Infor CloudSuite Industrial Enterprise Financials Customization Guide

Publication Information

Release: Infor CloudSuite Industrial Enterprise Financials 9.01

Publication date: October 17, 2016

Contents

About This Guide

This guide contains information and examples to help you customize Infor CloudSuite Industrial Enterprise Financials and change the mapping set up in the standard interface.

The first two chapters provide processing details for the two parts of the interface. The rest of the guide concentrates on what you need to know in order to customize SLSI. Later chapters in this guide build on concepts explained in earlier chapters, so for better understanding, read the chapters in order, like a textbook.

See the table below for content organization.

Organization

This table describes the chapters of the guide:

Section	Description
CloudSuite Industrial to SunSystems	Describes the flow through the SLSI interface in detail, including specifically what the SLSI Web application does during the process. It also includes some implementation details that you should be aware of.
SunSystems to CloudSuite Industrial	Describes the processing done by the three stored procedures provided with Infor CloudSuite Industrial Enterprise Financials. It also includes implementation details you should be aware of if you customize the stored procedures or add new SPs to bring data from SunSystems into CloudSuite Industrial.
Using Action Profiles	Explains the default action profiles in SLSI and how you can customize them by changing the underlying XSLs. You can customize action profiles for a specific target site/ business unit, object (XML document) and action (update or insert). You can also create new profiles to create additional record types in the SunSystems database.

Section	Description
Inbound and Outbound XML Schema	Gives examples of CloudSuite Industrial IDORequest XML documents that are sent to SLSI through replication, and examples of SunSystems SSML documents sent from SLSI to SunSystems Connect (SSC). We highlight the general schema used for each type of document and point to places where you can get additional information about the schema. This chapter also explains how and why some XMLs exported from CloudSuite Industrial require "flattening" before they reach SLSI.
Setting Process Defaults as Parameters in the XML	Lists and explains the SLSI process defaults you can choose during system setup, and change later. This chapter shows where the process defaults are stored as parameters in the inbound XML.
Customizing XSLs	Contains an extended example showing how changes to the default XSL can affect the outbound XML. This chapter also contains a pointer to some good tutorial and reference information on XSLs, as well as some tips about how the default XSLs use variables, functions, etc., to accomplish the desired mapping.
Default Mapping of Data Between CloudSuite Industrial and SunSystems	Shows the default mapping of the source data in CloudSuite Industrial to the target data in SunSystems done by the standard interface. Notes about transformations done to the data (calculations, truncations, defaults, and so on) are included. You can use this information while looking at the XSLs to better understand what the XSLs are doing.

Related Documents

Consult the following documentation and help files as needed. You can find the documents in the product documentation section of the Infor Xtreme Support portal, as described in "Contacting Infor" below.

- *CloudSuite Industrial Enterprise Financials Installation Guide*
- *CloudSuite Industrial Installation Guide*
- *SunSystems SQL Installation Guide*
- *SunSystems v6 - SQL Upgrade Guide*
- CloudSuite Industrial-SunSystems Interface (SLSI) online help
- CloudSuite Industrial online help
- SunSystems online help
- SunSystems Connect (SSC) Technical Reference online help

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal at http://www.infor.com/inforxtreme.

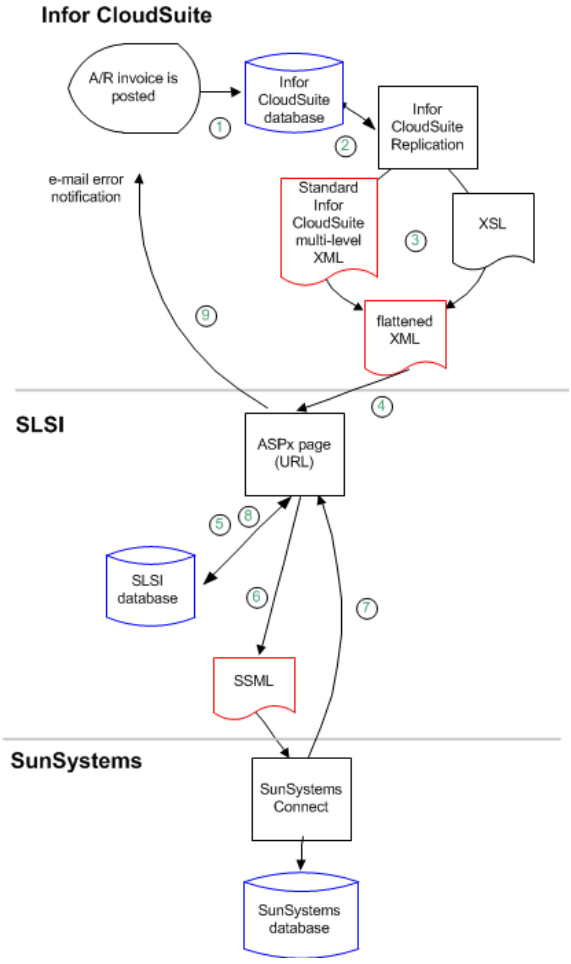
If we update this document after the product release, we will post the new version on this Web site. We recommend that you check this Web site periodically for updated documentation.

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

Chapter 1: CloudSuite Industrial to SunSystems: Processing Details

1

This chapter describes in detail the steps used in moving data from CloudSuite Industrial to Infor CloudSuite Industrial Enterprise Financials. The graphic below shows a sample data flow between CloudSuite Industrial and SunSystems. The process is automated and requires no user intervention unless an error occurs. The numbers in the flow correspond to the step descriptions on the following pages.



1 A CloudSuite Industrial user posts an A/R invoice. The record is saved in the CloudSuite Industrial application database.

2 Rules are set up in CloudSuite Industrial to replicate the EXTFIN category (which includes A/R invoice transactions) to a target site on the SUN intranet. See the *CloudSuite Industrial Enterprise Financials Installation Guide* for details.

   Non-transactional replication builds XML documents containing new or updated records. The initial version of the XML document uses the standard CloudSuite Industrial IDORequest format and may include nested data - for financial transactions that include both header and detail records. For examples of inbound XMLs and general information about the syntax and tags used in them, see “Inbound XML: CloudSuite Industrial IDORequest Schema” on page 27.

3 CloudSuite Industrial replication determines that the XML request is targeted for a site on the SUN intranet.

   For financial transactions, the system must find the XSL style sheet used to “flatten” the original XML document, and transform it (see “XSLs Used to Flatten Original XML Structure” on page 46). For a customer or vendor data transfer, there is no need to “flatten” this original XML document, so this transformation is skipped.

4 CloudSuite Industrial replication posts the inbound XML document to the SLSI Web application (ASP file) on the utility server where the SLSI interface is installed. (The path to this application is specified in the URL for the SUN intranet, as defined in the CloudSuite Industrial **Intranets** form.)

5 The SLSI Web application:

   - Parses the inbound IDORequest XML to extract the target site, collection object, action, and data.
   - Determines which SunSystems business unit the XML is intended for, using the target site and information from the SLSI Site to Business Unit Mappings form.
   - Extracts parameter information from the SLSI Process Defaults and Site to Business Unit Mappings forms, and adds those parameters to the XML. (See Chapter 5, “Setting Parameters in the XML.”)
   - Looks up the step(s) defined for this target site and object on the SLSI Action Profiles form. (See Chapter 3, “Using Action Profiles.”)
   - For each action profile, applies the associated XSL (stored in the SLSI database) to build the outbound request to update the SunSystems database.
   - Creates the batch header, batch details, and initial attempt record. These are stored in the SLSI database and can be accessed through the Batch Headers, Batch Details, and Batch Posting Attempts forms in SLSI.

6 For each action profile, the SLSI Web application posts an SSML request through SunSystems Connect (SSC).

7 SunSystems imports the SSML and attempts to update its database tables. It passes back a response XML.

8 The SLSI Web application receives the response XML, extracts any error information, and updates the Batch Posting Attempts record in the SLSI database. From this record, you can view the inbound IDORequest XML from CloudSuite Industrial (including the parameters added above), the outbound SunSystems SSML request, and the response XML returned by SSC.

It updates the status of the batch in the Batch Headers and Batch Details forms. In the Batch Posting Attempts form, it updates the status of the batch sequence and displays any error message for the sequence.

9 If a failure occurs, either during the transmission attempt or during the import into SunSystems, the SLSI Web application looks up in the SLSI database the Notification E-mail address specified for the target site (on the Site to Business Unit Mappings form). It then sends an e-mail containing the error information to that address.

Implementation Notes

- If a numeric overflow occurs to the right of the decimal point — for example, if a numeric CloudSuite Industrial field permitting 8 digits to the right of the decimal is mapped to a numeric SunSystems field permitting 3 digits to the right of the decimal — then SunSystems' built-in rounding will handle the overflow by rounding up when importing the value.
- For open credit and debit memos, the CloudSuite Industrial Apply To Invoice value (SunSystems analysis code 9) will be zero "0" and the CloudSuite Industrial Reference value (SunSystems analysis code 10) will be the credit or debit number.
- The setup of SunSystems transaction, base, and reporting currencies depends on your implementation. However, the A/P and A/R imports pass Transaction currency amounts, not Base currency amounts. The Base currency amount will be calculated based upon the exchange rate passed. The Ledger import passes both the Transaction and Base currency amounts. The Reporting currency (if set up) is not impacted by the imports.
- In order to successfully import a zero-amount ledger line during import of an A/R, A/P, or G/L transaction, the transformation sets the transaction's exchange rate (which maps to the SunSystems ConversionRate) to "0" (zero).
- The values for the following elements are set to 0 (zero) during an import to SunSystems when the elements are not included: tax_basis (this element is not applicable for header records); and freight, misc_charges, sales_tax, and sales_tax_2 (these elements are not applicable for "distribution" or "detail" records).
- When an analysis code is set to "Optional" on the SunSystems Chart of Accounts, and on the SunSystems Journal Types form's **Analysis** tab, a dimension is selected for the code, SunSystems' import process requires a value for the analysis code. So when an optional analysis code is not exported from CloudSuite Industrial, the CloudSuite Industrial export process specifies a value of "NOT USED".
- The CloudSuite Industrial multiple due date functionality is not supported when using Infor CloudSuite Industrial Enterprise Financials.
- Customer invoice approval and revision/pay days processing (if used) must be executed in CloudSuite Industrial, not SunSystems. The CloudSuite Industrial approval_status is not exported. Refer to the CloudSuite Industrial help for information about using these features.
- Views in the CloudSuite Industrial database contain subsets of the columns for the customer, custaddr, vendor, and vendaddr tables. Any changes to the data in these views will generate an XML document that is exported to the CloudSuite Industrial-SunSystems Interface (SLSI). This allows the system to generate XML documents only for changes that are pertinent to the interface.

If you need additional information from these CloudSuite Industrial tables to be passed to SunSystems, then you must drop and re-create the views in SQL with the appropriate changes. (RowPointer and keys for the table on which the view is based must be included in the view.) Once any view has been modified, you must regenerate the CloudSuite Industrial replication triggers to account for the change.

The views are named SLSICustomerView, SLSICustaddrView, SLSIVendorView, and SLSIVendaddrView.

The original SQL scripts used to add the views to the database are stored on the utility server in C:\Program Files\Infor\SyteLine\SLSI\ApplicationDb\Views.

To customize the views, you may want to modify and rerun these scripts.

Chapter 2: SunSystems to CloudSuite Industrial: Processing Details

2

This chapter describes in detail the processing done by the three stored procedures provided with the Infor CloudSuite Industrial Enterprise Financials interface:

- GetSunSystemsCurrencyCodesAndRatesSp
- GetSunSystemsCustomerPostedBalanceAndCreditLimitSp
- GetSunSystemsChartOfAccountsSp

These SPs are stored in the CloudSuite Industrial application database. You can schedule them to run on a regular basis that is determined by your system's requirements; they are not triggered by a CloudSuite Industrial application request. For information about how to set up and schedule the stored procedures, see the *CloudSuite Industrial Enterprise Financials Installation Guide*.

Get SunSystems Customer Posted Balance and Credit Limit

Both CloudSuite Industrial and SunSystems hold customer A/R posted balance information. In an integrated environment, A/R payments and related transactions are administered in SunSystems — not CloudSuite Industrial — and A/R invoices, credits and other transactions posted in CloudSuite Industrial are loaded into SunSystems. Because of this, SunSystems generally has the most accurate customer A/R posted balances.

How It Works

The SQL stored procedure GetSunSystemsCustomerPostedBalanceandCreditLimitSp provides a batch process to load SunSystems' customer A/R posted balances into CloudSuite Industrial. It does the following:

- Selects the posted balance (sum of the customer's open items) and credit limit for each customer in the SunSystems database.

- If a record exists in the CloudSuite Industrial customer and custaddr tables for the customer, the record is updated with the posted balance and credit limit from SunSystems.

Whenever this process is run, the CloudSuite Industrial customer A/R posted balances are accurate and in sync with SunSystems. In between the batch process runs, CloudSuite Industrial's customer A/R posted balance is automatically updated to include newly posted invoices, credits and related transactions but does not include any payments or adjustments that took place in SunSystems.

Implementation Notes

- CloudSuite Industrial uses its posted balance information for all customer credit checking, so the posted balance needs to be up-to-date.
- To intially transfer the current posted balance for each existing CloudSuite Industrial customer into SunSystems, you will need to enter transactions in SunSystems that assign the current posted balance for each existing CloudSuite Industrial customer. This should be done after all other information is entered and the interface is completely set up, but BEFORE the first time you run this stored procedure.
- The posted balance is calculated on an as-needed basis within SunSystems; it is not stored in a database field. So when the SP requests the posted balance, SunSystems has to calculate it at the time of the request for all customers. If this request is made too often, SunSystems system performance could be affected.
- The posted balance in CloudSuite Industrial may not exactly match the posted balance in SunSystems at any given time because:
  - SunSystems posted balances are exported to CloudSuite Industrial only upon request by the SP.
  - CloudSuite Industrial transactions that affect the SunSystems posted balances are imported into SunSystems only periodically (and usually less often than the posted balance requests are made).
  - Whenever customer invoice transactions are posted in CloudSuite Industrial, the customer posted balance that is stored in CloudSuite Industrial is updated automatically.
- If there is a concern about the accuracy of the CloudSuite Industrial customer A/R posted balance, it may be necessary to either schedule the batch process to run on a more frequent basis and/or enable CloudSuite Industrial users to log into SunSystems to view the customer A/R posted balance and related credit information.
- When retrieving from the SunSystems ledger table the amounts to sum to populate the CloudSuite Industrial posted balance, rows where ALLOCATION = A (allocated), P (paid) or C (corrections) are excluded.
- The stored procedure checks the currency code on the SunSystems ledger rows that are to be included. Depending on how that currency code compares with the CloudSuite Industrial parms.curr_code and CloudSuite Industrial custaddr.curr_code, the stored procedure takes one of the following actions:
  - If the CloudSuite Industrial customer's currency code equals the CloudSuite Industrial base currency code, the -(AMOUNT) value from the SunSystems ledger row is included in the posted balance.

- If the SunSystems ledger row currency code equals the CloudSuite Industrial customer's currency code, then the -(OTHER_AMT) value from the SunSystems ledger row is included in the posted balance.
- If neither of these is true, then the -(AMOUNT) value from the SunSystems ledger row is converted (using CloudSuite Industrial currency conversion logic and CloudSuite Industrial currency exchange rates) to the CloudSuite Industrial customer's currency based on the TRANS_DATETIME value in the SunSystems ledger row.

- The SunSystems customer record's User Area must contain the CloudSuite Industrial customer number (without the "D" prefix) for every CloudSuite Industrial customer record added in SunSystems. For any records where the User Area is not exactly equal to the CloudSuite Industrial customer number, the credit limit and posted A/R balance will not be brought into CloudSuite Industrial. For numeric customer numbers, the non-expanded number is stored in the SunSystems User Area, and the script that brings data over to CloudSuite Industrial will expand it. Do not update this field once it is set.
- CloudSuite Industrial's corporate customer features are not supported by Infor CloudSuite Industrial Enterprise Financials.
- Record lengths:
  - **Credit Limit** - SunSystems' maximum value for storage is larger than CloudSuite Industrial's maximum value for storage, which in turn is larger than CloudSuite Industrial's maximum value for display on a form.

    During import, if the credit limit value passed from SunSystems is too large to store in the CloudSuite Industrial database, a SQL error is logged. During form display, if the credit limit value stored in the CloudSuite Industrial database is too large to display in the field, CloudSuite Industrial displays an error on the form.

    The maximum credit limit value that can be displayed on a CloudSuite Industrial form without causing a display error depends on the currency format, which is specified in the Currency Codes form.

    The maximum value allowed in the CloudSuite Industrial database without causing a SQL error is defined by the datatype for the column custaddr.credit_limit. The datatype is AmountType, which is defined as "decimal 23,8" where 23 is the precision and 8 is the scale.
  - **Posted Balance** - This field (customer.posted_bal in the database) has the same display and database maximum value limitations as the **Credit Limit** field.
- If there is an error during processing, all updates occur with the exception of the specific values that have errors. For example, a customer's posted balance is still updated even though an error occurs when attempting to update the customer's credit limit. All other valid customer credit limit updates will occur. For specifics of possible error messages, see the troubleshooting information in the *CloudSuite Industrial Enterprise Financials Installation Guide*.

Get SunSystems Currency Codes and Rates

How It Works

The SQL stored procedure GetSunSystemsCurrencyCodesandRatesSp performs the following activities:

- Selects currency code data from the SunSystems database.
- If a record exists in the CloudSuite Industrial currency table for the currency code, the record is updated with description and places data from SunSystems.
- If the record did not already exist in CloudSuite Industrial, it will be inserted with the above data.
- If the currency code data selected from SunSystems was for daily rates:
  - Selects daily currency rate data from the SunSystems database.
  - The effective date within the CloudSuite Industrial currate table corresponds to the SunSystems Currency Daily Rates "Effective Date From" value.

    When pulling rates from SunSystems, CloudSuite Industrial ignores any values specified for "Effective Date To" for SunSystems Currency Daily Rates.
  - If a record exists in the CloudSuite Industrial currate table for the From Currency code, Effective Date, and To Currency code, the record is updated with buy rate and sell rate data from SunSystems.
  - If the record did not already exist in CloudSuite Industrial, it will be inserted with the above data.
- If the currency code data selected from SunSystems was not for daily rates, then period rates are used to update Currency Exchange Rates within CloudSuite Industrial:
  - Selects period currency rate data from the SunSystems database.
  - Formats effective date for CloudSuite Industrial from the SunSystems period.

    When importing exchange rate information for period-based rates from SunSystems, the effective date on the CloudSuite Industrial record is that of the first date in the specified accounting period. If a period rate has a period greater than 13, then the period is set to 13.
  - If a record exists in the CloudSuite Industrial currate table for the From Currency code, Effective Date, and To Currency code, the record is updated with buy rate and sell rate data from SunSystems.

Implementation Notes

- How often you schedule this stored procedure to run depends on how often you expect the currency rates to be updated in SunSystems.
- Record lengths:
  - Currency Code length can be larger in SunSystems (up to 5 characters) than in CloudSuite Industrial (up to 3 characters). Records containing currency codes greater than 3 characters will not be imported into CloudSuite Industrial.

- SunSystems' currency code is also mapped into CloudSuite Industrial's **Currency Symbol** field (3 characters).
- Currency description length can be larger in SunSystems (50) than in CloudSuite Industrial (40). This field is automatically truncated.
- For exchange rates, CloudSuite Industrial supports a maximum of 5 digits to the left of the decimal and 7 digits to the right of the decimal.

- The stored procedure only looks at SunSystems records FROM a given currency TO the CloudSuite Industrial domestic currency. SunSystems stores currency rate records that do not involve the domestic (base) currency; these records will never be imported to CloudSuite Industrial.
- SunSystems allows two different rate types for currency codes based on time frames: daily rates and period-based rates. Period-based rates, which apply for one accounting period, imply both an effective date and an obsolete date. (CloudSuite Industrial rates have an effective date but not an obsolete date.) When using period-based rates, the period may be left blank to define a rate that applies to all accounting periods. When a rate with a blank period value is copied into CloudSuite Industrial, the rate's effective date is set to 01/01/2000.
- SunSystems maintains only one rate for a specific currency, while CloudSuite Industrial maintains two (Buy Rate and Sell Rate). This one SunSystems rate is copied into both fields in CloudSuite Industrial.

Mapping Currency "Rate Is Divisor" to "Multiply or Divide"

In CloudSuite Industrial, the "Rate Is Divisor" field value is set for the currency code (in one place for all rates for one currency). In SunSystems, the equivalent "Multiply or Divide" field value is set on each currency rate (in multiple places).

When you change currency information in SunSystems and then use the stored procedure to pull the changed currency information into CloudSuite Industrial, the stored procedure will look at both the CloudSuite Industrial "Rate Is Divisor" field and the SunSystems "Multiply or Divide" field. The following chart shows the logic the stored procedure uses to determine the proper rate in the CloudSuite Industrial currate table:

	CloudSuite Industrial "Rate is Divisor" field not selected	**CloudSuite Industrial "Rate is Divisor" field selected**
SunSystems field set to "Multiply"	1 / SunSystems Rate	SunSystems Rate
SunSystems field set to "Divide"	SunSystems Rate	1 / SunSystems Rate

Get SunSystems Chart of Accounts

How It Works

The SQL stored procedure GetSunSystemsChartofAccountsSp performs the following activities:

- Selects account data from the SunSystems database (ignoring SunSystems debtor, creditor and client accounts since these are required only in SunSystems's unified ledger and not on the CloudSuite Industrial side).
- Converts the SunSystems account type to the CloudSuite Industrial account type as follows:

SunSystems Account Type	CloudSuite Industrial Account Type	Notes
3 - P&L (income and expenditure accounts)	R - Revenue	No way to tell within SunSystems whether a P&L account is expense or revenue
4 - Balance Sheet (asset and liability accounts)	A - Asset	No way to tell within SunSystems whether a balance sheet is for an asset or liability account
5 - Memo (used for entering non-financial and ratio allocation figures that do not form part of the trial balance)	Y - Analytical (only used if **Analytical Ledger** field is selected in CloudSuite Industrial General Parameters form)	

Note: CloudSuite Industrial account types D (Allocation), E (Expense), L (Liability), O (Owner's Equity) and S (Statistical) are not handled by this conversion.

- Converts the SunSystems analysis code parameters to CloudSuite Industrial unit code accessibility options as follows:

SunSystems Analysis Code Parameters ENTER_ANL_1 through ENTER_ANL_4	CloudSuite Industrial Unit Code Options access_unit1 through access_unit4
1 - Mandatory	R - Required
2 - Optional	A - Accessible
3 - Prohibited	N - No Access

  - If a record exists in the CloudSuite Industrial chart table for the account (acct), the record is updated with the following data from SunSystems: description, type, access_unit1, access_unit2, access_unit3, access_unit4, and obs_date.

obs_date is updated only if it is NULL on the current record and the status from the SunSystems record is greater than 1.

- If the record did not already exist in CloudSuite Industrial, it will be inserted with the above data.

Implementation Notes

- Although this stored procedure allows the loading of analytical accounts from SunSystems to CloudSuite Industrial, there is no flow of analytical ledger data from CloudSuite Industrial to SunSystems. You may choose to use SunSystems for analytical ledger-related processing; however, the Infor CloudSuite Industrial Enterprise Financials does not support the integration of the CloudSuite Industrial Analytical Ledger with SunSystems.
- Status is populated in only one direction. In SunSystems, you can change the status of an account to either “Suspended/Held” or “Closed/Complete,” and the Obsolete Date in CloudSuite Industrial will be set to the current date. However, if the status in SunSystems is changed back to “Open” or “Hidden,” CloudSuite Industrial will not be updated.
- Record length:
  - The **Account Number** field length can be larger in SunSystems (up to 15 characters) than in CloudSuite Industrial (up to 12 characters). If the SunSystems value is greater than 12 characters, the record will not be brought into CloudSuite Industrial.
  - The **Account Description** field length can be larger in SunSystems (50) than in CloudSuite Industrial (40). This field is automatically truncated.
- How often you schedule this stored procedure to run depends on how often you expect the chart of accounts information to be updated in SunSystems.

Customizing the Stored Procedures

You can modify these stored procedures in the same way as any other CloudSuite Industrial stored procedures.

You can also add new stored procedures to the CloudSuite Industrial database in order to import additional SunSystems data.

Chapter 3: Using Action Profiles

3

SLSI includes a set of action profiles that are used to:

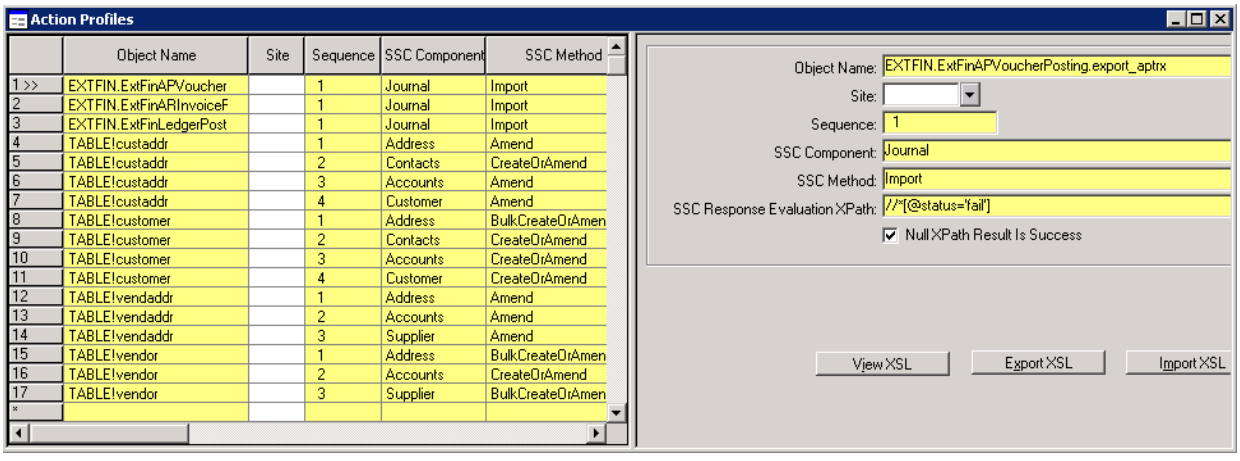
1 Match the incoming CloudSuite Industrial XML. Each profile includes a site/object combination. If the site/object matches the information in an incoming XML, then SLSI uses the profile to process the XML. If multiple action profiles match an incoming XML, then each of the profiles is run, in order by sequence number.

2 Use the XSL attached to the action profile's record to map the CloudSuite Industrial values to SunSystems values and to perform other transformations required to create an outbound SSML.

3 Specify what SunSystems Connect (SSC) component and method should be used when posting to SunSystems.

4 Specify how to interpret the results of the posting attempt.

Elements of the Action Profile Form



Each action profile contains this information:

- **Object Name** - This matches the IDO name on the CloudSuite Industrial XML document. The name appears after the Name attribute in the XML's UpdateCollection tag. For example, the object name might be: TABLE!vendor or EXTFIN.ExtFinARInvoicePosting.export_arinv.

- **Site** - Indicates the target site to which the profile should be applied. By default this is blank, indicating that the profile applies to all target sites, and therefore all business units. (Remember that target sites are mapped to business units through the Site to Business Unit Mappings form.)
- **Sequence** - If multiple profiles match an incoming XML, the XML is processed multiple times. This sequence number determines the order of processing.
- **SSC Component** - Specify the SunSystems Connect component to use when mapping the data to SunSystems. For example, the Address component exposes functions to create, modify, and query entries in the address tables. For more information about the available components, see the SSC Technical Reference online help.
- **SSC Method** - Specify the SunSystems Connect method to use when mapping the data to SunSystems. For a complete listing of available components and methods, see the SSC Technical Reference.
- **SSC Response Evaluation XPath** and **Null XPath Result Is Success** - The SunSystems SSC call associated with each step outputs an XML-formatted response. The **SSC Response Evaluation XPath** field contains an XPath query statement that is evaluated against the returned SSC response to determine whether the step succeeded or failed. How the system handles the result is determined by whether the **Null XPath Result is Success** field is selected:
  - If selected, the XPath query returns a set of nodes that relate to errors contained in the response. If the XPath query returns a count of zero nodes, the SSC call was successful. If the query returns one or more nodes, errors were raised during the SSC call, so the step failed. For example, on reference data, the query would look for the existence of any Status='Fail' attributes.
  - If cleared, and the XPath query returns a count of zero nodes, the SSC call failed.

    Reading the recommended XPath tutorial at http://www.w3schools.com may make this concept clearer.

The **View XSL** button on this form lets you look at the XSL that is stored in the database for the current action profile. It displays, properly formatted.

If you want to change the XSL, use the **Export XSL** button to save it to a file where you can edit it using an XML editor. When your changes are complete, use the **Import XSL** button to import the XSL back into the SLSI database.

What the Default Profiles Do

The following default action profiles have been defined.

CloudSuite Industrial Object (Sequence)	SunSystems Component	Description
A/P transactions	Journal	Imports any Accounts Payable transactions within the CloudSuite Industrial aptrx/aptrxd tables (vouchers or adjustments) that were posted to aptrxp (A/P Posted Transactions) into the SunSystems ledger. Payments are not included in the import.
A/R transactions	Journal	Imports any Accounts Receivable transactions within the CloudSuite Industrial arinv/arinvd tables (invoices, debit and credit memos) that were posted to artran (A/R Posted Transactions) into the SunSystems ledger. This includes project invoices and RMA credit memos. Payments, distributions that do not contain an account number, and finance charges are not included in the import.
Ledger transactions	Journal	Imports all journal records except AR DIST and AP DIST into the SunSystems ledger.
Customer (1)	Address	Imports customer number (as address code) and phone. Sets up the record for address information that is imported in a later step.
Customer (2)	Contacts	Imports contact name, customer number (as contact code), and phone.
Customer (3)	Accounts	Sets up account type and balance type. Imports customer number (as account code). Sets up record for later import of description.
Customer (4)	Customer	Sets up payment method and tax/sequence. Imports customer number (as customer code) and terms code. Sets up record for later import of description.
Custaddr (1)	Address	Imports customer number (as address code, short heading, and lookup code), name (as address line 1) and address, fax number, and URL.
Custaddr (2)	Contacts	Imports contact e-mail address and fax number.
Custaddr (3)	Accounts	Imports customer number (as account code, contact identifier, and user area), name (as lookup code, short heading, and description), and currency code.

CloudSuite Industrial Object (Sequence)	SunSystems Component	Description
Custaddr (4)	Customer	Imports customer number (as account, address code, customer code, and contact identifier), name (as lookup code, short heading, and name), and e-mail address. The customer name value is truncated to the first 50 characters imported from CloudSuite Industrial.
Vendor (1)	Address	Imports vendor number (as address code) and phone. Sets up the record for address information that is imported in a later step.
Vendor (2)	Accounts	Sets up account type and balance type. Imports vendor number (as account code) and currency code. Sets up record for later import of description.
Vendor (3)	Supplier	Sets up payment method and tax/sequence. Imports vendor number (as supplier code, and account code), currency code, terms code, and tax ID. Sets up record for later import of supplier name.
Vendaddr (1)	Address	Imports vendor number (as address code, short heading, and lookup code), name (as address line 1) and address, fax number, and URL.
Vendaddr (2)	Accounts	Imports vendor number (as account code) and name (as lookup code, short heading, and description).
Vendaddr (3)	Supplier	Imports vendor number (as account, address code, and supplier code), name (as lookup code, short heading, description, and name), and e-mail address. The vendor name value is truncated to the first 50 characters imported from CloudSuite Industrial.

Some of the inbound XMLs are processed multiple times, to import CloudSuite Industrial data into different SunSystems components (for example, the CloudSuite Industrial custaddr update XML is processed against the SunSystems Address, Customer, Accounts, and Contacts components).

Customizing Action Profiles

You can customize action profiles by:

- **Specifying a target site for the profile**. This restricts the action profile for use with XMLs that are destined for a specified site/business unit.
- **Changing the underlying XSL for a default profile**. By doing this, you can change the mapping done for a specific incoming XML, to map more or different fields or values. However, you are limited to using:
  - Name/Values included in the inbound IDORequest XML (many of the Name/Value pairs in the inbound XML are not currently mapped to anything in SunSystems),
  - Parameters added through SLSI Process Defaults, or
  - Constant values that you add in the XSL.

  See examples of changing the XSLs in Chapter 6, “Customizing XSLs."
- **Adding a new action profile**. For example, you might want to map some CloudSuite Industrial vendor bank information to the SunSystems Bank Details record for the supplier. The IDORequest XML from CloudSuite Industrial includes some bank information that is not used in the default action profiles. Bank information is stored in a SunSystems component (Bank Details) that is not currently part of an action profile. This would require a new action profile.

  When adding a profile, think about prerequisite data. In our example, what data needs to exist before the Bank Details record can be created? (There must be an existing Supplier record in order to map the Supplier code in the Bank Details record.) This helps you determine the step/sequence of the new action profile in relationship to existing action profiles.
- **Adding new data exports from CloudSuite Industrial to SunSystems**. This would require additional replication rules set up on the CloudSuite Industrial side, as well as new action profiles and XSLs.

Note: Before making changes to the action profiles, we recommend that you export all of the default Action Profile records from the form into a spreadsheet. Save a copy of the spreadsheet as a backup.

Remember that any change you make on the Action Profiles form, including changes to the XSLs, require that you restart IIS in order to clear the cache and load the changes into the SLSI ASP.

Chapter 4: Inbound and Outbound XML Schema

4

This chapter includes examples of CloudSuite Industrial IDORequest XML documents that are sent to SLSI through replication, and examples of SunSystems SSML documents sent from SLSI to SunSystems Connect (SSC).

We define the general schema used for each type of document and annotate the examples so you can get an idea of what each of the documents may include.

Inbound XML: CloudSuite Industrial IDORequest Schema

This section contains two examples of IDORequest XML documents sent from CloudSuite Industrial to SLSI. Inbound XML documents are stored in SLSI and can be accessed from the Batch Posting Attempts form through the **View (Incoming XML)** button.

IDORequest Structure

The following structure is representative of the majority of IDORequests used in SLSI.

XML Tag	Description
`<IDORequest>`	An IDO request must start with this tag.
`<RequestHeader Type="RequestType">`	Describes the type of IDO request. Currently the only type used by SLSI is UpdateCollection.
`<SourceName>`	Name of the site where the request originated.
`<TargetName>`	Name of the site the request is targeting.
`<RequestData>`	Contains the data for the request.

XML Tag	Description
`<UpdateCollection Name="Name">`	Contains the name of the IDO collection. This can be an IDO name or a table name in the format TABLE!*tablename*.
`<Items>`	Collection of multiple <Item> tags.
`<Item Action="Action">`	Each item in a collection has an assigned action: Insert, Update, or Delete. Delete is not currently used with SLSI.
`<Property Name="Name">`	Includes any properties to be updated, as well as all nonnullable properties that do not have a default value. Properties that have a default value, and that will not change as a result of the update, can be omitted.

For More Information

For detailed information about the syntax and tags used in CloudSuite Industrial IDORequest XML documents, see the document *Integrating IDOs with External Applications*, especially the chapter on XML IDO request and response documents.

Customer Update - Inbound XML Example

This XML document was created when an CloudSuite Industrial user created a customer, with the very minimum amount of field entries.

```
<IDORequest>
  <RequestHeader Type="UpdateCollection">
    <SourceName>MI</SourceName>
    <TargetName>SUNBU1</TargetName>
    <RequestData>
      <UpdateCollection Name="TABLE!customer">
        <Items>
          <Item Action="Update">
            <Property Name="cust_num">ZC00019</Property>
            <Property Name="cust_seq">0</Property>
          </Item>
        </Items>
      </UpdateCollection>
    </RequestData>
  </RequestHeader>
  <Parameters>
    <Parameter Name="SscUser">AMP</Parameter>
    <Parameter Name="SscBusinessUnit">BU1</Parameter>
    <Parameter Name="SscCompatibilityMode">0</Parameter>
    <Parameter Name="SscErrorOutput">1</Parameter>
    <Parameter Name="SscErrorThreshold">1</Parameter>
    <Parameter Name="SscBudgetCode">A</Parameter>
    <Parameter Name="SscAllowBalTran">N</Parameter>
    <Parameter Name="SscAllowOverBudget">Y</Parameter>
    <Parameter Name="SscAllowPostToSuspended">N</Parameter>
    <Parameter Name="SscBalancingOptions" />
    <Parameter Name="SscLayoutCode" />
    <Parameter Name="SscLoadOnly">N</Parameter>
    <Parameter Name="SscPostProvisional">N</Parameter>
    <Parameter Name="SscPostToHold">N</Parameter>
    <Parameter Name="SscPostingType">2</Parameter>
    <Parameter Name="SscPrint">N</Parameter>
    <Parameter Name="SscReportErrorsOnly">Y</Parameter>
    <Parameter Name="SscReportingAccount" />
    <Parameter Name="SscSuppressSubstitutedMessages">Y</Parameter>
    <Parameter Name="SscSuspenseAccount" />
    <Parameter Name="SscTransactionAmountAccount" />
  </Parameters>
</IDORequest>
```

SourceName = MI source site where the customer was created.

TargetName = SUNBU1 target site (mapped to the business unit in SLSI Site to Business Unit Mappings form).

Collection = TABLE!customer = the CloudSuite Industrial customer table.

Action = Update of

Properties = the columns in the customer table (via SLSICustomerView) that were updated. These will be mapped to SunSystems database columns by the XSL.

Normally there would be other properties, such as phone, billing terms, and so on. These are the Name/Value pairs you see in the Batch Details form.

Parameters = added to the XML by SLSI, using data from Site to Business Unit Mappings and Process Defaults (see Chapter 5, “Setting Parameters in the XML”).

A/R Invoice Posting - Inbound XML Example

This XML document was created when a CloudSuite Industrial user posted an A/R invoice. (Remember, this is not the original XML sent from CloudSuite Industrial. This XML has already been

transformed once before being stored in SLSI, in order to "flatten" the structure to remove parent-child hierarchies.)

```
<IDORequest>
  <RequestHeader Type="UpdateCollection">
    <SourceName>MI</SourceName>
    <TargetName>SUNBU1</TargetName>
    <RequestData>
      <UpdateCollection
Name="EXTFIN.ExtFinARInvoicePosting.export_arinv">
        <Items>
          <Item Action="Insert">
            <Property Name="HeaderDetailFlag">H</Property>
            <Property Name="ar_batch_id">4</Property>
            <Property Name="batch_seq">1</Property>
            <Property Name="cust_num">42</Property>
            <Property Name="inv_num">56</Property>
            <Property Name="inv_seq">0</Property>
            <Property Name="type">I</Property>
            <Property Name="co_num" />
            <Property Name="inv_date">2006-12-15</Property>
            <Property Name="due_date">2007-01-14</Property>
            <Property Name="acct">11000</Property>
            <Property Name="amount">10.00000000</Property>
            <Property Name="misc_charges">.00000000</Property>
            <Property Name="sales_tax">.00000000</Property>
            <Property Name="freight">.00000000</Property>
            <Property Name="ref">ARI 56</Property>
            <Property Name="terms_code">N30</Property>
            <Property Name="description">Invoice 56</Property>
            <Property Name="post_from_co">0</Property>
            <Property Name="exch_rate">1.0000000</Property>
            <Property Name="sales_tax_2">.00000000</Property>
            <Property Name="use_exch_rate">0</Property>
            <Property Name="tax_code1">NT</Property>
            <Property Name="tax_code2" />
            <Property Name="acct_unit1" />
            <Property Name="acct_unit2" />
            <Property Name="acct_unit3">1000</Property>
            <Property Name="acct_unit4" />
            <Property Name="fixed_rate">0</Property>
            <Property Name="rma">0</Property>
            <Property Name="pay_type">C</Property>
            <Property Name="draft_print_flag">0</Property>
            <Property Name="do_num" />
            <Property Name="processed">0</Property>
            <Property Name="curr_code">USD</Property>
```

SourceName = MI source site where the customer was created.

TargetName = SUNBU1 target site (mapped to the business unit in SLSI Site to Business Unit Mappings form).

Collection = the CloudSuite

Item = ledger line.
Action = insert.
This first item is the header, as indicated by HeaderDetailFlag=H.
All **properties** in this item relate to the header. These will be mapped to SunSystems database columns by the XSL. These are the Name/Value pairs you see in the Batch Details form.

```
        <Property Name="fiscal_year">2006</Property>
      </Item>
      <Item Action="Insert">
        <Property Name="HeaderDetailFlag">D</Property>
        <Property Name="ar_batch_id">4</Property>
        <Property Name="batch_seq">1</Property>
        <Property Name="cust_num">42</Property>
        <Property Name="inv_num">56</Property>
        <Property Name="inv_seq">0</Property>
        <Property Name="dist_seq">5</Property>
        <Property Name="acct">40000</Property>
        <Property Name="amount">10.00000000</Property>
        <Property Name="tax_code" />
        <Property Name="tax_basis">.00000000</Property>
        <Property Name="tax_system" />
        <Property Name="tax_code_e" />
        <Property Name="ref_type" />
        <Property Name="ref_num" />
        <Property Name="ref_line_suf">0</Property>
        <Property Name="ref_release">0</Property>
        <Property Name="acct_unit1" />
        <Property Name="acct_unit2" />
        <Property Name="acct_unit3">1000</Property>
        <Property Name="acct_unit4" />
      </Item>
    </Items>
  </UpdateCollection>
 </RequestData>
</RequestHeader>
<Parameters>
  <Parameter Name="SscUser">AMP</Parameter>
  <Parameter Name="SscBusinessUnit">SUN</Parameter>
  <Parameter Name="SscCompatibilityMode">0</Parameter>
  <Parameter Name="SscErrorOutput">1</Parameter>
  <Parameter Name="SscErrorThreshold">1</Parameter>
  <Parameter Name="SscBudgetCode">A</Parameter>
  <Parameter Name="SscAllowBalTran">N</Parameter>
  <Parameter Name="SscAllowOverBudget">Y</Parameter>
  <Parameter Name="SscAllowPostToSuspended">N</Parameter>
  <Parameter Name="SscBalancingOptions" />
  <Parameter Name="SscLayoutCode" />
  <Parameter Name="SscLoadOnly">N</Parameter>
  <Parameter Name="SscPostProvisional">N</Parameter>
  <Parameter Name="SscPostToHold">N</Parameter>
  <Parameter Name="SscPostingType">2</Parameter>
  <Parameter Name="SscPrint">N</Parameter>
  <Parameter Name="SscReportErrorsOnly">Y</Parameter>
  <Parameter Name="SscReportingAccount" />
  <Parameter Name="SscSuppressSubstitutedMessages">Y</Parameter>
```

Item = ledger line.
Action = insert.
This is a detail line, as indicated by HeaderDetailFlag=D. All properties in this item relate to the detail line. There could be multiple detail lines for the header. All **properties** in this item relate to the detail.

Parameters = added to the XML by SLSI, using data from Site to Business Unit Mappings and Process Defaults (see Chapter 5, “Setting Parameters in the XML”).

Outbound XML: SunSystems Connect SSML Schema

This section contains two examples of SunSystems Connect payload documents sent from SLSI to SunSystems. Outbound XML (SSML) documents are stored in SLSI and can be accessed from the Batch Posting Attempts form through the **View (Outgoing XML)** button.

Payload Structure

The following table is representative of the majority of input payloads.

XML Tag	Description
`<?xml version="1.0" encoding="UTF-8"?>`	An input payload must start with an XML declaration.
`<SSC>`	The XML declaration must be followed by an <SSC> element.
`<ErrorContext>`	Identifies what type of error handling is active for this payload file and overrides any system-wide defaults.
`<CompatibilityMode>.</ CompatibilityMode>`	Overrides the SSC installation default for the error message format.
`<ErrorOutput>.</ErrorOutput>`	Overrides the SSC installation default for the amount of message detail that SSC generates whenever a message is sent.
`<ErrorThreshold>.</ ErrorThreshold>`	Overrides the SSC installation default for the types of messages that appear in the output payload.
`<User>`	Identifies the SunSystems user name and must be provided for authentication. It specifies the User Id under which the payload runs.
`<Name>.</Name>`	The user ID specified here must exist in SunSystems as an Operator (unless you are using another method of authenticating users in which case it must exist in SSC User Manager) before running the payload.
`<SunSystemsContext>`	This element provides top level information for validation and applies to all transactions contained in the <Payload> element. Depending on the component, this element is where you specify the Business Unit, Budget or Ledger code for the payload.
`<BusinessUnit>.</BusinessUnit>`	

XML Tag	Description
<Payload>	A high level element used to flag the start of the payload records.
<*Element*>	A high level element used to flag the start of a single payload record. (Only a single record element is shown here).
<*Element*>.</*Element*>	Lower-level elements (records) that can have values assigned.

For More Information

For detailed information about the syntax and tags used in SunSystems SSC payload documents, see the “Payload Reference Guide” section of the SSC Technical Reference online help.

Customer Amend - Outbound XML Example

This XML document is created by SLSI after it receives a Customer Insert XML from CloudSuite Industrial and performs the XSL transformation prescribed by the appropriate action profile (CloudSuite Industrial Customer table to SunSystems Customer Amend).

```
<?xml version="1.0" encoding="utf-8" ?>
<SSC>
  <ErrorContext>
    <CompatibilityMode>0</CompatibilityMode>
    <ErrorOutput>1</ErrorOutput>
    <ErrorThreshold>1</ErrorThreshold>
  </ErrorContext>
  <User>
    <Name>AMP</Name>
  </User>
  <SunSystemsContext>
    <BusinessUnit>BU1</BusinessUnit>
  </SunSystemsContext>
  <Payload>
    <Customer>
      <PaymentMethod>1</PaymentMethod>
      <CustomerCode>DZC00019</CustomerCode>
      <Tax>
        <CustomerCode>DZC00019</CustomerCode>
        <SequenceNumber>1</SequenceNumber>
      </Tax>
    </Customer>
  </Payload>
</SSC>
```

ErrorContext = set from the parameter section of the inbound XML (via the Process Defaults). See Chapter 5, “Setting Parameters in the XML".

User = SunSystems user name, set on the Site to Business Unit Mappings form.

Business Unit = mapped based on data in Site to Business Unit Mappings form.

Payload = the collection of data to update. This payload updates the Customer component. The tags added in this section depend on: 1) the initial data contained in the CloudSuite Industrial XML, and 2) the mapping done by the XSL on the initial data.

Journal Insert - Outbound XML Example

This XML (SSML) document was created by SLSI after it received an A/R Invoice Posting XML from CloudSuite Industrial and performed the XSL transformation prescribed by the appropriate action profile (CloudSuite Industrial EXTFIN.ExtFinARInvoicePosting.export_arinv to SunSystems Journal component).

```
<?xml version="1.0" encoding="utf-8" ?>
<SSC>
  <ErrorContext>
    <CompatibilityMode>0</CompatibilityMode>
    <ErrorOutput>1</ErrorOutput>
    <ErrorThreshold>1</ErrorThreshold>
  </ErrorContext>
  <User>
    <Name>AMP</Name>
  </User>
  <SunSystemsContext>
    <BusinessUnit>BU1</BusinessUnit>
    <BudgetCode>A</BudgetCode>
  </SunSystemsContext>
  <MethodContext>
    <LedgerPostingParameters>
      <AllowBalTran>N</AllowBalTran>
      <AllowOverBudget>Y</AllowOverBudget>
      <AllowPostToSuspended>N</AllowPostToSuspended>
      <BalancingOptions />
      <DefaultPeriod>0122006</DefaultPeriod>
      <Description>Syteline AR Invoice Posting</Description>
      <JournalType>AR DI</JournalType>
      <LayoutCode />
      <LoadOnly>N</LoadOnly>
      <PostProvisional>N</PostProvisional>
      <PostToHold>N</PostToHold>
      <PostingType>2</PostingType>
      <Print>N</Print>
      <ReportErrorsOnly>Y</ReportErrorsOnly>
      <ReportingAccount />
      <SuppressSubstitutedMessages>Y</SuppressSubstitutedMessages>
      <SuspenseAccount />
      <TransactionAmountAccount />
    </LedgerPostingParameters>
  </MethodContext>
  <Payload>
    <Ledger>
      <Line>
        <JournalType>AR DI</JournalType>
        <AccountCode>D42</AccountCode>
        <TransactionAmount>10</TransactionAmount>
        <AnalysisCode5>NOT USED</AnalysisCode5>
        <DebitCredit>D</DebitCredit>
        <ConversionRate>1.0000000</ConversionRate>
        <AccountingPeriod>0122006</AccountingPeriod>
        <JournalSource>I</JournalSource>
```

ErrorContext = set from the parameter section of the inbound XML (via the Process Defaults). See Chapter 5, “Setting Parameters in the XML".

User = SunSystems user name, set in the Site to Business Unit Mappings form.

Business Unit = mapped based on data in Site to Business Unit Mappings form.

Ledger Posting Parameters = set from the parameter section of the inbound XML (via the Process Defaults). See Chapter 5, “Setting Parameters in the XML".

Payload = the collection of data to update. This payload updates the Ledger component. The tags added in this section depend on: 1) the initial data contained in the CloudSuite Industrial XML, and 2) the mapping done by the XSL on the initial data

Line = Header data.

Chapter 5: Setting Parameters in the XML

5

Before the XSL transformations are run, SLSI adds a Parameters section to the Infor CloudSuite Industrial Enterprise Financials IDO request XML, as shown in the example below.

```
<Parameters>
  <Parameter Name="SscUser">AMP</Parameter>
  <Parameter Name="SscBusinessUnit">BU1</Parameter>
  <Parameter Name="SscCompatibilityMode">0</Parameter>
  <Parameter Name="SscErrorOutput">1</Parameter>
  <Parameter Name="SscErrorThreshold">1</Parameter>
  <Parameter Name="SscBudgetCode">A</Parameter>
  <Parameter Name="SscAllowBalTran">N</Parameter>
  <Parameter Name="SscAllowOverBudget">Y</Parameter>
  <Parameter Name="SscAllowPostToSuspended">N</Parameter>
  <Parameter Name="SscBalancingOptions" />
  <Parameter Name="SscLayoutCode" />
  <Parameter Name="SscLoadOnly">N</Parameter>
  <Parameter Name="SscPostProvisional">N</Parameter>
  <Parameter Name="SscPostToHold">N</Parameter>
  <Parameter Name="SscPostingType">2</Parameter>
  <Parameter Name="SscPrint">N</Parameter>
  <Parameter Name="SscReportErrorsOnly">Y</Parameter>
  <Parameter Name="SscReportingAccount" />
  <Parameter Name="SscSuppressSubstitutedMessages">Y</Parameter>
  <Parameter Name="SscSuspenseAccount" />
  <Parameter Name="SscTransactionAmountAccount" />
</Parameters>
```

You will not see these parameters in the original XML that was sent from CloudSuite Industrial, but they are added by the Integration Service before the inbound XML is stored in the SLSI database. The parameters are taken from these forms in SLSI:

- Site to Business Unit Mappings
- Process Defaults

Changing the Parameter Values

To change these parameter values:

1 Change the values on the SLSI forms, as described below.

2 Since these values are cached in the ASP, you must also stop and restart IIS in order to clear the cache. On the utility server where SLSI is installed:

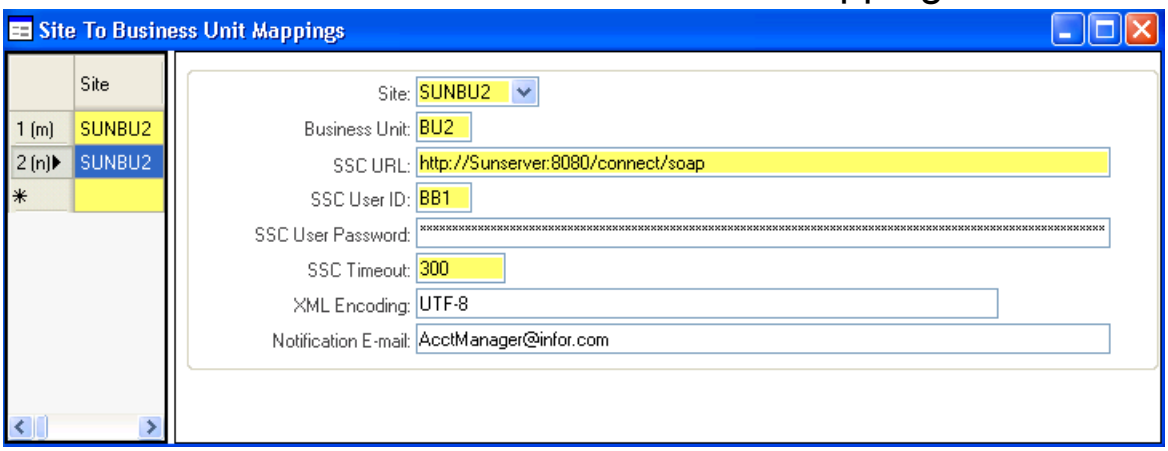
   a From Windows Administrative Tools, select **Internet Information Services**.

   b Right-click the server name and select **All Tables**.

   c Select **Restart IIS** and click **OK**.

   Be aware that restarting IIS will also affect any other applications running through IIS on this server.

Parameters on the Site To Business Unit Mappings Form



When an inbound IDORequest XML arrives, SLSI gets the target site name from the XML and searches the Site to Business Unit Mappings to find a record with a matching Site name. It then takes these values from that record and adds them to the Parameters section of the XML that gets stored in SLSI:

- **Business Unit** - becomes the SscBusinessUnit parameter in the IDORequest XML document.
- **SSC User ID and Password** - are used in the authentication process that is performed before sending the XML.

Parameters on the Process Defaults Form

Process Defaults

	Default Name	Default Value
1	SMTP Server	Server=AZTEC.qa.local;User=qat;Password=qat;Domain=QA
2	Report Output Obfuscation	NULL
3	Objects metadata linkage	Server=Vellagulf;Database=SLSI_Objects
4	Message Bus Outbound Queue	NULL
5	IDO Checkout Disabled	NULL
6	Keep Successful Event States	NULL
7	Number of Deadlock Retry Attempts	0
8	Collection Read Mode	UNCOMMITTED
9	Report Paper Size	DEFAULT
10	TaskMan Options	NULL
11	Enable Audit Logging	0
12	WinStudio Max Record Cap	NULL
13	SSC CompatibilityMode	0
14	SSC ErrorOutput	1
15	SSC ErrorThreshold	1
16	SSC BudgetCode	A
17	SSC AllowBalTran	N
18	SSC AllowOverBudget	Y
19	SSC AllowPostToSuspended	N
20	SSC BalancingOptions	NULL
21	SSC LayoutCode	NULL
22	SSC LoadOnly	N
23	SSC PostProvisional	N
24	SSC PostToHold	N
25	SSC PostingType	2
26	SSC Print	N
27	SSC ReportErrorsOnly	Y
28	SSC ReportingAccount	NULL
29	SSC SuppressSubstitutedMessages	Y
30	SSC SuspenseAccount	NULL
31	SSC TransactionAmountAccount	NULL
32	SLSI Integration Service URL	http://RIVIERA/SLSIIntegrationSvc/Resubmit.aspx

SLSI takes the following information as currently set in the Process Defaults and adds it to the Parameters section of the IDORequest XML document. These parameters must be available so that the provided XSL can transform them into the context tags that are set in the outbound XML sent to SSC.

For more information about any of these, refer to the SSC Technical Reference online help. Search on the part of the name after "SSC" - for example, search for CompatibilityMode.

Error Context Parameters

These tags dictate the level of error reporting that SSC will employ.

- **SSC CompatibilityMode** - Provides backward compatibility on the SSC error tags.
- **SSC ErrorOutput** - Determines the level of detail in the error messages that are sent back to SLSI from SSC.
- **SSC ErrorThreshold** - Determines what constitutes an error.

Budget Code Parameter

The SSC BudgetCode determines the ledger code to use for ledger imports.

Ledger Posting Parameters

These defaults apply only to journal imports and are used to drive the posting options:

- **SSC AllowBalTran** - Determines whether the posting rules allow for a balancing entry to be generated.
- **SSC AllowOverBudget** - Determines whether over-budget entries are allowed.
- **SSC AllowPostToSuspended** - Determines whether suspended accounts are allowed to receive values.
- **SSC BalancingOptions** - Determines whether posting has to balance by a given analysis dimension.
- **SSC LayoutCode** - Used only if posting auto-generates a printed output.
- **SSC LoadOnly** - Specifies whether to load transactions in the posting area but not process the posting.
- **SSC PostProvisional** - Determines whether to post the entries in provisional mode to the ledger.
- **SSC PostToHold** - Determines whether to post the entries as a held journal.
- **SSC PostingType** - Specifies the type of posting: validate only (value = 0), force posting (value = 1), or post if no errors (value = 2).
- **SSC Print** - Determines whether the posting is to generate a printed output.
- **SSC ReportErrorsOnly** - Determines whether only errors are included in the output.
- **SSC ReportingAccount** - Specifies the SunSystems account that receives reporting currency out of balance.
- **SSC SuppressSubstitutedMessages** - Determines whether to suppress warning messages to the effect of substitutions.
- **SSC SuspenseAccount** - Specifies the SunSystems account to receive erroneous transactions.
- **SSC TransactionAmountAccount** - Specifies the SunSystems account to receive transaction currency out of balance.

Chapter 6: Customizing XSLs

6

About This Chapter

This chapter contains:

- A list of things you should understand before attempting to customize XSLs.
- Tips about how our default XSLs are set up: how they use variables, cases and when statements, and functions to accomplish the desired mapping.
- An explanation of why some XMLs exported from CloudSuite Industrial require “flattening” before they reach SLSI, which XSLs are used to do this transformation, and why their filenames might require changes.
- An extended example showing how changes to the default XSL can affect the outbound XML. Use this example to understand the process and the areas to look for when changing or adding data transformations.

What You Need to Know

If you plan to customize the XSLs that transform the data, you should have a good understanding of the following areas:

- Become familiar with both the CloudSuite Industrial database structure and SunSystems database structure.
- Understand the CloudSuite Industrial IDO Request XML document structure, so you know what data is being sent, and in which tag. See “Inbound XML: CloudSuite Industrial IDORequest Schema” on page 27 for a brief description and examples. For additional information, see the document *Integrating IDOs with External Applications* on the Infor support Web site.
- Understand how XSL transformations are used. Some very good tutorials and reference material about XSLs (also sometimes referred to as XSLTs) are available on the World Wide Web. We currently recommend this site:

  http://www.w3schools.com/xsl/xsl_intro.asp
- Understand the SunSystems SSML document (payload) structure, so you know the structure in which SSC expects to receive data. See “Outbound XML: SunSystems Connect SSML Schema” on page 32 for a brief description and examples. For additional information, see the SSC Technical Reference online help.

- A general knowledge and understanding of SSC is required in order to make the appropriate selection of the SSC component method to use. For more information, see the SSC Technical Reference online help.
- The specific data mapping done by the XSLs attached to each default action profile is described in Chapter 7, “Mapping Details.” Use this mapping information - particularly the Notes - while looking at the XSLs to better understand what the XSLs are doing.

XSL Transformations: Tips and Tricks

Caution: Before changing any of the default XSLs, be sure to export them and save a copy as a backup.

What Data You Can Map

You can change the mapping done for a specific incoming XML, to map more or different fields or values. These are the types of data you can map to SunSystems:

- Name/Values included in the inbound IDORequest XML from CloudSuite Industrial. Many of the name/value pairs in the default inbound XML documents are not currently mapped to anything in SunSystems.

  You could also set up mapping from other CloudSuite Industrial IDOs to other existing SunSystems components, beyond the default ones we have used. To do this, you must set up replication rules in CloudSuite Industrial to send the IDORequest XML document to the target site/business unit. Then you must create SLSI action profiles and XSLs to map the values into SunSystems.
- Parameters changed in the SLSI Process Defaults form (see Chapter 5, “Setting Parameters in the XML”).
- Constant values that you add in the XSL.

Importing and Exporting XSLs for Editing

When editing the default XSLs (or creating new ones):

1 Use the SLSI Action Profiles **Export XSL** button to export existing XSL from the database to a file.

2 Open the XSL in a standard XML editor - for example, XMLSpy. The editor should provide syntax help and formatting that will make your job easier.

3 Use the SLSI Action Profiles **Import XSL** button to import the modified XSL back into the database from the file.

4 Make any needed changes to the Action Profile that contains the XSL.

5 Restart IIS in order to clear the SLSI ASP's cache and load the changes.

Using Variables

The XSLs use variables to reference repetitive source values or to handle multiple properties from the same or related records. Variables are defined and assigned using the following syntax:

```
<xsl:variable name="amount">
  <xsl:choose>
    <xsl:when test="../Property[@Name='amount']=''">0</xsl:when>
      <xsl:otherwise><xsl:value-of select="../Property[@Name='amount']"></
xsl:value-of></xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

The above example sets a variable named *amount* as follows:

- If the property named "amount" is unassigned in the source XML, the variable is set to 0.
- If the property named "amount" is defined in the source XML, the variable is set to its value.

The variable can then be referenced by its name prefixed with a dollar sign as in:

```
<xsl:element name="TransactionAmount"><xsl:value-of select="$amount"></xsl:value-
of></xsl:element>
```

Params Variable

A variable can also be used to reference an entire node from the source document. The XSLs do this when referencing the IDORequest/Parameters node. The variable is defined as follows at the top of the script:

```
<xsl:variable name="Params" select="IDORequest/Parameters"></xsl:variable>
```

This variable node is then referenced later in the script to set various SSC posting options. For example:

```
<xsl:element name="BusinessUnit"><xsl:value-of select="$Params/
Parameter[@Name='SscBusinessUnit']"></xsl:value-of></xsl:element>
```

FirstHeader Variable

Similarly, the first header row node is defined as a variable from the source document for both A/R and A/P. For example;

```
<xsl:variable name="FirstHeader" select="IDORequest/RequestHeader/RequestData/
UpdateCollection/Items/Item[Property[@Name='HeaderDetailFlag']='H']" />
```

This is then referenced in setting some of the posting options, such as the default period.

Calculations Using Variables

Variables are also used to perform some arithmetic calculations across multiple properties in the A/R XSL when setting the transaction amount by summing multiple values (amount, misc_charges, sales_tax, etc.).

Setting the Date Format

Near the end of each default XSL is a function (template) that sets the date in the format expected by SunSystems:

```
<xsl:template name="formatDate">
  <xsl:param name="dateIn"></xsl:param>
  <xsl:variable name="Day" select="substring($dateIn,9,2)"></xsl:variable>
  <xsl:variable name="Month" select="substring($dateIn,6,2)"></xsl:variable>
  <xsl:variable name="Year" select="substring($dateIn,1,4)"></xsl:variable>
  <xsl:value-of select="concat($Day,$Month,$Year)"></xsl:value-of>
</xsl:template>
```

Whenever a CloudSuite Industrial date is mapped to a SunSystems date in the XSL, this function is called. For example:

```
<xsl:when test="@Name='trans_date'">
  <xsl:element name="TransactionDate">
    <xsl:call-template name="formatDate">
      <xsl:with-param name="dateIn" select="."></xsl:with-param>
    </xsl:call-template>
  </xsl:element>
</xsl:when>
```

Linking Headers and Their Related Detail Records

The default XSLs used on the A/R and A/P transaction XML documents must deal with certain values that are set in the header but that are needed in the detail records as well. These values must be pulled from the header when creating the detail ledger entries. (Note that ledger transactions are all detail records, so this requirement does not apply.)

The XSLs handle this by using a reference that is common to both header and detail CloudSuite Industrial records. For A/R transactions, this common reference is inv_num. For A/P transactions, it is voucher.

For each source record a variable named *headerLink* is created using the common reference. For example, in the XSL that transforms A/P transactions, this variable is defined:

```
<xsl:variable name="headerLink" select="Property[@Name='voucher']"></xsl:variable>
```

After the headerLink variable is defined, we can use it to create other variables representing header values by navigating to the ancestor node and finding the header row using the common reference and the line type of H (for header).

This is shown in the following A/P example for the header purchase order number:

```
<xsl:variable name="headerPONum" select="ancestor::Items/
Item[Property[@Name='voucher']=$headerLink and
Property[@Name='HeaderDetailFlag']='H']/Property[@Name='po_num']">
</xsl:variable>
```

The headerPONum variable can then be referenced wherever the header PO number is required further in the script, as shown in this example:

```
<xsl:choose>
      <xsl:when test="@Name='voucher'">
            <xsl:element name="PermanentPostingReference"><xsl:value-of
select="$headerPONum"></xsl:value-of></xsl:element>
      </xsl:when>
</xsl:choose>
```

This code would be positioned under the test for the voucher field, since the voucher field exists for both the header and the detail. To see these code snippets in context, view the XSL for the ExtFinARInvoicePosting action profile.

Using “For-Each” and “Choose”

The For-Each construct is used to cycle through source elements. Use this to process each <Property> tag of the name/value pair from the source XML’s line/Item element. The source XML is made up of <Item> nodes containing a series of <Property> child nodes with a Name attribute identifying the field name. The For-Each loop cycles through each <Property> node in a given <Item> node. The Choose function is then used to evaluate the Name attribute and act upon that field. For example:

```
<xsl:template name="Address" match="Item">
   <xsl:element name="Address">
      <xsl:for-each select="Property">
         <xsl:choose>
            <xsl:when test="@Name='vend_num'">
              <xsl:element name="AddressCode">
                 <xsl:value-of select="concat('C', translate(normalize-space(.),
$smallcase, $uppercase))"></xsl:value-of>
              </xsl:element>
            </xsl:when>
...
```

Setting Upper and Lowercase

Near the beginning of each XSL are these lines:

```
<xsl:variable name="smallcase" select="'abcdefghijklmnopqrstuvwxyz'" />
<xsl:variable name="uppercase" select="'ABCDEFGHIJKLMNOPQRSTUVWXYZ'" />
```

The XSL then uses these variables when translating mapped data in cases where CloudSuite Industrial requires uppercase and SunSystems requires lowercase, or vice versa.

For example:

```
<xsl:when test="@Name='acct'">
  <xsl:element name="AccountCode"><xsl:value-of select="translate(normalize-
space(.), $smallcase, $uppercase)"></xsl:value-of></xsl:element>
  </xsl:when>
```

In the above example, the CloudSuite Industrial acct value is mapped to the SunSystems AccountCode. Any lowercase letters found in the value are translated to uppercase letters before the value is loaded into the AccountCode.

Trimming Spaces

To trim spaces from inbound values, use the `normalize-space(.)` function, as shown in the upper- and lowercase example, above.

Concatenating Strings

The function `concat()` concatenates string values. The following example adds a prefix to the address code:

```
<xsl:element name="AddressCode"><xsl:value-of select="concat('C',
translate(normalize-space(.), $smallcase, $uppercase))"></xsl:value-of></
xsl:element>
```

Using Substrings

The function `substring()` captures parts of a string value. The following example truncates the AddressLine1 name at 50 characters:

```
<xsl:element name="AddressLine1"><xsl:value-of select="substring(.,1,50)"></
xsl:value-of></xsl:element>
```

Source Document Positioning

Values are pulled from the source XML document using XPath notations:

- **"." (period)** defines the current node. In the following example, the "." stated in the select clause returns the value of the current node (fax_num):

  ```
  <xsl:when test="@Name='fax_num'">
        <xsl:element name="FaxNumber"><xsl:value-of select="."></xsl:value-of></
  xsl:element>
  </xsl:when>
  ```

- **"../"** defines the parent node. In the following example, the "../" stated in the select clause retrieves the value of the specific Property Name "inv_amt" when the current node is actually the

HeaderDetailFlag by navigating to the parent node (Item) and then finding the property name "inv_amt". This is especially useful when evaluating a given Property node and retrieving the value of another.

```
<xsl:when test="@Name='HeaderDetailFlag'">
  <xsl:choose>
    <xsl:when test="normalize-space(.)='H'">
      <xsl:element name="TransactionAmount"><xsl:value-of select="../
Property[@Name='inv_amt']"></xsl:value-of></xsl:element>
    </xsl:when>
  </xsl:choose>
</xsl:when>
```

- **"ancestor::"** defines the ancestor of the context node up to the root node. The following example navigates to the ancestor:: (Items) to then retrieve a given Item based on an XPath filter. This is especially useful to retrieve any specific node from the souce document - in this case, using a variable as a filter.

```
<xsl:variable name="headerTxt" select="ancestor::Items/
Item[Property[@Name='voucher']=$headerLink and
Property[@Name='HeaderDetailFlag']='H']/Property[@Name='txt']">
</xsl:variable>
```

Use these XPath positioning notations to reference the appropriate source nodes as needed.

XPath Filtering

The XSLs also use filtering, especially when testing for the property Name attribute:

```
<xsl:when test="@Name='vend_num'">
```

Filter is also used in other node value queries. For example:

```
<xsl:variable name="headerRef" select="ancestor::Items/
Item[Property[@Name='inv_num']=$headerLink and
Property[@Name='HeaderDetailFlag']='H']/Property[@Name='ref']">
</xsl:variable>
```

This example navigates to the root element of the context using the ancestor:: node positioning. It then finds the Items/Item node where the value of the Property attribute Name "inv_num" is equal to the variable named "headerLink" and the value of the Property attribute Name "HeaderDetailFlag" is H. It them retrieves the value of the Property Name "ref".

Specifying Additional Ledger Details

The <DetailLad> tag contains all the SunSystems additional ledger detail fields, which include the configurable dates and general descriptions.

XSLs Used to Flatten Original XML Structure

As originally exported from CloudSuite Industrial, A/P and A/R transaction IDO XMLs are set up with a parent-child hierarchy between headers and details. Two XSLs are used to "flatten" the structure of these XMLs so that the header and detail data is at the same level of hierarchy in the file. You should not need to customize these XSLs; however, you might have to change their filenames.

By default, these XSL files are named:

- SyteLineEXTFIN.ExtFinAPVoucherPosting.export_aptrxSunSystems.xsl
- SyteLineEXTFIN.ExtFinARInvoicePosting.export_arinvSunSystems.xsl

These files reside on the utility server in an XSL subfolder under the replication directory. The replication directory is specified in the Service Configuration Manager's **Replication** tab.

System Types and XSL File Naming Conventions

The naming convention for these files is

[*source_site_system_type*][*object_Name*][*target_site_system_type*]

where *source_site_system_type* and *target_site_system_type* are used in the site definitions on the Sites/Entities form.

If the source site is already set up in CloudSuite Industrial's Sites/Entities form using a system type other than CloudSuite Industrial, you must either change the system type for the site to CloudSuite Industrial, or change these XSL filenames to match the other system type. The only time this might be an issue is if you have different CloudSuite Industrial versions running at different sites that are performing non-transaction replication to each other, and you are supplying some other transforms to handle the schema changes.

For example, if your CloudSuite Industrial "from" site's system type is set to SL704, and you have other transforms that require that system type, then the XSL filenames above should be changed to SL704EXTFIN.ExtFinAPVoucherPosting.export_aptrxSunSystems.XSL and SL704EXTFIN.ExtFinARInvoicePosting.export_arinvSunSystems.XSL.

Handling Distributions without Account Numbers

The SyteLineEXTFIN.ExtFinARInvoicePosting.export_arinvSunSystems.xsl (above) also removes any distributions that do not contain a valid account number.

XSL Example: Customizing the Payment Method for Vendor/Supplier

First, we show you the default versions:

- A sample inbound IDORequest XML created when a new vendor record was saved in CloudSuite Industrial.
- The default XSL (from the SLSI Action Profiles form) that is used to transform the XML. In this default version, the payment method is always set to 0.
- The outbound XML (SSML) that is sent to SunSystems after transformation.

Then we show you the customized version:

- View modification. The example requires the CloudSuite Industrial vendor payment type in order to determine the SunSystems payment method to use. The vendor data in the XML is pulled from a limited view in the database (SLSIVendorView). Since that view currently does not include the vendor payment type (vendor.pay_type), you will need to add it to the view and then regenerate the replication triggers. (Modifying the views is only necessary when you need to reference customer, custaddr, vendor, or vendaddr data that is not in the views.)
- The original XSL, but with a modification to the payment type transformation that sets the payment method and priority based on the payment type coming from CloudSuite Industrial.
- The outbound XML that is sent to SunSystems after the modification.

The area that will be customized is highlighted in all of the examples below.

Sample Inbound XML from CloudSuite Industrial

An XML similar to the following is sent to SLSI when a CloudSuite Industrial user adds a vendor. Notice that the payment type is not set. Notice also the Parameters section at the end (see Chapter 5, “Setting Parameters in the XML”).

```
<IDORequest>
  <RequestHeader Type="UpdateCollection">
    <SourceName>MI</SourceName>
    <TargetName>SUN</TargetName>
    <RequestData>
      <UpdateCollection Name="TABLE!vendor">
        <Items>
          <Item Action="Insert">
            <Property Name="vend_num">92665</Property>
            <Property Name="phone" />
            <Property Name="curr_code">USD</Property>
            <Property Name="terms_code">2%</Property>
            <Property Name="tax_reg_num1">89711177777</Property>
            <Property Name="RowPointer">060E952B-EDCE-40D7-9BB6-
5B948D7B227A</Property>
          </Item>
        </Items>
      </UpdateCollection>
    </RequestData>
  </RequestHeader>
```

```
  <Parameters>
    <Parameter Name="SscUser">AMP</Parameter>
    <Parameter Name="SscBusinessUnit">SUN</Parameter>
    <Parameter Name="SscCompatibilityMode">0</Parameter>
    <Parameter Name="SscErrorOutput">1</Parameter>
    <Parameter Name="SscErrorThreshold">1</Parameter>
    <Parameter Name="SscBudgetCode">A</Parameter>
    <Parameter Name="SscAllowBalTran">N</Parameter>
    <Parameter Name="SscAllowOverBudget">Y</Parameter>
    <Parameter Name="SscAllowPostToSuspended">N</Parameter>
    <Parameter Name="SscBalancingOptions" />
    <Parameter Name="SscLayoutCode" />
    <Parameter Name="SscLoadOnly">N</Parameter>
    <Parameter Name="SscPostProvisional">N</Parameter>
    <Parameter Name="SscPostToHold">N</Parameter>
    <Parameter Name="SscPostingType">2</Parameter>
    <Parameter Name="SscPrint">N</Parameter>
    <Parameter Name="SscReportErrorsOnly">Y</Parameter>
    <Parameter Name="SscReportingAccount" />
    <Parameter Name="SscSuppressSubstitutedMessages">Y</Parameter>
    <Parameter Name="SscSuspenseAccount" />
    <Parameter Name="SscTransactionAmountAccount" />
  </Parameters>
</IDORequest>
```

Sample Default XSL

The following XSL is used in the default action profile defined for an CloudSuite Industrial Vendor to SunSystems Supplier insert. You may also want to reference the Vendor to Supplier mapping table on page 68, to better understand what this XSL is doing.

In the highlighted section below, the default XSL always sets the payment method to 1 no matter what payment method was entered in CloudSuite Industrial.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- script version 2.2 -->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
  <xsl:variable name="smallcase" select="'abcdefghijklmnopqrstuvwxyz'" />
  <xsl:variable name="uppercase" select="'ABCDEFGHIJKLMNOPQRSTUVWXYZ'" />
  <xsl:template match="/">
    <xsl:variable name="Params" select="IDORequest/Parameters"></xsl:variable>
    <xsl:element name="SSC">
      <xsl:element name="ErrorContext">
        <xsl:element name="CompatibilityMode"><xsl:value-of select="$Params/
Parameter[@Name='SscCompatibilityMode']"></xsl:value-of></xsl:element>
        <xsl:element name="ErrorOutput"><xsl:value-of select="$Params/
Parameter[@Name='SscErrorOutput']"></xsl:value-of></xsl:element>
        <xsl:element name="ErrorThreshold"><xsl:value-of select="$Params/
Parameter[@Name='SscErrorThreshold']"></xsl:value-of></xsl:element>
      </xsl:element>
      <xsl:element name="User">
        <xsl:element name="Name"><xsl:value-of select="$Params/
```

```
Parameter[@Name='SscUser']"></xsl:value-of></xsl:element>
      </xsl:element>
      <xsl:element name="SunSystemsContext">
        <xsl:element name="BusinessUnit"><xsl:value-of select="$Params/
Parameter[@Name='SscBusinessUnit']"></xsl:value-of></xsl:element>
      </xsl:element>
      <xsl:element name="Payload">
        <xsl:apply-templates select="IDORequest/RequestHeader/RequestData/
UpdateCollection/Items"></xsl:apply-templates>
      </xsl:element>
    </xsl:element>
  </xsl:template>
  <xsl:template name="Vendor" match="Item">
    <xsl:element name="Supplier">
      <xsl:element name="PaymentMethod">1</xsl:element>
      <xsl:choose>
        <xsl:when test="./@Action='Insert'">
          <xsl:element name="SupplierName">TBD</xsl:element>
        </xsl:when>
      </xsl:choose>
      <xsl:for-each select="Property">
        <xsl:choose>
          <xsl:when test="@Name='vend_num'">
            <xsl:element name="SupplierCode"><xsl:value-of
select="concat('C',translate(normalize-space(.), $smallcase, $uppercase))"></
xsl:value-of></xsl:element>
            <xsl:element name="AccountCode"><xsl:value-of
select="concat('C',translate(normalize-space(.), $smallcase, $uppercase))"></
xsl:value-of></xsl:element>
          </xsl:when>
          <xsl:when test="@Name='terms_code'">
            <xsl:element name="PaymentTermsGroupCode"><xsl:value-of select="."></
xsl:value-of></xsl:element>
          </xsl:when>
          <xsl:when test="@Name='curr_code'">
            <xsl:element name="DefaultCurrencyCode"><xsl:value-of select="."></
xsl:value-of></xsl:element>
          </xsl:when>
        </xsl:choose>
      </xsl:for-each>
      <Tax>
        <xsl:element name="SequenceNumber">1</xsl:element>
        <xsl:for-each select="Property">
          <xsl:choose>
            <xsl:when test="@Name='vend_num'">
              <xsl:element name="SupplierCode"><xsl:value-of
select="concat('C',translate(normalize-space(.), $smallcase, $uppercase))"></
xsl:value-of></xsl:element>
            </xsl:when>
            <xsl:when test="@Name='tax_reg_num1'">
              <xsl:element name="TaxIdentificationCode"><xsl:value-of
select="."></xsl:value-of></xsl:element>
            </xsl:when>
          </xsl:choose>
        </xsl:for-each>
      </Tax>
```

```
    </xsl:element>
  </xsl:template>
</xsl:stylesheet>
```

Sample Default Outbound XML to SunSystems

An SSML (SunSystems XML) similar to the following would be sent to SunSystems after the default XSL above was applied to the input XML. Notice that the payment method is set to 1.

```
<?xml version="1.0" encoding="utf-8" ?>
<SSC>
  <ErrorContext>
    <CompatibilityMode>0</CompatibilityMode>
    <ErrorOutput>1</ErrorOutput>
    <ErrorThreshold>1</ErrorThreshold>
  </ErrorContext>
  <User>
    <Name>AMP</Name>
  </User>
  <SunSystemsContext>
    <BusinessUnit>SUN</BusinessUnit>
  </SunSystemsContext>
  <Payload>
    <Supplier>
   <PaymentMethod>1</PaymentMethod>
      <SupplierName>TBD</SupplierName>
      <SupplierCode>C92665</SupplierCode>
      <AccountCode>C92665</AccountCode>
      <PaymentTermsGroupCode>2%</PaymentTermsGroupCode>
      <DefaultCurrencyCode>USD</DefaultCurrencyCode>
      <Tax>
        <SequenceNumber>1</SequenceNumber>
        <SupplierCode>C92665</SupplierCode>
        <TaxIdentificationCode>89711177777</TaxIdentificationCode>
      </Tax>
    </Supplier>
  </Payload>
</SSC>
```

View Modification

You need to add vendor.pay_type to SLSIVendorView in the CloudSuite Industrial application database. To do this, you can create a copy of the SQL script and modify it, then regenerate the replication triggers.

1 Make a copy of the SQL script SLSIVendorView.sql (found on the utility server in C:\Program Files\Infor\SyteLine\SLSI\ApplicationDb\Views). Name the new script SLSIVendorWithPayTypeView.sql.

2 Edit the new script:

```
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_NULLS ON
GO
IF OBJECT_ID('dbo.SLSIVendorView') IS NOT NULL
DROP VIEW dbo.SLSIVendorView
GO
/* $Header: /SyteLine/ApplicationDB/Views/SLSIVendorView.sql 4     7/01/10 10:42a
Dahn $ */
/*
*************************************************************
*                           NOTICE                          *
*                                                           *
*   THIS SOFTWARE IS THE PROPERTY OF AND CONTAINS           *
*   CONFIDENTIAL INFORMATION OF INFOR AND/OR ITS AFFILIATES  *
*   OR SUBSIDIARIES AND SHALL NOT BE DISCLOSED WITHOUT PRIOR  *
*   WRITTEN PERMISSION. LICENSED CUSTOMERS MAY COPY AND       *
*   ADAPT THIS SOFTWARE FOR THEIR OWN USE IN ACCORDANCE WITH  *
*   THE TERMS OF THEIR SOFTWARE LICENSE AGREEMENT.            *
*   ALL OTHER RIGHTS RESERVED.                              *
*                                                           *
*   (c) COPYRIGHT 2010 INFOR.  ALL RIGHTS RESERVED.         *
*   THE WORD AND DESIGN MARKS SET FORTH HEREIN ARE          *
*   TRADEMARKS AND/OR REGISTERED TRADEMARKS OF INFOR        *
*   AND/OR ITS AFFILIATES AND SUBSIDIARIES. ALL RIGHTS      *
*   RESERVED.  ALL OTHER TRADEMARKS LISTED HEREIN ARE       *
*   THE PROPERTY OF THEIR RESPECTIVE OWNERS.                *
*                                                           *
*************************************************************
*/
/* $Archive: /SyteLine/ApplicationDB/Views/SLSIVendorView.sql $
*
* Initial Revision for 106612
*
* $NoKeywords: $
*/
CREATE VIEW dbo.SLSIVendorView
AS
SELECT vend_num,
phone,
curr_code,
terms_code,
tax_reg_num1,
vendor.pay_type,
RowPointer
FROM vendor
GO
```

3 In SQL, run the new script against the CloudSuite Industrial database and then verify that the updated view displays the correct data.

4 In CloudSuite Industrial, go to the Replication Management form and regenerate replication triggers.

Modified Inbound XML from CloudSuite Industrial

After the view has been modified and triggers are regenerated, an XML similar to the following is sent to SLSI when an CloudSuite Industrial user adds a vendor. Notice that the payment type is now set (in this example, it is set to S for standard check).

```
<IDORequest>
  <RequestHeader Type="UpdateCollection">
    <SourceName>MI</SourceName>
    <TargetName>SUN</TargetName>
    <RequestData>
      <UpdateCollection Name="TABLE!vendor">
        <Items>
          <Item Action="Insert">
            <Property Name="vend_num">92665</Property>
            <Property Name="phone" />
            <Property Name="curr_code">USD</Property>
            <Property Name="terms_code">2%</Property>
            <Property Name="tax_reg_num1">89711177777</Property>
            <Property Name="pay_type">S</Property>
            <Property Name="RowPointer">060E952B-EDCE-40D7-9BB6-
5B948D7B227A</Property>
          </Item>
        </Items>
      </UpdateCollection>
    </RequestData>
  </RequestHeader>
  <Parameters>
    <Parameter Name="SscUser">AMP</Parameter>
    <Parameter Name="SscBusinessUnit">SUN</Parameter>
    <Parameter Name="SscCompatibilityMode">0</Parameter>
    <Parameter Name="SscErrorOutput">1</Parameter>
    <Parameter Name="SscErrorThreshold">1</Parameter>
    <Parameter Name="SscBudgetCode">A</Parameter>
    <Parameter Name="SscAllowBalTran">N</Parameter>
    <Parameter Name="SscAllowOverBudget">Y</Parameter>
    <Parameter Name="SscAllowPostToSuspended">N</Parameter>
    <Parameter Name="SscBalancingOptions" />
    <Parameter Name="SscLayoutCode" />
    <Parameter Name="SscLoadOnly">N</Parameter>
    <Parameter Name="SscPostProvisional">N</Parameter>
    <Parameter Name="SscPostToHold">N</Parameter>
    <Parameter Name="SscPostingType">2</Parameter>
    <Parameter Name="SscPrint">N</Parameter>
    <Parameter Name="SscReportErrorsOnly">Y</Parameter>
    <Parameter Name="SscReportingAccount" />
    <Parameter Name="SscSuppressSubstitutedMessages">Y</Parameter>
    <Parameter Name="SscSuspenseAccount" />
    <Parameter Name="SscTransactionAmountAccount" />
  </Parameters>
</IDORequest>
```

Modified XSL

The following modified XSL could be applied instead of the default XSL. In order to do this, you would export the default XSL from the Action Profiles form and edit it in a standard XML editor such as XSL Spy, so that you can take advantage of the editor's tools and syntax help. Then import the changed version back into the Action Profiles form.

Notice that the original XSL's line about the vendor payment method was removed. A new section (highlighted below) was added to select the vendor payment method and priority based on the payment type (S, M, W, T, N, or E) sent from CloudSuite Industrial.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- script version 2.2 -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/
Transform">

...

  <xsl:template name="Vendor" match="Item">
    <xsl:element name="Supplier">
      <xsl:choose>
        <xsl:when test="./@Action='Insert'">
          <xsl:element name="SupplierName">TBD</xsl:element>
        </xsl:when>
      </xsl:choose>
      <xsl:for-each select="Property">
        <xsl:choose>
          <xsl:when test="@Name='vend_num'">
            <xsl:element name="SupplierCode"><xsl:value-of
select="concat('C',translate(normalize-space(.), $smallcase,
$uppercase))"></xsl:value-of></xsl:element>
            <xsl:element name="AccountCode"><xsl:value-of
select="concat('C',translate(normalize-space(.), $smallcase,
$uppercase))"></xsl:value-of></xsl:element>
          </xsl:when>
          <xsl:when test="@Name='pay_type'">
           <xsl:choose>
             <xsl:when test=".='S'">
               <xsl:element name="PaymentMethod">0</xsl:element>
             </xsl:when>
             <xsl:when test=".='M'">
               <xsl:element name="PaymentMethod">99</xsl:element>
               <xsl:element name="Priority">U</xsl:element>
             </xsl:when>
             <xsl:when test=".='W'">
               <xsl:element name="PaymentMethod">99</xsl:element>
               <xsl:element name="Priority">D</xsl:element>
             </xsl:when>
             <xsl:when test=".='T'">
               <xsl:element name="PaymentMethod">99</xsl:element>
               <xsl:element name="Priority">F</xsl:element>
             </xsl:when>
             <xsl:when test=".='N'">
               <xsl:element name="PaymentMethod">99</xsl:element>
               <xsl:element name="Priority">E</xsl:element>
             </xsl:when>
             <xsl:when test=".='E'">
               <xsl:element name="PaymentMethod">1</xsl:element>
```

```
            </xsl:when>
          </xsl:choose>
          </xsl:when>
          <xsl:when test="@Name='terms_code'">
            <xsl:element name="PaymentTermsGroupCode"><xsl:value-of
select="."></xsl:value-of></xsl:element>
          </xsl:when>
          <xsl:when test="@Name='curr_code'">
            <xsl:element name="DefaultCurrencyCode"><xsl:value-of
select="."></xsl:value-of></xsl:element>
          </xsl:when>
        </xsl:choose>
      </xsl:for-each>

...

    </xsl:element>
  </xsl:template>
</xsl:stylesheet>
```

Modified Outbound XML to SunSystems

An SSML (SunSystems XML) similar to the following would be sent to SunSystems after the modified XSL above was applied to the input XML. Notice that the payment method is set to 0 because the CloudSuite Industrial pay_type was S, as requested in the modified XSL.

```
<?xml version="1.0" encoding="utf-8" ?>
  <SSC>
  <ErrorContext>
    <CompatibilityMode>0</CompatibilityMode>
    <ErrorOutput>1</ErrorOutput>
    <ErrorThreshold>1</ErrorThreshold>
  </ErrorContext>
  <User>
    <Name>AMP</Name>
  </User>
  <SunSystemsContext>
    <BusinessUnit>SUN</BusinessUnit>
  </SunSystemsContext>
  <Payload>
    <Supplier>
      <PaymentMethod>0</PaymentMethod>
      <SupplierName>TBD</SupplierName>
      <SupplierCode>C92665</SupplierCode>
      <AccountCode>C92665</AccountCode>
      <PaymentTermsGroupCode>2%</PaymentTermsGroupCode>
      <DefaultCurrencyCode>USD</DefaultCurrencyCode>
      <Tax>
        <SequenceNumber>1</SequenceNumber>
        <SupplierCode>C92665</SupplierCode>
        <TaxIdentificationCode>89711177777</TaxIdentificationCode>
      </Tax>
    </Supplier>
  </Payload>
</SSC>
```

Chapter 7: Mapping Details

7

Two types of mapping must be done between CloudSuite Industrial and SunSystems:

- **Mapping a CloudSuite Industrial site to a SunSystems business unit**. In each CloudSuite Industrial site that will pass data to SunSystems:

  a Create a “SUN” intranet that maps to the SLSI URL.

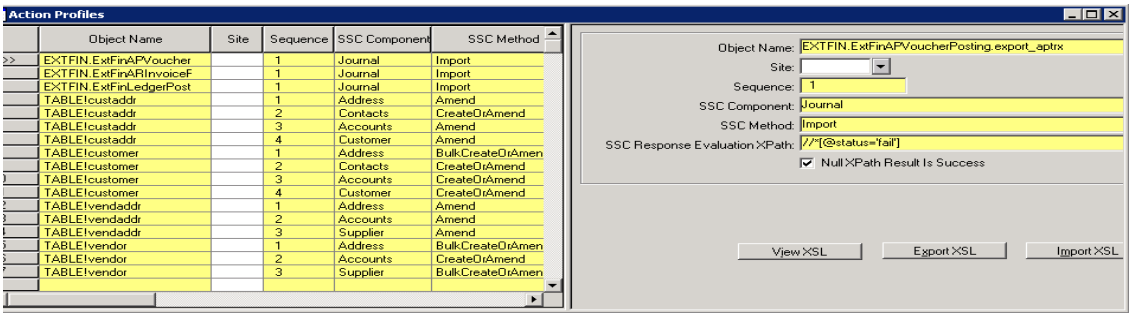
  b Add a “site” on that intranet for the business unit that will receive the site’s data.

  c Create replication rules to pass data from the CloudSuite Industrial site to the business unit site.

  d In SLSI’s Site to Business Unit Mappings form, set up the SSC connection to the business unit, using the “site” name you specified in CloudSuite Industrial.

  For more information, see the *CloudSuite Industrial Enterprise Financials Installation Guide*.

- **Mapping data from a site database to a business unit database**.This is done by XSLs tied to the SLSI Action Profiles.



The rest of this chapter describes the data mapping set up in the default XSLs. You may customize these mappings if necessary; however, be sure to make backups and test any customizations thoroughly.

Note: In the tables that follow, constant values are represented by values in square brackets [ ].

Parameters

Before the transformations are run, SLSI adds a Parameters section to the IDORequest XML, based on information in the Process Defaults and Site to Business Unit Mappings forms. (See Chapter 5, “Setting Parameters in the XML”). Where applicable, this data source is noted in the “Source”

columns of the tables that follow.

CloudSuite Industrial Customer to SunSystems Address

Source type: customer table
Sequence for type: 1

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format.
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail.
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only.
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Address	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AddressCode	cust_num	D + Trim and uppercase
		TelephoneNumber	Phone##3	
		AddressLine1	[TBD]	Since name is not part of CloudSuite Industrial customer record, temporarily set to TBD until address is created/updated. Only populated if the source item action is "Insert"

CloudSuite Industrial Customer to SunSystems Contact

Source type: customer table
Sequence for type: 2

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Customer	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	Name	contact##3	
		ContactCode	cust_num	D + Trim and uppercase
		TelephoneNumber	phone##3	

CloudSuite Industrial Customer to SunSystems Account

Source type: customer table
Sequence for type: 3

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Accounts	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AccountType	[0]	Debitor
		BalanceType	[1]	Open Item
		Description	[TBD]	Since name is not part of CloudSuite Industrial customer record, temporarily set to TBD until address is crated/ updated. Only populated if the source item action is "Insert"
		AccountCode	cust_num	D + Trim and uppercase

CloudSuite Industrial Customer to SunSystems Customer

Source type: customer table
Sequence for type: 4

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Customer	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	PaymentMethod	[1]	Bank Payment
		Description	[TBD]	Since name is not part of CloudSuite Industrial customer record, temporarily set to TBD until address is crated/ updated. Only populated if the source item action is "Insert"
		CustomerCode	cust_num	D + Trim and uppercase
		PaymentTermsGroupCode	terms_code	
		Tax/CustomerCode	cust_num	D + Trim and uppercase
		Tax/ SequenceNumber	[1]	
		Tax/TaxIdentification Code	tax_reg_num1	

CloudSuite Industrial Customer Address to SunSystems Address

Source type: custaddr table
Sequence for type: 1

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Address	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AddressCode	cust_num	D + Trim and uppercase
		AddressLine1	name	First 50 characters
		ShortHeading	name	First 15 characters and uppercase
		LookupCode	name	First 15 characters and uppercase
		AddressLine2	addr##1	
		AddressLine3	addr##2	
		AddressLine4	addr##3	
		AddressLine5	addr##4	
		Country	country	
		Area	county	
		FaxNumber	fax_num	
		PostalCode	zip	
		State	state	
		TownCity	city	
		WebPageAddress	internet_url	

CloudSuite Industrial Customer Address to SunSystems Contact

Source type: custaddr table
Sequence for type: 2

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Contacts	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	ContactCode	cust_num	D + Trim and uppercase
		EmailAddress	external_email_addr	
		FaxNumber	fax_num	

CloudSuite Industrial Customer Address to SunSystems Account

Source type: custaddr table
Sequence for type: 3

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Accounts	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AccountType	[0]	Debitor
		BalanceType	[1]	Open Item
		AccountCode	cust_num	D + Trim and uppercase
		ContactAddress/ AddressCode	cust_num	D + Trim and uppercase
		ContactAddress/ ContactIdentifier	cust_num	D + Trim and uppercase
		UserArea	cust_num	
		LookupCode	name	Trim and uppercase first 15 characters
		ShortHeading	name	Trim and uppercase first 15 characters
		DefaultCurrencyCode	curr_code	
		Description	name	First 50 characters

CloudSuite Industrial Customer Address to SunSystems Customer

Source type: custaddr table
Sequence for type: 4

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only

Area	Occurrence	Target	Source	Function/ Comment
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Customer	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	ActualAccount	cust_num	D + Trim and uppercase
		CompanyAddressCode	cust_num	D + Trim and uppercase
		CustomerCode	cust_num	D + Trim and uppercase
		ContactAddress/ AddressCode	cust_num	D + Trim and uppercase
		ContactAddress/ ContactIdentifier	cust_num	D + Trim and uppercase
		DefaultCurrencyCode	curr_code	
		LookupCode	name	Trim and uppercase first 15 characters
		ShortHeading	name	Trim and uppercase first 15 characters
		EMailAddress	external_email_addr	
		Name	name	First 50 characters
		Description	name	First 50 characters

CloudSuite Industrial Vendor to SunSystems Address

Source type: vendor table
Sequence for type: 1

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Address	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AddressCode	vend_num	C + Trim and uppercase
		TelephoneNumber	phone	
		AddressLine1	[TBD]	Since name is not part of CloudSuite Industrial vendor record, temporarily set to TBD until address is created/ updated. Only populated if the source item action is “Insert”.

CloudSuite Industrial Vendor to SunSystems Account

Source type: vendor table
Sequence for type: 2

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Accounts	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AccountType	[1]	Creditor
		BalanceType	[1]	Open Item
		Description	[TBD]	Since name is not part of CloudSuite Industrial vendor record, temporarily set to TBD until address is created/ updated. Only populated if the source item action is “Insert”.
		DefaultCurrency Code	curr_code	
		AccountCode	vend_num	C + Trim and uppercase

CloudSuite Industrial Vendor to SunSystems Supplier

Source type: vendor table
Sequence for type: 3

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Supplier	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	SupplierName	[TBD]	Since name is not part of CloudSuite Industrial vendor record, temporarily set to TBD until address is created/updated. Only populated if the source item action is "Insert"
		SupplierCode	vend_num	C + Trim and uppercase
		AccountCode	vend_num	C + Trim and uppercase
		DefaultCurrency Code	curr_code	
		PaymentMethod	[1]	
		PaymentTerms GroupCode	terms_code	
		Tax/Sequence Number	[1]	
		Tax/SupplierCode	vend_num	C + Trim and uppercase
		Tax/TaxIdentification Code	tax_reg_num1	

CloudSuite Industrial Vendor Address to SunSystems Address

Source type: vendaddr table
Sequence for type: 1

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Address	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AddressCode	vend_num	C + Trim and uppercase
		AddressLine1	name	First 50 characters
		LookupCode	name	First 15 characters and uppercase
		ShortHeading	name	First 15 characters and uppercase
		AddressLine2	addr##1	
		AddressLine3	addr##2	
		AddressLine4	addr##3	
		AddressLine5	addr##4	
		Country	country	
		Area	county	
		FaxNumber	fax_num	
		PostalCode	zip	
		State	state	
		TownCity	city	
		WebPageAddress	internet_url	

CloudSuite Industrial Vendor Address to SunSystems Account

Source type: vendaddr table
Sequence for type: 2

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Accounts	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AccountType	[1]	Creditor
		BalanceType	[1]	Open Item
		AccountCode	vend_num	C + Trim and uppercase
		Description	name	Trim and uppercase first 50 characters
		LookupCode	name	Trim and uppercase first 15 characters
		ShortHeading	name	Trim and uppercase first 15 characters

CloudSuite Industrial Vendor Address to SunSystems Supplier

Source type: vendaddr table
Sequence for type: 3

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/ SunSystemsContext	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	
SSC/Payload/ Supplier	One per source “Item” in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AccountCode	vend_num	C + Trim and uppercase
		CompanyAddress Code	vend_num	C + Trim and uppercase
		SupplierCode	vend_num	C + Trim and uppercase
		EMailAddress	external_email_addr	
		LookupCode	name	Trim and uppercase first 15 characters
		ShortHeading	name	Trim and uppercase first 15 characters

Area	Occurrence	Target	Source	Function/ Comment
		Description	name	Trim and uppercase first 50 characters
		SupplierName	name	Trim and uppercase first 50 characters

CloudSuite Industrial A/P Vouchers to SunSystems Journal

Source type: A/P Voucher Posting
Sequence for type: 1

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/SunSystems Context	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	Target SunSystems Business Unit
		BudgetCode	IDORequest/ Parameters/ SscBudgetCode (Process Defaults)	Default value A - Actual

Area	Occurrence	Target	Source	Function/ Comment
SSC/ LedgerPostingParameters	One per SSC call	AllowBalTran	IDORequest/ Parameters/ SscAllowBalTran (Process Defaults)	Default N - No
		AllowOverBudget	IDORequest/ Parameters/ SscAllowOverBudget (Process Defaults)	Default Y - Yes
		AllowPostToSuspended	IDORequest/ Parameters/ SscAllowPostToSuspended (Process Defaults)	Default N - No
		BalancingOptions	IDORequest/ Parameters/ SscBalancingOptions (Process Defaults)	Default Blank - No balance option
		DefaultPeriod	cur_per fiscal_year	Derived from value on first header record and "0" + cur_per + fiscal_year
		Description	[CloudSuite Industrial AP Voucher Posting]	
		JournalType	[AP DI]	
		LayoutCode	IDORequest/ Parameters/ SscLayoutCode (Process Defaults)	Default Blank - No layout code specified
		LoadOnly	IDORequest/ Parameters/ SscLoadOnly (Process Defaults)	Default N - No
		PostProvisional	IDORequest/ Parameters/ SscPostProvisional (Process Defaults)	Default N - No
		PostToHold	IDORequest/ Parameters/ SscPostToHold (Process Defaults)	Default N - No
		PostingType	IDORequest/ Parameters/ SscPostingType (Process Defaults)	Default 2 - Post if no errors

Area	Occurrence	Target	Source	Function/ Comment
SSC/ LedgerPostingParameters	One per SSC call	Print	IDORequest/ Parameters/ SscPrint (Process Defaults)	Default N - No
		ReportErrorsOnly	IDORequest/ Parameters/ SscReportErrorsOnly (Process Defaults)	Default Y - Yes
		ReportingAccount	IDORequest/ Parameters/ SscReportingAccount (Process Defaults)	Reporting account code to be specified for site
		SuppressSubstitutedMessages	IDORequest/ Parameters/ SscSuppress SubstitutedMessages (Process Defaults)	Default Y - Yes
		SuspenseAccount	IDORequest/ Parameters/ SscSuspenseAccount (Process Defaults)	Suspense account code to be specified for site
		TransactionAmount Account	IDORequest/ Parameters/ SscTransactionAmount Account (Process Defaults)	Transaction account code to be specified for site
SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AccountCode	HeaderDetailFlag vend_num acct	If HeaderDetailFlag = H then C + vend_num If HeaderDetailFlag = D then acct Trim and uppercase
		AccountingPeriod	cur_per fiscal_year voucher	Derived from value on related header record linked by voucher and formatted to three digits followed by fiscal_year

Area	Occurrence	Target	Source	Function/ Comment

SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AnalysisCode1	ap_acct_unit1 acct_unit1	If HeaderDetailFlag = H then
				● If ap_acct_unit1 blank then NOT USED
				● Else ap_acct_unit1
				If HeaderDetailFlag = D then
				● If acct_unit1 blank then NOT USED
				● Else acct_unit1
				Note: Trim and uppercase

		AnalysisCode2	ap_acct_unit2 acct_unit2	If HeaderDetailFlag = H then
				● If ap_acct_unit2 blank then NOT USED
				● Else ap_acct_unit2
				If HeaderDetailFlag = D then
				● If acct_unit2 blank then NOT USED
				● Else acct_unit2
				Note: Trim and uppercase

		AnalysisCode3	ap_acct_unit3 acct_unit3	If HeaderDetailFlag = H then
				● If ap_acct_unit3 blank then NOT USED
				● Else ap_acct_unit3
				If HeaderDetailFlag = D then
				● If acct_unit3 blank then NOT USED
				● Else acct_unit3
				Note: Trim and uppercase

Area	Occurrence	Target	Source	Function/ Comment

SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AnalysisCode4	ap_acct_unit4
			acct_unit4
				If HeaderDetailFlag = H then
				● If ap_acct_unit4 blank then NOT USED
				● Else ap_acct_unit4
				If HeaderDetailFlag = D then
				● If acct_unit4 blank then NOT USED
				● Else acct_unit4
				Note: Trim and uppercase

		AnalysisCode5	HeaderDetailFlag tax_code_e tax_code	If HeaderDetailFlag = H then NOT USED If HeaderDetailFlag = D then if tax_code_e is blank then if tax_code is blank then NOT USED otherwise tax_code_e
		AnalysisCode10	voucher	Trim and uppercase

		ConversionRate	exch_rate	If HeaderDetailFlag = H then
				● if exch_rate > 0 and inv_amt != 0 then exch_rate
				● Else 0
				If HeaderDetailFlag = D then
				● if header exch_rate > 0 and amount != 0 then header exch_rate
				● Else 0

		CurrencyCode	curr_code	Derived from value on related header record linked by voucher

Area	Occurrence	Target	Source	Function/ Comment

SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	DebitCredit	HeaderDetailFlag
			inv_amt
			amount
			type
			voucher
				If HeaderDetailFlag = H then
				If inv_amt < 0
				then "D"
				else "C"
				If HeaderDetailFlag = D then
				If amount < 0
				then "D"
				else "C"
				Note that type is pulled from the related header linked by voucher

		Description	txt	Derived from value on related header record linked by voucher
		DetailLad/ AccountCode	HeaderDetailFlag vend_num ap_acct	If HeaderDetailFlag = H then C + vend_num If HeaderDetailFlag = D then ap_acct Trim and uppercase
		DetailLad/ GeneralDate1	inv_date	Derived from value on related header record linked by voucher and formatted from YYYY-MM-DD to DDMMYYYY
		DetailLad/ GeneralDate2	dist_date	Derived from value on related header record linked by voucher and formatted from YYYY-MM-DD to DDMMYYYY
		DetailLad/ GeneralDescription1	inv_num	
		DetailLad/ GeneralDescription2	disc_pct	Derived from value on related header record linked by voucher and append percentage sign

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	DueDate	due_date	Derived from value on related header record linked by voucher and formatted from YYYY-MM-DD to DDMMYYYY
		Extension/ DiscountDate1	disc_date	Derived from value on related header record linked by voucher and formatted from YYYY-MM-DD to DDMMYYYY
		Extension/ DiscountPercent1	disc_pct	Derived from value on related header record linked by voucher
		JournalType	[AP DI]	
		JournalSource	type	Derived from value on related header record linked by voucher
		MemoAmount	tax_basis	If tax_basis is blank then 0 otherwise tax_basis
		PermanentPostingReference	po_num	Derived from value on related header record linked by voucher
		TransactionAmount	HeaderDetailFlag inv_amt amount	If HeaderDetailFlag = H then inv_amt If HeaderDetailFlag = D then amount
		TransactionDate	inv_date	Derived from value on related header record linked by voucher and formatted from YYYY-MM-DD to DDMMYYYY
		TransactionReference	ref	Derived from value on related header record linked by voucher and trim

CloudSuite Industrial A/R Invoice to SunSystems Journal

Source type: A/R Invoice Posting
Sequence for type: 1

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	
SSC/SunSystems Context	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	Target SunSystems Business Unit
		BudgetCode	IDORequest/ Parameters/ SscBudgetCode (Process Defaults)	Default value A - Actual
SSC/ LedgerPostingParameters	One per SSC call	AllowBalTran	IDORequest/ Parameters/ SscAllowBalTran (Process Defaults)	Default N - No
		AllowOverBudget	IDORequest/ Parameters/ SscAllowOverBudget (Process Defaults)	Default Y - Yes
		AllowPostToSuspended	IDORequest/ Parameters/ SscAllowPostToSuspended (Process Defaults)	Default N - No

Area	Occurrence	Target	Source	Function/ Comment
SSC/ LedgerPostingParameters	One per SSC call	BalancingOptions	IDORequest/ Parameters/ SscBalancingOptions (Process Defaults)	Default Blank - No balance option
		DefaultPeriod	cur_per fiscal_year	Derived from value on first header record and "0" + cur_per + fiscal_year
		Description	[CloudSuite Industrial AR Invoice Posting]	
		JournalType	[AR DI]	
		LayoutCode	IDORequest/ Parameters/ SscLayoutCode (Process Defaults)	Default Blank - No layout code specified
		LoadOnly	IDORequest/ Parameters/ SscLoadOnly (Process Defaults)	Default N - No
		PostProvisional	IDORequest/ Parameters/ SscPostProvisional (Process Defaults)	Default N - No
		PostToHold	IDORequest/ Parameters/ SscPostToHold (Process Defaults)	Default N - No
		PostingType	IDORequest/ Parameters/ SscPostingType (Process Defaults)	Default 2 - Post if no errors
		Print	IDORequest/ Parameters/ SscPrint (Process Defaults)	Default N - No
		ReportErrorsOnly	IDORequest/ Parameters/ SscReportErrorsOnly (Process Defaults)	Default Y - Yes
		ReportingAccount	IDORequest/ Parameters/ SscReportingAccount (Process Defaults)	Reporting account code to be specified for site

Area	Occurrence	Target	Source	Function/ Comment
SSC/ LedgerPostingParameters	One per SSC call	SuppressSubstitutedMessages	IDORequest/ Parameters/ SscSuppress SubstitutedMessages (Process Defaults)	Default Y - Yes
		SuspenseAccount	IDORequest/ Parameters/ SscSuspenseAccount (Process Defaults)	Suspense account code to be specified for site
		TransactionAmount Account	IDORequest/ Parameters/ SscTransactionAmount Account (Process Defaults)	Transaction account code to be specified for site
SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AccountCode	HeaderDetailFlag cust_num acct	If HeaderDetailFlag = H then D + cust_num If HeaderDetailFlag = D then acct Trim and uppercase
		AccountingPeriod	cur_per fiscal_year inv_num	Derived from value on related header record linked by inv_num and formatted to three digits followed by fiscal_year
		AnalysisCode1	acct_unit1	If blank then NOT USED otherwise Trim and uppercase
		AnalysisCode2	acct_unit2	If blank then NOT USED otherwise Trim and uppercase
		AnalysisCode3	acct_unit3	If blank then NOT USED otherwise Trim and uppercase
		AnalysisCode4	acct_unit4	If blank then NOT USED otherwise Trim and uppercase

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AnalysisCode5	HeaderDetailFlag tax_code_e tax_code	If HeaderDetailFlag = H then NOT USED If HeaderDetailFlag = D then if tax_code_e is blank then if tax_code is blank then NOT USED otherwise tax_code_e
		AnalysisCode9	apply_to_inv_num	Derived from value on related header record linked by inv_num and if blank then NOT USED otherwise Trim and uppercase
		AnalysisCode10	inv_num	Trim and uppercase

		ConversionRate	amount
			misc_charges
			sales_tax
			freight
			sales_tax_2
			exch_rate
				If HeaderDetailFlag = H then
				● If amount + misc_charges + sales_tax + freight + sales_tax_2 > 0 then exch_rate
				● Else 0
				If HeaderDetailFlag = D then
				● If amount > 0 then exch_rate
				● Else 0
				Note: exch_rate is derived from value on related header record linked by inv_num

		CurrencyCode	curr_code	Derived from value on related header record linked by inv_num

Area	Occurrence	Target	Source	Function/ Comment

SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	DebitCredit	HeaderDetailFlag
			amount
			misc_charges
			freight
			sales_tax
			sales_tax_2
			type
			inv_num
				If HeaderDetailFlag = H then
				● If amount + misc_charges + freight + sales_tax + sales_tax_2 < 0
				If type = 'D' then "C"
				If type = 'C' then "D"
				If type = 'I' then "C"
				● If amount + misc_charges + freight + sales_tax + sales_tax_2 >= 0
				If type = 'D' then "D"
				If type = 'C' then "C"
				If type = 'I' then "D"
				If HeaderDetailFlag = D then
				● If amount < 0
				If type = 'D' then "D"
				If type = 'C' then "C"
				If type = 'I' then "D"
				● If amount >= 0
				If type = 'D' then "C"
				If type = 'C' then "D"
				If type = 'I' then "C"
				Note that type is pulled from the related header linked by inv_num

		Description	description	Derived from value on related header record linked by inv_num

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	DetailLad/ AccountCode	HeaderDetailFlag cust_num acct	If HeaderDetailFlag = H then D + cust_num If HeaderDetailFlag = D then acct Trim and uppercase
		DetailLad/ GeneralDescription3	co_num	
		DetailLad/ GeneralDescription4	ref	
		DetailLad/ Period	cur_per fiscal_year inv_num	Derived from value on related header record linked by inv_num and formatted as "0" + cur_per + fiscal_year
		DueDate	due_date	Derived from value on related header record linked by inv_num and formatted from YYYY-MM-DD to DDMMYYYY
		JournalSource	type	Derived from value on related header record linked by inv_num
		JournalType	[AR DI]	
		MemoAmount	tax_basis	If tax_basis is blank then 0 otherwise tax_basis
		PermanentPostingReference	co_num	Derived from value on related header record linked by inv_num
		TransactionAmount	HeaderDetailFlag amount misc_charges sales_tax freight sales_tax_2	If HeaderDetailFlag = H then amount + misc_charges + sales_tax + freight + sales_tax_2 Otherwise amount
		TransactionDate	inv_date	Derived from value on related header record linked by inv_num and formatted from YYYY-MM-DD to DDMMYYYY

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	TransactionReference	ref	Derived from value on related header record linked by inv_num and trim
		Extension/ PaymentTermsCode	terms_code	Derived from value on related header record linked by inv_num and trim and uppercase

CloudSuite Industrial Ledger Transactions to SunSystems Journal

Source type: Financial Ledger Posting
Sequence for type: 1

Area	Occurrence	Target	Source	Function/ Comment
SSC/ErrorContext	One per SSC call	CompatibilityMode	IDORequest/ Parameters/ SscCompatibilityMode (Process Defaults)	Default value 0 - Messages to be generated in the SunSystems v5.2.n format
		ErrorOutput	IDORequest/ Parameters/ SscErrorOutput (Process Defaults)	Default value 1 - Generate messages with a normal level of detail
		ErrorThreshold	IDORequest/ Parameters/ SscErrorThreshold (Process Defaults)	Default value 1 - Include error messages only
SSC/User	One per SSC call	Name	IDORequest/ Parameters/SscUser (Site To Business Unit Mappings)	

Area	Occurrence	Target	Source	Function/ Comment
SSC/SunSystems Context	One per SSC call	BusinessUnit	IDORequest/ Parameters/ SscBusinessUnit (Site To Business Unit Mappings)	Target SunSystems Business Unit
		BudgetCode	IDORequest/ Parameters/ SscBudgetCode (Process Defaults)	Default value A - Actual
SSC/ LedgerPostingParameters	One per SSC call	AllowBalTran	IDORequest/ Parameters/ SscAllowBalTran (Process Defaults)	Default N - No
		AllowOverBudget	IDORequest/ Parameters/ SscAllowOverBudget (Process Defaults)	Default Y - Yes
		AllowPostToSuspended	IDORequest/ Parameters/ SscAllowPostToSuspended (Process Defaults)	Default N - No
		BalancingOptions	IDORequest/ Parameters/ SscBalancingOptions (Process Defaults)	Default Blank - No balance option
		DefaultPeriod	control_period control_year	Based on value from first item record and comprised of control period formatted to three digits followed by control year
		Description	[CloudSuite Industrial Ledger Posting]	
		JournalType	from_id	Based on value from first item record with trim and uppercase of first 5 characters
		LayoutCode	IDORequest/ Parameters/ SscLayoutCode (Process Defaults)	Default Blank - No layout code specified
		LoadOnly	IDORequest/ Parameters/ SscLoadOnly (Process Defaults)	Default N - No

Area	Occurrence	Target	Source	Function/ Comment
SSC/ LedgerPostingParameters	One per SSC call	PostProvisional	IDORequest/ Parameters/ SscPostProvisional (Process Defaults)	Default N - No
		PostToHold	IDORequest/ Parameters/ SscPostToHold (Process Defaults)	Default N - No
		PostingType	IDORequest/ Parameters/ SscPostingType (Process Defaults)	Default 2 - Post if no errors
		Print	IDORequest/ Parameters/ SscPrint (Process Defaults)	Default N - No
		ReportErrorsOnly	IDORequest/ Parameters/ SscReportErrorsOnly (Process Defaults)	Default Y - Yes
		ReportingAccount	IDORequest/ Parameters/ SscReportingAccount (Process Defaults)	Reporting account code to be specified for site
		SuppressSubstitutedMessages	IDORequest/ Parameters/ SscSuppressSubstituted Messages (Process Defaults)	Default Y - Yes
		SuspenseAccount	IDORequest/ Parameters/ SscSuspenseAccount (Process Defaults)	Suspense account code to be specified for site
		TransactionAmount Account	IDORequest/ Parameters/ SscTransactionAmount Account (Process Defaults)	Transaction account code to be specified for site

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AccountCode	acct	Trim and uppercase
		AccountingPeriod	cur_per fiscal_year	current period formatted to three digits followed by fiscal year
		AnalysisCode1	acct_unit1	If blank then NOT USED otherwise Trim and uppercase
		AnalysisCode2	acct_unit2	If blank then NOT USED otherwise Trim and uppercase
		AnalysisCode3	acct_unit3	If blank then NOT USED otherwise Trim and uppercase
		AnalysisCode4	acct_unit4	If blank then NOT USED otherwise Trim and uppercase
		AnalysisCode10	trans_num	Trim and uppercase
		PermanentPosting Reference	trans_num	
		BaseAmount	dom_amount	
		ConversionRate	dom_amount exch_rate	If dom_amount != 0 then exch_rate otherwise 0
		DebitCredit	dom_amount	If dom_amount >= 0 then D otherwise C
		CurrencyCode	curr_code	
		Description	ref	
		JournalSource	ref_type	
		JournalType	from_id	Trim and uppercase first 5 characters
		TransactionAmount	for_amount	
		TransactionDate	trans_date	Format from YYYY-MM-DD to DDMMYYYY
		TransactionReference	control_number	
		DetailLad/ AccountCode	acct	Trim and uppercase

Area	Occurrence	Target	Source	Function/ Comment
SSC/Payload/ Ledger/Line	One per source "Item" in IDORequest/ RequestHeader/ RequestData/ UpdateCollection/ Items	AccountingPeriod	control_period control_year	control period formatted to three digits followed by control year
		GeneralDescription3	trans_num	

Index

Z