

Programmierhilfen für die logischen Datenbanken PNP und PAP



HELP.PAXX

Release 4.6C



Copyright

© Copyright 2001 SAP AG. Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP AG oder deren Vertriebsfirmen angebotenen Software-Produkte können Software-Komponenten auch anderer Software-Hersteller enthalten.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] und SQL Server[®] sind eingetragene Marken der Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®] und OS/400[®] sind eingetragene Marken der IBM Corporation.

ORACLE[®] ist eine eingetragene Marke der ORACLE Corporation.

INFORMIX[®]-OnLine for SAP und Informix[®] Dynamic Server[™] sind eingetragene Marken der Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®] und Motif[®] sind eingetragene Marken der Open Group.

HTML, DHTML, XML, XHTML sind Marken oder eingetragene Marken des W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] ist eine eingetragene Marke der Sun Microsystems, Inc.

JAVASCRIPT[®] ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo und mySAP.com sind Marken oder eingetragene Marken der SAP AG in Deutschland und vielen anderen Ländern weltweit. Alle anderen Produkte sind Marken oder eingetragene Marken der jeweiligen Firmen.

Symbole

Symbol	Bedeutung
	Achtung
	Beispiel
	Empfehlung
	Hinweis
	Syntax
	Tip

Inhalt

Programmierhilfen für die logischen Datenbanken PNP und PAP.....	5
Definieren von Konstanten.....	6
Ausgetretene Mitarbeiter nicht berücksichtigen.....	7
Definieren des Auswahlzeitraums.....	8
Definieren des Datenauswahlzeitraums.....	9
Namensaufbereitung in HR-Reports.....	10
Namensaufbereitung gemäß nationalen Vorschriften.....	11
HR-Adreßaufbereitung nach postalischen Vorschriften.....	13
Prüfen und Ausgeben von Personal-Identifikationsnummern.....	14
Ausgeben des Listenzeitraums.....	16
Ausgeben des Listenstichtags.....	17
Initialisieren des Puffers für Zugriff auf PCLn.....	18
Import-/Export-Makros für Clusterdaten.....	19
Import-Makros.....	21
Export-Makros.....	23
Löschen von Abrechnungsergebnissen im Speicher.....	24
Lesen eines Infotyps einer Person in einem Zeitraum.....	25
Ersten Eintrag im Zeitraum zur Verfügung stellen.....	27
Letzten Eintrag im Zeitraum zur Verfügung stellen.....	29
Verändern von Personaldaten.....	31

Programmierhilfen für die logischen Datenbanken PNP und PAP

Verwendung

Programmierhilfen bieten Ihnen die Möglichkeit, gleichartige Anforderungen für Reports programmtechnisch abzuhandeln. Diese Module können Sie für eigene Reports nutzen.

Folgende Programmierhilfen stehen Ihnen für die logischen Datenbanken PNP und PAP zur Verfügung:

- Funktionsbausteine
- Makros

Makros bieten Ihnen den Vorteil, daß bei einer Änderung des Makros jeder Report, der dieses Makro verwendet, beim nächsten Aufruf automatisch neu generiert wird.

- RMAC-Makros

RMAC-Makros stellen spezifische Makros dar, die in der Komponente *Personalwirtschaft* (HR) genutzt werden. Sie haben den Nachteil, daß eine Änderung an einem RMAC-Makro manuell in allen Reports, die den RMAC-Makro verwenden, nachgezogen werden muß.



Wir empfehlen Ihnen, vorwiegend Makros oder Funktionsbausteine in Ihren Reports zu verwenden, soweit diese RMAC-Makros bereits ersetzen.

Funktionsumfang

Programmierhilfen stehen Ihnen für folgende Zeitpunkte des Programmablaufs in einem Report zur Verfügung:

- Daten-Definition
- Zum Zeitpunkt INITIALIZATION
- Zum Zeitpunkt START-OF-SELECTION
- Zum Zeitpunkt SELECTION (GET PERNR)
- Zum Zeitpunkt TOP-OF-PAGE
- Beliebiger Zeitpunkt
 - (z.B. SELECTION, END-OF-SELECTION, AT PFxx, ...)
 - Besorgen von Daten
 - Verarbeiten von Daten
 - Verändern von Daten

Definieren von Konstanten

Definieren von Konstanten

Verwendung

Die Parameter für die Datendefinition des niedrigsten im System stehenden Datums (LOW-DATE) und des höchsten im System vorhandenen Datums (HIGH-DATE) können Sie direkt im Programm eingeben. Verwenden Sie dazu folgende Definitionen:

- *CONSTANTS: LOW-DATE TYPE D VALUE '18000101'*
- *CONSTANTS: HIGH-DATE TYPE D VALUE '99991231'*



Die Konstanten repräsentieren Minusunendlich und Plusunendlich auf der Zeitachse und nicht konkrete Zeitpunkte. Beachten Sie, daß Sie mit diesen Daten nur vergleichen, nicht aber rechnen dürfen.

Ausgetretene Mitarbeiter nicht berücksichtigen

Verwendung

Wenn ausgetretene Mitarbeiter in der Datenselektion nicht berücksichtigt werden sollen, können Sie folgende Programmierhilfe verwenden:

RMAC-Makro: RP-SEL-EIN-AUS-INIT

Sie verwenden dieses RMAC-Makro zum Zeitpunkt INITIALIZATION.

Die Verwendung des RMAC-Makros *RP-SEL-EIN-AUS-INIT* bietet Ihnen folgende Vorteile:

- Der *RP-SEL-EIN-AUS-INIT* schlägt eine Selektion vor, die der Benutzer beim Start des Reports auf dem Selektionsbild verändern kann.
- Der *RP-SEL-EIN-AUS-INIT* läßt die Verarbeitung recht früh durch den Datenbankprozessor ausführen.
- Der *RP-SEL-EIN-AUS-INIT* bezieht sich auf das Zeitintervall (PN-BEGPS, PN-ENDPS).
- Die durch den *RP-SEL-EIN-AUS-INIT* initiierte Selektion vom Datenbankprozessor wird mit den anderen Datenbankselektionskriterien 'zeitgleich' verknüpft

Funktionsumfang

Wenn Sie das RMAC-Makro *RP-SEL-EIN-AUS-INIT* verwenden, schlägt das System Ihnen auf dem Reportselektionsbild vor, nur nicht ausgetretene Mitarbeiter zu selektieren. Im Feld *Status Beschäftigung* ist beim Aufruf eines Reports automatisch ein Vorschlagswert eingetragen, mit dessen Eingabe das System nur nicht ausgetretene Mitarbeiter selektiert.

Parameter

keine

Prüfung

keine

Voraussetzung

keine

Beispiel

```
...  
INITIALIZATION.  
  RP-SEL-EIN-AUS-INIT.  
START-OF-SELECTION.
```

Definieren des Auswahlzeitraums

Definieren des Auswahlzeitraums

Verwendung

Mit der globalen Variable *PNPTIMED* können Sie zum Zeitpunkt *INITIALIZATION* den Auswahlzeitraum für das Selektionsbild setzen. Beispielsweise können Sie Ihr Programm so gestalten, daß das System dem Benutzer bereits beim Aufruf eines Reports vorschlägt, eine tagesaktuelle Auswertung vorzunehmen.

Funktionsumfang

Mögliche Ausprägungen der Variable *PNPTIMED* sind:

- D für *Stichtag*
- M für *Monat*
- Y für *Jahr*
- P für *Vergangenheit*
- F für *Zukunft*

Beispiel

```
INITIALIZATION.  
...  
    PNPTIMED = 'D'.
```



Die ABAP-Anweisung bewirkt, daß das System beim Aufruf des Reports auf dem Reportselektionsbild eine stichtagsbezogene Auswertung vorschlägt.

Definieren des Datenauswahlzeitraums

Verwendung

Um die Datenauswahlzeiträume einzelner Infotypen zu ändern, steht Ihnen folgende Programmierhilfe zur Verfügung:

Makro: *RP_SET_DATA_INTERVAL*

Sie verwenden das Makro zum Zeitpunkt INITIALIZATION.

Funktionsumfang

Mit dem Makro *RP_SET_DATA_INTERVAL* können Sie zum Zeitpunkt INITIALIZATION den Datenauswahlzeitraum für einzelne Infotypen verändern.

Parameter

IN	1)	Infotyp
	2)	beg: Beginn des Zeit-Intervalls
	3)	end: Gültigkeitsende des Zeit-Intervalls
OUT		keine

Prüfung

keine

Beispiel

```
INITIALIZATION.
  RP_SET_DATA_INTERVAL 0001 SY_DATUM SY_DATUM
  START_OF_SELECTION.
```

Namensaufbereitung in HR-Reports

Namensaufbereitung in HR-Reports

Verwendung

Für die Aufbereitung des Namensformats in HR-Reports steht Ihnen folgende Programmierhilfe zur Verfügung:

Funktionsbaustein: *RP_SET_NAME_FORMAT*

Die HR-Namensaufbereitung wird mittels des Funktionsbausteins *RP_EDIT_NAME* zum Zeitpunkt SELECTION (GET PERNR) vorgenommen. Dieses Unterprogramm bereitet den Namen gemäß Eintragungen in der Tabelle *HR-Namensaufbereitung* (T522N) auf. Da die HR-Namensaufbereitung individuell vom Programm abhängig sein kann, steht der Funktionsbaustein *RP_SET_NAME_FORMAT* zum Zeitpunkt START-OF-SELECTION unmittelbar in Verbindung mit der HR-Namensaufbereitung.



Diesen Funktionsbaustein sollten Sie in allgemeinen HR-Reports anwenden, bei denen eine Namensliste von Mitarbeitern ausgegeben wird. (z.B. *Mitarbeiterliste* (RPLMIT00), *Dauer der tariflichen Zugehörigkeit* (RPLTRF00), *Vorschläge zur tariflichen Umstufung* (RPLTRF10)).

Funktionsumfang

Der Funktionsbaustein *RP_SET_NAME_FORMAT* ermittelt das jeweilige Report-Format aus der Tabelle *Format für HR-Namensaufbereitung* (T522F). Ist in dieser Tabelle keine Eintragung für den entsprechenden HR-Report vorgesehen, so wird das Report-Format auf '01' (DEFAULT) gesetzt.

Parameter:

IN :	Reportname
OUT:	Report-Format aus T522F (Default: '01') für T522N

Prüfung:

Tabelle *Format für HR-Namensaufbereitung* (T522F)

Namensaufbereitung gemäß nationalen Vorschriften

Verwendung

Für die Aufbereitung des Namens nach nationalen Gesichtspunkten steht Ihnen folgende Programmierhilfe zur Verfügung:

Funktionsbaustein: *RP_EDIT_NAME*



Sie sollten den Funktionsbaustein *RP_EDIT_NAME* in allen HR-Reports anwenden, bei denen die vollständigen Namen von Mitarbeitern gedruckt werden.

Sie verwenden den Funktionsbaustein zum Zeitpunkt SELECTION.

Funktionsumfang

Der Funktionsbaustein *RP_EDIT_NAME* dient zur Aufbereitung des Namens gemäß den jeweiligen nationalen Vorschriften in HR-Standard-Auswertungen.

Die Aufbereitung des Namens wird laut den Eintragungen der Tabelle *HR-Namensaufbereitung* (T522N) vorgenommen.

Die HR-Namensaufbereitung hängt außerdem von folgendem ab:

- MOLGA (Buchungskreis, Personalbereich, Personalteilbereich) aus der Sicht *Personal-/Personalteilbereich* (V_T001P)
- Report-Format aus Tabelle *Format für HR-Namensaufbereitung* (T522F) (Default: '01')
- Herkunft der Daten
 - Infotyp Daten zur Person (0002)
 - Infotyp Familie/Bezugsperson (0021)
 - Infotyp Family data (0148)
- Kennzeichnung Namensaufbereitung aus Informationstyp *Daten zur Person* (0002) (Feld *Sonderform* (KNZNM))

Parameter

IN	1)	Infotyp P0001	Org. Zuordnung (P0001-ENAME)
	2)	Infotyp P0002	Daten zur Person
		Infotyp P0021	Familie/Bezugsperson
		Infotyp P0148	Family data (Japan)
	3)	MOLGA	(T001P-MOLGA)
	4)	LANGU	Sprache für Anrede
			SPACE - SAP-Anmelde-Sprache SY-LANGU
			x - Empfängersprache P0002-SPRSL

Namensaufbereitung gemäß nationalen Vorschriften

	5)	\$\$FORMAT	Namens-Format für Aufbereitung >
	6)	NAMEL	Maximale Länge von EDIT-NAME
OUT	1)	\$EDIT-NAME	Aufbereitete Übergabe-Leiste für Namen
	2)	\$RET-CODE	Return-Code von Aufbereitung des Namens

Prüfung

- Tabelle *Format für HR-Namensaufbereitung* (T522F)
- Tabelle *HR-Namensaufbereitung* (T522N)

HR-Adreßaufbereitung nach postalischen Vorschriften

Verwendung

Um Adressen laut den postalischen Vorschriften aufzubereiten, können Sie folgende Programmierhilfe verwenden:

Funktionsbaustein: *HR_MAKE_ADDRESS*

Sie verwenden den Funktionsbaustein zum Zeitpunkt SELECTION.

Funktionsumfang

Der Funktionsbaustein *HR_MAKE_ADDRESS* ruft den Funktionsbaustein *ADDRESS_INTO_PRINTFORM* auf. Dieser bereitet die Anschrift eines Mitarbeiters gemäß international-postalischen Vorschriften auf.

Wie die Adresse im einzelnen bearbeitet wird, können Sie in der Dokumentation zum Funktionsbaustein *ADDRESS_INTO_PRINTFORM* lesen.

Parameter

IN	1) Infotyp P0001	Org. Zuordnung (P0001-BUKRS für T001F)
	2) Infotyp P0002	Daten zur Person (P0002-ANRED für Anrede)
	3) Infotyp P0006	Daten für Anschrift
	4) P0001-ENAME	Name gemäß nationalen Vorschriften (in Verbindung mit RP_EDIT_NAME)
	5) LINE_COUNT	Anzahl der ausgegebenen Zeilen
	6) Name	RP_EDIT_NAME oder CNAME aus Infotyp <i>Daten zur Person</i> (0002)
OUT	ADRS	Struktur der aufbereiteten Adresse

Prüfung

keine

Prüfen und Ausgeben von Personal-Identifikationsnummern

Verwendung

Personal-Identifikationsnummern werden vor allem von den HR-Landesversionen genutzt (z.B. USA, Kanada, Dänemark u.a.) und sind dort von wesentlicher Bedeutung. Z.B. kann im Infotyp *Daten zur Person* (0002) als Personal-Identifikationsnummer die Sozialversicherungs- oder Steuernummer eines Mitarbeiters hinterlegt sein.

In manchen Reports kann es sinnvoll sein, daß das System Personal-Identifikationsnummern mit berücksichtigt.

Für die Prüfung und Ausgabe der Personal-Identifikationsnummer aus dem Infotyp *Daten zur Person* (0002) stehen Ihnen folgende Programmierhilfen zur Verfügung:

- Funktionsbaustein: *RP_FETCH_ALTERNATE_PERNR*
- RMAC-Makro: *RP-WRITE-ALTER-PERID*



Diese Programmierhilfen sollten Sie in allen allgemeinen HR-Reports gemeinsam anwenden (z.B. *Mitarbeiterliste* (RPLMIT00), *Dauer der tariflichen Zugehörigkeit* (RPLTRF00), *Vorschläge zur tariflichen Umstufung* (RPLTRF10)).

Sie verwenden die Programmierhilfen zum Zeitpunkt START-OF-SELECTION.

Funktionsumfang

Zur Prüfung der Personal-Identifikationsnummer aus dem Infotyp *Daten zur Person* (0002) zieht das System den Funktionsbaustein *RP_FETCH_ALTERNATE_PERID* heran.

Parameter *RP_FETCH_ALTERNATE_PERNR*

OUT:	1) SW-ALTER-PERID 2) RETCODE	Schalter alternative PERID (YES/NO) Return-Code aus RE549B
------	---------------------------------	---

Prüfung

- Transaktion: *HR: Merkmale* (PE03)
- Merkmal: *Berücksichtigung der Personal-Id. aus Infotyp 0002* (PERNO)
- Der Funktionsbaustein bestimmt in Abhängigkeit vom benutzerbezogenen Molgawert (T513A), ob eine Personal-Identifikationsnummer vorliegt.



Wenn Sie zusätzlich das RMAC-Makro *RP-WRITE-ALTER-PERID* verwenden, werden die Personal-Identifikationsnummern vom System geprüft und ausgegeben.

Das RMAC-Makro *RP-WRITE-ALTER-PERID* sorgt zusätzlich dafür, daß Personal-Identifikationsnummern in der Ausgabeliste des Reports in einer separaten Ausgabezeile angezeigt werden.

Prüfen und Ausgeben von Personal-Identifikationsnummern

Parameter *RP-WRITE-ALTER-PERID*

IN	1) \$\$\$W-ALTER-PERID	Schalter alternative PERID (YES/NO)
	2) P0002	Infotyp <i>Daten zur Person</i> (0002)
	3) Druckoption	x direkt im RMAC-Baustein oder SPACE indirekt im Anwendungs-Report
OUT	1) PERID	Personal-Identifikation aus P0002
	2) \$LENGTH	Länge der Personal-Identifikation

Prüfung

- Transaktion: *HR: Merkmale* (PE03)
- Merkmal: *Berücksichtigung der Personal-Id. aus Infotyp 0002* (PERNO)
- Var. Argument: Benutzerbezogener Wert aus Tabelle *Benutzerbezogene Werte* (T513A) (MOLGA)

Ausgeben des Listenzeitraums

Ausgeben des Listenzeitraums

Verwendung

Wenn Sie in der Ausgabeliste eines HR-Reports den Zeitraum mit anzeigen lassen möchten, können Sie folgende Programmierhilfe nutzen:

RMAC-Makro: *RP-ZEITRAUM*

Sie verwenden das RMAC-Makro zum Zeitpunkt TOP-OF-PAGE.

Funktionsumfang

Das RMAC-Makro *RP-ZEITRAUM* gibt in einer HR-Standard-Liste den Zeitraum (PN-BEGDA, PN-ENDDA) in der obersten Zeile jeder Seite der Liste aus.

Parameter

keine

Prüfung

keine

Beispiel

```

...
TOP-OF-PAGE.
  IF PN-BEGDA EQ PN-ENDDA.
    RP-STICHTAG.
  ELSE.
    RP-ZEITRAUM.
  ENDIF.
    
```

Ausgeben des Listenstichtags

Verwendung

Wenn Sie den Listenstichtag in der Ausgabeliste eines HR-Reports anzeigen lassen möchten, können Sie folgende Programmierhilfe verwenden:

RMAC-Makro: *RP-STICHTAG*

Sie verwenden das RMAC-Makro zum Zeitpunkt TOP-OF-PAGE.

Funktionsumfang

Wenn Sie das RMAC-Makro *RP-STICHTAG* verwenden, gibt das System in der Ausgabeliste des Reports den Stichtag (PN-ENDDA) in der obersten Zeile für jede Seite der Liste aus.

Parameter

keine

Prüfung

keine

Beispiel

```
(RP-STICHTAG)
...
TOP-OF-PAGE.
  IF PN-BEGDA EQ PN-ENDDA.
    RP-STICHTAG.
  ELSE.
    RP-ZEITRAUM.
  ENDIF.
```

Initialisieren des Puffers für Zugriff auf PCLn

Initialisieren des Puffers für Zugriff auf PCLn

Verwendung

Initialisiert den Hauptspeicher-Puffer für den Zugriff auf die Dateien PCLn (n=1,2,3,4) und wird im Zusammenhang mit den [Import-Makros \[Seite 21\]](#) und [Export-Makros \[Seite 23\]](#) verwendet.

Makro: *RP-INIT-Buffer*



Detaillierte Information über den Import/Export von Dateien finden Sie in [Import/Export Dateien im HR \[Extern\]](#)

Parameter

keine

Prüfung

keine

Integration

Folgende Includes beinhalten die Datenbeschreibung des Puffers und müssen in den Report eingebunden werden, der die Daten auf die Datenbank schreiben bzw. von der Datenbank lesen soll:

- RPPPXD00x
- RPPPXD10
- RPPPXM00



Lesen Sie dazu das Beispiel [Starten der Abrechnung im Testmodus \[Extern\]](#)

Import-/Export-Makros für Clusterdaten

Definition

Programme die Daten eines Clusters (z.B. RX) verarbeiten, greifen nicht selbständig auf das Cluster zu. Der Zugriff auf die Daten geschieht über eine definierte Schnittstelle, die durch Makros realisiert wird.

Import-/Export-Makros bilden die Grundlage für die Strukturbeschreibung eines Clusters. Die Strukturbeschreibung eines Clusters enthält die Auflistung der dort gespeicherten Daten und deren Reihenfolge.



Weitere Informationen zur Strukturbeschreibung finden Sie unter [Dateien PCL1, PCL2, PCL3 und PCL4 \[Extern\]](#).

Verwendung

Der Einsatz von Makros garantiert, daß der Zugriff auf die Clusterdaten zu gleichbleibenden Bedingungen stattfindet. Inkonsistenzen werden so vermieden.

Wenn sich die Art und Weise ändern soll, wie Daten gelesen oder gespeichert werden, muß durch den Einsatz von Makros diese Änderungen nur an einer Stelle – im betreffenden Makro – vorgenommen werden.



Das Coding der Makros muß im Programm vorhanden sein. Makros werden über Includes in die Reports eingebunden.

Struktur

Es gilt folgende **Namenskonvention**:

RP-aaa-bb-cc

Dabei gilt:

- *aaa* gibt die Art des Makros an:
 - IMP** Importmakro
 - EXP** Export-Makro
- *bb* gibt die Datenbanktabelle an, in der die Daten gespeichert werden:
 - C1** Datenbankobjekt PCL1
 - C2** Datenbankobjekt PCL2
 - C3** Datenbankobjekt PCL3
 - C4** Datenbankobjekt PCL4
- *cc* benennt das Cluster:
 - RX** Clusterobjekt RX

Import-/Export-Makros für Clusterdaten

RD Clusterobjekt RD

B2 Clusterobjekt B2

usw.

Makros werden über die ABAP-Befehle DEFINE/END-OF-DEFINITION definiert.

Import-Makros

Verwendung

Lesen Felder, Feldleisten oder interne Tabellen von der Datenbank.

Makro: *RP-IMP-Cn-xx*

Die Makros verwenden zur Verwaltung der Datenbanktabellen PCLn den ABAP-Befehl `IMPORT`. Die PCLn werden mit dem eindeutigen Schlüssel [xx-Key \[Extern\]](#) gelesen.



Benutzen Sie zum Import der Daten nur die Makros *RP-IMP-Cn-xx* oder den Funktionsbaustein `PYXX_READ_PAYROLL_RESULTS`.

Beim Import mit Makros liest das System die Datensätze nicht direkt von der Datei PCLn, sondern überprüft im Puffer-Directory, ob schon ein Satz mit gleichem Schlüssel im Hauptspeicher liegt. Wenn das nicht der Fall ist, wird der Satz von der PCLn in den Puffer gelesen und aus dem Puffer im Report bereitgestellt.

Bei erfolgreichem Import wird ein Returncode `RP-IMP-xy-SUBRC = 0` gesetzt. Beim gepufferten Lesen wird eine Prüfung auf Cluster-Berechtigung durchgeführt. Standard-Importprogramme finden Sie unter der Namenskonvention `RPCLSTxy` (xy = Clustername).



Die wichtigsten Importmakros *RP-IMP-C2-Rx* für die Abrechnungscluster Rx finden Sie in den Includes `RPCnRxx0` (z.B. *RP-IMP-C2-RU* im Include `RPC2RUU0`).

Parameter

Keine

Der entsprechende [xx-Key \[Extern\]](#) muß vor dem Aufruf gefüllt werden.

Prüfung

keine

Voraussetzung

Abhängig von der bereits gelesenen Datenmenge muß der Hauptspeicher-Puffer mit dem Makro [RP-INIT-BUFFER \[Seite 18\]](#) initialisiert worden sein.



Detaillierte Informationen über den Import-/Export von Dateien im HR erhalten Sie in [Import-/Export Dateien im HR \[Extern\]](#)

Beispiel

Ein Importmakro, das aus dem Cluster TT liest, könnte folgendermaßen aussehen:



```
DEFINE RP-IMP-C2-TT.
```

Import-Makros

```
IMPORT TABLE_01
FIELDSTRING_01
TABLE_02
TABLE_03
FROM DATABASE PCL2 (TT) ID TT-KEY.
END-OF-DEFINITION
```

In diesem Beispiel werden folgende Objekte aus dem Cluster TT der Datenbanktabelle PCL2 gelesen:

- Interne Tabellen TABLE_01, TABLE_02, TABLE_03
- Feldleiste FIELDSTRING_01

Export-Makros

Verwendung

Legen Felder, Feldleisten oder interne Tabellen auf der Datenbank ab.

Makro: *RP-EXP-Cn-xx*

Die Makros verwenden zur Verwaltung der Datenbanktabellen PCLn den ABAP-Befehl `EXPORT`. Die Zuordnung erfolgt über den eindeutigen Schlüssel [xx-Key \[Extern\]](#).



Benutzen Sie zum Export der Daten nur die Makros *RP-EXP-Cn-xx* oder den Funktionsbaustein `PYXX_WRITE_PAYROLL_RESULTS`.

Beim Export mit Makros schreibt das System die Datensätze nicht direkt auf die Datenbank, sondern in einen Hauptspeicher-Puffer. Am Ende des Programms werden die gepufferten Sätze auf der entsprechenden PCLn-Datenbank gesichert.

Das Sichern der gepufferten Sätze kann auch durch die Routine `Verbuchen` erzwungen werden.



Die wichtigsten Exportmakros *RP-EXP-C2-Rx* für die Abrechnungscluster *Rx* finden Sie in den Includes `RPCnRxx0` (z.B. *RP-EXP-C2-RU* im Include `RPC2RUU0`).

Parameter

Keine

Der entsprechende [xx-Key \[Extern\]](#) muß vor dem Aufruf gefüllt werden.

Prüfung

keine

Voraussetzung

Abhängig von der bereits gelesenen Datenmenge muß der Hauptspeicher-Puffer mit dem Makro [RP-INIT-BUFFER \[Seite 18\]](#) initialisiert worden sein.



Der Baustein *RP-EXP-C2-RA* schreibt die Abrechnungsergebnisse aus dem Speicher in den Hauptspeicher-Puffer zurück (Datei `PCL2`, Cluster `RA`).



Detaillierte Informationen über den Import-/Export von Dateien im HR finden Sie unter [Import-/Export Dateien im HR \[Extern\]](#)

Löschen von Abrechnungsergebnissen im Speicher

Löschen von Abrechnungsergebnissen im Speicher

Verwendung

Löscht die im Speicher gehaltenen Daten des Clusters xx der Datei PCLn. Die Struktur bleibt im Speicher erhalten.

Makro: *RP-REF-Cn-xx*



Die Daten im Hauptspeicher-Puffer und in der Datenbank bleiben erhalten.

Parameter

Der Baustein hat keinen Parameter. Es muß aber der entsprechende [xx-Key \[Extern\]](#) vor dem Aufruf gefüllt werden.

Prüfung:

keine



Der Baustein RP-REF-C2-RA löscht die im Speicher gehaltenen Abrechnungsergebnisse des Clusters RA der Datei PCL2.

Lesen eines Infotyps einer Person in einem Zeitraum

Verwendung

Um alle Datensätze eines Infotyps einer Person in einem gegebenen Zeitraum in eine interne Infotyp-Tabelle zu stellen, können Sie folgende Programmierhilfe verwenden:

Makro: *RP_READ_INFOTYPE*

Das Makro definieren Sie mit Hilfe des Sprachelements INFOTYPES.

Sie können das Makro in allen Programmen zu beliebigen Zeitpunkten oder auch in Funktionsbausteinen verwenden.



In der Datenbank PNP sollte normalerweise ein Infotyp mit GET PERNR gelesen werden. Das Lesen mit Hilfe des Makros *RP_READ_INFOTYPE* stellt eine Ausnahme dar.

Darüber hinaus können Sie den Funktionsbaustein *HR_READ_INFOTYPE* benutzen. Wie Sie den Funktionsbaustein einsetzen, lesen Sie in der Dokumentation zum Funktionsbaustein.

Voraussetzungen

- Der Gültigkeitsbeginn des Zeitraumes muß kleiner oder gleich dem Gültigkeitsende des Zeitraumes sein.
- Gültigkeitsbeginn und -ende sind korrekte Datumsangaben (am besten vom Type DATE).
- Die Infotyp-Tabelle muß zur Infotypnummer passen.
- Das das Makro verwendende Programm muß das Include DBPNPMAC beinhalten.

Funktionsumfang

Das Makro *RP_READ_INFOTYPE* sorgt dafür, daß alle Datensätze eines Infotyps einer Person in einem gegebenen Zeitraum in eine interne Infotyp-Tabelle gestellt werden.

Parameter

RP_READ_INFOTYPE pernr infty inftytab beg end

IN :	1)	Personalnummer der gewünschten Person
	2)	Infotypnummer des gewünschten Infotyps
	3)	Name der internen Infotyp-Tabelle
	4)	Gültigkeitsbeginn des Zeit-Intervalls
	5)	Gültigkeitsende des Zeit-Intervalls
OUT:	1)	PNP-SW-FOUND = 0, falls kein passender Datensatz im Datenbestand gefunden wird

Lesen eines Infotyps einer Person in einem Zeitraum

		PNP-SW-FOUND = 1, falls mindestens ein passender Datensatz im Datenbestand gefunden wird
	2)	PNP-SW-AUTH-SKIPPED-RECORD = 0, falls die HR-Berechtigungsprüfung keinen Datensatz wegen fehlender Berechtigung zurückgehalten hat
		PNP-SW-AUTH-SKIPPED-RECORD = 1, falls die HR-Berechtigungsprüfung mindestens einen Datensatz wegen fehlender Berechtigung zurückgehalten hat
	3)	interne Infotyp-Tabelle, gefüllt mit allen passenden Datensätzen, für die der Benutzer Berechtigung hat (evtl. leer)

Prüfung

keine

Beispiel

```
(RP_READ_INFOTYPE pernr infty inftytab beg end)
INFOTYPES: 0001.
RP-LOWDATE-HIGHDATE.
DATA: PERNR LIKE P0001-PERNR.
DATA: BEGDA LIKE P0001-BEGDA, ENDDA LIKE P0001-ENDDA.
PERNR = '12345678'.
BEGDA = LOW-DATE + 15.
ENDDA = HIGH-DATE - 5.
RP-READ-INFOTYPE PERNR 0001 P0001 BEGDA ENDDA.
IF PNP-SW-AUT-SKIPPED-RECORD EQ '1'.
  WRITE: / 'keine ausreichende Berechtigung'. STOP.
ENDIF.
IF PNP-SW-FOUND EQ '0'.
  WRITE: / 'Informationstyp 0001 fehlt'. STOP.
ENDIF.
```

Ersten Eintrag im Zeitraum zur Verfügung stellen

Verwendung

Um aus einer internen Infotyp-Tabelle den ersten Eintrag eines gewünschten Zeitraumes (evtl. eines gewünschten Subtyps) in den Tabellen-Kopf-Eintrag zu stellen, können Sie folgende Programmierhilfe verwenden:

Makro: *RP_PROVIDE_FROM_FRST*

Sie definieren das Makro mit Hilfe des Sprachelements INFOTYPES.

Sie benutzen das Makro *RP_PROVIDE_FROM_FRST* in Programmen der logischen Datenbanken PNP und PAP, in denen aus einer Infotyp-Tabelle der zeitlich erste Datensatz eines Zeitraumes (evtl. eines Subtyps) gelesen werden soll. Die Infotyp-Tabelle wurde vorher gefüllt (z.B. mit GET PERNR oder *RP_READ_INFOTYPE*). Sinnvollerweise ist das Makro nur anwendbar, falls der Infotyp die Zeitbindung 1 oder 2 hat.

Voraussetzungen

- Der Gültigkeitsbeginn des Zeitraumes muß kleiner oder gleich dem Gültigkeitsende des Zeitraumes sein.
- Gültigkeitsbeginn und -ende sind korrekte Daten (am besten vom Type DATE).
- Die Infotyp-Tabelle ist aufsteigend sortiert. Anderenfalls erhalten Sie den ersten passenden Tabelleneintrag, der nicht dem ersten zeitlichen entsprechen muß.

Funktionsumfang

Aus einer internen Infotyp-Tabelle wird der erste Eintrag eines gewünschten Zeitraumes in den Tabellen-Kopf-Eintrag gestellt.

Parameter

RP_PROVIDE_FROM_FRST infitytab subty beg end

IN :	1)	Name der internen Tabelle
	2)	gewünschter Subtyp oder SPACE , falls kein Subtyp spezifiziert werden soll
	3)	Gültigkeitsbeginn des Zeit-Intervalls
	4)	Gültigkeitsende des Zeit-Intervalls

Ersten Eintrag im Zeitraum zur Verfügung stellen

OUT:	1)	PNP-SW-FOUND: hat den Wert 0, falls im angegebenen Zeitraum kein passender Eintrag in der Infotyp-Tabelle ist, 1 sonst.
	2)	der passende Tabellen-Kopf-Eintrag, falls PNP-SW-FOUND = 1 oder einen initialen Tabellen-Kopf-Eintrag, falls PNP-SW-FOUND = 0

Prüfung

keine

Beispiel

```
(RP_PROVIDE_FROM_FRST infytyab subty beg end)
RP_PROVIDE_FROM_FIRST P0021 '1' PN-BEGDA PN-ENDDA.
IF PNP-SW-FOUND EQ '1'.
...
oder
RP_PROVIDE_FROM_FRST P0001 SPACE PN-BEGDA PN-ENDDA.
IF PNP-SW-FOUND EQ '0'.
WRITE: / 'Fehler: Org. Zuordnung fehlt'. REJECT.
ENDIF.
```

Letzten Eintrag im Zeitraum zur Verfügung stellen

Verwendung

Um aus einer internen Infotyp-Tabelle den letzten Eintrag eines gewünschten Zeitraumes (evtl. eines gewünschten Subtyps) in den Tabellen-Kopf-Eintrag zu stellen, können Sie folgende Programmierhilfe verwenden:

Makro: *RP_PROVIDE_FROM_LAST*

Sie definieren das Makro mit Hilfe des Sprachelements INFOTYPES.

Sie verwenden das Makro *RP_PROVIDE_FROM_LAST* in Programmen der logischen Datenbanken PNP und PAP, in denen aus einer Infotyp-Tabelle der zeitlich letzte Datensatz eines Zeitraumes (evtl. eines Subtyps) gelesen werden soll. Die Infotyp-Tabelle wurde vorher gefüllt (z.B. mit GET PERNR oder *RP_READ_INFOTYPE*). Sinnvollerweise wenden Sie das Makro nur an, falls der Infotyp (evtl. Subtyp) die Zeitbindung 1 oder 2 hat.

Voraussetzungen

- Der Gültigkeitsbeginn des Zeitraumes muß kleiner oder gleich dem Gültigkeitsende des Zeitraumes sein.
- Gültigkeitsbeginn und -ende sind korrekte Daten (am besten vom Type DATE).
- Die Infotyp-Tabelle ist aufsteigend sortiert. Anderenfalls erhalten Sie den letzten passenden Tabelleneintrag, der nicht unbedingt dem zeitlich letzten entspricht.

Funktionsumfang

Das Makro *RP_PROVIDE_FROM_LAST* sorgt dafür, daß der letzte Eintrag eines gewünschten Zeitraums in den Tabellen-Kopf-Eintrag der Ausgabeliste eines Reports gestellt wird.

Parameter

RP_PROVIDE_FROM_LAST infitytab subty beg end

IN :	1) Name der internen Tabelle
	2) gewünschter Subtyp oder <i>SPACE</i> , falls kein Subtyp spezifiziert werden soll
	3) Gültigkeitsbeginn des Zeit-Intervalls
	4) Gültigkeitsende des Zeit-Intervalls
OUT :	1) PNP-SW-FOUND: hat den Wert 0, falls im angegebenen Zeitraum kein passender Eintrag in der Infotyp-Tabelle ist, 1 sonst.
	2) der passende Tabellen-Kopf-Eintrag, falls PNP-SW-FOUND = 1 oder der initiale Tabellen-Kopf-Eintrag, falls PNP-SW-FOUND = 0

Prüfung

keine

Letzten Eintrag im Zeitraum zur Verfügung stellen

Beispiel

```
(RP_PROVIDE_FROM_LAST inftytab subty beg end)
RP_PROVIDE_FROM_LAST P0021 '1' PN-BEGDA PN-ENDDA.
IF PNP-SW-FOUND EQ '1'.
...
oder
RP_PROVIDE_FROM_LAST P0001 SPACE PN-BEGDA PN-ENDDA.
IF PNP-SW-FOUND EQ '0'.
WRITE: / 'Fehler: Org. Zuordnung fehlt'. REJECT.
ENDIF.
```



Der (nicht implementierte) PROVIDE-FROM-FINAL ist ein Spezialfall des PROVIDE-FROM-LAST :

```
PROVIDE-FROM-FINAL inftytab subty beg end =
RP_PROVIDE_FROM_LAST inftytab subty end end
```

Verändern von Personaldaten

Verwendung

Wenn Personaldaten verändert werden sollen, können Sie folgende Programmierhilfe verwenden:

Makro: *RP_UPDATE*

Das Makro verwenden Sie als Schnittstelle zwischen Reports, die Datenbankänderungen der Infotyp-Tabelle vornehmen. Sie setzen das Makro ausschließlich zum Ändern von Datensätzen ein. Das Hinzufügen oder Löschen von Datensätzen ist nicht erlaubt.

Sie verwenden das Makro nur in Programmen, die Daten verändern. Sie können das Makro zu einem beliebigen Zeitpunkt des Programmablaufs benutzen.

Voraussetzungen

- Alle Datensätze, die verändert werden sollen, sind auch in der Tabelle mit dem Originalzustand enthalten.
- Keine Key-Änderungen, d.h. kein Hinzufügen und Löschen von Daten.

Funktionsumfang

Parameter

RP_UPDATE old-table new-table

IN :	1) Name der internen Tabelle mit den Infotypsätzen vor Veränderung.
	2) Name der internen Tabelle mit den Infotypsätzen nach Veränderung.

Prüfung

keine

Beispiel

RP-UPDATE OLD-P0003 P0003.