

4.8.1 CAMT: Zu berücksichtigende Referenzobjekte

🌐 Sprache: **Deutsch** · [English](#)

📖 4.8.1 CAMT: Zu berücksichtigende Referenzobjekte

Die Methode `getReferences()` der abstrakten Klasse `AbstractCAMT054Logic` wird vor dem Importvorgang ausgeführt.

REFERENZ

INTEGRATOREN

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

Auf dieser Seite

- [Verwandte Seiten](#)

Die Methode `getReferences()` der abstrakten Klasse `AbstractCAMT054Logic` wird vor dem Importvorgang ausgeführt. Sie dient dazu, alle bereits existierenden Objekte einzulesen, die beim Import der Bankumsätze als mögliche Referenzobjekte herangezogen werden sollen (bspw. Kundenrechnungen, die noch offen sind und nicht bereits zu einem Bankumsatz zugeordnet wurden).

Das Standardverhalten des Nucleus berücksichtigt alle Referenzobjekte, die sich in einem jener Status befinden, die über die Methode `getSourceStates()` der Klasse `ReferenceFacade` definiert sind ([siehe dazu Abschnitt 4.7.3](#)).

Soll das Standardverhalten für die eigene Anwendung modifiziert, ergänzt oder erweitert werden, so ist eine eigene Methode `getReferences()` in der Klasse `CAMT054Logic` oder einer eigenen Implementierung der `AbstractCAMT054Logic` zu erstellen.

```
/**
 * Fetches all references that need to be considered during the CAMT.054 import process, e.g.
 * all open client billings/invoices that have not yet been linked to other bank transactions
 *
 * @note A dummy implementation of <code>org.nuclet.mt940.logic.AbstractMt940Logic</code> has
 * been provided with this class here, its methods are meant to be implemented with user
 * specific behaviour.
 *
 * @return all references that need to be considered during the CAMT.054 import process; the return
 * value comes as as <code>Map</code> in which the <code>BusinessObject</code> represents the
 * map's key while the text, which is meant to be used to link the reference to a bank
 * transaction, represents the map's value
 *
 * @throws BusinessException might be thrown in case of errors or other exceptions
 */
public Map<BusinessObject, String> getReferences() throws BusinessException
{
    final HashMap<BusinessObject, String> mpReferences = new HashMap<BusinessObject, String>();

    final Query<ClientBilling> qryGetReferencesStateAB = QueryProvider.create(ClientBilling.
class)
        .where(ClientBilling.NuclosState.eq(ProcessClientBillingSM.State_AB.getId()))
        .and(ClientBilling.Facturated.eq(Boolean.FALSE));

    final List<Rechnung> lstReferencesStateAB = QueryProvider.execute(qryGetReferencesStateAB);

    for (final ClientBilling clientBilling : lstReferencesStateAB) {
        mpReferences.put(clientBilling, clientBilling.getBillingNumber());
    }

    final Query<ClientBilling> qryGetReferencesStateXY = QueryProvider.create(ClientBilling.
class);
        .where(ClientBilling.NuclosState.eq(ProcessClientBillingSM.State_XY.getId()))
        .and(ClientBilling.Facturated.eq(Boolean.FALSE));

    final List<ClientBilling> lstReferencesStateXY = QueryProvider.execute
(qryGetReferencesStateXY);

    for (final ClientBilling clientBilling : lstReferencesStateXY) {
        mpReferences.put(clientBilling, clientBilling.getBillingNumber());
    }

    return mpReferences;
}
```

Verwandte Seiten

Nuclet: CAMT(sind im Handelsnuclet verfügbar)

CAMT-Datenblatt

[Öffnen](#)

Schnittstellen

Alle Integrationen

[Öffnen](#)

