Infor CloudSuite Industrial System Administration Guide Cloud Edition

Release 9.01.x

Publication Information

Release: Infor Cloudsuite Business (Industrial) 9.01.x
Publication Date: July 31, 2017
Document code: csbi_9.01.x_csbiag_cl_csind_en-us

Contents

About this guide

This guide contains background or supplemental information to answer questions you may encounter as you manage and maintain CloudSuite Industrial. CloudSuite Industrial offers you, as System Administrator, considerable power to manage the system and its users in accordance with good business practices and company policy. While this guide assists users of both cloud and on-premise versions of CloudSuite Industrial, cloud users can not perform all tasks. Any task that involves access to the utility server or the database server must be performed by a cloud administrator.

This manual is intended as a reference. For instructions on how to install the CloudSuite Industrial system, see the *Infor CloudSuite Industrial Installation Guide*. For information about using the Cloud Suite Industrial system, see the CloudSuite Industrial online help. Consult this manual when you have questions about CloudSuite Industrial design or architecture.

Additional documentation

The most current version of all documentation is available on the Infor support web pages.

Online help

CloudSuite Industrial online help gives you instant access to procedures and information about forms and fields. You can access help from CloudSuite Industrial forms, from other topics within Help, or from the search (index) function. Select **Help > > Contents and Index** from the CloudSuite Industrial title bar to open the help, or use the **F1** key to get help on any form or field.

Developer-level help

To access the help for developers, select **Help > Customizing Forms**.

System requirements and prerequisite knowledge

For the most up-to-date list of software and hardware requirements for Infor products, see the *Technical Requirements for Multi-Tenant Cloud Edition* Guide.

Contacting us: plan your communication

To make sure the correct analyst is assigned to your case and to expedite the resolution of your questions, please have the following information available when you call us:

- Your company name and phone number
- CloudSuite Industrial version release and point release
- Database software version and release, if applicable
- Platform or environment (Example: Windows 2012)
- Functional area (Examples: Production, Administration, etc.)
- What you were doing (Example: Printing a report)
- What type of data you were accessing or trying to access (Example: Customer data)
- If you received an error message, the full message text and error number
- If data was entered into a form other than by using a keyboard, test data entry into the form using a keyboard before calling so that we will know if there is a problem with the form or with the method of data entry.
- If you are calling back on an existing case, the case number

Sign up for support

If you are not currently on support and would like more information on your support options, please call your customer account representative. If you are not sure who your account representative is, contact Infor Customer Service.

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal.

If we update this document after the product release, we will post the new version on this website. We recommend that you check this website periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

License management

1

As System Administrator, you need to manage your license agreement, and CloudSuite Industrial allows you to do that. Use these forms to manage your license:

- License Management form
- Licensed Modules form
- User Modules form

For more information about licensing, see the *Infor CloudSuite Industrial Licensing Administration Guide*.

Multi-site

2

In CloudSuite Industrial, each site's database does not have to be connected to any other database in order to function. Data is transferred through a process called replication. The system manages its own replication which is different from SQL's replication. We do not support standard SQL replication as a means to transfer data from one site to another.

The connection between sites can be either of these:

- Transactional: This is a constant live connection from one site to another. Any update is made to the target site immediately. The down side is that if the target site is down for any reason, the transfer of data does not occur.
- Non-transactional: There is no live connection between sites. If using a non-transactional (also referred to as delayed replication or asynchronous replication) setup, data that is shared must be replicated.

For more detailed information on replication and multi-site use, see these documents on Infor Xtreme:

- *Infor Mongoose Replication Reference*
- *Infor CloudSuite Industrial Multi-Site Planning Guide*
- *Infor CloudSuite Industrial Multi-Site Implementation Guide*

Authorizations

3

You may limit user access to certain forms or limit what they can do on certain forms by way of authorizations. Use the following four forms to manage these authorizations:

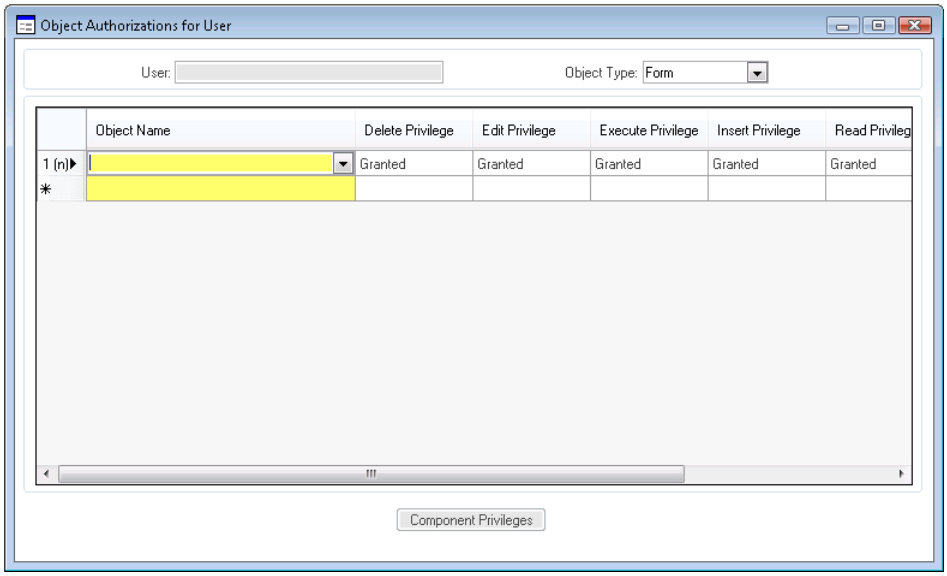
- Object Authorizations for User
- Object Authorization for Group
- Users
- Groups

Object Authorizations for User

Use this form to set up form-level security for a user or to set up security for a user based on a middle-tier IDO. An object in this case is either the name of the form or the name of the IDO.

If the user specified on this form is part of a group, the user authorizations that you set up on this form override any group authorizations that apply to the same user.

You can open the form right from the Explorer window, or you can click **User Authorizations** on the Users form.



The following privileges can be granted or revoked for the user for a specified form in the Object Authorizations For User form:

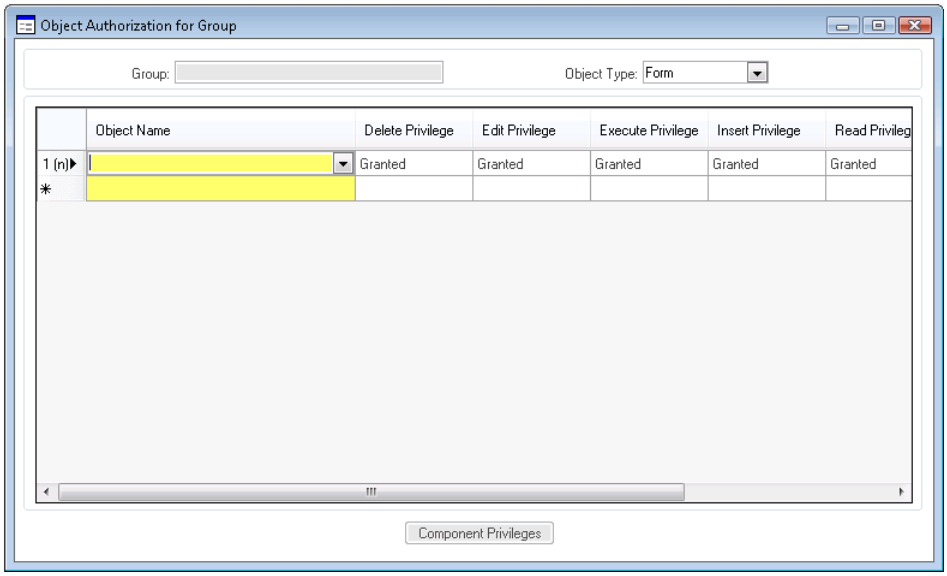
- **Delete** controls whether the user can delete records.
- **Edit** controls whether the user can edit existing data.
- **Execute** controls all privileges, including whether the user can open the form. When execute is revoked, all privileges are unavailable.
- **Insert** controls whether the user can insert records.
- **Read** controls whether the user can read data.
- **Bulk Update** controls whether the user can perform a bulk update, such as multiple find and replace operations.
- **Update** controls whether the user can save changes to existing data.

Object Authorization for Group

Use this form to set up form-level security for a group of users or to set up security for a group based on a middle-tier IDO.

Any user authorizations for individuals will override the group authorizations defined on this form.

You can open the form right from the Explorer, or you can click **Group Authorizations** on the Groups form.

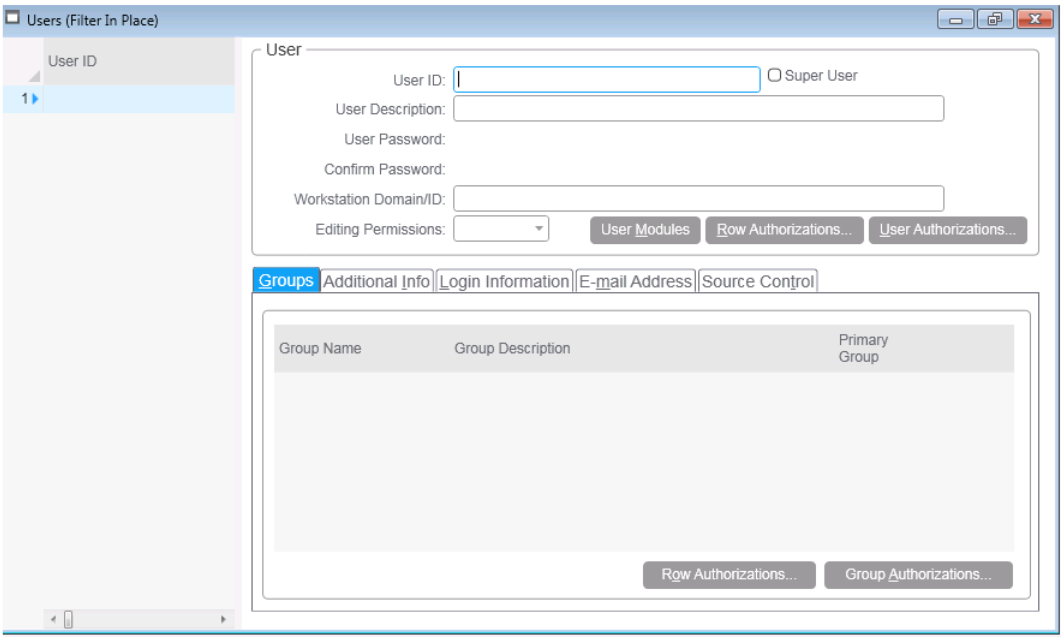


The following privileges can be granted or not granted for the group in the Object Authorization For Groups form:

- Delete controls whether the user can delete records.
- Edit controls whether the user can edit existing data.
- Execute controls all privileges, including whether the user can open the form. When execute is not granted, all privileges are unavailable.
- Insert controls whether the user can insert records.
- Read controls whether the user can read data.
- Update controls whether the user can save changes to existing data.
- Bulk Update controls whether the user can perform a bulk update, such as multiple find and replace operations.

Users

Use this form to manage users. You can use this form to create users, though most of the time you will use IFS to create users.



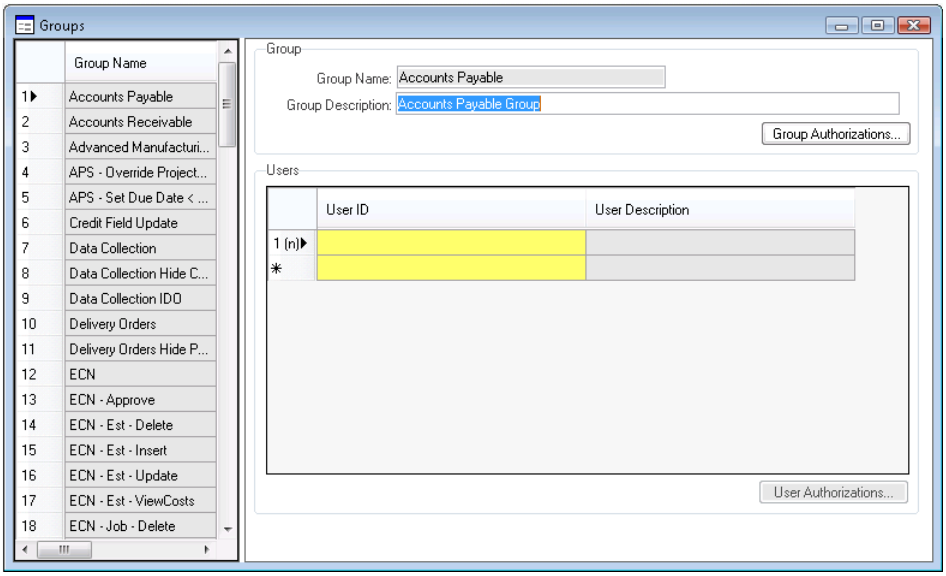
You can also specify:

- Passwords needed to log on to the application
- Workstation IDs so users bypass the logon dialog box
- Email addresses so notifications can be sent about automated tasks
- Editing permissions that determine whether users can enter edit mode to create or customize forms
- Security authorizations for this user at a form level or a component level
- Groups to which the user belongs and security authorizations for that group
- Additional information about the user that the application needs.

Note: Initially, only the supplied default system administrator user ID can create or delete other user IDs. Any user who is added to the System Administration group or designated as a super user can access the Users form and change the password for any other user. Users can change their own passwords on the User Information form.

Groups

Use the Groups form to create groups and to assign user IDs to them. These groups can be organized in any way that makes sense for your company or organization -- by location, by organization, by job description, and so on.



Note: Although the system allows you to modify or delete the default groups, doing so may cause future conversion problems while upgrading and other problems. We recommend you copy the records from the default group to a new group name and modify that. Do not delete or modify default groups.

Assign users to groups so you can then create group authorizations that apply to every individual/user ID in the group.

Create a Super User

A Super User can run all forms and perform all actions on all forms for which they hold a license. In some cases, actions may have to be performed by the 'sa' user account. For example, a user with an CloudSuite Industrial Entity module license can not access the User Modules form. In this case, you need to access the User Modules form with the 'sa' user account.

If you create a Super User, you do not need to set any other authorizations for this user. The Super User status overrides all other types of authorizations.

1. Open the Users form.
2. Select the desired User ID.
3. Select the **Super User** field.
4. Save the record.

Assign a user to a group

1. Open the Users form.

2 Select the desired User ID.

3 On the **Groups** tab, select the Group Name from the drop down list.

4 Save the record.

Note: When you assign a user to a group (for example, user Bob is assigned to the Accounts Payable Group), that user gets access to all the forms associated with that group. If you want user Bob to have most of the access associated with that group but not all, you need to edit the user authorizations for user Bob. See"Edit authorizations for a user in a group" on page 17

Assign a user to a primary group

In general, it is easier to first create the group, then assign user authorizations. This allows you to assign each user a Primary Group that specifies authorizations without going through each user's authorizations form by form.

1 On the Users form, select the desired User ID.

2 On the **Groups** tab, select the group name from the Group Name drop down list.

3 Select the **Primary Group** field.

When selected, this field indicates that the corresponding user group is a primary group. The system uses the group designated as the Primary Group to load the correct version of customized forms for the selected user.

Users may belong to more than one group, but only one group may be designated as the Primary Group, and only the Primary Group is used for loading group versions of a customized form. For example, suppose user Bob is a member of the Accounts Payable and Accounts Receivable groups with the Accounts Receivable group marked as Bob’s primary group. If the user Bob launches a form, the system looks for a group-level customized version of that form for the Accounts Receivable group, not the Accounts Payable group. If such a customized version exists, and if there isn't also a user-level customized version for user Bob, then that is the form the system will display.

View group authorizations

1 Open the Groups form.

2 Select the desired group name.

3 Click **Group Authorizations**. The Object Authorization for Group form is displayed.

4 In the **Object Name** field, select an object. The privileges are listed in the columns to the right, and those privileges are either granted or not granted.

Edit authorizations for a user in a group

User authorizations override group authorizations. So for example, you might want user Bob to have all of the authorizations of the accounts payable group, except for access to the Accounting Periods form.

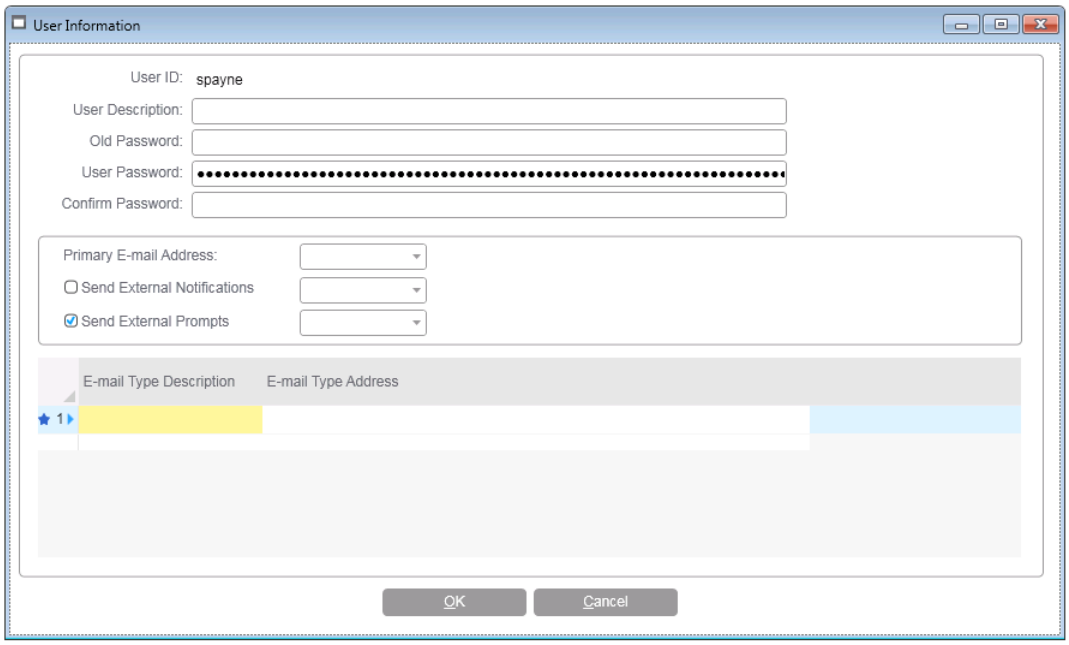
1. Assign a user to a group on the Users form.
   See "Assign a user to a group" on page 15
2. Open the Users form.
3. Select the user whose authorizations you want to edit.
4. Click **User Authorizations**.
   The Object Authorizations for User form is displayed.
5. In the **Object Name** field, select the object you wish to edit for this user. In this example, it is AccountingPeriods.
6. Depending on what you want, change the desired privileges either to **Granted** or **Revoked**.

Forms security

Form level security is checked when a user opens a form. The AccountAuthorizations table is queried to see what privileges have been granted to the user or to the group to which the user belongs.

Change passwords

Any user who is added to the System Administration group or designated as a super user can access the Users form and change the password there for any other user. Users without such authorization can change their own passwords on the User Information form.


User Information
User ID:
spayne
User Description:
Old Password:
User Password:
Confirm Password:
Primary E-mail Address:
Send External Notifications
Send External Prompts
E-mail Type Description
E-mail Type Address
1
OK
Cancel

Recovering after a system crash

4

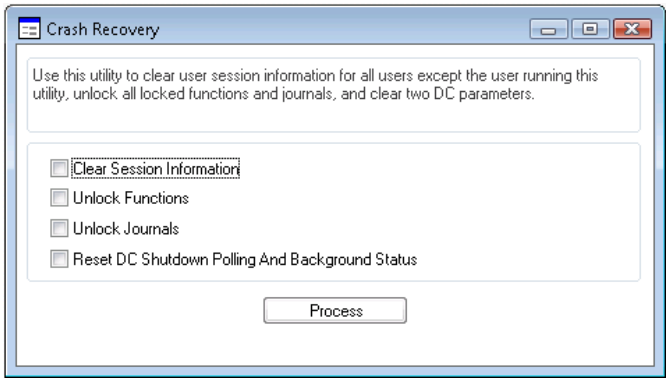
To get everything working properly again in the event of a system crash, you may have to do any or all of the following:

- Unlock locked functions
- Unlock locked journals
- Recover locked tokens.

You can go to the individual forms mentioned above or you can run the Crash Recovery utility. It is not always necessary to run the Crash Recovery utility.

Crash Recovery utility

Use this utility to clear user sessions for all users except the user running this utility, unlock all locked functions and journals, and clear two DC parameters.

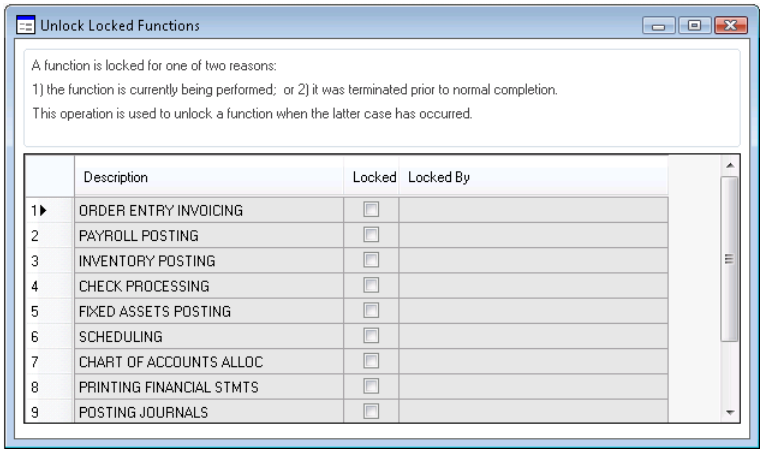


You do not need to run this utility following a database server crash. The functions of this utility are performed automatically when the database server is rebooted.

Note: Any users still logged in are kicked out of the system when you run this utility. DO NOT run this utility against a database that is still being accessed by users.

Unlock Locked Functions

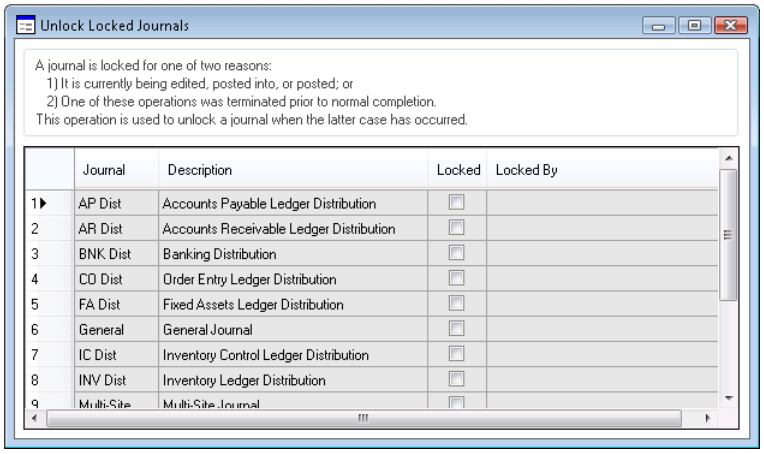
If the **Locked** field is selected next to the function, it is locked. If the Locked field is cleared next to the function, it is unlocked.



Use the Unlock Locked Functions utility to unlock functions that were somehow aborted prior to normal termination. If a Locked message displays when you enter certain processes, use this utility to unlock it.

Unlock Locked Journals

If the **Locked** field is selected next to the journal, it is locked. If the **Locked** field is cleared next to the journal, it is unlocked.



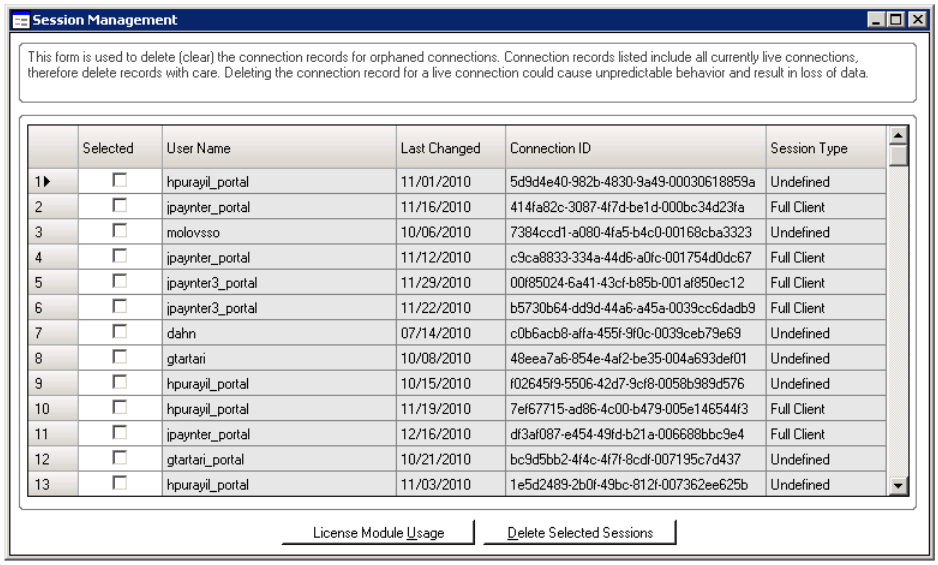
Use the Unlock Locked Journals utility to unlock journals that are locked as a result of one of these operations being terminated prior to normal completion:

- Edit a journal

- Post information into a journal
- Post a journal

Recover locked tokens

If CloudSuite Industrial terminates unexpectedly, it is possible that license tokens in use at the time are locked, thus preventing users from logging in. In this case, use the Session Management form to free up the locked tokens.



	Selected	User Name	Last Changed	Connection ID	Session Type
1	☐	hpurayil_portal	11/01/2010	5d9d4e40-982b-4830-9a49-00030618859a	Undefined
2	☐	jpaynter_portal	11/16/2010	414fa82c-3087-4f7d-be1d-000bc34d23fa	Full Client
3	☐	molovsso	10/06/2010	7384ccd1-a080-4fa5-b4c0-00168cba3323	Undefined
4	☐	jpaynter_portal	11/12/2010	c9ca8833-334a-44d6-a0fc-001754d0dc67	Full Client
5	☐	jpaynter3_portal	11/29/2010	00f85024-6a41-43cf-b85b-001af850ec12	Full Client
6	☐	jpaynter3_portal	11/22/2010	b5730b64-dd9d-44a6-a45a-0039cc6dadb9	Full Client
7	☐	dahn	07/14/2010	c0b6acb8-affa-455f-9f0c-0039ceb79e69	Undefined
8	☐	gtartari	10/08/2010	48eea7a6-854e-4af2-be35-004a693def01	Undefined
9	☐	hpurayil_portal	10/15/2010	f02645f9-5506-42d7-9cf8-0058b989d576	Undefined
10	☐	hpurayil_portal	11/19/2010	7ef67715-ad86-4c00-b479-005e146544f3	Full Client
11	☐	jpaynter_portal	12/16/2010	df3af087-e454-49fd-b21a-006688bbc9e4	Full Client
12	☐	gtartari_portal	10/21/2010	bc9d5bb2-4f4c-4f7f-8cdf-007195c7d437	Undefined
13	☐	hpurayil_portal	11/03/2010	1e5d2489-2b0f-49bc-812f-007362ee625b	Undefined

Setting up an audit log

5

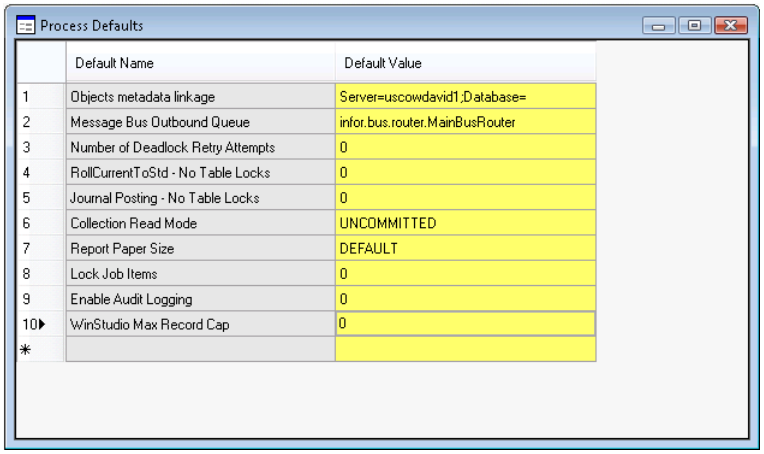
Use the audit log to see which user performed what action at what time and on what form.

Use these three forms to manage your audit logging:

- Process Defaults
- Audit Log Types
- Audit Log

Process Defaults

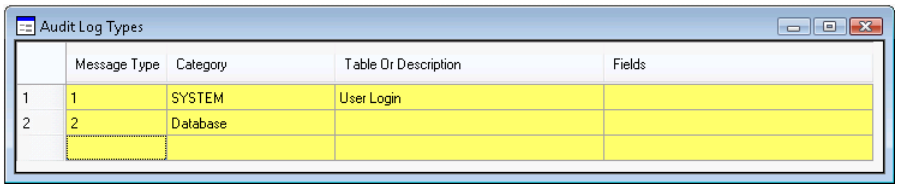
Use the Process Defaults form to enable audit logging.



Set the **Default Value** field to **1** for **Enable Audit Logging**. A default value of **0** turns the audit log functionality off.

Audit Log Types

Use the Audit Log Types form to create types of messages to include in the audit log. By default, the system generates messages when users log on (type 1) and when users open a form (type 2). Developers can create other types of messages that are generated by event handlers of the Add Entry to Audit Log response type.



The Audit Log Types form displays the following information:

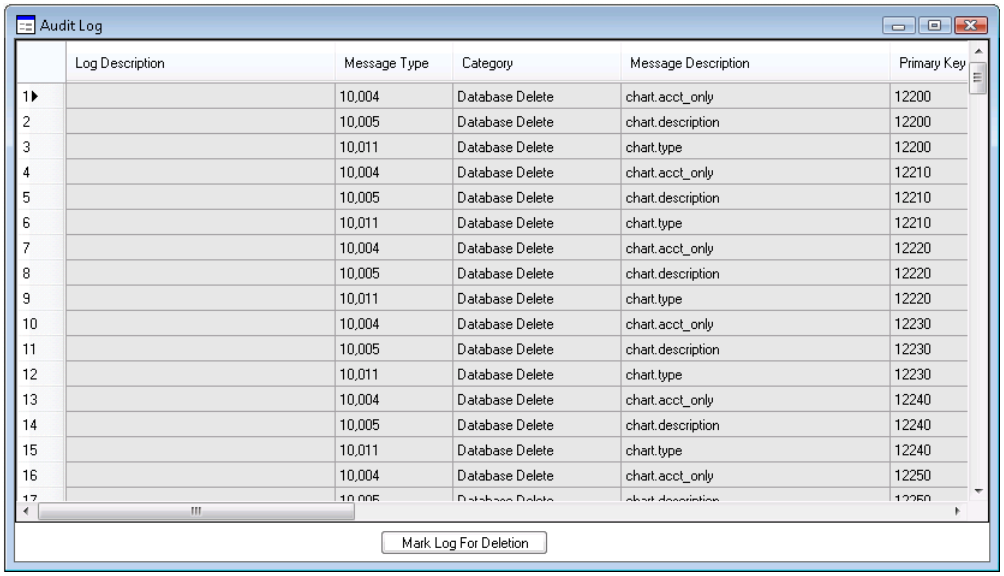
- **Message Type** indicates the type of message in the audit log. By default, messages about users logging on are **1**, messages about users opening forms are **2**, and messages about users modifying forms or objects are **3**. Custom messages are numbered 10,000 and greater.
- **Category** identifies the type of auditing. Three values can appear in this field:
  - **System**: Types of events are User Login and Open Form. You cannot select a category of **System**.
  - **Custom Form Event**: This allows programmers to add audit logging to any event handler; for example, when a user selects a tab on a form.
  - **Database**: Database displays when records are added, updated, and deleted, and the message displays `Database: Add`, `Database: Update`, or `Database: Delete`, depending on the action performed.
- **Table or Description**. For System and Custom Form Events, this field describes the event being logged. You can change the contents of this field at any time. The value in this field appears in the Audit Log.

  For an entry where **Category** is set to **Database**, this field indicates the table that will have an audit log.
- **Fields** is used only if the **Category** is **Database**. This can be a specific database field (cust_num, for example) or an asterisk (*). Entry of an asterisk indicates that all fields in the table should be entered in the audit log.

Audit Log

Use the Audit Log form to view and delete messages in the audit log.

The Audit Log form displays this information:



- **Log Description** describes the type of audit log. For example, if you opened the Purchase Order Lines form, the **Log Description** lists the form name "PurchaseOrderLines".
- **Message Type** indicates the type of message in the audit log. By default, messages about users logging on are **1**, messages about users opening forms are **2**, and messages about users modifying forms or objects are **3**. Custom messages are numbered 10,000 and greater.
- **Category** identifies the type of auditing. Three values can appear in this field:
  - **System**: Types of events are User Login and Open Form. You cannot select a category of **System**.
  - **Custom Form Event**: This allows programmers to add audit logging to any event handler; for example, when a user selects a tab on a form.
  - **Database**: Database displays when records are added, updated, and deleted, and the message displays `Database: Add`, `Database: Update`, or `Database: Delete`, depending on the action performed.
- **Message Description** is a description of the message type. The description will be one of three vales:
  - **User Login**
  - **Open Form**
  - **Custom Message**
- **Primary Key** identifies the database record being changed. It contains the fields that make up the primary key. When multiple fields make up a key, the values are concatenated and separated by dashes. For example, Purchase Order P000000001 Line 5 Release 2 appears as P000000001-5-2.
- **Old Value** is only used when **Category** is set to **Database**. Based on the activity - Add, Update, or Delete - the following is displayed:
  - Add: Blank
  - Update: Value prior to update
  - Delete: Value prior to delete

- **New Value** is only used **Category** is set to **Database**. Based on the activity - Add, Update, or Delete - the following is displayed:
  - Add: Value after the add
  - Update: Value after the update
  - Delete: Blank
- **User Name** shows The user ID whose actions generated the audit log message is displayed.
- **Date/Time** is the system date and time that the audit log entry was made.

Improving performance

6

This chapter presents information on how to improve the performance of your system. In general, techniques for improving performance are designed to reduce unnecessary processing, network traffic, and blocking. The techniques minimize:

- The number of unneeded records stored in tables
- The number of records retrieved in queries
- The number of locks on records in queries
- The duration of locks on records
- The size and duration of transactions
- The fragmentation of tables and indexes.
- Avoiding timeouts.

Customizations

Customizations to CloudSuite Industrial should be evaluated for performance along with standard product components. You should ensure that indexes for new tables are designed correctly and maintained adequately. If a custom feature performs slower than when it was first implemented, determine whether unneeded records are causing the performance reduction.

Custom reports and processes should be evaluated to see if they are reading the least number of records. BI queries should be similarly evaluated for efficiency.

User actions

End users' practices can slow CloudSuite Industrial performance. Actions such as querying an unlimited number of records into a form, specifying overly broad query criteria in reports, and running unneeded reports increase network traffic and can tax database resources.

Reduce the number of rows returned in queries

Users can choose to retrieve all rows or any specified maximum number of rows in queries. This option overrides a default limit set on queries that return data records and items in drop-down lists. However, unlimited queries can degrade system performance or exceed the resources of the utility server and the client machine.

The process default **WinStudio max record cap** allows you to set a system-wide limit on the number of records or drop-down list items that users can query into forms. See the help for the Process Defaults form.

Reduce the scope of reports

Report users should be sure to set criteria in a such way that the system returns only the information needed for the purposes of the report. Users should limit the range of time frames and other criteria to prevent needless processing.

Users should avoid running unnecessary reports.

Replication

When you use replication, set up an intranet with a master site and share certain _all tables. This allows other sites to use views into shared _all tables on the master site, reducing replication traffic between the sites. Advantages and requirements are described in the *Multi-Site Planning Guide*. The process is described in the *Multi-Site Implementation Guide*.

Locking and blocking

Locking prevents users from reading data being changed by other users, and prevents multiple users from changing the same data at the same time. If locking is not used, data within the database may become logically incorrect, and queries executed against that data may produce unexpected results. SQL Server enforces locking automatically. Locking can occur at record, page, or table level.

Blocking occurs when one user holds a lock and a second user requires a conflicting lock type. This forces the second user to wait, blocked by the first. Typically, the second user sees an hourglass while trying to process or save records. Most blocking problems happen because a single process holds locks for an extended period of time, causing a chain of blocked processes. A design goal is to minimize the amount of time a record is locked to reduce the potential blocking of another user.

A deadlock arises when two processes have data locked, and each process cannot release its lock until the other process has released its lock. SQL Server rolls back one of the transactions and then allows the other transaction to continue.

Save each modified row in a separate transaction

You can specify that WinStudio save each modified row in a separate transaction. By default, without this setting, all modified rows are sent to the mid-tier to be processed within a single transaction.

Saving one row per transaction can alleviate blocking problems in some forms. Whether the setting enhances performance depends on the complexity of a form's save operation and the speed of the network connection. Forms that require highly complex saves may benefit from the setting, especially if connection speed is adequate. Forms that require simple save operations are less likely to benefit, especially if the additional network traffic required in saving one row at a time slows the system.

In default WinStudio behavior, if one row fails, the entire transaction is rolled back. When you save each row in a separate transaction and a row fails, all previous rows remain committed.

Note: If your form design requires that all modified rows be committed as a unit, with processing on the unit before and/or after a save operation, saving rows in separate transactions may not be appropriate. If an error occurs on a row, some rows may be committed and others not committed.

The following forms save rows in separate transactions:

- Job Orders
- Customer
- Vendor
- Item
- Job Operations
- Job Materials
- Current Operations
- Current Materials
- Purchase Orders
- Purchase Order Lines
- Customer Orders
- Customer Order Lines

For instructions on setting this feature, see the information about Save One Row Per Transaction in the help.

Set the collection read mode (Transaction Isolation Level)

You can specify whether form queries read committed or uncommitted data by setting the **Collection Read Mode**. The setting applies to queries that load primary collections, secondary collections, and

lists, and to in-collection validations. It also applies to background-task queries that generate reports and to background-task stored procedures. The setting does not affect SQL SELECT operations or other processing coded in stored procedure (method) calls. The default **Collection Read Mode** value, **UNCOMMITTED**, corresponds to the Transact-SQL statement **SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED**.

With the **UNCOMMITTED** setting, which allows the reading of uncommitted data, users do not have to wait for other long-running transactions that access the same dataset to complete before their queries can complete.

With the **COMMITTED** setting, a query reads committed data and returns only data for which the query can get a shared lock.

The base, system-wide transaction isolation level is set on the Process Defaults form. Note that if this setting is unsuitable for all forms and tasks, you can override it for selected forms and tasks. You can set the isolation level for individual reports and stored procedures on the Background Task Definitions form. You can also override the system setting at the form level in WinStudio edit mode. See the Cloud Suite Industrial help for information about setting the Read Mode for a collection and about setting transaction isolation levels.

Prevent locking of the journal table during mass journal posting

Posting a large number of transactions with the Mass Journal Posting form can set an exclusive lock on the journal table. This prevents users from inserting data into the table until the posting is complete.

The process default **Journal Posting - No Table Locks** allows you to override this behavior. The value **1** prevents an exclusive table lock from being taken during mass journal posting. The value **0** does not prevent a lock. The setting applies to both forms and the background task that runs journal posting without middleware.

See the help about Process Defaults for more information.

Prevent blocking of other processes when rolling current costs to standard costs

Normally, the Roll Current Cost to Standard Cost utility processes all current costing data at one time. When processing large amounts of data, this can block other processes, such as adding jobs, adding CO lines, using the Purchase Order Receiving form, or opening the Customer Order Lines form.

The process default **Roll Current to Standard - No Table Locks** allows you to override this behavior. Changing this setting from **0** (the default) to **1** can eliminate this blocking. You should understand, however, that selecting this option can also slow down the Roll Current Cost to Standard Cost utility processing.

Prevent deadlocks on the item table during certain operations

Deadlocks on the item table can occur during certain operations involving bills of materials that contain many items. The **Lock Job Items** process default determines whether job items are locked during these operations. With the value **1**, operations such as releasing a job, which copies the bill of materials, and posting a job will lock all item records in blocks according to the operation number. The default value, **0**, does not lock item records.

See the help about Process Defaults for more information.

Avoid long delays from deadlocks

A high value for the process default **Number of Deadlock Retry Attempts** can cause users to experience excessive delays from deadlocks. The value 3 is a recommended starting point.

See the help about Process Defaults for more information.

Populating an empty database

7

Note: For more information about the forms listed in this section, refer to the online help.

When populating an empty database, it is recommended that you enter information on these forms in the order in which they appear:

Users

Use this form to register users to the application. A user ID is required for each user who logs on to the application.

Object Authorizations for User

Use this form to set up form-level security for a user or to set up security for a user based on a middle-tier IDO.

Chart of Accounts

Use the Chart of Accounts form to define account numbers which will be used throughout the system to record, track, and report costs. The General Ledger is tied to other parts of the system through the Chart of Accounts and the journals.

Accounting Periods

Use the Accounting Periods form to maintain the accounting periods used by General Ledger.

Bank Reconciliations

CloudSuite Industrial maintains a transaction history of all activity against the bank checking accounts your company uses. You can view this information through the Bank Reconciliations form, and also prepare bank reconciliations by tracking what has been recorded by the bank.

Financial Statement Definition

You can define your own financial statements that best suit your reporting needs. Using the Financial Statement Definition form, you can add new financial statements. You then use the Financial Statement Definition Columns and Financial Statement Line Definition forms to define the content and format of a given statement.

Financial Statement Definition - Columns

See Financial Statement Definition.

Financial Statement Line Definition

See Financial Statement Definition.

General Parameters

Use the General Parameters form to set the parameters used throughout the system.

Shop Floor Control Parameters

Use the Shop Floor Control Parameters form to set up default settings for shop floor data and for running the Scheduler.

Order Entry Parameters

Use the Order Entry Parameters form to set default values for customer order entry.

Inventory Parameters

Use the Inventory Parameters form to set parameters in your inventory system for use throughout the entire system. Changing these parameters can affect the way the system is run and how it handles inventory.

Planning Parameters

Use the Planning Parameters form to enable features and options used throughout the MRP and APS planning functions.

Purchasing Parameters

Use the Purchasing Parameters form to enter default values for use throughout Purchasing.

Accounts Payable Parameters

Use the Accounts Payable Parameters form to specify the accounts to use in the General Ledger. These accounts are used throughout A/P for distributions to the G/L.

Accounts Receivable Parameters

Use the Accounts Receivable Parameters form to enter the default parameter values the system applies throughout Accounts Receivable.

Departments

Use the Departments form to maintain a list of all departments referenced by work center and employee records. The **Department** field also appears on the Fixed Assets Class Codes form. You use departments to group work centers for application of overhead rates and direct labor cost. To report departmental shop floor capacity, combine all work centers of the department.

Product Codes

Use product codes to group similar types of items and assign each group an identifying code. Product codes can override system-wide parameters for a subset of items.

Distribution Accounts

Distribution accounts are a set of accounts grouped by warehouse and product code.

Tax Codes

Use the Tax Codes form to specify the percentage of tax to charge to a customer and the G/L account to which the sales tax is to be posted.

Tax Systems

Use the Tax Systems form to establish the tax system and Tax Codes used in processing Value Added Tax or sales tax related information. The Value Added Tax function processes items in Customer Order Entry, Purchasing, Accounts Receivable, and Accounts Payable.

Tax Parameters

Use the Tax Parameters form to define global (applying to both tax systems) switches and dates, and to set optional tax data printing options.

Prov/States

Use the Prov/States form to enter state or province abbreviations.

Billing Terms

Use the Billing Terms form to identify and maintain billing terms to be applied to customers. The **Billing Terms** code displays on the Customers and Vendors forms. You can identify default billing terms for each customer or vendor.

Miscellaneous Receipt Reason Codes

Use the Miscellaneous Receipt Reason Codes form to track the entry of miscellaneous receipt transactions. For example, you could use the code **RTS** to indicate a return to stock.

Miscellaneous Issues Reason Codes

Use the Miscellaneous Issues Reason Codes form to track the entry of miscellaneous issue transactions. For example, you could use the code **TES** to indicate the material was issued for testing.

Locations

Use the Locations form to maintain the list of valid places for inventory. These locations are used when establishing inventory balances and processing inventory transactions. The system initially creates a location of **Stock** and strongly recommends that you not delete this record, since several CloudSuite Industrial programs assume that **Stock** exists as a default.

Unit of Measure Codes

Use the Unit of Measure Codes form to maintain all units of measure that can be associated with an item. Transactions can then be tracked in various units. A base unit of measure is assigned to each item on the Items form.

Shift Codes

Use the Shift Codes form to maintain codes that identify particular shifts for use in payroll administration. You can create as many shifts as needed based on the starting and ending times for a flex-time work schedule. Use the Scheduling Shifts form to define shifts to schedule resources for working on operations.