUR;
ENDIF;

EVALR DSKORABA = %CHAR(IHDISV);
IF IHDISV <> *ZERO;
EVALR DSKORABAC= '1';
ENDIF;

EVALR DSKOFRACHT= %CHAR(IHFRT);
DSKOUSER = IHUSR;

DATA = FODSKO00;
WRITE FOSENDF1;

ENDSR;
//-----
// Versorgung Satzart 'Colli'
//-----
BEGSR SR_COLLI_GES;

CLEAR FODSPS00;
POS_COUNTER = *ZERO;

EMERGENCY/QRPGLESRC(FOR001R)
DSKOEORT = DSKOKORT;
DSKOEOLAND = DSKOKLAND;
ENDIF;

DSKORGBEZ= %CHAR(IHORD);
EVALR DSKOBRGEW= %CHAR(IHGWGH);
EVALR DSKONEGEW= %CHAR(IHNBWGH);

// Vorbelegung Währung
IF IHCUR = *BLANK;
DSKOWAEH = 'EUR';
ELSE;
DSKOWAEH = IHCUR;
ENDIF;

EVALR DSKORABA = %CHAR(IHDISV);
IF IHDISV <> *ZERO;
EVALR DSKORABAC= '1';
ENDIF;

EVALR DSKOFRACHT= %CHAR(IHFRT);
DSKOUSER = IHUSR;

DATA = FODSKO00;
WRITE FOSENDF1;

ENDSR;
//-----
// Versorgung Satzart 'Colli'
//-----
BEGSR SR_COLLI_GES;

CLEAR FODSPS00;
POS_COUNTER = *ZERO;

// Lesen INVLDP zur Ermittlung der Einzelrechnungsnumm
SETLL KEYINVAL1 INVALDR;

```

- In der Version **CMOne** Next Generation gibt es die Funktion, Änderungen nach Möglichkeit vollautomatisch in die anderen ausgecheckten Quellen einzumischen.
- Zu jedem Objekt können Vorlauf-, Nachlauf- oder modifizierte von den allgemeinen Definitionen abweichende Erstellungsbefehle angegeben werden.
- Objektreferenzen
 - Alles, was OS/400 an CL-Befehlen und API`s bietet, um Objektreferenzen zu ermitteln, wird genutzt wie z.B. DSPPGMREF.
 - Zusätzlich wird der Quellcode gescannt und gefundene Objektreferenzen übernommen, wie z.B. die Verwendung externer Datenstrukturen und COPY-Strecken in RPG Programmen.
 - Objektreferenzen, die **CMOne** nicht über die genannten Wege ermitteln kann, können über die Direktive %REF% direkt im Quellcode vom Programmierer hinterlegt werden. Diese werden dann übernommen. Bei Objekten, denen keine Quelle zugeordnet ist können solche Referenzen in den benutzerdefinierten Objektreferenzen eingetragen werden.
 - Die Objektreferenzen können für ein bestimmtes Objekt oder kumuliert für mehrere anzugebende Objekte dargestellt werden.
- In SQL-Scripts kann als Qualifizierung zur Datei der Wert *LIBL angegeben werden. **CMOne** setzt zur Ausführungszeit des Scripts den Wert *LIBL in die entsprechende Bibliothek um. Eine feste Qualifizierung mit dem Bibliotheksnamen entfällt hierdurch.
- Formen können auf ferne Systeme gesendet werden über:
 - SNADS
 - TCP/IP FTP
 - Band

- Es können Zeitfenster definiert werden, zu denen Formen installiert werden sollen.
- Eine Fehlerbehandlung bei einem Formlauf kann erfolgen nach:
 - Fehlerbehandlung *RECOVER (Beim ersten Fehler, der auftritt werden alle bereits gemachten erfolgreichen Änderungen zurückgesetzt).
 - Fehlerbehandlung *CONTINUE (Nach einem Fehler wird mit der nächsten Formposition fortgefahren).
 - Fehlerbehandlung *MSGRPY (Nach einem Fehler wird der Job auf den Status MSGW gesetzt und der Systembediener hat eine Aktion vorzugeben).
 - Die Fehlerbehandlungsmaßnahme kann auf Formkopfebene sowie auf Formpositionsebene erfolgen.
- Vor- und Nachlaufbefehle auf Formebene
- Formlaufaktionen
 - Kopieren der Befehle von der Ausgangsumgebung zur Zielumgebung
 - Kopieren der Quelle von der Ausgangsumgebung zur Zielumgebung
 - Kopieren des Objektes von der Ausgangsumgebung zur Zielumgebung
 - Kompilierung des Objektes in der Zielumgebung
 - Löschen der Befehle in der Ausgangsumgebung
 - Löschen der Quellen in der Ausgangsumgebung
 - Löschen der Objekte in der Ausgangsumgebung
 - Erstellen von Teildateien zu physischen und logischen Dateien in der Zielumgebung anhand der ersetzten Datei
 - Kopieren der Daten aus Teildateien zu physischen Dateien in die Zielumgebung anhand der ersetzten Datei
 - Modifikation des Objekteigners in der Zielumgebung
 - Modifikation der Objektberechtigung in der Zielumgebung
- Formen können nach einem Formlauf punktuell oder generell zurückgesetzt werden. Das bedeutet, dass der Zustand vor dem Formlauf wiederhergestellt wird (Recovery-Funktion).

Form	Datum-Erstellung	Uhrzeit-Erstellung	Benutzer-Erstell...	Datum-Formlauf	Uhrzeit-Formlauf	Benutzer-Forml...	Status-Formlauf	Status-Formvert...	SUM	JO	NIO	Beschreibung
D010892	23.09.2009	10:32:15	STRIETHOLT	23.09.2009	10:32:21	CMONE	*FNS-OK	*FNS-OK	1	1	---	Anpassung neue Zolllogik Abmeldung/Faktura
D010752	09.06.2009	21:26:58	STRIETHOLT	09.06.2009	21:37:03	CMONE	*FNS-OK	*FNS-OK	1	1	---	Anpassung neue Zolllogik Abmeldung/Faktura
D0107RW	09.06.2009	12:57:24	STRIETHOLT	09.06.2009	14:23:21	CMONE	*FNS-OK	*FNS-OK	1	1	---	Anpassung neue Zolllogik Abmeldung/Faktura

- Ein Formlaufjob kann nach einem Abbruch wieder gestartet werden. Die Formlaufaktionen werden hierbei an genau der Stelle wieder aufgesetzt, an der sie abgebrochen wurden.
- Nach einem Formlauf in einer Produktionsumgebung werden Rückmeldungen über den Status der Form zurück an den Rechner gesendet, von dem aus die Form verteilt wurde.
- Über *SPECIAL-Form Bereiche können Bibliotheken definiert werden, in die Objekte zum abschließenden Test auf den Produktionsrechner gestellt werden, ohne dass eine Rückführung in die Basisumgebung stattfindet.
-

- Befehls-, Quellen- und Objektarchiv, das sowohl online als auch offline vorgehalten werden kann. Online bedeutet, Speicherung im Plattenspeicher und Offline bedeutet, Speicherung auf Band nach vorheriger Auslagerung.
- Unterstützung von IBM`s CODE/400
- Einbindung von externen Programmen an definierten Ausgangspunkten (Exit-Point's) wie z.B. beim Hinzufügen von Objekten auf das Arbeitsblatt oder zum Zeitpunkt des Formlaufes.
- Wird die Mehrsprachigkeit einer Anwendung durch die Verwendung von Nachrichtendateien erreicht, so unterstützt **CMOne** dieses durch eine Vielzahl von Funktionen wie z.B. Wörterbücher, Übersetzungsvorschläge u.s.w.
- Deutsche und englische Benutzeroberfläche.
- Nutzung von modernen Anwendungsoberflächen wie WDSC und RDi in der Version **CMOne** Next Generation
- u.v.m.