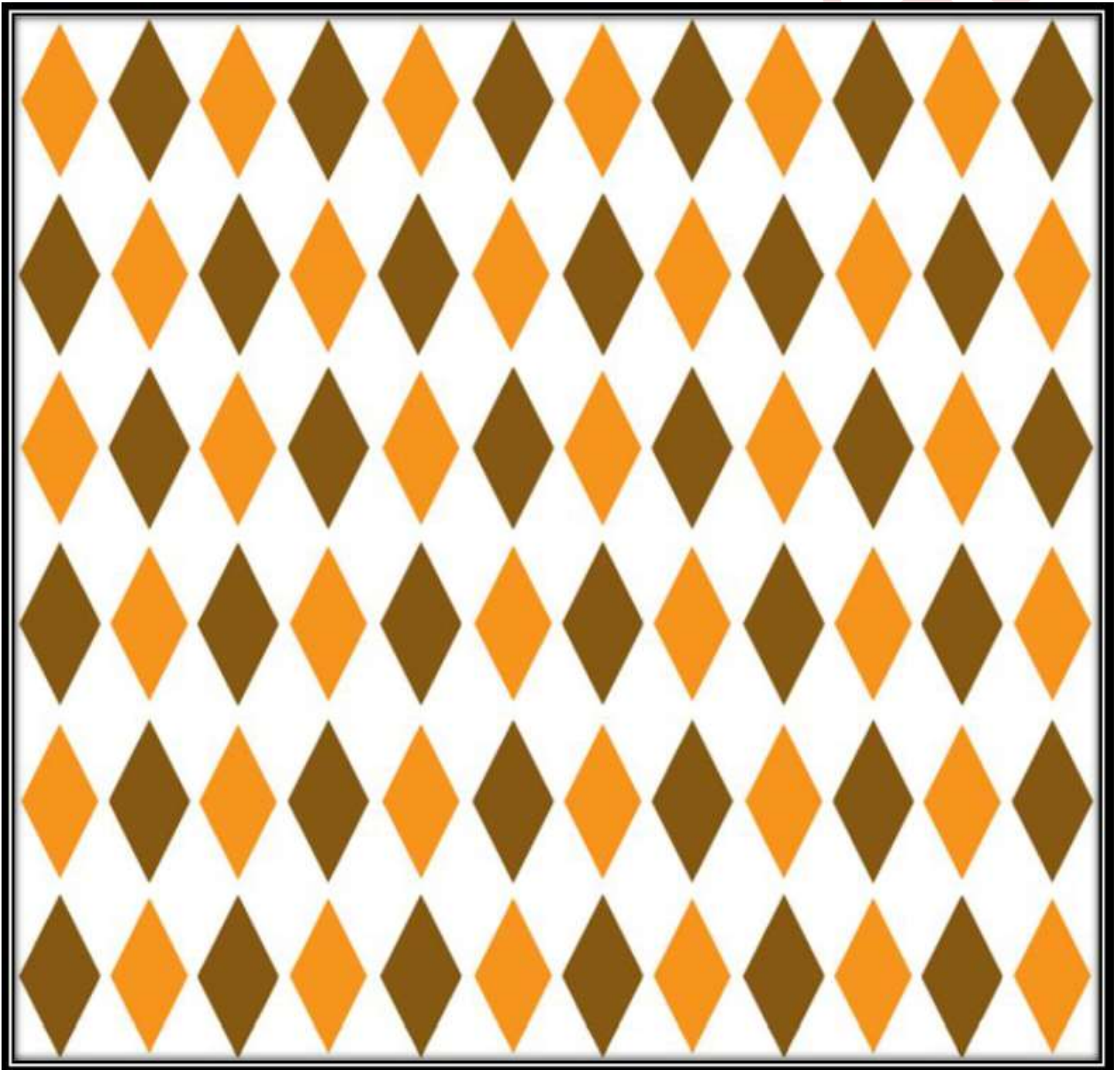


OpenERP **CUSTOMER DEPOSITS** MODULE ♦

Manual for Version 1.7 **BETA**

The **account_deposit** module is an addon module for OpenERP 7.0, the well-known integrated enterprise resource planning (ERP) software that is developed by OpenERP s.a.



Please note:

This module is currently in beta. It has been tested internally and with a few customers, but is being released in order to get community feedback.

We DO NOT recommend installing this module in a production environment.

To send your feedback to the developers of this module:

If you have a current support contract with Ursa, contact us using one of the standard methods available:

<http://www.ursainfosystems.com/helpdesk.html>

If you do not have a current Support contract with Ursa, please use Launchpad to send messages to the development team:

<https://answers.launchpad.net/openerp-shared/+addquestion>

Remember to indicate which module your question relates to.

To report bugs you encounter with this module:

If you have a current support contract with Ursa, contact us using one of the standard methods available:

<http://www.ursainfosystems.com/helpdesk.html>

If you do not have a current Support contract with Ursa, please use Launchpad to send bug reports to the development team.

<https://bugs.launchpad.net/openerp-shared/+filebug>

Remember to indicate which module your question relates to.

Limited development resources are available to address bugs from users who do not have a current support contract with Ursa Information Systems. While we read every bug report submitted, we may not be able to personally respond to or address your bug report.

CONTENTS

1. License	3
2. Introduction	4
3. User Experience Changes	5
4. User Interface Changes	5
5. Version Information	6
6. Hosted Location	6
7. OpenERP Server Installation	6
8. OpenERP User Interface Installation	7
9. Configuration	8
10. Example Workflow	9
11. Support and Feedback	12
12. Reporting Bugs	12
13. About Ursa Information Systems	13

Online version of this document is at **not available at beta**

Last updated: April 15th, 2013

1. License

Copyright (C) Ursa Information Systems – www.ursainfosystems.com

Copyright (C) Tiny SPRL – tiny.be

This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

You should have received a copy of the GNU Affero General Public License along with this program. If not, see <http://www.gnu.org/licenses/>

2. Introduction

This module supports the recording of advanced customer deposits. These occur when a vendor receives money from their customer before a service is provided or shipment of goods is made. These funds are liabilities due to the fact that if the performance of the sale is not completed the customer would want a refund of their deposit because they have received nothing.

By default, Customer Payments credit AR. They either reconcile the full or partial amount of a specific Invoice, or remain until the balance is applied to an Invoice. This may not be a desirable way to handle Customer Deposits, and in some cases is not compliant with jurisdictional accounting practices (such as GAAP).

These types of transactions are commonly referred to as Unearned Revenue, Unearned Income and Deferred Revenue. They should be shown in the liability section of the balance sheet. No revenue (profit & loss statement) effect should occur until the revenue has been earned. When the revenue has been earned (shipment of goods or services performed) the liability is removed and the revenue is recognized. As these deposits are received they need to be specifically tracked by who made it so they may be applied against their open AR balances.

Users can elect to record a payment as a regular AR credit, or as a Deposit. They can also elect to default all payments as Deposits. Customer who pay more than the required amount for an Invoice may consider this money owed to them (again, a liability) instead of an AR credit.

By default, this kind of Customer Payment would need to be managed via a series of Manual Journal Entries.

3. User Experience Changes

This module streamlines the user experience for recording customer deposits. It is not necessary to record them manually.

The standard workflows are relatively unchanged – checking a checkbox and optionally updating the deposit account is all that the user has to do.

4. User Interface Changes

Partner Configuration

A new field is added to the Accounting tab of form view of the Customer record. The Deposit Account is the account used as the default receivable account when a payment is processed as an Advanced Deposit.

The screenshot shows the 'Accounting' tab of a Customer record form. The 'Deposit Account' field is highlighted with a red border and contains the value '250000 Advanced Deposits'. Other fields include 'Fiscal Position', 'Latest Full Reconciliation Date', 'Account Receivable' (120010 Account Receivable), 'Account Payable' (200010 Account Payable), 'Supplier Payment Term', 'Customer Payment Term', 'Total Receivable' (0.00), and 'Credit Limit' (0.00).

Field	Value
Fiscal Position	[Dropdown]
Latest Full Reconciliation Date	[Calendar]
Account Receivable	120010 Account Receivable
Account Payable	200010 Account Payable
Deposit Account	250000 Advanced Deposits
Supplier Payment Term	[Dropdown]
Customer Payment Term	[Dropdown]
Total Receivable	0.00
Total Payable	0.00
Credit Limit	0.00

4. User Interface Changes (continued)

Customer Payments

A new option is added to Customer Payment form view to (checked by default) to indicate if this payment is to be recorded as an Advanced Deposit. A new field to indicate the Receivable Account is also added, the value (by default) is taken from the new Deposit Account field on the partner record, but can be modified.

Customer	Customer	Date	04/16/2013
Paid Amount	0.00	Payment Ref	e.g. 003/10
Payment Method	Bank (USD)	Memo	e.g. Invoice SAJ/0042
		Record as Deposit	<input checked="" type="checkbox"/>
		Receivable Account	250000 Advanced Deposits
		Account	100002 Bank

BETA PRE

5. Version Information

This version was written and tested for OpenERP 7.0 and works equally well on Windows and Linux distributions – as it has no platform specific code.

This version was released on April 16th, 2013.

6. Hosted Location

The source code is available at:

http://bazaar.launchpad.net/~rcarnes/openerp-shared/7.0/files/head:/beta/account_deposit/

7. OpenERP Server Installation

This module installs as a standard OpenERP module.

The module needs to physically reside on your OpenERP server. This means the files need to be copied to a drive that the server uses to store OpenERP modules (also called addons).

There are several methods to copy the source for this module to your OpenERP server. The entire openerp-shared collection can be branched or checked out, or you can download the ZIP file for just this module (**not available at beta**):

This task may need to be performed by your System Administrator. Be sure to check that the permissions on the module folder and the files within it are set so that the server can read everything properly.

For more information on installing modules or using Launchpad to checkout or branch a collection of modules, refer to the following links:

<http://help.openerp.com>

<https://help.launchpad.net/>

8. OpenERP UI Installation

Once your OpenERP server has the module and can access it, you need to install it via the User Interface. The following instructions assume the use of the Web Client.

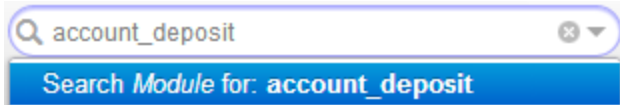
A: Get OpenERP to recognize the new module, by navigating the menu sequence *Settings* → *Modules* → *Update Module List*.

B: Navigate the menu sequence *Settings* → *Modules* → *Installed Modules*.

C: Clear the search filter by clicking the X next to the word Installed.



D: Enter and search for the term *account_deposit* in the search box to locate the module.



E: Locate the module and click the *Install* button.



9. Configuration

The module assumes that any accounts used for Advanced Deposits are configured as *Internal Type: Receivable* and *Account Type: Payable*. Since this is an AR account, *Allow Reconciliation* should be checked.

Account Code and Name

250000 - Advanced Deposits

Parent	250 Customer Deposits	Debit	0.00
Internal Type	Receivable	Credit	0.00
Account Type	Payable	Balance	0.00
Active	<input checked="" type="checkbox"/>		
Default Taxes			
Allow Reconciliation	<input checked="" type="checkbox"/>		

Internal Notes

BETA

10. Example Workflow

A. Record a Customer Payment for \$100:

BNK2/2013/0001

Customer	Customer	Date	04/16/2013
Paid Amount	100.00	Payment Ref	
Payment Method	Bank (USD)	Memo	
		Record as Deposit	<input checked="" type="checkbox"/>
		Receivable Account	250000 Advanced Deposits
		Account	100002 Bank

Payment Information | Journal Items

Credits

Journal Item	Account	Date	Due Date	Original Amount	Open Balance	Full Reconcile	Allocation
							0.00

Difference Amount	100.00
Payment Difference	Keep Open

The Journal Items created for this Payment:

Journal	Effective date	Name	Reference	Partner	Account	Tax	Journal Entry	Debit	Credit	Reconcile Ref
Bank (USD)	04/16/2013	/	BNK220130001	Customer	100002 Bank		BNK2/2013/0001	100.00	0.00	
Bank (USD)	04/16/2013	BNK2/2013/0001	BNK220130001	Customer	250000 Advanced Deposits		BNK2/2013/0001	0.00	100.00	

10. Example Workflow (CONTINUED)

B. Later (after work has been invoiced) record a payment to apply the deposit:

BNK2/2013/0002

Customer	Customer	Date	04/16/2013
Paid Amount	0.00	Payment Ref	
Payment Method	Bank (USD)	Memo	
		Record as Deposit	<input checked="" type="checkbox"/>
		Receivable Account	250000 Advanced Deposits
		Account	100002 Bank

Payment Information | Journal Items

Credits

Journal Item	Account	Date	Due Date	Original Amount	Open Balance	Full Reconcile	Allocation
00001 (00001)	120010 Account Receivable	04/16/2013	04/16/2013	100.00	100.00	<input checked="" type="checkbox"/>	100.00

100.00

Debits

Journal Item	Account	Date	Due Date	Original Amount	Open Balance	Full Reconcile	Allocation
BNK2/2013/0001 (BNK220130001)	250000 Advanced Deposits	04/16/2013		100.00	100.00	<input checked="" type="checkbox"/>	100.00

100.00

Difference Amount	0.00
Payment Difference	Keep Open



10. Example Workflow (CONTINUED)

B. (Continued)

The Journal Items created for this Payment:

Journal	Effective date	Name	Reference	Partner	Account	Tax	Journal Entry	Debit	Credit	Reconcile Ref
Bank (USD)	04/16/2013	/	BNK220130002	Customer	100002 Bank		BNK2/2013/0002	0.00	0.00	
Bank (USD)	04/16/2013	BNK2/2013/0001	BNK220130002	Customer	250000 Advanced Deposits		BNK2/2013/0002	100.00	0.00	A21
Bank (USD)	04/16/2013	00001	BNK220130002	Customer	120010 Account Receivable		BNK2/2013/0002	0.00	100.00	A22

BETA RELEASE

11. Support and Feedback

If you have a current support contract with Ursa, contact us using one of the standard methods available:

<http://www.ursainfosystems.com/helpdesk.html>

If you do not have a current Support contract with Ursa, please use Launchpad to send messages to the development team:

<https://answers.launchpad.net/openerp-shared/+addquestion>

Remember to indicate which module your question relates to.

Limited support resources are available to users who do not have a current support contract with Ursa Information Systems. While we read every question submitted, we may not be able to personally respond or to support you.

12. Reporting Bugs

If you have a current support contract with Ursa, contact us using one of the standard methods available:

<http://www.ursainfosystems.com/helpdesk.html>

If you do not have a current Support contract with Ursa, please use Launchpad to send bug reports to the development team.

<https://bugs.launchpad.net/openerp-shared/+filebug>

Remember to indicate which module your question relates to.

Limited development resources are available to address bugs from users who do not have a current support contract with Ursa Information Systems. While we read every bug report submitted, we may not be able to personally respond to or address your bug report.

13. About Ursa Information Systems

Ursa Information Systems is an OpenERP Gold Partner in the United States.

We use our extensive experience to bring you high-level consulting services and a personalized implementation of OpenERP to optimize and automate your current operations.

Using the science of Operations Research, we bring a structured approach to help you model and implement new business processes, allowing for continued growth and success.

Using OpenERP, we implement an intuitive and powerful system that helps automate information sharing between business units, reducing redundancy and increasing efficiency. We have created a tradition of successful implementations of OpenERP and would value the opportunity to help make you more successful.

Find us on the web at <http://www.ursainfosystems.com>