Deltek

Deltek Time & Expense™8.0

Technical Guide

September 1, 2020

This edition published September 2020.

Contents

Overview

Adding Custom Notes to This Guide

If you would like to add custom notes to this guide that are specific to your company, Adobe® Reader® X provides this ability. If you do not already use Adobe Reader X, you can download it here free from Adobe.

To add a custom note using Adobe Reader X, complete the following steps:

1. On the Reader toolbar, click **Comment** at the far right.
2. In the **Annotations** pane that displays, click **Sticky Note**. The cursor changes to match the button.
3. Position the cursor at the location in the guide where you want the note to appear, and click. A note icon is inserted at the location and a text box pops up.
4. Enter your information in the text box.
5. Continue adding notes as needed.
6. Save the document.

Note: Deltek recommends that you save the document to a slightly different filename so as to keep the original from being overwritten.

When reading the document, cursor over a note icon to see the information. Double-click a note icon to edit the information.

If You Need Assistance

If you need assista7nce installing, implementing, or using Deltek Time & Expense, Deltek makes a wealth of information and expertise readily available to you.

Customer Services

For over 30 years, Deltek has maintained close relationships with client firms, helping with their problems, listening to their needs, and getting to know their individual business environments. A full range of customer services has grown out of this close contact, including the following:

- Extensive self-support options through the Customer Care Connect Web portal.
- Phone and email support from Customer Care analysts
- Technical services
- Consulting services
- Custom programming
- Classroom, on-site, and Web-based training

Note: Find out more about these and other services from the Customer Care Connect site.

Customer Care Connect Site

The Deltek Customer Care Connect site is a support Web site for Deltek customers who purchase an Ongoing Support Plan (OSP).

The following are some of the many options that the Customer Care Connect site provides:

- Search for product documentation, such as release notes, install guides, technical information, online help topics, and white papers
- Ask questions, exchange ideas, and share knowledge with other Deltek customers through the Deltek Connect Customer Forums
- Access Cloud specific documents and forums
- Download the latest versions of your Deltek products
- Search Deltek's knowledge base
- Submit a support case and check on its progress
- Transfer requested files to a Customer Care analyst
- Subscribe to Deltek communications about your products and services
- Receive alerts of new Deltek releases and hot fixes
- Use Quick Chat to submit a question to a Customer Care analyst online

Note: For more information regarding Deltek Customer Care Connect, refer to the online help available from the Web site.

Access Customer Care Connect

To access the Customer Care Connect site, complete the following steps:

1. Go to http://support.deltek.com.
2. Enter your Customer Care Connect **Username** and **Password**.
3. Click **Log In.**

> **Note:** If you do not have a username and password for the Customer Care Connect site, contact your firm's Deltek Time & Expense Administrator.
>
> If you forget your username or password, you can click the **Account Assistance** button on the login screen for help.

Introduction

Welcome to Deltek Time & Expense, one of the premier Web-based time collection and expense collection systems available today. It provides the power and flexibility to support companies large or small regardless of financial, payroll, or Enterprise Resource Planning (ERP) system used.

This Technical Guide is a valuable reference document that covers the technical aspects of the system. It covers everything from the system architecture to the database with in-depth discussions on many specific topics such as interfacing. As you implement or use Deltek Time & Expense, remember these pointers:

- The online help is a valuable resource once the system is operational. Its topics help employees, supervisors, administrators, and other users get the most out of the system.
- The Getting Started Guides provide valuable information on implementing the system and on its underlying concepts. The main audience for the Getting Started Guides consists of administrators and implementers. Please note that separate Getting Started Guides exist for Deltek Time and Deltek Expense.
- You can find technical details regarding the installation of the system in the installation guide, including information on Lightweight Directory Access Protocol (LDAP) and Single Sign-On (SSO).

Note: Single Sign On (SSO) Functionality is a feature that you can choose to install with your Deltek T&E instance. This functionality relies on third party applications (for example, Microsoft IIS & ISA, or BEA Weblogic) to allow users to pass the necessary information through the infrastructure to facilitate the SSO capability. This is not a feature that Deltek installs with the core application.

Depending on the characteristics and complexities of the environment that supports your Deltek applications, this functionality can be challenging to install and might take hours or days to complete. Deltek Technical Consulting Services can assist you in managing a multi-vendor team that may be required to achieve a successful installation.

If you experience difficulties implementing this functionality, contact Deltek Customer Care. They will put you in contact with a Deltek Consulting Manager, who will outline your options and help you engage a qualified technical consultant to work with your internal IT organization.

Custom Stored Procedures

Overview

Time & Expense provides the capability to call specialized stored procedures to invoke custom logic such as additional validation routines or database updates. These stored procedures include two types: “Pre” and “Post.”

“Pre” Stored Procedures

“Pre” stored procedures are called prior to a standard process or database update. They enable you to customize the validation process by adding validation logic in the stored procedure. Any errors or warnings resulting from this additional validation are automatically included with the validation results reported by the standard system validations.

Note: Warnings provide information but do not stop processing, while errors prevent the timesheet or expense report from being signed or saved to the database until corrected.

These are the currently available “Pre” stored procedures:

- Timesheet Pre-Save
- Timesheet Pre-Sign
- Timesheet Pre-Approve
- Expense Report Pre-Sign
- Expense Report Pre-Save
- Expense Authorization Pre-Sign
- Expense Authorization Pre-Save

“Post” Stored Procedures

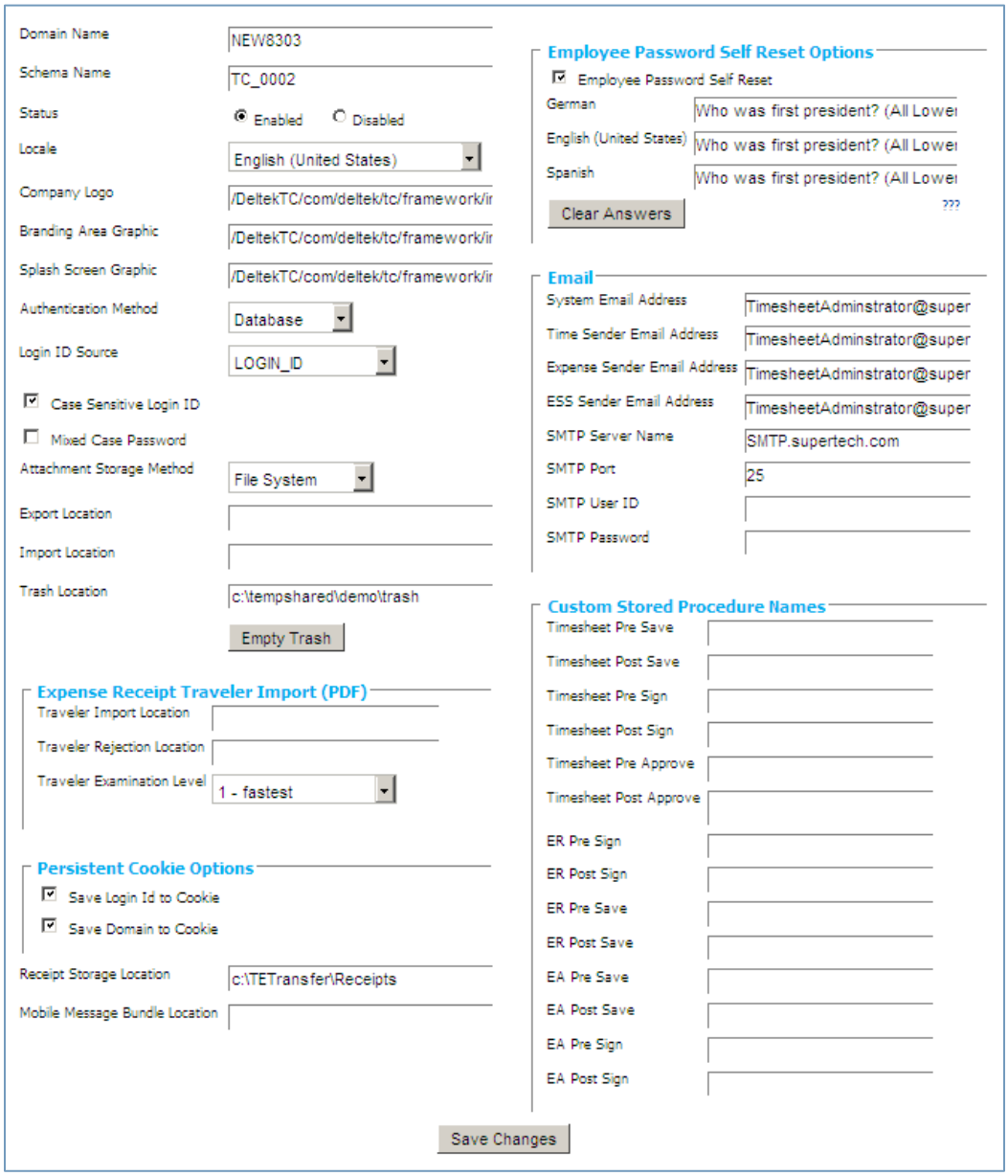
“Post” stored procedures are called after a process or database update occurs. These stored procedures are normally used to do additional database updates after standard processing is completed.

These are the currently available “Post” stored procedures:

- Timesheet Post-Save
- Timesheet Post-Sign
- Timesheet Post-Approve
- Expense Report Post-Save
- Expense Report Post-Sign
- Expense Authorization Post-Save
- Expense Authorization Post-Sign

Setup

To set up Time & Expense to use a custom stored procedure, use the Technical Console to identify which type of stored procedure you want and specify the name of the stored procedure. You may use one or all of the callouts. To identify the stored procedure to the callout, use the Technical Console for the domains for which you want the logic to be executed.



The name of the stored procedure must adhere to the naming conventions supported by your database software. Do not enter the stored procedure name in the Technical Console until the stored procedure has been written, compiled, and stored in the appropriate database and schema. After you enter the name and save it in the Technical Console, you must restart the server.

The actual writing of the stored procedure can be handled in any of three ways:

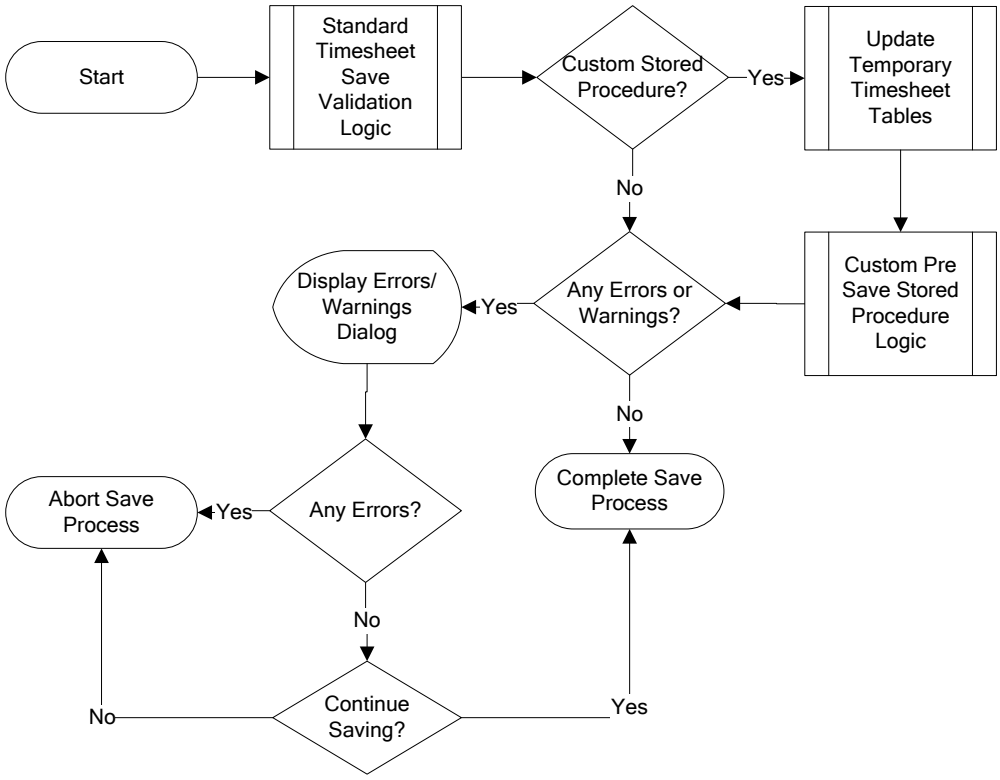
- If you have database IT staff familiar with PL/SQL (Oracle) or Transact-SQL (Microsoft SQL Server), you can develop the stored procedures yourself.
- You can arrange to have Deltek, Inc. create the store procedure for you under a custom programming contract.
- You can engage a third-party consultant to create the stored procedure for you.

Regardless of your approach, you need to develop detailed requirements for what you want the stored procedure to do.

Timesheet Pre-Save

Logic Flow

If a Timesheet Pre-Save stored procedure is specified in the Technical Console for the domain, it is called after the standard validation routine before a timesheet save. If a Timesheet Pre-Save procedure exists, the logic flow is as shown in the following illustration:



When the user clicks **Save**, the following occurs:

- Standard Timesheet Save Validation Logic
- Custom Stored Procedure
- Update Temporary Timesheet Tables
- Custom Stored Procedure Logic
- Any Errors or Warnings?
- Display Errors/Warnings Dialog

- Complete Save Process

Standard Timesheet Save Validation Logic

This is the standard Time & Expense logic for validating timesheet data before saving it. The validations performed vary depending on your configuration. These are the possible validations:

- Start/End Date Validation
- Comment Required Validation
- Future Hours Rules
- Inactive Charge Validation
- Leave Balance Validation

Custom Stored Procedure?

The system checks to see if a Timesheet Pre-Save stored procedure has been identified in the Technical Console for the data domain.

Update Temporary Timesheet Tables

For each timesheet database table, a temporary version of the table exists. If a stored procedure is to be called, the system inserts records into the temporary table so the stored procedure can use them to see what the employee's timesheet will look like after the save. These records are deleted after the save is completed (or is stopped because of an error). These tables are updated:

- TEMP_TS
- TEMP_TS_LINE
- TEMP_TS_CELL
- TEMP_TS_DAY
- TEMP_TS_START_STOP

Custom Stored Procedure Logic

The system calls the customized stored procedure containing your special logic. When it does, it passes the following information in the stored procedure call:

- Employee ID of the user saving the timesheet (Alphanumeric 20)
- Employee ID of the employee for whom the timesheet was created (Alphanumeric 20)
- Year of the timesheet period (Alphanumeric 4)
- Period of the timesheet period (Alphanumeric 3)
- Timesheet schedule code (Alphanumeric 10)
- Timesheet class of the timesheet employee (Alphanumeric 20)
- Work schedule code of the timesheet employee (Alphanumeric 10)
- Country code of the user saving the timesheet (Alphanumeric 8)
- Language code of the user saving the timesheet (Alphanumeric 2)

The above parameters tell the stored procedure what timesheet is about to be saved, who is saving it, and what locale (Country/Language) is associated with that user. The stored procedure must then return the following:

- Error or warning indicator (E or W) (Alphanumeric 1)
- Error description (Alphanumeric 2000)

The stored procedure can return multiple errors and warnings.

Any Errors or Warnings?

The system checks to see if either the standard validation logic or the custom stored procedure logic has reported any errors or warnings.

Display Errors/Warnings Dialog

The system displays an Errors/Warnings dialog box if there are any errors or warnings. If there are any errors, the user cannot save the timesheet. If there are only warnings, they can choose whether or not to save the timesheet. If there are no errors or warnings, the Errors/Warnings dialog box does not open.

Complete Save Process

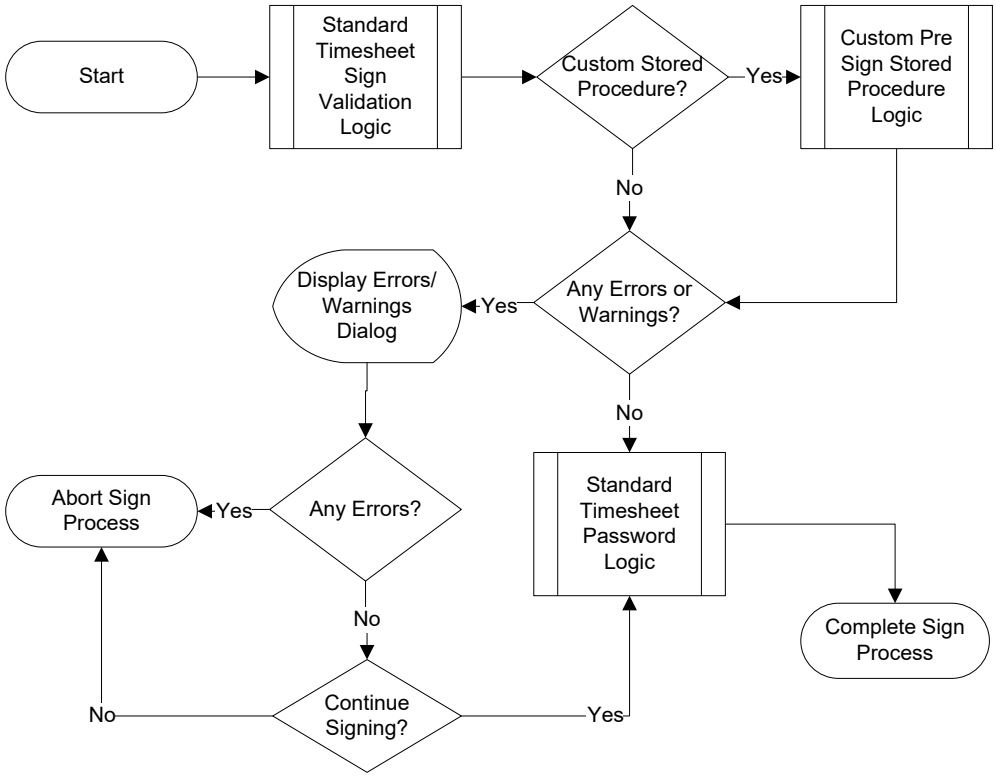
The system performs these additional steps to complete the save process:

- Delete records from temporary tables
- Carry out revision audit logic
- Update timesheet tables
- Call custom Timesheet Post-Save stored procedure (if applicable)
- Call workflow engine

Timesheet Pre-Sign

Logic Flow

If a Timesheet Pre-Sign stored procedure is specified in the Technical Console for the domain, it is called after the standard validation routine before timesheet signing. If a Timesheet Pre-Sign procedure exists, the logic flow is as shown in the following illustration.



When the user initiates the signature process, the following occurs:

- Standard Timesheet Sign Validation Logic
- Custom Stored Procedure?
- Custom Stored Procedure Logic
- Any Errors or Warnings
- Display Errors/Warnings Dialog
- Standard Timesheet Password Logic
- Complete Sign Process

Standard Timesheet Sign Validation Logic

This is the standard Time & Expense logic for validating timesheet data before completing the signature process. The validations performed vary depending on your configuration. These are the possible validations:

- Daily Regular Hour Rules
- Weekly Regular Hour Rules
- Timesheet Period Hour Rules
- UDT01 Regular Hour Rules
- Time In/Out Rules
- Overtime Rules
- Holiday Rules

- Vacation Rules

Custom Stored Procedure?

The system checks to see if a Timesheet Pre-Sign stored procedure has been identified in the Technical Console for the data domain.

Custom Stored Procedure Logic

The system calls the customized stored procedure containing your special logic. When it does, it passes the following information in the stored procedure call:

- Employee ID of the user signing the timesheet (Alphanumeric 20)
- Employee ID of the employee for whom the timesheet was created (Alphanumeric 20)
- Year of the timesheet period (Alphanumeric 4)
- Period of the timesheet period (Alphanumeric 3)
- Timesheet schedule code (Alphanumeric 10)
- Timesheet class of the timesheet employee (Alphanumeric 20)
- Work schedule code of the timesheet employee (Alphanumeric 10)
- Country code of the user saving the timesheet (Alphanumeric 8)
- Language code of the user saving the timesheet (Alphanumeric 2)

The above parameters tell the stored procedure what timesheet is about to be signed, who is signing it, and what locale (Country/Language) is associated with that user. The stored procedure must then return the following:

- Error or warning indicator (E or W) (Alphanumeric 1)
- Error description (Alphanumeric 2000)

The stored procedure can return multiple errors and warnings.

Any Errors or Warnings?

The system checks to see if either the standard validation logic or the custom stored procedure logic has reported any errors or warnings.

Display Errors/Warnings Dialog

The system displays an Errors/Warnings dialog box if there are any errors or warnings. If there are any errors, the user cannot sign the timesheet. If there are only warnings, they can choose whether or not to sign the timesheet. If there are no errors or warnings, the Errors/Warnings dialog box does not open.

Standard Timesheet Password Logic

Whether the user must provide a password to complete the timesheet signing is based on system configuration. If it is required, a dialog box opens, and they must enter a valid password.

Complete Sign Process

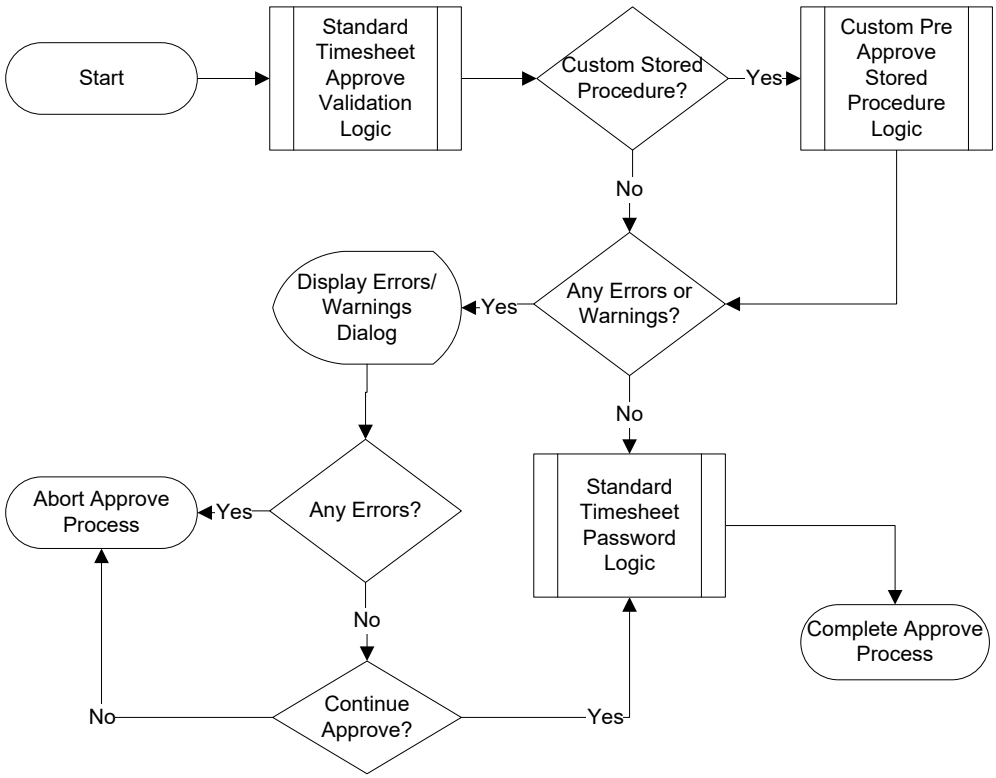
The system performs these additional steps to complete the signing process:

- Update timesheet tables
- Call custom Timesheet Post-Sign stored procedure (if applicable)
- Call workflow engine

Timesheet Pre-Approve

Logic Flow

If a Timesheet Pre-Approve stored procedure is specified in the Technical Console for the domain, it is called after the standard validation routine before timesheet approval. If a Timesheet Pre-Approve procedure exists, the logic flow is as shown in the following illustration.



When the user initiates the approval process, the following occurs:

- Standard Timesheet Approval Validation Logic
- Custom Stored Procedure?
- Custom Stored Procedure Logic
- Any Errors or Warnings?
- Display Errors/Warnings Dialog
- Standard Timesheet Password Logic
- Complete Approval Process

Standard Timesheet Approve Validation Logic

This is the standard Time & Expense logic for validating timesheet data before completing the approval process. Currently, no standard validations are performed when a timesheet is approved.

Custom Stored Procedure?

The system checks to see if a Timesheet Pre-Approve stored procedure has been identified in the Technical Console for the data domain.

Custom Stored Procedure Logic

The system calls the customized stored procedure containing your special logic. When it does, it passes the following information in the stored procedure call:

- Employee ID of the user approving the timesheet (Alphanumeric 20)
- Employee ID of the employee for whom the timesheet was created (Alphanumeric 20)
- Year of the timesheet period (Alphanumeric 4)
- Period of the timesheet period (Alphanumeric 3)
- Timesheet schedule code (Alphanumeric 10)
- Timesheet class of the timesheet employee (Alphanumeric 20)
- Work schedule code of the timesheet employee (Alphanumeric 10)
- Country code of the user saving approving the timesheet (Alphanumeric 8)
- Language code of the user saving approving the timesheet (Alphanumeric 2)

The above parameters tell the stored procedure what timesheet is about to be approved, who is approving it, and what locale (Country/Language) is associated with that user. The stored procedure must then return the following:

- Error or warning indicator (E or W) (Alphanumeric 1)
- Error description (Alphanumeric 2000)

The stored procedure can return multiple errors and warnings.

Any Errors or Warnings?

The system checks to see if either the standard validation logic or the custom stored procedure logic has reported any errors or warnings.

Display Errors/Warnings Dialog

The system displays an Errors/Warnings dialog box if there are any errors or warnings. If there are any errors, the user cannot approve the timesheet. If there are only warnings, they can choose whether or not to approve the timesheet. If there are no errors or warnings, the Errors/Warnings dialog box does not open.

Standard Timesheet Password Logic

Whether the user must provide a password to complete the timesheet approval is based on system configuration. If it is required, a dialog box opens, and they must enter a valid password.

Complete Approve Process

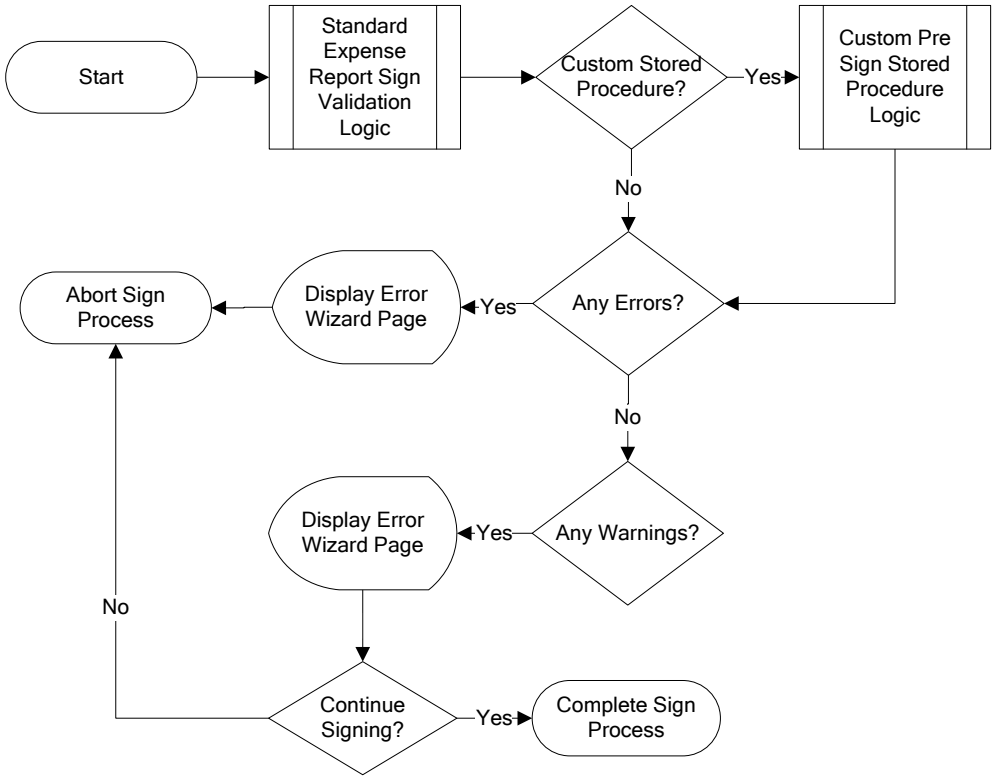
The system performs these additional steps to complete the approval process:

- Update timesheet tables
- Call custom Timesheet Post-Approve stored procedure (if applicable)
- Call workflow engine

Expense Report Pre-Sign

Logic Flow

If an Expense Report Pre-Sign stored procedure is specified in the Technical Console for the domain, it is called after the standard validation routine preceding the signing of an expense report. If an Expense Report Pre-Sign procedure exists, the logic flow is as follows:



When the user initiates the signature process, the following occurs:

- Standard Expense Report Sign Validation Logic
- Custom Stored Procedure?
- Custom Stored Procedure Logic
- Any Errors?
- Any Warnings?
- Complete Sign Process

Standard Expense Report Sign Validation Logic

This is the standard Time & Expense logic for validating expense report data before completing the signature process. The validation performed is Incomplete Child Expenses.

Custom Stored Procedure?

The system checks to see if an Expense Report Pre-Sign stored procedure has been identified in the Technical Console for the data domain.

Custom Stored Procedure Logic

The system calls the customized stored procedure containing your special logic. When it does, it passes the following information in the stored procedure call:

- Expense report ID (Alphanumeric 10)
- Employee ID of the user signing the expense report (Alphanumeric 20)
- Country code of the user signing the expense report (Alphanumeric 8)
- Language code of the user signing the expense report (Alphanumeric 2)

The above parameters tell the stored procedure what expense report is about to be signed, who is signing it, and what locale (Country/Language) is associated with that user. The stored procedure must then return the following:

- Error or warning indicator (E or W) (Alphanumeric 1)
- Error description (Alphanumeric 2000)

The stored procedure can return multiple errors and warnings.

Any Errors?

The system displays an Errors page if either the standard validation logic or the custom stored procedure logic reports any errors. If there are errors, the user cannot sign the expense report.

Any Warnings?

The system displays a Warnings page if either the standard validation logic or the custom stored procedure logic reports any warnings. If there are warnings but no errors, they can choose whether or not to sign the expense report.

Complete Sign Process

The system performs these additional steps to complete the signing process:

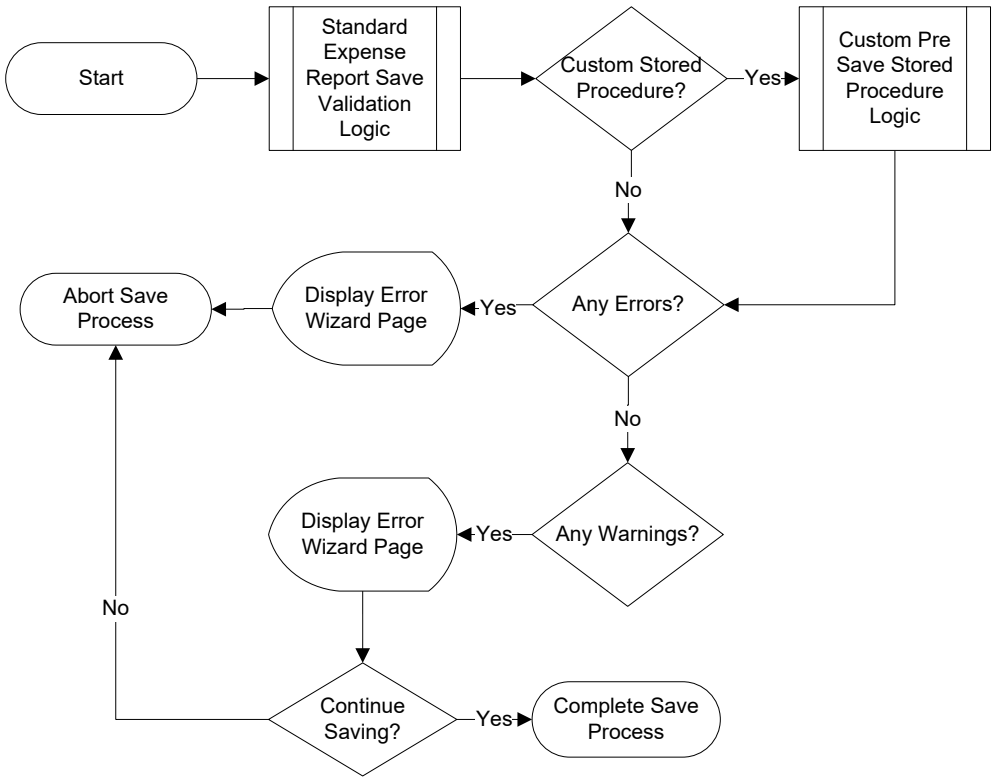
- User-directed workflow
- Receipt information gathering
- Company due processing
- Password logic
- Workflow engine
- Receipt traveler print

- Custom Expense Report Post-Sign stored procedure (if applicable)

Expense Report Pre-Save

Logic Flow

If an Expense Report Pre-Save stored procedure is specified in the Technical Console for the domain, it is called after the standard validation routine preceding the saving of expense information when **Finish** is clicked in the expense report wizard. If an Expense Pre-Save procedure exists, the logic flow is as follows:



When the user clicks **Finish**, the following occurs:

- Standard Expense Report Wizard Save Validation Logic
- Custom Stored Procedure?
- Update Temporary Expense Report Tables
- Custom Stored Procedure Logic
- Any Errors?
- Any Warnings?
- Complete Save Process

Standard Expense Report Wizard Save Validation Logic

This is the standard Time & Expense logic for validating expense report data before saving it. Currently, there are no standard validations performed.

Custom Stored Procedure?

The system checks to see if an Expense Pre-Save stored procedure has been identified in the Technical Console for the data domain.

Update Temporary Expense Report Tables

For each expense report database table, a temporary version of the table exists. If a stored procedure is to be called, the system inserts records into the temporary table so the stored procedure can use them to see what the employee's expense report will look like after the save. These records are deleted after the save process completes (or is stopped because of an error).

These tables are updated:

- TEMP_EXP_RPT
- TEMP_EXP_RPT_EXPENSE
- TEMP_EXP_RPT_EXPENSE_CHARGE
- TEMP_EXP_RPT_EXPENSE_DAY
- TEMP_EXP_RPT_EXPENSE_OTHER
- TEMP_EXP_RPT_TASKS
- TEMP_EXP_RPT_TASKS_EMPL

Custom Stored Procedure Logic

The system calls the customized stored procedure containing your special logic. When it does, it passes the following information in the stored procedure call:

- Expense report ID (Alphanumeric 10)
- Employee ID of the user saving the expense report (Alphanumeric 20)
- Expense ID (Integer)
- Country code of the user saving the expense report (Alphanumeric 8)
- Language code of the user saving the expense report (Alphanumeric 2)

The above parameters tell the stored procedure what expense report and expense is about to be saved, who is saving it, and what locale (Country/Language) is associated with that user. The stored procedure must then return the following:

- Error or warning indicator (E or W) (Alphanumeric 1)
- Error description (Alphanumeric 2000)
- The stored procedure can return multiple errors and warnings.

Any Errors?

The system displays an Errors page if either the standard validation logic or the custom stored procedure logic reports any errors. If there are errors, the user cannot save the expense report.

Any Warnings?

The system displays a Warnings page if either the standard validation logic or the custom stored procedure logic reports any warnings. If there are warnings but no errors, they can choose whether or not to save the expense report.

Complete Save Process

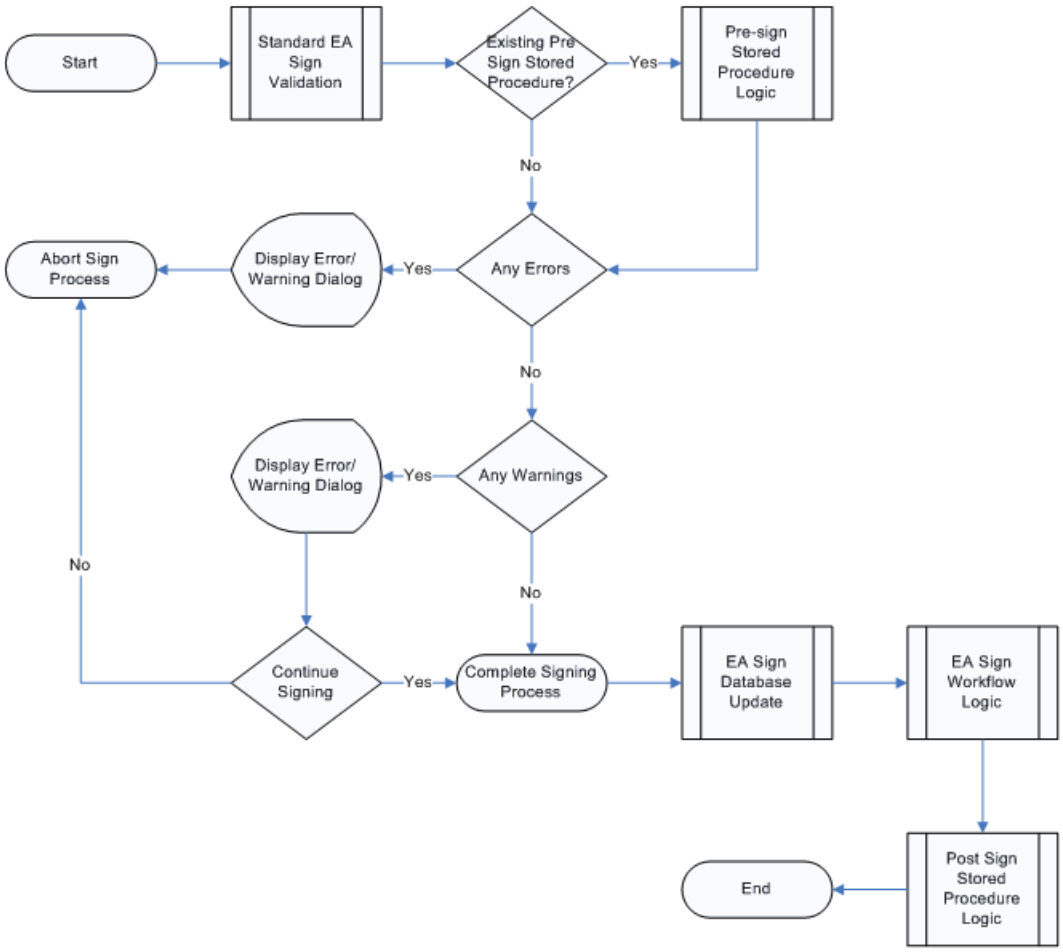
The system performs these additional steps to complete the save process:

- Carry out revision audit logic
- Delete records from temporary tables
- Update expense report tables
- Call workflow engine
- Call custom Expense Post-Save stored procedure (if applicable)

Expense Authorization Pre-Sign

Logic Flow

If an Expense Authorization Pre-Sign stored procedure is specified in the Technical Console for the domain, it is called after the standard validation routine preceding the signing of an Expense Authorization. If an Expense Authorization Pre-Sign procedure exists, the logic flow is as follows:



When the user initiates the signature process, the following occurs:

- Standard Expense Authorization Sign Validation Logic
- Custom Stored Procedure?
- Custom Stored Procedure Logic
- Any Errors?
- Any Warnings?
- Complete Sign Process

Custom Stored Procedure?

The system checks to see if an Expense Authorization Pre-Sign stored procedure has been identified in the Technical Console for the data domain.

Custom Stored Procedure Logic

The system calls the customized stored procedure containing your special logic. When it does, it passes the following information in the stored procedure call:

- Expense Authorization ID (Alphanumeric 10)

- Employee ID of the user signing the Expense Authorization (Alphanumeric 20)
- Country code of the user signing the Expense Authorization (Alphanumeric 8)
- Language code of the user signing the Expense Authorization (Alphanumeric 2)

The above parameters tell the stored procedure what Expense Authorization is about to be signed, who is signing it, and what locale (Country/Language) is associated with that user. The stored procedure must then return the following:

- Error or warning indicator (E or W) (Alphanumeric 1)
- Error description (Alphanumeric 2000)

The stored procedure can return multiple errors and warnings.

Any Errors?

The system displays an Errors page if either the standard validation logic or the custom stored procedure logic reports any errors. If there are errors, the user cannot sign the Expense Authorization.

Any Warnings?

The system displays a Warnings page if either the standard validation logic or the custom stored procedure logic reports any warnings. If there are warnings but no errors, they can choose whether or not to sign the Expense Authorization.

Complete Sign Process

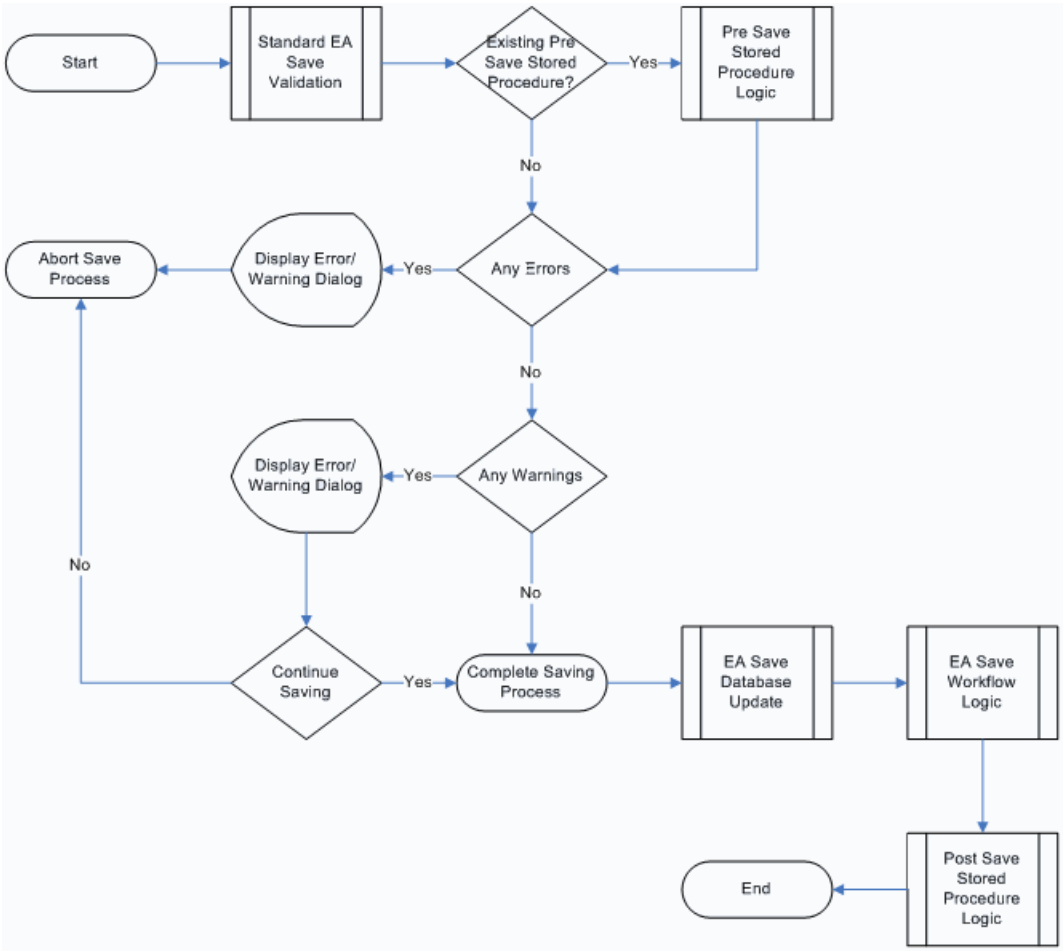
The system performs these additional steps to complete the signing process:

- User-directed workflow
- Cash Advance processing
- Password logic
- Workflow engine
- Custom Expense Authorization Post-Sign stored procedure (if applicable)

Expense Authorization Pre-Save

Logic Flow

If an Expense Authorization Pre-Save stored procedure is specified in the Technical Console for the domain, it is called after the standard validation routine preceding the saving of expense authorization information when **Finish** is clicked in the Expense Authorization wizard. If an Expense Authorization Pre-Save procedure exists, the logic flow is as follows:



Standard Expense Authorization Wizard Save Validation Logic

This is the standard Time & Expense logic for validating Expense Authorization data before saving it. Currently, there are no standard validations performed.

Custom Stored Procedure?

The system checks to see if an Expense Pre-Save stored procedure has been identified in the Technical Console for the data domain.

Update Temporary Expense Authorization Tables

For each Expense Authorization database table, a temporary version of the table exists. If a stored procedure is to be called, the system inserts records into the temporary table so the stored procedure can use them to see what the employee's Expense Authorization will look like after the save. These records are deleted after the save process completes (or is stopped because of an error).

These tables are updated:

- TEMP_EXP_AUTH
- TEMP_EXP_AUTH_EXPENSE

- TEMP_EXP_ AUTH_EXPENSE_CHARGE
- TEMP_EXP_ AUTH_EXPENSE_DAY
- TEMP_EXP_ AUTH_TASKS
- TEMP_EXP_ AUTH_TASKS_EMPL

Custom Stored Procedure Logic

The system calls the customized stored procedure containing your special logic. When it does, it passes the following information in the stored procedure call:

- Employee ID of the user saving the Expense Authorization (Alphanumeric 20)
- Expense Authorization ID (Alphanumeric 10)
- Expense ID (Integer)
- Country code of the user saving the Expense Authorization (Alphanumeric 8)
- Language code of the user saving the Expense Authorization (Alphanumeric 2)

The above parameters tell the stored procedure what Expense Authorization is about to be saved, who is saving it, and what locale (Country/Language) is associated with that user. The stored procedure must then return the following:

- Error or warning indicator (E or W) (Alphanumeric 1)
- Error description (Alphanumeric 2000)
- The stored procedure can return multiple errors and warnings.

Any Errors?

The system displays an Errors page if either the standard validation logic or the custom stored procedure logic reports any errors. If there are errors, the user cannot save the Expense Authorization.

Any Warnings?

The system displays a Warnings page if either the standard validation logic or the custom stored procedure logic reports any warnings. If there are warnings but no errors, they can choose whether or not to save the Expense Authorization.

Complete Save Process

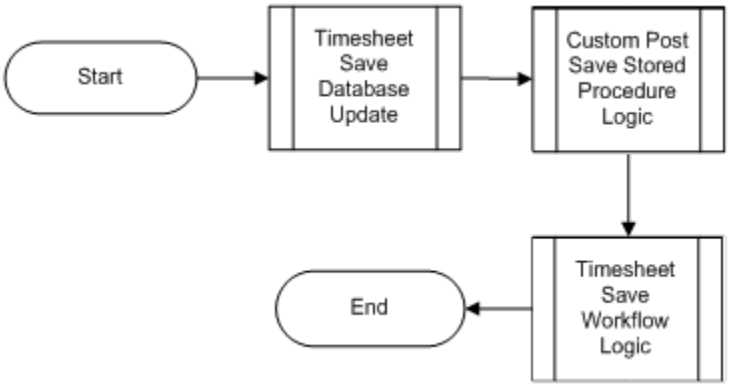
The system performs these additional steps to complete the save process:

- Carry out revision audit logic
- Delete records from temporary tables
- Update Expense Authorization tables
- Call workflow engine
- Call custom Expense Post-Save stored procedure (if applicable)

Timesheet Post Save

Logic Flow

If a stored procedure for the Timesheet Post-Save is specified in the Technical Console for the domain, it displays after a timesheet has been successfully saved. If a Timesheet Post-Save procedure exists, the logic flow is as follows:



After performing the database update, Time & Expense calls the workflow engine to initiate any workflow processing. It then checks to see if a Timesheet Post-Save stored procedure has been identified in the Technical Console for the data domain. If one exists, the system calls the stored procedure. When it does, it passes the following information in the stored procedure call:

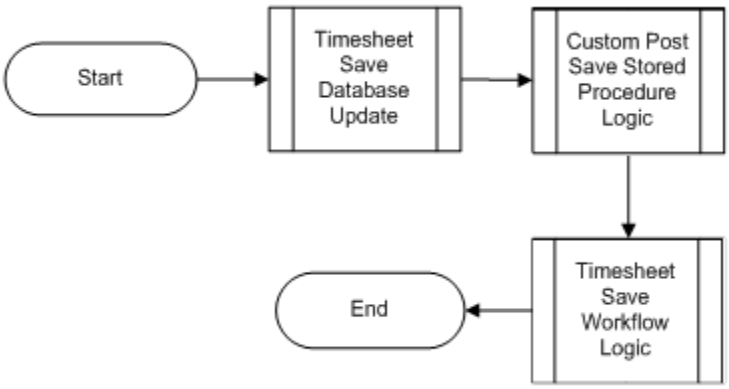
- Employee ID of the user saving the timesheet (Alphanumeric 20)
- Employee ID of the employee for whom the timesheet was created (Alphanumeric 20)
- Year of the timesheet period (Alphanumeric 4)
- Period of the timesheet period (Alphanumeric 3)
- Timesheet schedule code (Alphanumeric 10)
- Timesheet class of the timesheet employee (Alphanumeric 20)
- Work schedule code of the timesheet employee (Alphanumeric 10)

The above parameters tell the stored procedure which timesheet was saved, who saved it, and what locale (Country/Language) is associated with that user. The stored procedure must then return an empty result set containing two columns: one alphanumeric (1) and the other alphanumeric (2000). (If the result set contains data, that data is ignored.)

Timesheet Post-Sign

Logic Flow

If a stored procedure for a Timesheet Post-Sign is specified in the Technical Console for the domain, it displays after a timesheet has been successfully signed. If a Timesheet Post-Sign procedure exists, the logic flow is as follows:



After performing the database update, Time & Expense calls the workflow engine to initiate any workflow processing. It then checks to see if a Timesheet Post-Sign stored procedure has been identified in the Technical Console for the data domain. If one exists, the system calls the stored procedure. When it does, it passes the following information in the stored procedure call:

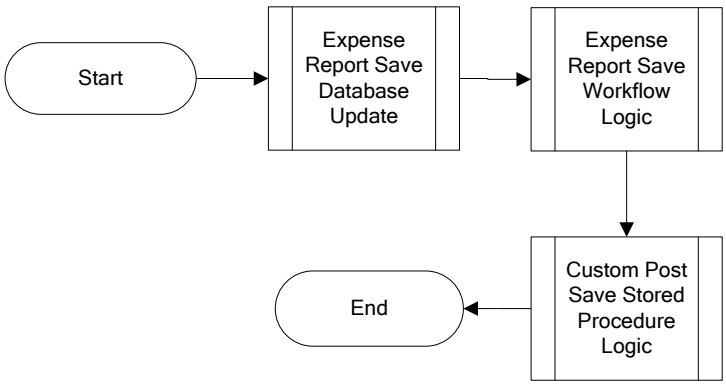
- Employee ID of the user saving the timesheet (Alphanumeric 20)
- Employee ID of the employee for whom the timesheet was created (Alphanumeric 20)
- Year of the timesheet period (Alphanumeric 4)
- Period of the timesheet period (Alphanumeric 3)
- Timesheet schedule code (Alphanumeric 10)
- Timesheet class of the timesheet employee (Alphanumeric 20)
- Work schedule code of the timesheet employee (Alphanumeric 10)

The above parameters tell the stored procedure which timesheet was signed, who signed it, and what locale (Country/Language) is associated with that user. The stored procedure must then return an empty result set containing two columns: one alphanumeric (1) and the other alphanumeric (2000). (If the result set contains data, that data is ignored.)

Expense Report Post-Save

Logic Flow

If a stored procedure for an Expense Report Post-Save is specified in the Technical Console for the domain, it is called after an expense report has been successfully saved. If an Expense Post-Save procedure exists, the logic flow is as follows:



After performing the database update, Time & Expense calls the workflow engine to initiate any workflow processing. It then checks to see if an Expense Post-Save stored procedure has been identified in the Technical Console for the data domain. If one exists, the system calls the stored procedure. When it does, it passes the following information in the stored procedure call:

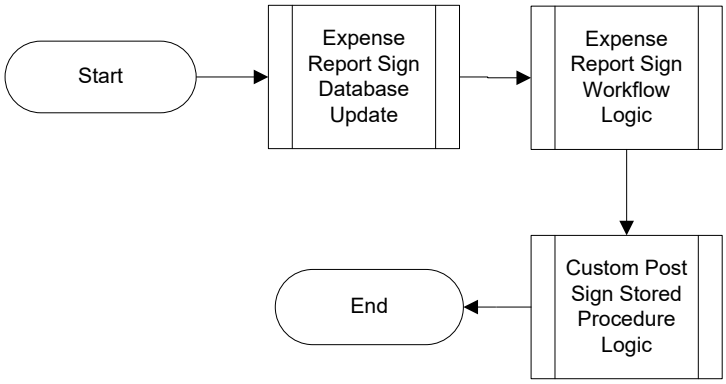
- Employee ID of the user saving the expense report (Alphanumeric 20)
- Expense report ID (Alphanumeric 10)
- Expense ID (Integer)
- Country code of the user saving the expense report (Alphanumeric 8)
- Language code of the user saving the expense report (Alphanumeric 2)

The above parameters tell the stored procedure what expense report and expense was saved, who saved them, and what locale (Country/Language) is associated with that user. The stored procedure must then return an empty result set containing two columns: one alphanumeric (1) and the other alphanumeric (2000). (If the result set contains data, that data is ignored.)

Expense Report Post-Sign

Logic Flow

If a stored procedure for an Expense Report Post-Sign is specified in the Technical Console for the domain, it is called after an expense report has been successfully signed. If an Expense Report Post-Sign procedure exists, this logic flow follows:



After performing the database update, Time & Expense calls the workflow engine to initiate any workflow processing. It then checks to see if an Expense Report Post-Sign stored procedure has been identified in the Technical Console for the data domain. If one exists, the system calls the stored procedure. When it does, it passes the following information in the stored procedure call:

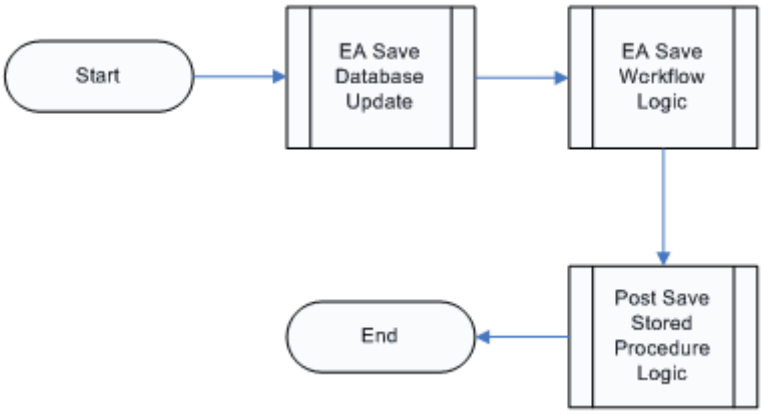
- Expense report ID (Alphanumeric 10)
- Employee ID of the user signing the expense report (Alphanumeric 20)
- Country code of the user signing the expense report (Alphanumeric 8)
- Language code of the user signing the expense report (Alphanumeric 2)

The above parameters tell the stored procedure what expense report was signed, who signed it, and what locale (Country/Language) is associated with that user. The stored procedure must then return an empty result set containing two columns: one alphanumeric (1) and the other alphanumeric (2000). (If the result set contains data, that data is ignored.)

Expense Authorization Post-Save

Logic Flow

If a stored procedure for an Expense Authorization Post-Save is specified in the Technical Console for the domain, it is called after an Expense Authorization has been successfully saved. If an Expense Authorization Post-Save procedure exists, the logic flow is as follows:



After performing the database update, Time & Expense calls the workflow engine to initiate any workflow processing. It then checks to see if an Expense Authorization Post-Save stored procedure has been identified in the Technical Console for the data domain. If one exists, the system calls the stored procedure. When it does, it passes the following information in the stored procedure call:

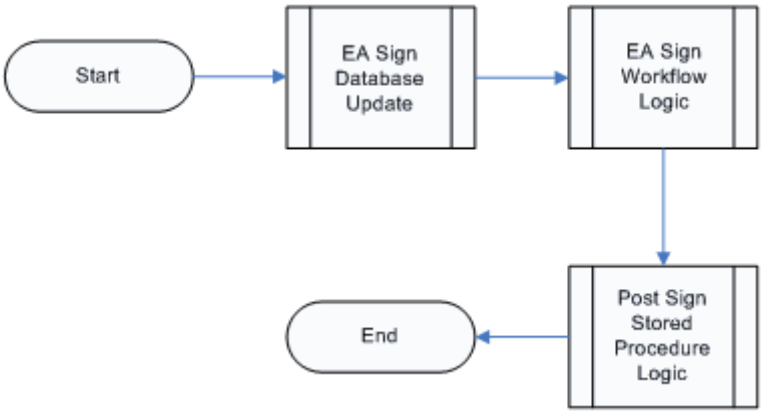
- Employee ID of the user saving the Expense Authorization (Alphanumeric 20)
- Expense ID (Integer)
- Country code of the user saving the Expense Authorization (Alphanumeric 8)
- Language code of the user saving the Expense Authorization (Alphanumeric 2)

The above parameters tell the stored procedure what Expense Authorization and expense was saved, who saved them, and what locale (Country/Language) is associated with that user. The stored procedure must then return an empty result set containing two columns: one alphanumeric (1) and the other alphanumeric (2000). (If the result set contains data, that data is ignored.)

Expense Authorization Post-Sign

Logic Flow

If a stored procedure for an Expense Authorization Post-Sign is specified in the Technical Console for the domain, it is called after an Expense Authorization has been successfully signed. If an Expense Authorization Post-Sign procedure exists, this logic flow follows:



After performing the database update, Time & Expense calls the workflow engine to initiate any workflow processing. It then checks to see if an Expense Authorization Post-Sign stored procedure has been identified in the Technical Console for the data domain. If one exists, the system calls the stored procedure. When it does, it passes the following information in the stored procedure call:

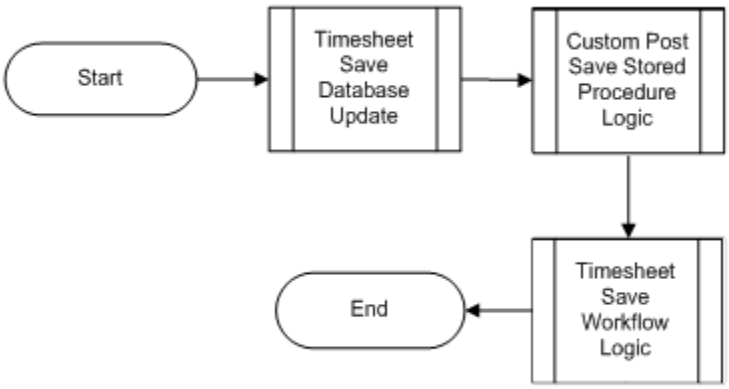
- Expense Authorization ID (Alphanumeric 10)
- Employee ID of the user signing the Expense Authorization (Alphanumeric 20)
- Country code of the user signing the Expense Authorization (Alphanumeric 8)
- Language code of the user signing the Expense Authorization (Alphanumeric 2)

The above parameters tell the stored procedure what Expense Authorization was signed, who signed it, and what locale (Country/Language) is associated with that user. The stored procedure must then return an empty result set containing two columns: one alphanumeric (1) and the other alphanumeric (2000). (If the result set contains data, that data is ignored.)

Timesheet Post-Approve

Logic Flow

If a stored procedure for a Timesheet Post-Approve is specified in the Technical Console for the domain, it is called after a timesheet has been successfully approved. If a Timesheet Post-Approve procedure exists, this logic flow follows:



After performing the database update, Time & Expense calls the workflow engine to initiate any workflow processing. It then checks to see if a Timesheet Post-Approve stored procedure has been identified in the Technical Console for the data domain. If one exists, the system calls the stored procedure. When it does, it passes the following information in the stored procedure call:

- Employee ID of the user who approved the timesheet (Alphanumeric 20)
- Employee ID of the employee for whom the timesheet was created (Alphanumeric 20)
- Year of the timesheet period (Alphanumeric 4)
- Period of the timesheet period (Alphanumeric 3)
- Timesheet schedule code (Alphanumeric 10)
- Timesheet class of the timesheet employee (Alphanumeric 20)
- Work schedule code of the timesheet employee (Alphanumeric 10)

The above parameters tell the stored procedure what timesheet was approved, who approved it, and what locale (Country/Language) is associated with that user. The stored procedure must then return an empty result set containing two columns: one alphanumeric (1) and the other alphanumeric (2000). (If the result set contains data, that data is ignored.)

Oracle Technical Notes

Understanding Result Sets

All procedures return a result set with either empty results or one or more rows. These characteristics are described below.

Note: The result set is a reference cursor (refcursor object type).

Empty Results

- An empty result set for “Pre” procedures means no warnings or errors were found.
- An empty result set for “Post” procedures means the procedure ended normally.

One or More Rows

- A result set containing rows for ”Pre” procedures means warnings or errors were found. Each row is marked as an error or warning.
- A result set containing rows for “Post” procedures is ignored by the application and processed as if the procedure completed normally.

Creating Oracle Stored Procedures

Here are some guidelines for creating Oracle stored procedures:

- Set default values when the variable/object is declared.
- Use date format for Julian conversions.
- Use as few cursors as possible and do as much “work” with each as possible.
- Use explain plan or another tool to tune queries you create. Include new indexes if you find your access methods require them.

Timesheet “Pre” Stored Procedures

Input Parameters

When a timesheet-related “Pre” stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the system user who is accessing the timesheet. This could be the employee for the displayed timesheet, a supervisor, or an administrator.
2	P_TS_EMPL_ID	The employee ID of the employee for whom the timesheet was created. This may be identical to the P_LOGIN_EMPL_ID.
3	P_YEAR_NO_CD	The year number code for the timesheet.
4	P_PERIOD_NO_CD	The period number code for the timesheet.
5	P_TS_SCHEDULE_CD	The timesheet schedule code for the timesheet.
6	P_CLASS_CD	The timesheet class code for the employee for whom the timesheet was created.
7	P_WORK_SCEHDULE_CD	The work schedule code for the employee for whom the timesheet was created.
8	P_COUNTRY_CD	The country code of the user.
9	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

The “Pre” stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done in the final reference cursor.

Be aware that some versions of Oracle have limited capability to sort reference cursor objects.

Timesheet “Post” Stored Procedures

Input Parameters

When a timesheet-related “Post” stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the system user who is accessing the timesheet. This could be the employee for the displayed timesheet, a supervisor, or an administrator.
2	P_TS_EMPL_ID	The employee ID of the employee for whom the timesheet was created. This may be identical to the P_LOGIN_EMPL_ID.
3	P_YEAR_NO_CD	The year number code for the timesheet.
4	P_PERIOD_NO_CD	The period number code for the timesheet.
5	P_TS_SCHEDULE_CD	The timesheet schedule code for the timesheet.
6	P_CLASS_CD	The timesheet class code for the employee for whom the timesheet was created.
7	P_WORK_SCHEDULE_CD	The work schedule code for the employee for whom the timesheet was created.

Return Values

The “Post” stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they will always return a null result set. The column definitions are the same as those for a “Pre” stored procedure:

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

Expense Report Pre-Save Stored Procedures

Input Parameters

When an Expense Report Pre-Save stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the user who is accessing the expense report. This may be the employee for the displayed expense report, a supervisor, or an administrator.
2	P_EXP_RPT_ID	The expense report ID. This is the unique identifier for the expense report.
3	P_EXP_ID	The expense ID. This is the unique identifier for the individual expense.
4	P_COUNTRY_CD	The country code of the user.
5	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

The Expense Pre-Save stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error

	Name	Description
		W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done in the final reference cursor.

Be aware that some versions of Oracle have limited capability to sort reference cursor objects.

Expense Report Post-Save Stored Procedures

Input Parameters

When an Expense Report Post-Save stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the user accessing the expense report. This may be the employee for the displayed expense report, a supervisor, or an administrator.
2	P_EXP_RPT_ID	The expense report ID. This is the unique identifier for the expense report.
3	P_EXP_ID	The expense ID. This is the unique identifier for the individual expense.
4	P_COUNTRY_CD	The country code of the user.
5	P_LANGUAGE_CD	The language code of the user.

Return Values

The Expense Report Post-Save stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they always return a null result set. The column definitions are the same as those for an Expense Pre-Save stored procedure.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

Expense Report Pre-Sign Stored Procedures

Input Parameters

When an Expense Report Pre-Sign stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_EXP_RPT_ID	The expense report ID. This is the unique identifier for the expense report.
2	P_LOGIN_EMPL_ID	The employee ID of the user who is accessing the expense report. This may be the employee for the displayed expense report, a supervisor, or an administrator.
3	P_COUNTRY_CD	The country code of the user.
4	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

Expense Report Pre-Sign stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done in the final reference cursor.

Be aware that some versions of Oracle have limited capability to sort reference cursor objects.

Expense Report Post-Sign Stored Procedures

Input Parameters

When an Expense Report Post-Sign stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_EXP_RPT_ID	The expense report ID. This is the unique identifier for the expense report.
2	P_LOGIN_EMPL_ID	The employee ID of the user accessing the expense report. This may be the employee for the displayed expense report, a supervisor, or an administrator.
3	P_COUNTRY_CD	The country code of the user.
4	P_LANGUAGE_CD	The language code of the user.

Return Values

The Expense Report Post-Sign stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they always return a null result set. The column definitions are the same as those for an Expense Report Pre-Sign stored procedure:

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

Expense Authorization Pre-Save Stored Procedures

Input Parameters

When an Expense Authorization Pre-Save stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the user who is accessing the Expense Authorization. This may be the employee for the displayed Expense Authorization, a supervisor, or an administrator.
2	P_ EA_ID	The Expense Authorization ID. This is the unique identifier for the Expense Authorization.
3	P_ EA_EXPENSE_ID	The expense ID. This is the unique identifier for the individual expense.
4	P_COUNTRY_CD	The country code of the user.
5	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

The Expense Authorization Pre-Save stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done in the final reference cursor.

Be aware that some versions of Oracle have limited capability to sort reference cursor objects.

Expense Authorization Post-Save Stored Procedures

Input Parameters

When an Expense Authorization Post-Save stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the user accessing the Expense Authorization. This may be the employee for the displayed Expense Authorization, a supervisor, or an administrator.
2	P_EA_ID	The Expense Authorization ID. This is the unique identifier for the Expense Authorization.
3	P_EA_EXPENSE_ID	The expense ID. This is the unique identifier for the individual expense.
4	P_COUNTRY_CD	The country code of the user.
5	P_LANGUAGE_CD	The language code of the user.

Return Values

The Expense Authorization Post-Save stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they always return a null result set. The column definitions are the same as those for an Expense Pre-Save stored procedure.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

Expense Authorization Pre-Sign Stored Procedures

Input Parameters

When an Expense Authorization Pre-Sign stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_EA_ID	The Expense Authorization ID. This is the unique identifier for the Expense Authorization.
2	P_LOGIN_EMPL_ID	The employee ID of the user who is accessing the Expense Authorization. This may be the employee for the displayed Expense Authorization, a supervisor, or an administrator.
3	P_COUNTRY_CD	The country code of the user.
4	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

Expense Authorization Pre-Sign stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error

	Name	Description
		W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done in the final reference cursor.

Be aware that some versions of Oracle have limited capability to sort reference cursor objects.

Expense Authorization Post-Sign Stored Procedures

Input Parameters

When an Expense Authorization Post-Sign stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_EA_ID	The Expense Authorization ID. This is the unique identifier for the Expense Authorization.
2	P_LOGIN_EMPL_ID	The employee ID of the user accessing the Expense Authorization. This may be the employee for the displayed Expense Authorization, a supervisor, or an administrator.
3	P_COUNTRY_CD	The country code of the user.
4	P_LANGUAGE_CD	The language code of the user.

Return Values

The Expense Authorization Post-Sign stored procedures must return a result set. For Oracle, this is a reference cursor object. If the release you are using includes stored procedures, they always return a null result set. The column definitions are the same as those for an Expense Authorization Pre-Sign stored procedure:

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

SQL Server Technical Notes

Understanding Result Sets

All procedures return a result set with either empty results or one or more rows. These characteristics are described below.

Note: The result set should be a simple SELECT statement.

Empty Results

- An empty result set for “Pre” procedures means no warnings or errors were found.
- An empty result set for “Post” procedures means the procedure ended normally.

One or More Rows

- A result set containing rows for ”Pre” procedures means warnings or errors were found. Each row is marked as an error or warning.
- A result set containing rows for “Post” procedures is ignored by the application and processed as if the procedure completed normally.

Timesheet “Pre” Stored Procedures

Input Parameters

When a timesheet-related “Pre” stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the system user who is accessing the timesheet. This could be the employee for the displayed timesheet, a supervisor, or an administrator.
2	P_TS_EMPL_ID	The employee ID of the employee for whom the timesheet was created. This may be identical to the P_LOGIN_EMPL_ID.
3	P_YEAR_NO_CD	The year number code for the timesheet.
4	P_PERIOD_NO_CD	The period number code for the timesheet.
5	P_TS_SCHEDULE_CD	The timesheet schedule code for the timesheet.
6	P_CLASS_CD	The timesheet class code for the employee for whom the timesheet was created.
7	P_WORK_SCEHDULE_CD	The work schedule code for the employee for whom the timesheet was created.
8	P_COUNTRY_CD	The country code of the user.
9	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

The "Pre" stored procedures must return a result set. If the release you are using includes stored procedures, they always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done by the stored procedure.

Timesheet "Post" Stored Procedures

Input Parameters

When a timesheet-related "Post" stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the system user who is accessing the timesheet. This could be the employee for the displayed timesheet, a supervisor, or an administrator.
2	P_TS_EMPL_ID	The employee ID of the employee for whom the timesheet was created. This may be identical to the P_LOGIN_EMPL_ID.
3	P_YEAR_NO_CD	The year number code for the timesheet.
4	P_PERIOD_NO_CD	The period number code for the timesheet.
5	P_TS_SCHEDULE_CD	The timesheet schedule code for the timesheet.
6	P_CLASS_CD	The timesheet class code for the employee for whom the timesheet was created.
7	P_WORK_SCEHDULE_CD	The work schedule code for the employee for whom the timesheet was created.

Return Values

The “Post” stored procedures must return a result set. If the release you are using includes stored procedures, they always return a null result set.

The column definitions are the same as those for a “Pre” stored procedure:

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

Expense Pre-Save Stored Procedures

Input Parameters

When an Expense Pre-Save stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the user who is accessing the expense report. This may be the employee for the displayed expense report, a supervisor, or an administrator.
2	P_EXP_RPT_ID	The expense report ID. This is the unique identifier for the expense report.
3	P_EXP_ID	The expense ID. This is the unique identifier for the individual expense.
4	P_COUNTRY_CD	The country code of the user.
5	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

The Expense Pre-Save stored procedures must return a result set. If the release you are using includes stored procedures, they will always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error

	Name	Description
		W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done by the stored procedure.

Expense "POST" Save Stored Procedures

Input Parameters

When an Expense Post-Save stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_LOGIN_EMPL_ID	The employee ID of the user accessing the expense report. This may be the employee for the displayed expense report, a supervisor, or an administrator.
2	P_EXP_RPT_ID	The expense report ID. This is the unique identifier for the expense report.
3	P_EXP_ID	The expense ID. This is the unique identifier for the individual expense.
4	P_COUNTRY_CD	The country code of the user.
5	P_LANGUAGE_CD	The language code of the user.

Return Values

The Expense Post-Save stored procedures must return a result set. If the release you are using includes stored procedures, they always return a null result set. The column definitions are the same as those for an Expense Pre-Save stored procedure:

	Name	Description
1	STS_E_OR_W	A one-character code either: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

Expense Report Pre-Sign Stored Procedures

Input Parameters

When an Expense Report Pre-Sign stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_EXP_RPT_ID	The expense report ID. This is the unique identifier for the expense report.
2	P_LOGIN_EMPL_ID	The employee ID of the user who is accessing the expense report. This may be the employee for the displayed expense report, a supervisor, or an administrator.
3	P_COUNTRY_CD	The country code of the user.
4	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

Expense Report Pre-Sign stored procedures must return a result set. If the release you are using includes stored procedures, they always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done by the stored procedure.

Expense Report Post-Sign Stored Procedures

Input Parameters

When an Expense Report Post-Sign stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_EXP_RPT_ID	The expense report ID. This is the unique identifier for the expense report.
2	P_LOGIN_EMPL_ID	The employee ID of the user accessing the expense report. This may be the employee for the displayed expense report, a supervisor, or an administrator.
8	P_COUNTRY_CD	The country code of the user.
9	P_LANGUAGE_CD	The language code of the user.

Return Values

The Expense Report Post-Sign stored procedures must return a result set. If the release you are using includes stored procedures, they always return a null result set. The column definitions are the same as those for an Expense Report Pre-Sign stored procedure:

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

Expense Authorization Pre-Save Stored Procedures

Input Parameters

When an Expense Authorization Pre-Save stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_ LOGIN_EMPL_ID	The employee ID of the user who is accessing the expense authorization. This may be the employee for the displayed expense authorization, a supervisor, or an administrator.
2	P_ EA_ID	The expense authorization ID. This is the unique identifier for the expense authorization.
3	P_ EA_EXPENSE_ID	The expense ID. This is the unique identifier for the individual expense.
4	P_COUNTRY_CD	The country code of the user.
5	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

The Expense Authorization Pre-Save stored procedures must return a result set. If the release you are using includes stored procedures, they will always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done by the stored procedure.

Expense Authorization "POST" Save Stored Procedures

Input Parameters

When an Expense Authorization Post-Save stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_ LOGIN_EMPL_ID	The employee authorization ID of the user accessing the expense report. This may be the employee for the displayed expense report, a supervisor, or an administrator.
2	P_ EA_ID	The expense authorization ID. This is the unique identifier for the expense authorization.
3	P_ EA_EXPENSE_ID	The expense ID. This is the unique identifier for the individual expense.
4	P_COUNTRY_CD	The country code of the user.
5	P_LANGUAGE_CD	The language code of the user.

Return Values

The Expense Authorization Post-Save stored procedures must return a result set. If the release you are using includes stored procedures, they always return a null result set. The column definitions are the same as those for an Expense Authorization Pre-Save stored procedure:

	Name	Description
1	STS_E_OR_W	A one-character code either: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

Expense Authorization Report Pre-Sign Stored Procedures

Input Parameters

When an Expense Authorization Report Pre-Sign stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_ EA_ID	The expense authorization ID. This is the unique identifier for the expense authorization.
2	P_LOGIN_EMPL_ID	The employee ID of the user who is accessing the expense authorization. This may be the employee for the displayed expense authorization, a supervisor, or an administrator.
3	P_COUNTRY_CD	The country code of the user.
4	P_LANGUAGE_CD	The language code of the user.

Note: The data is provided by the calling application and cannot be changed by the site. However, the stored procedure can use the data provided to access other tables in the schema for any additional information needed.

Return Values

Expense Authorization Pre-Sign stored procedures must return a result set. If the release you are using includes stored procedures, they always return a null result set. The column definitions for a row that contains data are listed below. Returned rows are displayed to the user as warnings or errors.

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: Any special formatting must be performed by the stored procedure. For example, to report the line and date causing the message, the timesheet line number and the timesheet day must be concatenated with any textual message in the STS_TEXT column. Sequencing for display purposes must also be done by the stored procedure.

Expense Authorization Report Post-Sign Stored Procedures

Input Parameters

When an Expense Authorization Post-Sign stored procedure is called, these parameters are passed to the procedure:

	Name	Description
1	P_EA_ID	The expense authorization ID. This is the unique identifier for the expense report.
2	P_LOGIN_EMPL_ID	The employee ID of the user accessing the expense authorization. This may be the employee for the displayed expense report, a supervisor, or an administrator.
8	P_COUNTRY_CD	The country code of the user.
9	P_LANGUAGE_CD	The language code of the user.

Return Values

The Expense Authorization Post-Sign stored procedures must return a result set. If the release you are using includes stored procedures, they always return a null result set. The column definitions are the same as those for an Expense Authorization Pre-Sign stored procedure:

	Name	Description
1	STS_E_OR_W	A one-character code: E – Error W – Warning
2	STS_TEXT	The text of the error or warning message. Limit is 2000 characters

Note: If non-null rows are returned, they are ignored.

Known Errors and Cautions

No Result Set Returned

Condition

When you implement a custom stored procedure, the application displays this “procedure not found” error message:

- Timesheet Pre-save stored procedure not found. (E1193)

Cause

The stored procedure must return a result set even if the result set is empty. Thus, the error can occur due to either of the following:

- The stored procedure exists and is called, but it does not return a result set.
- The stored procedure is missing from the schema.

Solution

Verify that the stored procedure exists in the schema and that it always returns a result set, even when the result set is empty.

Sample Oracle Timesheet Pre-Sign Procedure

This sample includes a standalone execution at the end.

```
-- ------------------------------------------------
--                                                -
--          TIMESHEET PARCKAGE BODY               -
--                                                -
-- ------------------------------------------------
CREATE OR REPLACE PACKAGE BODY TIMESHEET_SIGN_PKG
AS
-- ------------------------------------------------
--                                                -
-- TIMESHEET PRESIGN CUSTOM                       -
--   Created:  June, 2006                         -
-- ------------------------------------------------
PROCEDURE TIMESHEET_PRESIGN_PROC
( P_LOGIN_EMPL_ID    IN VARCHAR2,
P_TS_EMPL_ID       IN VARCHAR2,
P_YEAR_NO_CD       IN VARCHAR2,
P_PERIOD_NO_CD     IN VARCHAR2,
P_TS_SCHEDULE_CD   IN VARCHAR2,
P_CLASS_CD         IN VARCHAR2,
P_WORK_SCHEDULE_CD IN VARCHAR2,
P_COUNTRY_CD       IN VARCHAR2,
P_LANGUAGE_CD      IN VARCHAR2,
P_CURSOR  IN OUT   RET_REFCURSOR )
IS
--
lv_LOGIN_EMPL_ID       VARCHAR2(20) := P_LOGIN_EMPL_ID;
lv_TS_EMPL_ID          VARCHAR2(20) := P_TS_EMPL_ID;
lv_YEAR_NO_CD          VARCHAR2(4)  := P_YEAR_NO_CD;
lv_PERIOD_NO_CD        VARCHAR2(3)  := P_PERIOD_NO_CD;
lv_TS_SCHEDULE_CD      VARCHAR2(10) := P_TS_SCHEDULE_CD;
lv_CLASS_CD            VARCHAR2(20) := P_CLASS_CD;
lv_WORK_SCHEDULE_CD    VARCHAR2(10) := P_WORK_SCHEDULE_CD;
```

```
lv_COUNTRY_CD          VARCHAR2(8)   := P_COUNTRY_CD;
lv_LANGUAGE_CD         VARCHAR2(2)   := P_LANGUAGE_CD;
lv_obj_index           NUMBER        := 0;
lv_TIMESHEET_SIGN_OBJ  TAB_TIMESHEET_SIGN_OBJ :=
TAB_TIMESHEET_SIGN_OBJ();
lv_STS_E_OR_W          VARCHAR2(1);
lv_STS_TEXT            VARCHAR2(2000);
--
  Lv_line_cnt                  INTEGER:= 0;
  Lv_line_no                   NUMBER(10);
  Lv_charge_cd                 VARCHAR2(100);
  Lv_udt02                     VARCHAR2(50);
BEGIN
  -- ------------------------------------------------
  -- put all edit exceptions into one query.        -
  -- for a specific UDT02 on a line, each cell      -
  -- must have a zero in entered_hrs column else    -
  -- we format an output line as a fatal error      -
  -- ------------------------------------------------
  FOR cell_rec IN
       (SELECT T2.HRS_DT HRS_DT,
               T2.CORRECT_FL CORRECT_FL,
               T2.S_CELL_STATUS_CD S_CELL_STATUS_CD,
               T2.ENTERED_HRS ENTERED_HRS,
               T2.APPROVED_HRS APPROVED_HRS,
               T2.PRORATED_HRS PRORATED_HRS,
               T2.LINE_NO LINE_NO
         FROM TS_LINE T1, TS_CELL T2
             WHERE T1.EMPL_ID        = P_TS_EMPL_ID
               AND T1.TS_SCHEDULE_CD = P_TS_SCHEDULE_CD
               AND T1.YEAR_NO_CD     = P_YEAR_NO_CD
               AND T1.PERIOD_NO_CD   = P_PERIOD_NO_CD
               AND T1.UDT02_ID       LIKE 'LEAVE%'
               AND T1.EMPL_ID        = T2.EMPL_ID
               AND T1.TS_SCHEDULE_CD = T2.TS_SCHEDULE_CD
               AND T1.YEAR_NO_CD     = T2.YEAR_NO_CD
               AND T1.PERIOD_NO_CD   = T2.PERIOD_NO_CD
               AND T1.LINE_NO        = T2.LINE_NO
               AND T2.ENTERED_HRS   != 0
         ORDER BY T2.LINE_NO, T2.HRS_DT)
   LOOP
        lv_STS_E_OR_W     := 'E';
           lv_STS_TEXT       := 'Testing.->Line_NO '
                                ||to_char(cell_rec.LINE_NO)
                                ||' entering '
                                ||to_char(cell_rec.ENTERED_HRS)
                                ||' hours for date '
```

```
                                ||to_char(cell_rec.HRS_DT,'YYYY-MON-
DD')
                                ||' FOR UDT02 THIS MUST BE ZERO!';
           lv_obj_index      := lv_obj_index + 1;
           lv_TIMESHEET_SIGN_OBJ.extend;
           lv_TIMESHEET_SIGN_OBJ(lv_obj_index) :=
               SCALER_TIMESHEET_SIGN_OBJ(lv_STS_E_OR_W,lv_STS_TEXT);
  END LOOP;
  -- ------------------------------------------------
  --                 END CUSTOM                      -
  -- ------------------------------------------------
 OPEN P_CURSOR FOR SELECT *
    FROM TABLE (CAST ( lv_TIMESHEET_SIGN_OBJ as TAB_TIMESHEET_SIGN_OBJ)
) ;
EXCEPTION
  When OTHERS Then
    RAISE_APPLICATION_ERROR(-20303, SUBSTR(SQLERRM,1,200));
END TIMESHEET_PRESIGN_PROC;
END TIMESHEET
END TIMESHEET_SIGN_PKG;
/
show errors


-- standalone execution
variable x refcursor
exec TIMESHEET_SIGN_PKG.TIMESHEET_PRESIGN_PROC
(‘10010','10010','2003','001','B','HRN','TEST','us','US',:x)
print
```

Sample SQL Server Timesheet Pre-Sign Procedure

This sample includes a standalone execution at the end.

```
print '---test existence of PROCEDURE timesheet_presign_proc'
IF EXISTS (SELECT name FROM sysobjects
    WHERE name = 'TIMESHEET_PRESIGN_PROC' AND type = 'P'
     AND user_name (uid) = current_user )
   BEGIN
PRINT '---dropping existing PROCEDURE timesheet_presign_proc'
     DROP PROCEDURE TIMESHEET_PRESIGN_PROC
   END
   ELSE
   BEGIN
print '---PROCEDURE timesheet_presign_proc does not exist'
   END
```

```
go
print '---creating PROCEDURE timesheet_presign_proc'
go
CREATE PROCEDURE TIMESHEET_PRESIGN_PROC
   @P_LOGIN_EMPL_ID     VARCHAR(20),
   @P_TS_EMPL_ID        VARCHAR(20),
   @P_YEAR_NO_CD        VARCHAR(4),
   @P_PERIOD_NO_CD      VARCHAR(3),
   @P_TS_SCHEDULE_CD    VARCHAR(10),
   @P_CLASS_CD          VARCHAR(20),
   @P_WORK_SCHEDULE_CD  VARCHAR(10),
   @P_COUNTRY_CD        VARCHAR(8),
   @P_LANGUAGE_CD       VARCHAR(2)
AS SET NOCOUNT ON
BEGIN
  DECLARE
  @lv_LOGIN_EMPL_ID       VARCHAR(20),
  @lv_TS_EMPL_ID          VARCHAR(20),
  @lv_YEAR_NO_CD          VARCHAR(4),
  @lv_PERIOD_NO_CD        VARCHAR(3),
  @lv_TS_SCHEDULE_CD      VARCHAR(10),
  @lv_CLASS_CD            VARCHAR(20),
  @lv_WORK_SCHEDULE_CD    VARCHAR(10),
  @lv_COUNTRY_CD          VARCHAR(8),
  @lv_LANGUAGE_CD         VARCHAR(2),
  @lv_count               INTEGER,
  @lv_obj_index           INTEGER,
  @lv_have_e_or_w         VARCHAR(1),
  @lv_my_sql              NVARCHAR(2000),
  @lv_qualify             NVARCHAR(756),
  @lv_my_cursor           NVARCHAR(4000),
  @lv_ParmDefinition      NVARCHAR(1000),
  @lv_MODIFIED_BY         VARCHAR(20),
  @lv_STS_E_OR_W          VARCHAR(1),
  @lv_STS_TEXT            VARCHAR(2000),
  @HRS_DT                 DATETIME,
  @CORRECT_FL             VARCHAR(1),
  @S_CELL_STATUS_CD       VARCHAR(1),
  @ENTERED_HRS            DECIMAL(8,2),
  @APPROVED_HRS           DECIMAL(8,2),
  @PRORATED_HRS           DECIMAL(8,2),
  @LINE_NO                INTEGER
DECLARE
  @ErrorWarningTable      TABLE (
  STS_E_OR_W              VARCHAR(1)  NULL,
  STS_TEXT                VARCHAR(2000) NULL)
--
```

```
--
  SET @lv_qualify      =
        N'"'+ CAST (@@servername as nvarchar)+ N'"' + N'.'+ N'"' +
      CAST(db_name() as nvarchar) + N'"' + N'.' +
      CAST(current_user as nvarchar) + N'.'
  SET @lv_qualify = N''
  SET @lv_ParmDefinition =
       N'@nP_TS_EMPL_ID     VARCHAR(20),
         @nP_YEAR_NO_CD     VARCHAR(4),
         @nP_PERIOD_NO_CD   VARCHAR(3),
         @nP_TS_SCHEDULE_CD  VARCHAR(10)'
  SET @lv_my_sql =
       N'DECLARE read_line_cell CURSOR for SELECT
                T2.HRS_DT ,
                T2.CORRECT_FL,
                T2.S_CELL_STATUS_CD,
                T2.ENTERED_HRS,
                T2.APPROVED_HRS,
                T2.PRORATED_HRS,
                T2.LINE_NO
         FROM TS_LINE T1, TS_CELL T2
             WHERE T1.EMPL_ID        =  @nP_TS_EMPL_ID
               AND T1.TS_SCHEDULE_CD =  @nP_TS_SCHEDULE_CD
               AND T1.YEAR_NO_CD     =  @nP_YEAR_NO_CD
               AND T1.PERIOD_NO_CD   =  @nP_PERIOD_NO_CD
               AND T1.UDT02_ID       LIKE ''LEAVE%''
               AND T1.EMPL_ID        = T2.EMPL_ID
               AND T1.TS_SCHEDULE_CD = T2.TS_SCHEDULE_CD
               AND T1.YEAR_NO_CD     = T2.YEAR_NO_CD
               AND T1.PERIOD_NO_CD   = T2.PERIOD_NO_CD
               AND T1.LINE_NO        = T2.LINE_NO
               AND T2.ENTERED_HRS   != 0
          ORDER BY T2.LINE_NO, T2.HRS_DT for read only'
  EXECUTE sp_executesql @lv_my_sql ,@lv_ParmDefinition,
         @nP_TS_EMPL_ID     = @P_TS_EMPL_ID,
         @nP_YEAR_NO_CD     = @P_YEAR_NO_CD,
         @nP_PERIOD_NO_CD   = @P_PERIOD_NO_CD,
         @nP_TS_SCHEDULE_CD = @P_TS_SCHEDULE_CD
  OPEN read_line_cell
  WHILE ( 1 = 1 )
  BEGIN
    FETCH read_line_cell INTO
         @HRS_DT,
         @CORRECT_FL,
         @S_CELL_STATUS_CD,
         @ENTERED_HRS,
         @APPROVED_HRS,
```

```
          @PRORATED_HRS,
          @LINE_NO
      IF (@@fetch_status <> 0)
      BEGIN
        CLOSE read_line_cell
        DEALLOCATE read_line_cell
        BREAK
      END
      SET @lv_STS_E_OR_W  = 'E'
      SET @lv_STS_TEXT    =
            'Testing.->Line_NO '
          + CAST(@LINE_NO AS VARCHAR(10))
          + ' entering '
          + CAST(@ENTERED_HRS AS VARCHAR(12))
          + ' hours for date '
          + CAST(@HRS_DT AS VARCHAR(22))
          + ' FOR UDT02 THIS MUST BE ZERO!'
      INSERT INTO @ErrorWarningTable VALUES (@lv_STS_E_OR_W,
@lv_STS_TEXT)
  END
  SELECT * from @ErrorWarningTable
RETURN
END
GO
print '---PROCEDURE timesheet_PRESIGN_proc completed'
GO


-- test standalone executions
exec timesheet_presign_proc
'10010','10010','2003','001','B','HRN','TEST','us','US'
```

Weblogic Server

Overview

When you run Deltek Time & Expense on multiple Weblogic server instances, some of the administrative functions require extra attention. The following sections give a brief description of all areas that may require additional or modified steps.

Areas Requiring Additional or Modified Steps

System Installation

Deltek Time & Expense must be installed on each application server instances. The database setup should be executed once on the shared database.

Apply Upgrade Technical Console Function

Upgrades don't affect the Weblogic server, so an upgrade needs to be performed only once from one Weblogic instance. First, make sure that only one Weblogic instance is running, then apply the upgrade. Then restart/start all Weblogic instances.

Apply Service Pack Technical Console Function

Service packs may affect both Weblogic server instances and the database. For this reason, you should apply them to each Weblogic instance one at a time. If there are custom stored procedures in the system, reapply them to all application domains after applying service packs to all Weblogic instances.

Applying Quick Fixes

Quick fixes can contain program file changes, Weblogic configuration changes, and database changes. If a quick fix contains only program file changes, you should apply it to each Weblogic server instance separately.

If a quick fix contains Weblogic configuration changes and it requires running the Apply Service Pack technical console function, apply the quick fix to all weblogic instances separately.

If there are database changes included in a quick fix, run the Apply Service Pack technical console function at least one time through any of the Weblogic instances.

If you ran the Apply Service Pack technical console function while applying the quick fix and there are custom stored procedures in the system, reapply these custom stored procedures to all application domains after applying the service packs to all Weblogic instances.

Create New Domain Technical Console Function

After creating a new domain, apply all necessary changes to the central database and to the local Weblogic instance on which the new domain was created. You will need to replicate the Weblogic configuration changes to the other Weblogic instances.

To accomplish this, follow these steps on all Weblogic instances that were not used to execute the Create New Domain technical console function:

1. Go to weblogic console/DeltekTE/Services/JDBC/Connection Pools.

2. Click the Clone icon next to the connection pool called TC_0001_POOL.
3. Set the name of the connection pool to the new schema name created for the new Deltek Time & Expense domain + _POOL (e.g., TC_0002_POOL).
4. Enter the password for the schema in the *Password* and *Confirm Password* fields.
5. Click Clone.
6. Go to weblogic console/DeltekTE/Services/JDBC/Data Source.
7. Set the name of the data source to the new schema name created for the new Deltek Time & Expense domain + _DSN (e.g., TC_0002 _DSN).
8. Set JNDI name to the same string as the data source name above.
9. Select the newly create connection pool in the *Pool Name* field
10. Click Clone.

Delete Domain Technical Console Function

After deleting a domain, apply all necessary changes to the central database and to the local Weblogic instance on which the domain was deleted. You will need to replicate these Weblogic configuration changes to the other Weblogic instances.

To accomplish this, follow these steps on all Weblogic instances that were not used to execute the Delete Domain technical console function:

1. Go to weblogic console/DeltekTE/Services/JDBC/Data Source.
2. Click the Delete icon next to the name of the data source that includes the name of the database schema that hosted the deleted Deltek Time & Expense domain.
3. Click Yes.
4. Go to weblogic console/DeltekTE/Services/JDBC/Connection Pools.
5. Click the Delete icon next to the name of the connection pool that includes the name of the database schema that hosted the deleted Deltek Time & Expense domain.
6. Click Yes.

Copy Domain Technical Console Function

Execute this function only once through any of the Weblogic instances. Copying domains makes changes only in the shared database.

Apply License Technical Console Function

Execute this function only once through any of the Weblogic instances. Applying licenses makes changes only in the shared database.

Running Scheduled Imports

Configuration information on scheduled imports is stored in the database, so all running Weblogic instances look at the same information and will start up the import process at the same time. If these separate Weblogic instances are looking at local, rather than shared, import directories, this won't be a problem because only one of them will see the import data files and start processing the import.

However, if these separate Weblogic instances are looking at the same network folder that contains the import data files, more then one of them may start importing. This situation can lead to unexpected behavior. To prevent a server from importing, map its import network drive and folder to an empty directory. Do this for each server that you don't want running the import process. Every time an import starts up, only the server that is mapped to the real import folder will grab the import files. The other servers will see empty directories and will not start processing. Note that the actual path string that all the servers will be looking at is the same. This information comes from the database. Only the mappings will be different on the servers that run Weblogic.

If you are running imports through the database, you must disable scheduled imports on all but one server. To disable scheduled imports on a Weblogic server, remove the import scheduler service that starts up scheduled imports. You can do this by un-deploying the startup java class that creates this scheduler service every time Weblogic is started. On servers that have the "DeltekTCDSIScheduler" startup class un-deployed, scheduled imports are not available. If you go to the Import console and try to change any of the settings related to scheduled imports, you will receive an error message, which can be safely ignored.

Follow these steps to disable scheduled imports on a Weblogic server instance:

1. Confirm that you are running Time and Expense with ESS 8.0 or higher.
2. Open the *DeltekTE\applications\TC\META-INF\weblogic-application.xml* file in Notepad or a similar text editor.
3. Comment out the Listener section as shown below. Type the lines in bold exactly as shown.

   <!--

   <listener>

   <listener-class>com.deltek.tc.dsi.schedulemanager.DSIScheduler</listener-class>

   </listener>

   -->
4. Restart Weblogic server.

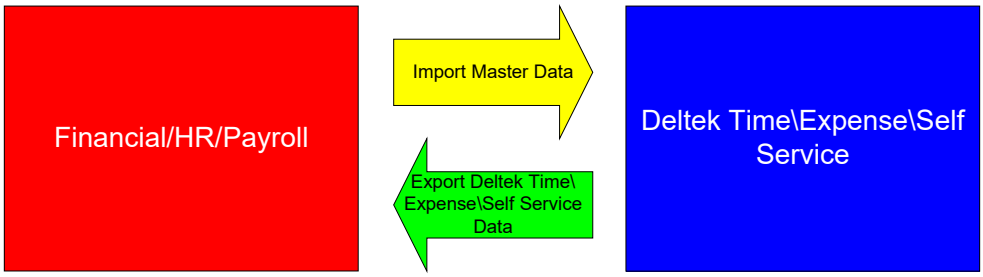
Do not use the import console on servers that have the import scheduler disabled*.* This includes scheduling or on-demand import. To perform these tasks, always connect to the WebLogic server where the import scheduler is running. If you inadvertently use the Import Console on a server that has the import scheduler disabled, you will receive a E9002 error message if you change any schedule settings. This is expected and you can ignore the error.

If any scheduled imports are pending when you disable the import scheduler, an error message will be logged after you restart the WebLogic server. This is expected and the error can be ignored.

Import/Export

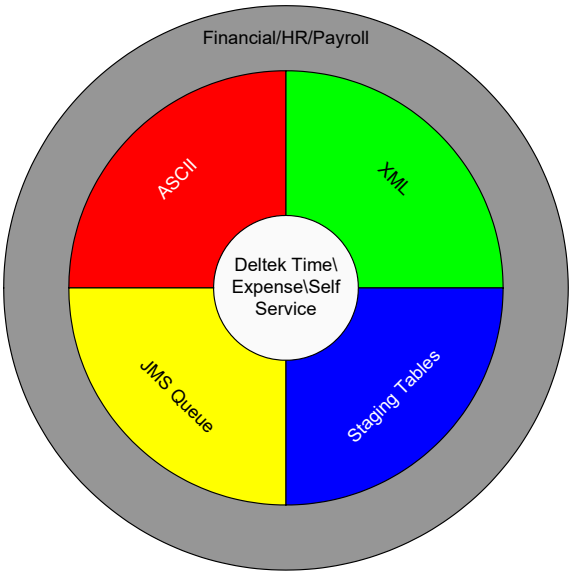
Overview

The Deltek Time & Expense interface is comprised of two data flows. The first is an inbound flow, where master data is imported from source systems. The master data comprises such information as employees, projects, etc. The second flow is the outbound flow, where timesheet and expense data is exported from Deltek Time & Expense to outside systems. This timesheet data is comprised mainly of employee charges and hours. The expense data is comprised mainly of expense report information.



Import Master Data

The import flow into Deltek Time & Expense provides the system with master data. This master data defines your organization, employees, and projects. Although you can use Deltek Time & Expense to maintain this information, it is more likely that you already maintain this information in an existing system(s). The more information that you import into the system, the better.



How the interface occurs and is executed is very flexible. The system accepts a mixture of import types and accepts imports from multiple source systems.

Import Types

The import accepts master data in any of four import types: ASCII, XML, Staging Table, and Java Message Service (JMS) Queue.

ASCII

Comma-delimited ASCII files are commonly used for interfacing. This import type is excellent for batch-type interfaces where the source system cannot be configured to use a transactional-based interface or when the systems cannot communicate directly. This interface type is also a good choice when you are initially loading the master data from the source systems into Deltek Time & Expense. See the “Import ASCII Layouts” chapter in this guide for the supported ASCII layouts.

Note: The location or directory where the system looks for the ASCII files is configured within the system. See the documentation for the Import Console screen in the online help for further details.

The location to which processed ASCII files are moved is configured within the system. See the documentation for the Import Console screen in the online help for further details.

The ASCII file layouts support multiple versions. Deltek Time Collection 3.x/4.x/5.x layouts are supported for backwards compatibility.

Staging Tables

Deltek Time & Expense has "Staging Tables" within its database that you can use as a drop-off point for master data. This type of interface is an excellent choice when systems can communicate. You can use it in both batch and transactional environments. See the “Import Staging Table Layouts” chapter in this guide for the table layouts.

Note: All staging tables begin with "IMPORT_" and can be accessed via the TC_IMPORT login account.

XML

The system supports the import of XML files. The advantages are very similar to those of ASCII files, but XML files are more commonly used for interfacing with newer source systems. See the “Import XML Schemas” chapter in this guide for XML layouts.

Note: The location or directory where the system looks for XML files is configured within the system. See the documentation for the Import Console screen in the online help for further details.

The location to which processed XML files are moved is configured within the system. See the documentation for the Import Console screen in the online help for further details.

JMS Queue

Java Message Service (JMS) Queue is an application program interface from Sun Microsystems that supports formal communication between computers on a network. Think of the queue as an inbox to which you send import transactions. This type of interface is an excellent choice for transactional-based interfacing, especially if source systems can communicate with JMS queues.

Execution Modes

Deltek Time & Expense supports three types of execution modes. These are:

On Demand

When using the On Demand mode, you execute the import by selecting a pushbutton within the system. See the documentation for the Import Console screen in the online help for further details. On-demand execution is available for the following import types:

- ASCII
- XML

- Staging Tables

Scheduled

When you use the Scheduled mode, the system automatically executes the import according to a schedule. The schedule is based on an interval defined within the system. See the documentation for the Import Console screen in the online help for further details. Scheduled execution is available for the following import types:

- ASCII
- XML
- Staging Tables

When Received

When you use this mode, the system processes import transactions as soon as they are received. Transactions are processed in a first-in, first-out order. JMS Queue imports use this mode of execution.

Transaction Types

Each import transaction name indicates which type of transaction it is. The valid transaction types are as follows:

- **Add** - The data item being downloaded is new and is added to Deltek Time & Expense.
- **Change** - The data item being downloaded is an update to an existing item in the system.
- **Delete** - The item being downloaded is deleted from the system.
- **Batch** - The item is added or updated depending on whether it exists in the system.

You can use all the above transaction types with any of the import types. There are two categories of transaction types. One processes specific transactions, and the other passes data only in a batch. If your system can generate different types of transactions, using the transaction types above provides the best synchronization between systems. Batch transactions cannot delete data; they can only provide new data or update existing data.

Note: Batch transactions can update an item to inactive. Although this does not delete the item, it does mark it as unusable.

Data

Deltek Time & Expense can receive 33 different sets of data. As mentioned earlier, the more data interfaced, the better. Below are descriptions of these data flows.

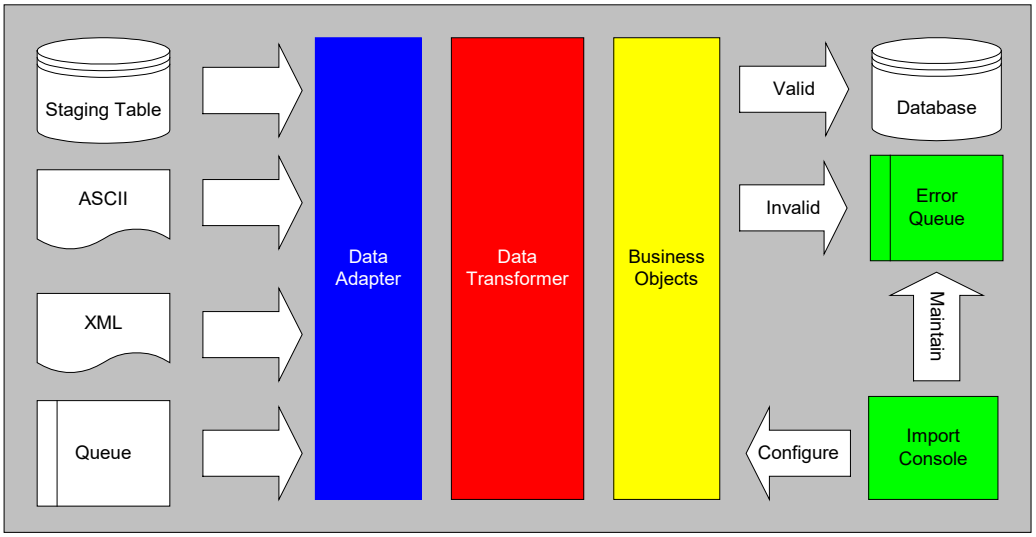
- Employee Data — This data set consists of identification information for each employee/user. Typically, this data originates from a Human Resources (HR) system.
- Employee History Data — This set contains sensitive employee data (e.g., the date an employee changed departments). Typically, this data originates either from an HR or Payroll system.
- Employee Leave Data — This data set contains the employee's leave balances for the tracked leave. It can consist of beginning year balances, accruals, and/or adjustments.
- Employee Group Data — You can use this data set to build employee groups through the import process. You can create/update the group itself, and assign members (employees) and owners (supervisors/administrators). Typically, this information does not exist anywhere in an

organization except on organizational charts. It is possible that enough data exists to create the groups and assign members, and therefore you must assign primary and backup supervisors/administrators within Deltek Time Collection.

- UDTXX Data (01 - 10) — These data sets contain the UDT master data, where XX equals one of the 10 user-defined tables. The contents of a data set depend on how the UDTXX is defined.
- UDT01 Supervisor Data — This data set contains the supervisor information for UDT01 records. The Charge Security and Line Level Approval features use this information.
- UDT02 Supervisor Data — This data set contains the supervisor information for UDT02 records. The Charge Security and Line Level Approval features use this information.
- UDT01/UDT02 Link Data — This data set contains the link information for UDT01 and UDT02.
- UDT01/UDT07 Link Data — This data set contains the link information for UDT01 and UDT07.
- UDT01/UDT09 Link Data — This data set contains the link information for UDT01 and UDT09.
- UDT02/UDT07 Link Data — This data set contains the link information for UDT02 and UDT07.
- UDT02/UDT09 Link Data — This data set contains the link information for UDT02 and UDT09.
- UDT09/UDT03 Link Data — This data set contains the link information for UDT09 and UDT03.
- Timesheet Invoicing Data — This data set contains invoicing information that indicates which timesheets are included on certain invoices. The Billing Backup feature uses this information.
- Charge Tree Data — You can use this data set to build charge trees through the import process. You can create/update trees, branches, and charges; assign employee groups to trees and branches; and assign employees to charges. Typically, this data originates from some type of project/job costing system.
- Timesheet Rejection Data — This data set returns rejected timesheet transactions so that timesheets can be appropriately marked and users notified. If this data set is not available, timesheet rejections must be procedurally handled within the system. See the documentation for the Change TS Status screen in the online help for further details.
- Costpoint Company Data — This data set contains Costpoint company codes. They are used for the Costpoint multicompany feature in Deltek Time & Expense.
- Currency Schedules Data — This data set contains currency exchange rates. They are used for currency conversions.
- Expense Report Payments — This data set contains payment information for expense reports.
- Expense Rejection Data — This data set returns rejected expense transactions so that expense reports can be appropriately marked and users notified.
- Leave Type Data — This data set contains leave types. You can specify whether they are vacation or holiday types.
- Per Diem Schedule Data — This data set can contain the government's Continental United States (CONUS) or Outside Continental United States (OCONUS) per diem rates.
- Tax Codes — This data set can contain the tax codes used by the Deltek Expense when you need to track taxes, such as VAT, for certain expenses.
- Receipt Images — This import is a catalog of the Adobe .PDF files that contain images of expense report receipts.

Import Components

The following diagram details the components that make up the import:



Data Adapter

The Data Adapter is the first component in the import routine. It gathers, validates, and packages import transactions/files. It checks to make sure that the import transactions are in the proper format/layout. Each time an on-demand or scheduled import is executed, the adapter looks for transactions that import table entries, ASCII files, and XML files. The adapter processes queue entries as they are received. If the adapter finds errors in the layout of the import transaction, the system records an error in the Error Queue; otherwise, it passes the import transactions to the Data Transformer for further processing.

Data Transformer

The Data Transformer accepts the import transactions from the Data Adapter. It checks to see whether a transaction is complete and required elements have been provided. It also provides defaults for any missing elements. It determines whether a batch transaction will be processed as an Add or Change transaction. If the transaction is invalid, the system records an error in the Error Queue. If the transaction is valid, it is packaged and submitted to the appropriate Business Object for processing.

Business Objects

Business Objects validate the import transactions against the business rules defined in the system. For example, a particular Business Object validates transactions that add employees. The system uses the same Business Object when you add employees in the Employee Information application. If the transaction is invalid, the system records it in the Error Queue. If it is valid, the system updates the Deltek Time Collection database.

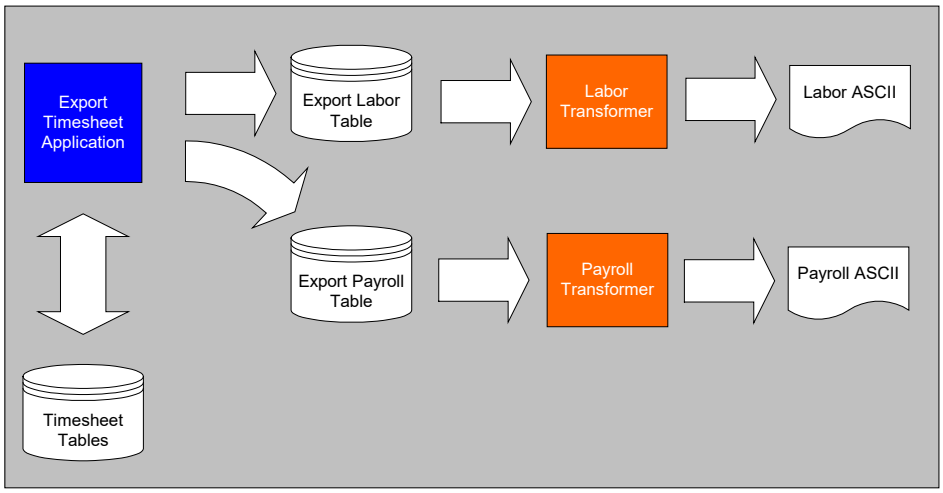
Import Console

Administrators can use the Import Console application in the Interfaces menu to configure, schedule, and execute an import. In addition, they can view, print, and clear the error queue.

An Edit Transaction table function is available. You can select a record in the Error Log table then select the Edit Transaction function to make changes to the record. Select the Update/Retry pushbutton to re-submit the record for processing. See the documentation for the Import Console screen in the online help for further details.

Export Time Data

The Export Timesheet process sends employee timesheet data to an outside system. The diagram below provides an overview.



The Export Timesheets process selects the appropriate timesheets and drops the timesheet transactions into a labor table, a payroll table, or both. Whether the labor or payroll tables are updated depends on the system configuration. (In addition to the present topic, see the online documentation for the Manage Timesheet Classes screen and the Export Timesheets screen for additional system configuration information.)

After the timesheet transactions are exported, they either remain in staging tables for pickup, or the export application calls a transformer that transforms the table entries into a specific ASCII layout.

Export Timesheets Application

Selection Criteria

The administrator selects which timesheets to export using the following criteria:

- Timesheet period
- Export Work Schedules
- Time record date range
- Costpoint company (if you are using the Costpoint multicompany feature)
- Employee group
- Timesheet class

> **Note:** As part of the selection criteria, you assign a batch ID to the export.

Process

After providing the selection criteria, the administrator clicks **Process** to do the following:

- Gather timesheets - The application gathers the hours from approved or, optionally, in-process timesheets that have not been previously exported.

- Mark timesheets as unavailable - As the timesheets are gathered, the application marks them as unavailable so no one can change them during the export process.

Note: The option to export in-process timesheets is only available if you are exporting to Costpoint 5.1 or a later version or if you are exporting to an application other than Costpoint. It is not available for earlier versions of Costpoint.

Reporting

Once timesheets have been processed, the administrator can print summary or detail reports that show the hours that will be exported.

Export

If satisfied with the reports, the administrator then clicks **Export**. The export function does the following:

- Writes timesheet transactions to the appropriate staging tables
- Marks timesheets as processed and as available for corrections.
- Calls the transformer, if appropriate

Note: You can find layouts of the staging tables (EXPORT_LABOR and EXPORT_PAYROLL) in the "Data Dictionary" chapter and the "Export Staging Table Layouts" chapter of this guide.

Timesheet Transformers

The system determines whether transformation is needed based on the settings you specify on the Export Options tab of the Configure Time Settings screen. The following sections describe these settings.

Note: For Time & Expense version 7.1.2 and later, the Export Options tab supersedes the ExportTimesheet.properties file. If you are using version 7.1.2 or higher, you must use this tab to configure all of your timesheet export options. When you first open the Export Timesheets screen after upgrading to 7.1.2 or later, Time & Expense automatically converts the settings in your ExportTimesheet.properties file (if one exists) to settings on the Export Options tab.

UDT Export Table

For each UDT you are using, you can indicate in the table of UDTs if you want to export charges entered for any of the UDT codes. You can also list codes in **Export Filter** for which you do not want to export charges.

Check the **Export** check box for those UDTs you want to export. If **Export** is checked, Time & Expense will export all UDT codes of that type, except for any you specify in **Export Filter**. This is normally done so that the destination system can supply a default value for the omitted UDT type.

Note: These settings will cause Costpoint to calculate missing values using Costpoint default logic.

If you have a specific UDT code that you do not wish to export, enter that value in **Export Filter** (UDT Options subtask) for the appropriate UDT cell. Any charge information entered for that code will be omitted from the transformed files.

If you have a range of UDT codes that should not be exported, you can use a wildcard character in **Export Filter** to remove values that start or end with a specific character or characters. Enter a percent sign (%) after the characters to remove any UDT codes that start with those characters. Entering the percent sign before the character or characters filters out any UDT codes that end with those characters.

Labor Distribution: ASCII Layout

Your selection in **ASCII Layout** under **Labor Distribution** specifies whether or not the labor distribution transformation will occur and, if it does, the type and version of ASCII file created.

These are the available options:

- Costpoint 5.1/5.2
- Costpoint 6X
- Costpoint 7X or greater
- Generic - Detail
- Generic - Summary
- None

If you select **None**, the exported timesheet labor records will not be transformed, but will instead stay in the staging tables. If you select a value other than **None**, the transformer will create an ASCII file that contains the records for the exported batch.

Costpoint Salary Proration

This field is only available when the ASCII Layout option is *Costpoint 7.x or greater*. Select this check box to enable the Salary Proration feature, which enables you to export timesheet data to Costpoint work schedules. You must also select the **Export Work Schedules** check box in **T Time » Timesheet Interfaces » Export Timesheets**.

> **Note:** When **Costpoint Salary Proration** is selected, the **Summarize** check box is disabled if the ASCII Layout is set to either *Costpoint 6.x* or *Costpoint 7.x* and if the **Export In-Process Timesheets** option is not selected.

Export Labor Amounts

If Time & Expense is configured to track Rate 1 information, check **Export Labor Amounts** to include Rate 1 values in the transformed export file.

If you do not check **Export Labor Amounts**, zeroes will be downloaded for the rate.

Summarize

If you are exporting to Costpoint (if **ASCII Layout** contains one of the Costpoint versions), check **Summarize** if you want to summarize the export data by timesheet line. The **Timesheet Date** field will contain the timesheet date supplied in the Export Timesheets screen. The **Notes** field will contain the line comment.

If you do not check **Summarize**, the records will not be summarized. They will instead be created at the cell level. The **Timesheet Date** field will contain the cell date, and the **Notes** field will contain cell comments.

Labor Distribution: Recast UDT10

If you are going to export time to another application for labor distribution reporting, select the Recast UDT10 check box if you want the option to "recast" hours entered for one UDT10 code to another UDT10 code.

"Recasting" results in the creation of another export record that is essentially identical to the original export record except that it is associated with a different UDT10 code and labor costs are recalculated based on the rate and other information associated with that UDT10 code. The UDT10 code to which a record is recast is specified when you set up the original UDT10 code in Time & Expense.

If the second UDT10 code also has a recast code specified, the export process creates a third record for that code. This process continues until a recast record is created for a UDT10 code that has no recast code.

Using recasting, you can simplify data entry for your employees while still being able to reclassify or subdivide that time for project labor reporting (to break the hours down into regular and overtime, for example).

> **Note:** This option is primarily for those exporting to applications other than Costpoint. Costpoint have an equivalent feature built into them.

Payroll: ASCII Layout

Your selection in **ASCII Layout** under **Payroll** specifies whether or not the payroll transformation will occur and, if it does, the type and version of ASCII file created.

These are the available options:

- ADP
- ADP Rate
- None

If you select **None**, the exported timesheet payroll records will not be transformed, but will instead stay in the staging tables. If you select a value other than **None**, the transformer will create an ASCII file that contains the records for the exported batch.

ADP Batch

If you export a payroll file to ADP, use **ADP Batch** to select an option for building batch IDs for ADP exports. These are the options:

- Batch ID — If you use this value, the system uses the batch ID provided in the Export Timesheets screen.
- Code 1 — The system uses the value in **Code 1** on the Miscellaneous subtask of the Manage Resource Information screen.

- Code 2 — The system uses the value in **Code 2** on the Miscellaneous subtask of the Manage Resource Information screen.
- Code 3 — The system uses the value in **Code 3** on the Miscellaneous subtask of the Manage Resource Information screen.

Note: The program creates the ASCII file in the Export directory. The name of the labor file is "<Batch ID>.ets" where <Batch ID> is the batch ID supplied in the Export Timesheets screen.

The name of the payroll (ADP) file is "EPI" + <Payroll ID>+<Payroll Batch Seq NO>.csv where <Payroll ID> is the payroll ID of the employee on the record (from the Manage Resource Information screen) and <Batch ID> is the batch ID supplied in the Export Timesheets screen.

You can find layouts of ASCII files created by the timesheet transformers in the “Transformed Export ASCII Layouts” chapter in this guide.

Payroll: Recast UDT10

If you are going to export time to a payroll application, select the Recast UDT10 check box if you want the option to "recast" hours entered for one UDT10 code to another UDT10 code.

"Recasting" results in the creation of another export record that is essentially identical to the original export record except that it is associated with a different UDT10 code and pay amounts are recalculated based on the rate and other information associated with that UDT10 code. The UDT10 code to which a record is recast is specified when you set up the original UDT10 code in Time & Expense.

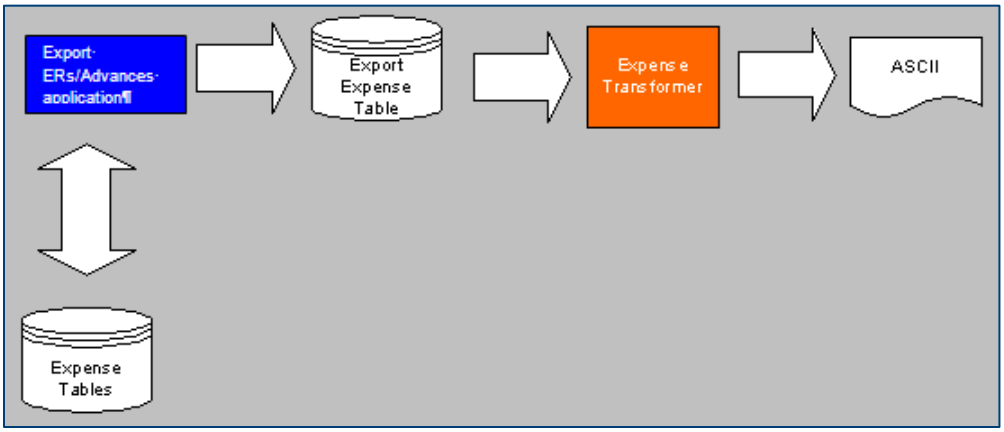
If the second UDT10 code also has a recast code specified, the export process creates a third record for that code. This process continues until a recast record is created for a UDT10 code that has no recast code.

Using recasting, you can simplify data entry for your employees while still being able to reclassify or subdivide that time for payroll processing (to break the hours down into regular and overtime, for example).

Note: This option is primarily for those exporting to applications other than Costpoint. Costpoint have an equivalent feature built into them.

Export Expense Data

The export process sends expense data to an outside system. The diagram below provides an overview.



The Export ERs/Advances application selects the appropriate expense reports and/or advances and drops the transactions into an expense staging table. The expense transaction remains in the staging table for pickup, or the export application can call a transformer that transforms the table entries into a specific ASCII layout.

Export ERs/Advances Application

Selection Criteria

Administrators can select which expense reports and/or advances they want to export using the following criteria:

- Costpoint Company (if you are using the Costpoint multicompany feature)
- Date Range
- Employee Group
- Expense Class
- Expense Report Type

Note: As part of the selection criteria, the administrator assigns a batch ID to the export.

Process

After providing the selection criteria, the administrator processes the request. This function performs the following steps:

- Gathers Expense Reports - The application gathers the expense report information from approved expense reports that have not been previously exported.
- Gathers Advances - The application gathers the advance information from approved expense authorizations that have not been previously exported.
- Marks Expense Reports Unavailable - As the expense reports are gathered, they are marked as unavailable so that they cannot be changed during the process.

Reporting

Once expense reports have been processed, the administrator can print reports that detail the expenses and/or advances that will be exported. Reports can be summary or detail.

Export

If satisfied with the reports, the administrator exports the expense reports and/or advances. This function performs the following steps:

- Writes expense and advance transactions to appropriate staging tables.
- Marks expense reports as processed and as available for corrections.
- Calls the Transformer, if appropriate.

Note: You can find the layout of the staging table (EXPORT_EXPENSE) in the “Export Staging Table Layouts” chapter of this guide.

Expense Report Transformers

The system determines whether transformation is needed by inspecting a file contained in the Export directory (you define the directory in the General Options tab of the Configure General Settings screen). The name of the file must be "ExportExpense.properties." The file contains the following:

EXPORT=<VALUE>

The valid values are:

- COSTPOINT 7X or greater
- COSTPOINT 6X
- NONE

If you do not specify a transformation value or the application cannot find a properties file, the exported expense report records will stay in the staging table.

In addition to specifying the layout of the transformed ASCII file, you can fill out one or more of the User-Defined Table (UDT) fields with spaces by placing the following parameters in the properties file:

EXPORT_UDTXX=N
For example, if you want UDT03 to be removed from ASCII file so that the destination system can provide a default, use the following parameter:
EXPORT_UDT03=N

Note: If you use these parameters, Costpoint calculates the missing values using Costpoint defaulting logic.

When called by the Export Expense Report Application, the transformer creates an ASCII file that contains the records for the batch exported.

Note: The program creates the ASCII file in the Export directory. The name of the file is "<Batch ID>.exp" where <Batch ID>is the ID supplied on the Export Expense Reports screen.

Note: You can find layouts of ASCII files created by the expense transformers in “Transformed Export ASCII Layout” chapter of this guide.

Import ASCII Layouts

Overview

Version

Deltek Time & Expense was designed to retain compatibility with previous ASCII file layouts as existing layouts are modified. The system accomplishes this by placing a version number in the first column in the ASCII file. The version number starts with 1 and is incremented any time we modify the import layout. Version 0 is assumed when files are named according to Deltek Time 3.x/4.x standards. In Expense imports, version 0 is used when you import from a third party.

File Name Convention

The import file name must adhere to Windows® file-system naming conventions, and must be readable by the import program (i.e., it must be located in a suitable directory and have appropriate permissions). The file must follow the following naming convention:

XXXXXXXXXX_YYYYMMDDHHMMSS.CSV

- XXXXXXXXXX — An acronym for the file being imported. For example, "EMPL" for employee.
- YYYYMMDDHHMMSS — A time stamp that indicates when the file was created. The format is year/month/day/hours/minutes/seconds. For example: 20011002125959

Note: You can use older file names if the file is in the Deltek Time 3.x/4.x format.

In addition to the above, you can import currency schedules, per diem schedules, and receipt images using the following naming conventions. Please see the detailed layout later in this chapter for more information.

Manage Currency Schedules

CURRENCY_SCHEDULE_YYY_XXXXXXXXXX_YYYYMMDD.TXT

- YYY — The base currency.
- XXXXXXXXXX — The currency schedule code. The length of the code varies from 1 to 10 positions.
- YYYYMMDD — The effective date for the schedule.

Manage Per Diem Schedules

You can import different files for the per diem schedules. They are based on the government's Continental United States (CONUS) and Outside Continental United States (OCONUS) rates.

- CONNOW.TXT — Military and Civilian
- CONUSNM.TXT — Civilian
- CONUSMIL.TXT — Military
- OCONUS.TXT — Military and Civilian
- OCONUSNM.TXT — Civilian

Fax Attachments

You can import .PDF images of receipts for expense reports.

> **Note:** The PDF must contain all attachments identified on the cover sheet which was printed. If later .PDF is imported for an expense report, it will replace the previous image.

Additionally, employees may manually attach other documents and images to the expense report, expenses, or expense authorization.

Character Content

The data in the import files must contain only printable ASCII characters. You cannot use control characters, non-printable characters, or "packed" data fields.

One record per line

Each line of the import file constitutes a single record, or row, in the database. You must terminate lines with a carriage return and line feed.

Field Separation

Separate the fields in the record with a comma. There are two exceptions: commas are not required for the per diem schedules import or the version 0 layout of the currency schedules import. Fields in the government's CONUS and OCONUS files are separated by a ";".

Field Length

The records in an import file are of variable length. The fields are only as wide as the data requires (there are no trailing spaces between field data and delimiter characters). The version 0 layout of the currency schedules import is an exception.

Field Types

Each record in the import file contains one or more data fields, each separated by the comma delimiter. Data fields within the record are of a specified data type; there are format and value restrictions for each of the supported data types, as outlined below:

Character

- Format — Alphanumeric printable characters.
- Restrictions — Less than or equal to specified length. Excess characters are deleted.

Smallint

- Format — Numeric characters (0-9) and optional leading sign indicator (+ or -). A positive sign (+) is assumed if a specific sign indicator is not present.
- Restrictions — Values within the range -32,767 to +32,767. Out-of-range values cause import errors.

Integer

- Format — Numeric characters (0-9) and optional leading sign indicator (+ or -). A positive sign is assumed if a specific sign indicator is not present.
- Restrictions — Values within the range -2,147,483,647 to +2,147,483,647. Out-of-range values cause import errors.

Numeric

- Format — Numeric characters (0-9), optional leading sign indicator (+ or -), and decimal point indicator (.). A positive sign is assumed if a specific sign indicator is not present. Decimal fields are stated in terms of precision (total number of digits) and scale (fields to the right of the decimal point). For example, Numeric (14,2) means 12 digits to the left and 2 digits to the right of the decimal point. The decimal point is optional (assumes zeros to the right of the decimal point). If you provide fewer digits after the decimal point than specified for the field, trailing zeros are assumed.
- Restrictions — Values within the range are determined by the field specification. For example, Numeric (14,2) can have 12 or fewer digits to the right of the imported (or implied) decimal point and 2 or fewer digits to the right of the decimal point. Out-of-range values cause import errors.

Date

- Format — You can import dates in the following formats: mm/dd/yyyy or mm-dd-yyyy.
- Restrictions — Invalid dates (for example, 12/32/1995) cause import errors.
- Data fields with null values (the lack of a value) are specified with adjacent comma delimiter.
- Dates for version 0 ASCII layouts must be in YYYYMMDD format, with the exception of Per Diem Schedules, which are in MM/DD/YYYY format.

Employee Layout

The following section contains the layouts for the Employee ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the Employee ASCII file, ET_EMPL.TXT.

Field	Data Type	Description	Notes
1	CHAR (12)	Employee ID	
2	CHAR (30)	Last Name	
3	CHAR (30)	First Name	
4	CHAR (1)	Middle Initial	
5	CHAR (11)	Government ID	
6	CHAR (12)	Vendor ID	
7	CHAR (8)	Hire Date	YYYYMMDD

Field	Data Type	Description	Notes
8	CHAR (8)	Termination Date	YYYYMMDD
9	CHAR (10)	N/A	
10	CHAR (10)	User-Defined Code #1	
11	CHAR (10)	User-Defined Code #2	
12	CHAR (8)	User-Defined Date #1	YYYYMMDD
13	CHAR (8)	User-Defined Date #2	YYYYMMDD
14	NUMERIC (14,2)	N/A	
15	NUMERIC (14,2)	N/A	
16	CHAR (1)	Active Flag	"Y" or "N"
17	CHAR (254)	E-mail Address	Internet address
18	CHAR (15)	N/A	
19	CHAR (12)	Payroll Employee ID	
20	CHAR (20)	N/A	
21	CHAR (30)	Home Phone Number	
22	CHAR (30)	Work Phone Number	
23	CHAR (30)	Fax Number	
24	CHAR (30)	Pager number	
25	CHAR (20)	Login ID	

Version 1

The following table contains the layout of the Employee ASCII file, EMPL_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1." This number will be incremented as the layout is modified.
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add C — Change D — Delete B — Batch
3	CHAR (20)	Employee ID	
4	CHAR (30)	Last Name	
5	CHAR (30)	First Name	
6	CHAR (1)	Middle Initial	
7	CHAR (20)	Government ID	

Field	Data Type	Description	Notes
8	CHAR (12)	Vendor ID	
9	CHAR (20)	Payroll ID	
10	CHAR (20)	Payroll Employee ID	
11	DATE	Hire Date	
12	DATE	Termination Date	
13	CHAR (30)	Home Phone Number	
14	CHAR (30)	Work Phone Number	
15	CHAR (30)	Fax Number	
16	CHAR (30)	Mobile Phone Number	
17	CHAR (30)	Pager Number	
18	CHAR (254)	E-mail Address	Internet address
19	CHAR (20)	Login ID	
20	CHAR (10)	Security Role Code	
21	CHAR (20)	User-Defined Code #1	
22	CHAR (20)	User-Defined Code #2	
23	CHAR (20)	User-Defined Code #3	
24	DATE	User-Defined Date #1	
25	DATE	User-Defined Date #2	
26	DATE	User-Defined Date #3	
27	CHAR (1)	Active Flag	"Y" or "N"

Version 2

The following table contains the layout of the Employee ASCII file, EMPL_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2." This number will be incremented as the layout is modified.

Field	Data Type	Description	Notes
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add C — Change D — Delete B — Batch
3	CHAR (20)	Employee ID	
4	CHAR (30)	Last Name	
5	CHAR (30)	First Name	
6	CHAR (1)	Middle Initial	
7	CHAR (20)	Government ID	
8	CHAR (12)	Vendor ID	
9	CHAR (20)	Payroll ID	
10	CHAR (20)	Payroll Employee ID	
11	DATE	Hire Date	
12	DATE	Termination Date	
13	CHAR (30)	Home Phone Number	
14	CHAR (30)	Work Phone Number	
15	CHAR (30)	Fax Number	
16	CHAR (30)	Mobile Phone Number	
17	CHAR (30)	Pager Number	
18	CHAR (254)	E-mail Address	Internet address
19	CHAR (20)	Login ID	
20	CHAR (10)	Security Role Code	
21	CHAR (20)	User-Defined Code #1	
22	CHAR (20)	User-Defined Code #2	
23	CHAR (20)	User-Defined Code #3	
24	DATE	User-Defined Date #1	
25	DATE	User-Defined Date #2	
26	DATE	User-Defined Date #3	

Field	Data Type	Description	Notes
27	CHAR (1)	Active Flag	"Y" or "N"
28	CHAR (3)	Currency Code	

Version 3

The following table contains the layout of the Employee ASCII file, EMPL_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "3."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add C — Change D — Delete B — Batch
3	CHAR (20)	Employee ID	
4	CHAR (30)	Last Name	
5	CHAR (30)	First Name	
6	CHAR (1)	Middle Initial	
7	CHAR (20)	Government ID	
8	CHAR (12)	Vendor ID	
9	CHAR (20)	Payroll ID	
10	CHAR (20)	Payroll Employee ID	
11	DATE	Hire Date	
12	DATE	Termination Date	
13	CHAR (30)	Home Phone Number	
14	CHAR (30)	Work Phone Number	
15	CHAR (30)	Fax Number	
16	CHAR (30)	Mobile Phone Number	
17	CHAR (30)	Pager Number	
18	CHAR (254)	E-mail Address	Internet address
19	CHAR (20)	Login ID	
20	CHAR (10)	Security Role Code	
21	CHAR (20)	User-Defined Code #1	

Field	Data Type	Description	Notes
22	CHAR (20)	User-Defined Code #2	
23	CHAR (20)	User-Defined Code #3	
24	DATE	User-Defined Date #1	
25	DATE	User-Defined Date #2	
26	DATE	User-Defined Date #3	
27	CHAR (1)	Active Flag	"Y" or "N"
28	CHAR (3)	Currency Code	
29	CHAR(30)	Location	
30	CHAR(30)	Title	

Notes

For Add transactions, the following fields are required:

- Employee ID
- Last Name
- First Name
- Government ID

For Change transactions, the following field is required:

- Employee ID

For Delete transactions, the following field is required:

- Employee ID

For Batch transactions, the following fields are required (they cannot be null):

- Employee ID
- Last Name
- First Name
- Government ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- Login ID — Default Employee ID
- Security Role — Default depends on system configuration.
- Active Flag — Default is "Y"
- Currency Code — Default depends on Deltek Expense configuration.

Employee History Layout

The following section contains the layouts for the Employee History ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the Employee History ASCII file, ET_EMPLH.TXT.

Field	Data Type	Description	Notes
1	CHAR (12)	Employee ID	
2	CHAR (8)	Effective Date	YYYYMMDD
3	CHAR (10)	Employee Class Code	
4	CHAR (4)	Timesheet Schedule Code	
5	CHAR (30)	Default UDT01	
6	CHAR (30)	Default UDT02	
7	CHAR (20)	Default UDT03	
8	CHAR (20)	Default UDT04	
9	CHAR (20)	Default UDT05	
10	CHAR (20)	Default UDT06	
11	CHAR (20)	Default UDT07	
12	CHAR (20)	Default UDT08	
13	CHAR (20)	Default UDT09	
14	CHAR (20)	Default UDT10	
15	NUMERIC (15,5)	N/A	
16	NUMERIC (15,5)	N/A	

Version 1

The following table contains the layout of the Employee History ASCII file, EMPL_HISTORY_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add C — Change D — Delete B — Batch
3	CHAR (20)	Employee ID	

Field	Data Type	Description	Notes
4	DATE	Effective Date	
5	CHAR (10)	Work Schedule Code	
6	CHAR (10)	Employee Class Code	
7	CHAR (4)	Timesheet Schedule Code	
8	CHAR (50)	Default UDT01	
9	CHAR (50)	Default UDT02	
10	CHAR (20)	Default UDT03	
11	CHAR (20)	Default UDT04	
12	CHAR (20)	Default UDT05	
13	CHAR (20)	Default UDT06	
14	CHAR (20)	Default UDT07	
15	CHAR (20)	Default UDT08	
16	CHAR (20)	Default UDT09	
17	CHAR (20)	Default UDT10	

Version 2

The following table contains the layout of the Employee History ASCII file, EMPL_HISTORY_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add C — Change D — Delete B — Batch
3	CHAR (20)	Employee ID	
4	DATE	Effective Date	
5	CHAR (10)	Work Schedule Code	
6	CHAR (10)	Timesheet Class Code	
7	CHAR (4)	Timesheet Schedule Code	
8	CHAR (50)	Default UDT01	
9	CHAR (50)	Default UDT02	

Field	Data Type	Description	Notes
10	CHAR (20)	Default UDT03	
11	CHAR (20)	Default UDT04	
12	CHAR (20)	Default UDT05	
13	CHAR (20)	Default UDT06	
14	CHAR (20)	Default UDT07	
15	CHAR (20)	Default UDT08	
16	CHAR (20)	Default UDT09	
17	CHAR (20)	Default UDT10	
18	CHAR (10)	Costpoint Company Code	
19	CHAR (10)	UDT06's Costpoint Company Code	
20	CHAR (10)	UDT07's Costpoint Company Code	
21	NUMERIC (15,5)	User-Defined Rate 1	
22	NUMERIC (15,5)	User-Defined Rate 2	
23	NUMERIC (15,5)	Fringe Reduction Rate	
24	CHAR (20)	Expense Class Code	

Version 3

The following table contains the layout of the Employee History ASCII file, EMPL_HISTORY_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "3."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add C — Change D — Delete B — Batch
3	CHAR (20)	Employee ID	
4	DATE	Effective Date	
5	CHAR (10)	Work Schedule Code	

Field	Data Type	Description	Notes
6	CHAR (10)	Timesheet Class Code	
7	CHAR (4)	Timesheet Schedule Code	
8	CHAR (50)	Default UDT01	
9	CHAR (50)	Default UDT02	
10	CHAR (20)	Default UDT03	
11	CHAR (20)	Default UDT04	
12	CHAR (20)	Default UDT05	
13	CHAR (20)	Default UDT06	
14	CHAR (20)	Default UDT07	
15	CHAR (20)	Default UDT08	
16	CHAR (20)	Default UDT09	
17	CHAR (20)	Default UDT10	
18	CHAR (10)	Costpoint Company Code	
19	CHAR (10)	UDT06's Costpoint Company Code	
20	CHAR (10)	UDT07's Costpoint Company Code	
21	NUMERIC (15,5)	User-Defined Rate 1	
22	NUMERIC (15,5)	User-Defined Rate 2	
23	NUMERIC (15,5)	Fringe Reduction Rate	
24	CHAR (20)	Expense Class Code	
25	CHAR(20)	UDT11	
26	CHAR(20)	UDT12	
27	CHAR(20)	UDT13	
28	CHAR(20)	UDT14	
29	CHAR(20)	UDT15	

Notes

For all transactions, the following fields are required:

- Employee ID
- Effective Date

For Add transactions, the system will use the following defaulting logic if the column is blank:

- Work Schedule — Default depends on Deltek Time configuration.
- Timesheet Schedule — Default depends on Time configuration.
- Timesheet Class — Default depends on Time configuration.
- Costpoint Company — Set to "1" if you are not using the Costpoint multicompany feature.
- UDT06 Costpoint Company
  - Set to "1" if you are not using the Costpoint multicompany feature.
  - If you are using the Costpoint multicompany feature and a default UDT06 exists, set to the associated UDT06 Costpoint company.
  - If you are using the Costpoint multicompany feature and a default UDT06 does not exist, set to null.
- UDT07 Costpoint Company
  - Set to "1" if you are not using the Costpoint multicompany feature.
  - If you are using the Costpoint multicompany feature and a default UDT07 exists, set to the associated UDT07 Costpoint company.
  - If you are using the Costpoint multicompany feature and a default UDT07 does not exist, set to null.
- User-Defined Rate 1 — "0.00"
- User-Defined Rate 2 — "0.00"
- Fringe Reduction Rate — "0.00"
- Expense Class Code — Default depends on Deltek Expense configuration.

Employee Leave Layout

The following section contains the layouts for the Employee Leave ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the Employee Leave ASCII file, ET_EMPLL.TXT.

Field	Data Type	Description	Notes
1	CHAR (12)	Employee ID	
2	CHAR (30)	Leave Type Code	
3	CHAR (15)	Transaction Type Code	Valid Values: BEG ACCRUED
4	CHAR (8)	Transaction Date	YYYYMMDD
5	NUMERIC (10,4)	Leave Hours	

Version 1

The following table contains the layout of the Employee Leave ASCII file, EMPL_LEAVE_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add C — Change D — Delete B — Batch
3	CHAR (20)	Employee ID	
4	CHAR (10)	Leave Type Code	
5	DATE	Transaction Date	
6	CHAR (10)	Transaction Type Code	The type of transaction. The valid values are: BEG ACCRUED ADJUSTMENT
7	NUMERIC (10,4)	Leave Hours	
8	CHAR (60)	Adjustment Text	

Notes

For all transactions, the following fields are required:

- Employee ID
- Leave Type
- Transaction Type Code
- Transaction Date

For Adjustment transactions, the following field is required:

- Adjustment Text

Employee Groups Layout

The following section contains the layouts for the Employee Groups ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 1

The following table contains the layout of the Employee Groups ASCII file, EMPL_GROUP_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add C — Change D — Delete B — Batch
3	CHAR (25)	Group Code	
4	CHAR (1)	Import Type	Indicates the type of record being imported. The valid values are: G — Group M — Employee O — Supervisor
5	CHAR (30)	Group Description	
6	CHAR (10)	Group Type	
7	CHAR (20)	Member's Employee ID	
8	CHAR (20)	Supervisor's Employee ID	
9	CHAR (10)	Supervisor Type	The type of supervisor. The valid options are as follows: PSPVSR — Primary Supervisor BSPVSR — Backup Supervisor PADMIN — Primary Administrator BADMIN — Primary Administrator
10	CHAR (1)	Modify Timesheet Flag	
11	CHAR (1)	Approve Flag	

Notes

For Add Transactions, the following fields are required:

- Import Type = "G"
  - Group Code
  - Group Description
  - Group Type Code

- Import Type = "M"
  - Group Code
  - Employee ID
- Import Type = "O"
  - Group Code
  - Supervisor Employee ID
  - Supervisor Type

For Change Transactions, the following fields are required:

- Import Type = "G"
  - Group Code
- Import Type = "M"
  - Group Code
  - Employee ID
- Import Type = "O"
  - Group Code
  - Supervisor Employee ID
  - Supervisor Type

For Delete Transactions, the following fields are required:

- Import Type = "G"
  - Group Code
- Import Type = "M"
  - Group Code
  - Employee ID
- Import Type = "O"
  - Group Code
  - Supervisor Employee ID

For Batch Transactions, the following fields are required:

- Import Type = "G"
  - Group Code
  - Group Description
  - Group Type Code
- Import Type = "M"
  - Group Code
  - Employee ID
- Import Type = "O"
  - Group Code

  - Supervisor Employee ID
  - Supervisor Type

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- Import Type = "G"
  - None
- Import Type = "M"
  - None
- Import Type = "O"
  - Supervisor Type — "PSPVSR"
  - Modify Timesheet Flag – "N"
  - Approve Flag = "Y"

Version 2

The following table contains the layout of the Employee Groups ASCII file, EMPL_GROUP_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add C — Change D — Delete B — Batch
3	CHAR (25)	Group Code	Identifying code for the employee group.
4	CHAR (1)	Import Type	Indicates the type of record being imported. The valid values are: G — Group M — Employee O — Supervisor
5	CHAR (30)	Group Description	Description of the employee group.
6	CHAR (10)	Group Type	The type of the employee group.
7	CHAR (20)	Member's Employee ID	Employee ID of a member of the employee group.
8	CHAR (20)	Supervisor's Employee ID	Employee ID of a supervisor of the employee group.

Field	Data Type	Description	Notes
9	CHAR (10)	Functional Role	The type of functional role the supervisor holds.

Notes

For Add Transactions, the following fields are required:

- Import Type = "G"
  - Group Code
  - Group Description
  - Group Type Code
- Import Type = "M"
  - Group Code
  - Employee ID
- Import Type = "O"
  - Group Code
  - Supervisor Employee ID

For Change Transactions, the following fields are required:

- Import Type = "G"
  - Group Code
- Import Type = "M"
  - Group Code
  - Employee ID
- Import Type = "O"
  - Group Code
  - Supervisor Employee ID

For Delete Transactions, the following fields are required:

- Import Type = "G"
  - Group Code
- Import Type = "M"
  - Group Code
  - Employee ID
- Import Type = "O"
  - Group Code
  - Supervisor Employee ID

For Batch Transactions, the following fields are required:

- Import Type = "G"

    - Group Code
    - Group Description
    - Group Type Code
- Import Type = "M"
    - Group Code
    - Employee ID
- Import Type = "O"
    - Group Code
    - Supervisor Employee ID

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- Import Type = "G"
    - None
- Import Type = "M"
    - None
- Import Type = "O"
    - Functional Role — "PSPVSR"

UDT01 Layout

The following section contains the layouts for the UDT01 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT01 ASCII file, ET_UDT01.TXT.

Field	Data Type	Description	Notes
1	CHAR (30)	UDT01 ID	
2	CHAR (20)	UDT01 Abbreviation	
3	CHAR (120)	UDT01 Name	
4	CHAR (10)	UDT01 Type	
5	CHAR (1)	UDT02 Link Flag	
6	CHAR (1)	UDT07 Link Flag	"Y" or "N"
7	CHAR (1)	UDT09 Link Flag	"Y" or "N"
8	CHAR (1)	UDT07 Required Flag	"Y" or "N"
9	CHAR (8)	Start Date	YYYYMMDD
10	CHAR (8)	End Date	YYYYMMDD
11	CHAR (1)	Active Flag	"Y" or "N"

Field	Data Type	Description	Notes
12	CHAR (1)	N/A	

Version 1

The following table contains the layout of the UDT01 ASCII file, UDT01_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT01 ID	
4	CHAR (120)	UDT01 Name	
5	CHAR (10)	UDT01 Type	
6	CHAR (20)	User-Defined Code #1	
7	CHAR (20)	User-Defined Code #2	
8	CHAR (20)	User-Defined Code #3	
9	DATE	User-Defined Date #1	
10	DATE	User-Defined Date #2	
11	DATE	User-Defined Date #3	
12	CHAR (1)	UDT02 Link Flag	"Y" or "N"
13	CHAR (1)	UDT07 Link Flag	"Y" or "N"
14	CHAR (1)	UDT09 Link Flag	"Y" or "N"
15	CHAR (1)	UDT07 Required Flag	"Y" or "N"
16	DATE	Start Date	
17	DATE	End Date	
18	CHAR (1)	Active Flag	"Y" or "N"
19	CHAR (1)	Allow Charging Flag	"Y" or "N"
20	DATE	Inactive Date	

Field	Data Type	Description	Notes
21	CHAR (1)	Date Edit Type	The types of edit that will be performed on start and end dates. The valid options are: H — Hard edit S — Soft edit N — No Edit

Notes

For Add/Batch transactions, the following fields are required:

- UDT01 ID
- UDT01 Name

For Change/Delete transactions, the following field is required:

- UDT01 ID

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- UDT01 Type Code — Default depends on system configuration.
- Date Edit Type — Default depends on system configuration.
- UDT02 Link Flag — "N"
- UDT07 Link Flag — "N"
- UDT07 Required Flag — "N"
- Active Flag — "Y"
- Allow Charging Flag — "Y"

Version 2

The following table contains the layout of the UDT01 ASCII file, UDT01_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT01 ID	
4	CHAR (120)	UDT01 Name	
5	CHAR (10)	UDT01 Type	
6	CHAR (20)	User-Defined Code #1	
7	CHAR (20)	User-Defined Code #2	

Field	Data Type	Description	Notes
8	CHAR (20)	User-Defined Code #3	
9	DATE	User-Defined Date #1	
10	DATE	User-Defined Date #2	
11	DATE	User-Defined Date #3	
12	CHAR (1)	UDT02 Link Flag	"Y" or "N"
13	CHAR (1)	UDT07 Link Flag	"Y" or "N"
14	CHAR (1)	UDT09 Link Flag	"Y" or "N"
15	CHAR (1)	UDT07 Required Flag	"Y" or "N"
16	DATE	Start Date	
17	DATE	End Date	
18	CHAR (1)	Active Flag	"Y" or "N"
19	CHAR (1)	Allow Charging Flag	"Y" or "N"
20	DATE	Inactive Date	
21	CHAR (1)	Date Edit Type	The types of edit that will be performed on start and end dates. The valid options are: H — Hard edit S — Soft edit N — No Edit
22	CHAR (1)	Line Approve Flag	"Y" or "N"

Version 3

The following table contains the layout of the UDT01 ASCII file, UDT01_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "3."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT01 ID	
4	CHAR (120)	UDT01 Name	
5	CHAR (10)	UDT01 Type	

Field	Data Type	Description	Notes
6	CHAR (20)	User-Defined Code #1	
7	CHAR (20)	User-Defined Code #2	
8	CHAR (20)	User-Defined Code #3	
9	DATE	User-Defined Date #1	
10	DATE	User-Defined Date #2	
11	DATE	User-Defined Date #3	
12	CHAR (1)	UDT02 Link Flag	"Y" or "N"
13	CHAR (1)	UDT07 Link Flag	"Y" or "N"
14	CHAR (1)	UDT09 Link Flag	"Y" or "N"
15	CHAR (1)	UDT07 Required Flag	"Y" or "N"
16	DATE	Start Date	
17	DATE	End Date	
18	CHAR (1)	Active Flag	"Y" or "N"
19	CHAR (1)	Allow Charging Flag	"Y" or "N"
20	DATE	Inactive Date	
21	CHAR (1)	Date Edit Type	The types of edit that will be performed on start and end dates. The valid options are: H — Hard edit S — Soft edit N — No Edit
22	CHAR (1)	Line Approve Flag	"Y" or "N"
23	CHAR (20)	UDT01 Abbreviation	

Notes

For Add/Batch transactions, the following fields are required:

- UDT01 ID
- UDT01 Name

For Change/Delete transactions, the following field is required:

- UDT01 ID

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- UDT01 Type Code — Default depends on system configuration.
- Date Edit Type — Default depends on system configuration.
- UDT02 Link Flag — "N"
- UDT07 Link Flag — "N"
- UDT07 Required Flag — "N"
- Active Flag — "Y"
- Allow Charging Flag — "Y"
- Line Approve Flag — "N"

For Add/Batch/Change transactions:

- If UDT07 Link flag = "Y," UDT07 required flag will be set to "Y."

UDT02 Layout

The following section contains the layouts for the UDT02 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT02 ASCII file, ET_UDT02.TXT.

Field	Data Type	Description	Notes
1	CHAR (30)	UDT02 ID	
2	CHAR (20)	UDT02 Abbreviation	
3	CHAR (120)	UDT02 Name	
4	CHAR (10)	N/A	
5	CHAR (1)	UDT02 Link Flag	"Y" or "N"
6	CHAR (1)	UDT07 Link Flag	"Y" or "N"
7	CHAR (1)	UDT09 Link Flag	"Y" or "N"
8	CHAR (1)	UDT07 Required Flag	"Y" or "N"
9	CHAR (8)	Start Date	YYYYMMDD
10	CHAR (8)	End Date	YYYYMMDD
11	CHAR (1)	Active Flag	"Y" or "N"
12	CHAR (1)	N/A	
13	CHAR (20)	N/A	

Version 1

The following table contains the layout of the UDT02 ASCII file, UDT02_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT02 ID	
4	CHAR (120)	UDT02 Name	
5	CHAR (20)	User-Defined Code #1	
6	CHAR (20)	User-Defined Code #2	
7	CHAR (20)	User-Defined Code #3	
8	DATE	User-Defined Date #1	
9	DATE	User-Defined Date #2	
10	DATE	User-Defined Date #3	
11	CHAR (1)	UDT02 Link Flag	"Y" or "N"
12	CHAR (1)	UDT07 Link Flag	"Y" or "N"
13	CHAR (1)	UDT09 Link Flag	"Y" or "N"
14	CHAR (1)	UDT07 Required Flag	"Y" or "N"
15	DATE	Start Date	
16	DATE	End Date	
17	CHAR (1)	Active Flag	"Y" or "N"
18	CHAR (1)	Allow Charging Flag	"Y" or "N"
19	DATE	Inactive Date	
20	CHAR (1)	Date Edit Type	The types of edit that will be performed on start and end dates. The valid options are: H — Hard edit S — Soft edit N — No Edit

Notes

For Add/Batch transactions, the following fields are required:

- UDT02 ID

- UDT02 Name

For Change/Delete transactions, the following field is required:

- UDT02 ID

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- Date Edit Type — Default depends on the system configuration.
- UDT01 Link Flag — "N"
- UDT07 Link Flag — "N"
- UDT07 Required Flag — "N"
- Active Flag — "Y"
- Allow Charging Flag — "Y"

Version 2

The following table contains the layout of the UDT02 ASCII file, UDT02_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT02 ID	
4	CHAR (120)	UDT02 Name	
5	CHAR (20)	User-Defined Code #1	
6	CHAR (20)	User-Defined Code #2	
7	CHAR (20)	User-Defined Code #3	
8	DATE	User-Defined Date #1	
9	DATE	User-Defined Date #2	
10	DATE	User-Defined Date #3	
11	CHAR (1)	UDT02 Link Flag	"Y" or "N"
12	CHAR (1)	UDT07 Link Flag	"Y" or "N"
13	CHAR (1)	UDT09 Link Flag	"Y" or "N"
14	CHAR (1)	UDT07 Required Flag	"Y" or "N"
15	DATE	Start Date	
16	DATE	End Date	

Field	Data Type	Description	Notes
17	CHAR (1)	Active Flag	"Y" or "N"
18	CHAR (1)	Allow Charging Flag	"Y" or "N"
19	DATE	Inactive Date	
20	CHAR (1)	Date Edit Type	The types of edit that will be performed on start and end dates. The valid options are: H — Hard edit S — Soft edit N — No Edit
21	CHAR (1)	Line Approve Flag	"Y" or "N"

Version 3

The following table contains the layout of the UDT02 ASCII file, UDT02_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "3."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT02 ID	
4	CHAR (120)	UDT02 Name	
5	CHAR (20)	User-Defined Code #1	
6	CHAR (20)	User-Defined Code #2	
7	CHAR (20)	User-Defined Code #3	
8	DATE	User-Defined Date #1	
9	DATE	User-Defined Date #2	
10	DATE	User-Defined Date #3	
11	CHAR (1)	UDT02 Link Flag	"Y" or "N"
12	CHAR (1)	UDT07 Link Flag	"Y" or "N"
13	CHAR (1)	UDT09 Link Flag	"Y" or "N"
14	CHAR (1)	UDT07 Required Flag	"Y" or "N"

Field	Data Type	Description	Notes
15	DATE	Start Date	
16	DATE	End Date	
17	CHAR (1)	Active Flag	"Y" or "N"
18	CHAR (1)	Allow Charging Flag	"Y" or "N"
19	DATE	Inactive Date	
20	CHAR (1)	Date Edit Type	The types of edit that will be performed on start and end dates. The valid options are: H — Hard edit S — Soft edit N — No Edit
21	CHAR (1)	Line Approve Flag	"Y" or "N"
22	CHAR (10)	Costpoint Company Code	
23	CHAR (20)	UDT02 Abbreviation	

Notes

For Add/Batch transactions, the following fields are required:

- UDT02 ID
- UDT02 Name

For Change/Delete transactions, the following field is required:

- UDT02 ID

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- Date Edit Type — Default depends on the system configuration.
- UDT01 Link Flag — "N"
- UDT07 Link Flag — "N"
- UDT07 Required Flag — "N"
- Active Flag — "Y"
- Allow Charging Flag — "Y"
- Line Approve Flag — "N"
- Costpoint Company Code — Set to "1" if you are not using the Costpoint multicompany feature.

For Add/Batch/Change transactions:

- If UDT07 Link flag = "Y," UDT07 required flag will be set to "Y."

Version 4

The following table contains the layout of the UDT02 ASCII file, UDT02_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "4."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT02 ID	
4	CHAR (120)	UDT02 Name	
5	CHAR (20)	User-Defined Code #1	
6	CHAR (20)	User-Defined Code #2	
7	CHAR (20)	User-Defined Code #3	
8	DATE	User-Defined Date #1	
9	DATE	User-Defined Date #2	
10	DATE	User-Defined Date #3	
11	CHAR (1)	UDT02 Link Flag	"Y" or "N"
12	CHAR (1)	UDT07 Link Flag	"Y" or "N"
13	CHAR (1)	UDT09 Link Flag	"Y" or "N"
14	CHAR (1)	UDT07 Required Flag	"Y" or "N"
15	DATE	Start Date	
16	DATE	End Date	
17	CHAR (1)	Active Flag	"Y" or "N"
18	CHAR (1)	Allow Charging Flag	"Y" or "N"
19	DATE	Inactive Date	
20	CHAR (1)	Date Edit Type	The types of edit that will be performed on start and end dates. The valid options are: H — Hard edit S — Soft edit N — No Edit

Field	Data Type	Description	Notes
21	CHAR (1)	Line Approve Flag	"Y" or "N"
22	CHAR (10)	Costpoint Company Code	
23	CHAR (20)	UDT02 Abbreviation	
24	CHAR (1)	Type	T- Time, E — Expense, B — Both

Notes

For Add/Batch transactions, the following fields are required:

- UDT02 ID
- UDT02 Name

For Change/Delete transactions, the following field is required:

- UDT02 ID

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- Date Edit Type — Default depends on the system configuration.
- UDT01 Link Flag — "N"
- UDT07 Link Flag — "N"
- UDT07 Required Flag — "N"
- Active Flag — "Y"
- Allow Charging Flag — "Y"
- Line Approve Flag — "N"
- Costpoint Company Code — Set to "1" if you are not using the Costpoint multicompany feature.
- Type — "B"
- For Add/Batch/Change transactions:
- If UDT07 Link flag = "Y," UDT07 required flag will be set to "Y."

UDT03 Layout

The following section contains the layouts for the UDT03 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT03 ASCII file, ET_UDT03.TXT.

Field	Data Type	Description	Notes
1	CHAR (20)	UDT03 ID	

Field	Data Type	Description	Notes
2	CHAR (30)	UDT03 Name	
3	CHAR (1)	Active Flag	"Y" or "N"

Version 1

The following table contains the layout of the UDT03 ASCII file, UDT03_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT03 ID	
4	CHAR (30)	UDT03 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Notes

For Add/Batch transactions, the following fields are required:

- UDT03 ID
- UDT03 Name

For Change/Delete transactions, the following field is required:

- UDT03 ID

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- Active Flag — "Y"

UDT04 Layout

The following section contains the layouts for the UDT04 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT04 ASCII file, ET_UDT04.TXT.

Field	Data Type	Description	Notes
1	CHAR (20)	UDT04 ID	
2	CHAR (30)	UDT04 Name	
3	CHAR (1)	Active Flag	"Y" or "N"

Version 1

The following table contains the layout of the UDT04 ASCII file, UDT04_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT04 ID	
4	CHAR (30)	UDT04 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Version 2

The following table contains the layout of the UDT04 ASCII file, UDT04_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT04 ID	
4	CHAR (30)	UDT04 Name	
5	CHAR (1)	Active Flag	"Y" or "N"
6	CHAR (10)	Costpoint Company Code	
7	NUMERIC (15,5)	UDT04 Rate	

Notes

For Add/Batch transactions, the following fields are required:

- UDT04 ID
- UDT04 Name

For Change/Delete transactions, the following field is required:

- UDT04 ID

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- Active Flag — "Y"
- Costpoint Company Code — Set to "1" if you are not using the Costpoint multicompany feature.
- UDT04 Rate — "0.00"

UDT05 Layout

The following section contains the layouts for the UDT05 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT05 ASCII file, ET_UDT05.TXT.

Field	Data Type	Description	Notes
1	CHAR (20)	UDT05 ID	
2	CHAR (30)	UDT05 Name	
3	CHAR (1)	Active Flag	"Y" or "N"

Version 1

The following table contains the layout of the UDT05 ASCII file, UDT05_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT05 ID	
4	CHAR (30)	UDT05 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Version 2

The following table contains the layout of the UDT05 ASCII file, UDT05_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT05 ID	
4	CHAR (30)	UDT05 Name	
5	CHAR (1)	Active Flag	"Y" or "N"
6	CHAR (10)	Costpoint Company Code	

Notes

For Add/Batch transactions, the following fields are required:

- UDT05 ID
- UDT05 Name
- For Change/Delete transactions, the following field is required:
- UDT05 ID
- For Add transactions, the system will use the following defaulting logic if the column is blank:
- Active Flag — "Y"
- Costpoint Company Code — Set to "1" if you are not using the Costpoint multicompany feature.

UDT06 Layout

The following section contains the layouts for the UDT06 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT06 ASCII file, ET_UDT06.TXT.

Field	Data Type	Description	Notes
1	CHAR (20)	UDT06 ID	
2	CHAR (30)	UDT06 Name	
3	CHAR (1)	Active Flag	"Y" or "N"

Version 1

The following table contains the layout of the UDT06 ASCII file, UDT06_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT06 ID	
4	CHAR (30)	UDT06 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Version 2

The following table contains the layout of the UDT06 ASCII file, UDT06_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."

Field	Data Type	Description	Notes
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT06 ID	
4	CHAR (30)	UDT06 Name	
5	CHAR (1)	Active Flag	"Y" or "N"
6	CHAR (10)	Costpoint Company Code	

Notes

For Add/Batch transactions, the following fields are required:

- UDT06 ID
- UDT06 Name
- Costpoint Company Code
- For Change/Delete transactions, the following fields are required:
- UDT06 ID
- Costpoint Company Code

For Add transactions, the system will provide the following default if the column is blank:

- Active Flag — "Y"

UDT07 Layout

The following section contains the layouts for the UDT07 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT07 ASCII file, ET_UDT07.TXT.

Field	Data Type	Description	Notes
1	CHAR (20)	UDT07 ID	
2	CHAR (30)	UDT07 Name	
3	NUMERIC (15,5)	N/A	
4	CHAR (1)	Active Flag	"Y" or "N"
5	NUMERIC (15,5)	N/A	

Version 1

The following table contains the layout of the UDT07 ASCII file, UDT07_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT07 ID	
4	CHAR (30)	UDT07 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Version 2

The following table contains the layout of the UDT07 ASCII file, UDT07_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT07 ID	
4	CHAR (30)	UDT07 Name	
5	CHAR (1)	Active Flag	"Y" or "N"
6	CHAR (10)	Costpoint Company Code	
7	NUMERIC (15,5)	Rate 1	
8	NUMERIC (15,5)	Rate 2	

Notes

For Add/Batch transactions, the following fields are required:

- UDT07 ID
- UDT07 Name
- Costpoint Company Code

For Change/Delete transactions, the following fields are required:

- UDT07 ID
- Costpoint Company Code

For Add transactions, the system will use the following defaulting logic if the column is blank:

- Active Flag — "Y"
- Rate 1 — "0.00"
- Rate 2 — "0.00"

UDT08 Layout

The following section contains the layouts for the UDT08 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT08 ASCII file, ET_UDT08.TXT.

Field	Data Type	Description	Notes
1	CHAR (20)	UDT08 ID	
2	CHAR (30)	UDT08 Name	
3	CHAR (1)	Active Flag	"Y" or "N"

Version 1

The following table contains the layout of the UDT08 ASCII file, UDT08_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT08 ID	
4	CHAR (30)	UDT08 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Notes

For Add/Batch transactions, the following fields are required:

- UDT08 ID
- UDT08 Name

For Change/Delete transactions, the following field is required:

- UDT08 ID

For Add transactions, the system will provide the following default if the column is blank:

- Active Flag — "Y"

UDT09 Layout

The following section contains the layouts for the UDT09 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT09 ASCII file, ET_UDT09.TXT.

Field	Data Type	Description	Notes
1	CHAR (20)	UDT09 ID	
2	CHAR (20)	N/A	
3	CHAR (30)	UDT09 Name	
4	CHAR (1)	Active Flag	"Y" or "N"
5	CHAR (1)	N/A	

Version 1

The following table contains the layout of the UDT09 ASCII file, UDT09_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT09 ID	
4	CHAR (30)	UDT09 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Version 2

The following table contains the layout of the UDT09 ASCII file, UDT09_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT09 ID	
4	CHAR (30)	UDT09 Name	
5	CHAR (1)	Active Flag	"Y" or "N"
6	CHAR (10)	Costpoint Company Code	
7	CHAR (20)	UDT09 Abbreviation	

Notes

For Add/Batch transactions, the following fields are required:

- UDT09 ID
- UDT09 Name

For Change/Delete transactions, the following field is required:

- UDT09 ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- Active Flag — "Y"
- Costpoint Company Code — Set to "1" if you are not using the Costpoint multicompany feature.

UDT10 Layout

The following section contains the layouts for the UDT10 ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the UDT10 ASCII file, ET_UDT10.TXT.

Field	Data Type	Description	Notes
1	CHAR (20)	UDT10 ID	
2	CHAR (30)	UDT10 Name	
3	CHAR (1)	Overtime Flag	
4	CHAR (20)	Overtime Pay Type	
5	CHAR (1)	Cost Only Flag	"Y" or "N"
6	NUMERIC (12,4)	N/A	
7	NUMERIC (15,4)	N/A	
8	CALC (15)	N/A	
9	CHAR (1)	Comments Required Flag	"Y" or "N"
10	CHAR (1)	Active Flag	"Y" or "N"

Version 1

The following table contains the layout of the UDT10 ASCII file, UDT10_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT10 ID	
4	CHAR (30)	UDT10 Name	

Field	Data Type	Description	Notes
5	CHAR (1)	Overtime Flag	"Y" or "N"
6	CHAR (20)	Overtime Pay Type	
7	CHAR (1)	Cost Only Flag	"Y" or "N"
8	CHAR (1)	Comments Required Flag	"Y" or "N"
9	CHAR (30)	Misc Labor Code	
10	CHAR (30)	Misc Payroll Code	
11	CHAR (30)	Payroll Earnings Code	
12	CHAR (10)	Payroll Earnings Mapping Code	
13	CHAR (1)	Active Flag	"Y" or "N"

Version 2

The following table contains the layout of the UDT10 ASCII file, UDT10_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT10 ID	
4	CHAR (30)	UDT10 Name	
5	CHAR (1)	Overtime Flag	"Y" or "N"
6	CHAR (20)	Overtime Pay Type	
7	CHAR (1)	Cost Only Flag	"Y" or "N"
8	CHAR (1)	Comments Required Flag	"Y" or "N"
9	CHAR (30)	Misc Labor Code	
10	CHAR (30)	Misc Payroll Code	
11	CHAR (30)	Payroll Earnings Code	
12	CHAR (10)	Payroll Earnings Mapping Code	
13	CHAR (1)	Active Flag	"Y" or "N"
14	NUMERIC (15,5)	Rate 1 Factor Amount	

Field	Data Type	Description	Notes
15	NUMERIC (15,5)	Rate 1 Fixed Amount	
16	CHAR (5)	Rate 1 Calculation Code	The method by which the user-defined rate 1 amount is calculated. The valid values are as follows: CALC1 — (Hours x Rate x Factor) + Amount CALC2 — (Hours x Rate x Factor) + (Hours x Amount) CALC3 — (Hours x Rate x Factor) + (Hours x Amount x Factor)
17	NUMERIC (15,5)	Rate 2 Factor Amount	
18	NUMERIC (15,5)	Rate 2 Fixed Amount	
19	CHAR (5)	Rate 2 Calculation Code	The method by which the user-defined rate 2 amount is calculated. The valid values are as follows: CALC1 — (Hours x Rate x Factor) + Amount CALC2 — (Hours x Rate x Factor) + (Hours x Amount) CALC3 — (Hours x Rate x Factor) + (Hours x Amount x Factor)
20	CHAR (20)	Labor Recast Code	
21	CHAR (20)	Payroll Recast Code	
22	CHAR (1)	Prorate Flag	"Y" or "N"

Version 3

The following table contains the layout of the UDT10 ASCII file, UDT10_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "3."

Field	Data Type	Description	Notes
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT10 ID	
4	CHAR (30)	UDT10 Name	
5	CHAR (1)	Overtime Flag	"Y" or "N"
6	CHAR (20)	Overtime Pay Type	
7	CHAR (1)	Cost Only Flag	"Y" or "N"
8	CHAR (1)	Comments Required Flag	"Y" or "N"
9	CHAR (30)	Misc Labor Code	
10	CHAR (30)	Misc Payroll Code	
11	CHAR (30)	Payroll Earnings Code	
12	CHAR (10)	Payroll Earnings Mapping Code	
13	CHAR (1)	Active Flag	"Y" or "N"
14	NUMERIC (15,5)	Rate 1 Factor Amount	
15	NUMERIC (15,5)	Rate 1 Fixed Amount	
16	CHAR (5)	Rate 1 Calculation Code	The method by which the user-defined rate 1 amount is calculated. The valid values are as follows: CALC1 — (Hours x Rate x Factor) + Amount CALC2 — (Hours x Rate x Factor) + (Hours x Amount) CALC3 — (Hours x Rate x Factor) + (Hours x Amount x Factor)
17	NUMERIC (15,5)	Rate 2 Factor Amount	
18	NUMERIC (15,5)	Rate 2 Fixed Amount	

Field	Data Type	Description	Notes
19	CHAR (5)	Rate 2 Calculation Code	The method by which the user-defined rate 2 amount is calculated. The valid values are as follows: CALC1 — (Hours x Rate x Factor) + Amount CALC2 — (Hours x Rate x Factor) + (Hours x Amount) CALC3 — (Hours x Rate x Factor) + (Hours x Amount x Factor)
20	CHAR (20)	Labor Recast Code	
21	CHAR (20)	Payroll Recast Code	
22	CHAR (1)	Prorate Flag	"Y" or "N"
23	CHAR(1)	Future Flag	"Y" or "N"

Notes

For Add/Batch transactions, the following fields are required:

- UDT10 ID
- UDT10 Name

For Change/Delete transactions, the following field is required:

- UDT10 ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- Overtime Flag — "N"
- Cost Only Flag — "N"
- Comments Required Flag — "N"
- Payroll Earnings Mapping Code — "N"
- Active Flag — "Y"
- Rate 1 Factor Amount — "0.00"
- Rate 1 Fixed Amount — "0.00"
- Rate 1 Calculation Code — "CALC1"
- Rate 2 Factor Amount — "0.00"
- Rate 2 Fixed Amount — "0.00"
- Rate 2 Calculation Code — "CALC1"

- Prorate Flag — "Y"
- Future Flag – ”Y”

UDT11 Layout

The following section contains the layout for the UDT11 ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 2

The following table contains the layout of the UDT11 ASCII file, UDT12_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT11 ID	
4	CHAR (30)	UDT11 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Notes

For Add/Batch transactions, the following fields are required:

- UDT11 ID
- UDT11 Name

For Change/Delete transactions, the following field is required:

- UDT11 ID

For Add transactions, the system will provide the following default if the column is blank:

- Active Flag — "Y"

UDT12 Layout

The following section contains the layout for the UDT12 ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 2

The following table contains the layout of the UDT12 ASCII file, UDT12_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch

Field	Data Type	Description	Notes
			C — Change D — Delete
3	CHAR (20)	UDT12 ID	
4	CHAR (30)	UDT12 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Notes

For Add/Batch transactions, the following fields are required:

- UDT12 ID
- UDT12 Name

For Change/Delete transactions, the following field is required:

- UDT12 ID

For Add transactions, the system will provide the following default if the column is blank:

- Active Flag — "Y"

UDT13 Layout

The following section contains the layout for the UDT13 ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 2

The following table contains the layout of the UDT13 ASCII file, UDT13_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT13 ID	
4	CHAR (30)	UDT13 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Notes

For Add/Batch transactions, the following fields are required:

- UDT13 ID
- UDT13 Name

For Change/Delete transactions, the following field is required:

- UDT13 ID

For Add transactions, the system will provide the following default if the column is blank:

- Active Flag — "Y"

UDT14 Layout

The following section contains the layout for the UDT14 ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 2

The following table contains the layout of the UDT14 ASCII file, UDT14_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT14 ID	
4	CHAR (30)	UDT14 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Notes

For Add/Batch transactions, the following fields are required:

- UDT14 ID
- UDT14 Name

For Change/Delete transactions, the following field is required:

- UDT14 ID

For Add transactions, the system will provide the following default if the column is blank:

- Active Flag — "Y"

UDT15 Layout

The following section contains the layout for the UDT15 ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 2

The following table contains the layout of the UDT15 ASCII file, UDT15_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch

Field	Data Type	Description	Notes
			C — Change D — Delete
3	CHAR (20)	UDT15 ID	
4	CHAR (30)	UDT15 Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Notes

For Add/Batch transactions, the following fields are required:

- UDT15 ID
- UDT15 Name

For Change/Delete transactions, the following field is required:

- UDT15 ID

For Add transactions, the system will provide the following default if the column is blank:

- Active Flag — "Y"

UDT01 Supervisors Layout

The following section contains the layout for the UDT01 Supervisors ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 1

The following table contains the layout of the UDT01 Supervisors ASCII file, UDT01_SPVSR_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT01 ID	UDT01 ID that is being assigned to the supervisor.
4	CHAR (20)	Supervisor Employee ID	Employee ID of supervisor being assigned.
5	CHAR (10)	Functional Role	The type of functional role the supervisor holds.

Notes

For all transactions, the following fields are required:

- UDT01 ID
- Supervisor Employee ID
- Functional Role

UDT02 Supervisors Layout

The following section contains the layout for the UDT02 Supervisors ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 1

The following table contains the layout of the UDT02 Supervisors ASCII file, UDT02_SPVSR_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT02 ID	UDT02 ID that is being assigned to the supervisor.
4	CHAR (20)	Supervisor Employee ID	Employee ID of supervisor being assigned.
5	CHAR (10)	Functional Role	The type of functional role the supervisor holds.

Notes

For all transactions, the following fields are required:

- UDT02 ID
- Supervisor Employee ID
- Functional Role

UDT01/UDT02 LINK Layout

The following section contains the layouts for the UDT01/UDT02 LINK ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the LINK12 ASCII file, ET_LK12.TXT.

Field	Data Type	Description	Notes
1	CHAR (30)	UDT01 ID	
2	CHAR (30)	UDT02 ID	

Version 1

The following table contains the layout of the LINK12 ASCII file, LINK12_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT01 ID	Wildcard or Actual UDT
4	CHAR (50)	UDT02 ID	Wildcard or Actual UDT

Notes

For all transactions, the following fields are required:

- UDT01 ID
- UDT02 ID

UDT01/UDT07 LINK

The following section contains the layouts for the UDT01/UDT07 LINK ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the LINK17 ASCII file, ET_LK17.TXT.

Field	Data Type	Description	Notes
1	CHAR (30)	UDT01 ID	
2	CHAR (20)	UDT07 ID	
3	CHAR (30)	UDT07 Name	
4	NUMERIC (15,5)	N/A	

Version 1

The following table contains the layout of the LINK17 ASCII file, LINK17_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."

Field	Data Type	Description	Notes
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT01 ID	Wildcard or Actual UDT
4	CHAR (20)	UDT07 ID	Actual UDT
5	CHAR (30)	UDT07 Name	

Version 2

The following table contains the layout of the LINK17 ASCII file, LINK17_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT01 ID	Wildcard or Actual UDT
4	CHAR (20)	UDT07 ID	Actual UDT
5	CHAR (30)	UDT07 Name	
6	NUMERIC (15,5)	Rate 1	
7	NUMERIC (15,5)	Rate 2	
8	CHAR (10)	Costpoint Company Code	

Notes

For Change/Batch transactions, the following fields are required:

- UDT01 ID
- UDT07 ID
- UDT07 Name

For Add/Delete transactions, the following fields are required:

- UDT01 ID
- UDT07 ID

For Add transactions, the system uses the following defaulting logic if the column is blank:

- UDT07 Name — Value from UDT07.udt07_name
- Rate 1 — "0.00"
- Rate 2 — "0.00"
- Costpoint Company — Set to "1" if you are not using the Costpoint multicompany feature.

UDT01/UDT09 LINK Layout

The following section contains the layouts for the UDT01/UDT09 LINK ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the LINK19 ASCII file, ET_LK19.TXT.

Field	Data Type	Description	Notes
1	CHAR (30)	UDT01 ID	
2	CHAR (20)	UDT09 ID	

Version 1

The following table contains the layout of the LINK19 ASCII file, LINK19_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT01 ID	Wildcard or Actual UDT
4	CHAR (20)	UDT09 ID	Wildcard or Actual UDT

Notes

For all transactions, the following fields are required:

- UDT01 ID
- UDT09 ID

UDT02/UDT07 LINK Layout

The following section contains the layouts for the UDT02/UDT07 LINK ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the LINK27 ASCII file, ET_LK27.TXT.

Field	Data Type	Description	Notes
1	CHAR (30)	UDT02 ID	
2	CHAR (20)	UDT07 ID	

Field	Data Type	Description	Notes
3	CHAR (30)	UDT07 Name	
4	NUMERIC (15,5)	N/A	
5	NUMERIC (15,5)	N/A	

Version 1

The following table contains the layout of the LINK27 ASCII file, LINK27_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT02 ID	Wildcard or Actual UDT
4	CHAR (20)	UDT07 ID	Actual UDT
5	CHAR (30)	UDT07 Name	

Version 2

The following table contains the layout of the LINK27 ASCII file, LINK27_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT02 ID	Wildcard or Actual UDT
4	CHAR (20)	UDT07 ID	Actual UDT
5	CHAR (30)	UDT07 Name	
6	NUMERIC (15,5)	Rate 1	
7	NUMERIC (15,5)	Rate 2	
8	CHAR (10)	Costpoint Company	

Notes

For Change/Batch transactions, the following fields are required:

- UDT02 ID
- UDT07 ID
- UDT07 Name

For Add/Delete transactions, the following fields are required:

- UDT02 ID
- UDT07 ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- UDT07 Name — Value from UDT07.udt07_name
- Rate 1 — "0.00"
- Rate 2 — "0.00"
- Costpoint Company — Set to "1" if you are not using the Costpoint multicompany feature.

UDT02/UDT09 LINK Layout

The following section contains the layouts for the UDT02/UDT09 LINK ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the LINK29 ASCII file, ET_LK29.TXT.

Field	Data Type	Description	Notes
1	CHAR (30)	UDT02 ID	
2	CHAR (20)	UDT09 ID	

Version 1

The following table contains the layout of the LINK29 ASCII file, LINK29_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (50)	UDT02 ID	Wildcard or Actual UDT
4	CHAR (20)	UDT09 ID	Wildcard or Actual UDT

Notes

For all transactions, the following fields are required:

- UDT02 ID
- UDT09 ID

UDT09/UDT03 LINK Layout

The following section contains the layouts for the UDT09/UDT03 LINK ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the LINK93 ASCII file, ET_LK93.TXT.

Field	Data Type	Description	Notes
1	CHAR (20)	UDT09 ID	
2	CHAR (20)	UDT03 ID	
3	CHAR (30)	UDT03 Name	

Version 1

The following table contains the layout of the LINK93 ASCII file, LINK93_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	UDT09 ID	Wildcard or Actual UDT
4	CHAR (20)	UDT03 ID	Actual UDT
5	CHAR (30)	UDT03 Name	

Notes

For Change/Batch transactions, the following fields are required:

- UDT09 ID
- UDT03 ID
- UDT03 Name

For Add/Delete transactions, the following fields are required:

- UDT09 ID
- UDT03 ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- UDT03 Name — Value from UDT03.udt03_name

Charge Trees Layout

The following section contains the layouts for the Charge Trees ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the Charge Tree ASCII file, ET_CHRG.TXT.

Field	Data Type	Description	Notes
1	CHAR (10)	Branch Code	
2	CHAR (30)	Branch Description	
3	CHAR (1)	N/A	
4	CHAR (45)	Charge Code	
5	CHAR (120)	Charge Description	
6	CHAR (10)	N/A	
7	CHAR (30)	Default UDT01	
8	CHAR (30)	Default UDT02	
9	CHAR (20)	Default UDT03	
10	CHAR (20)	Default UDT04	
11	CHAR (20)	Default UDT05	
12	CHAR (20)	Default UDT06	
13	CHAR (20)	Default UDT07	
14	CHAR (20)	Default UDT08	
15	CHAR (20)	Default UDT09	
16	CHAR (20)	Default UDT10	
17	CHAR (8)	N/A	
18	CHAR (8)	N/A	
19	CHAR (12)	Employee ID	
20	NUMERIC (8,2)	N/A	
21	NUMERIC (15,5)	N/A	
22	NUMERIC (15,5)	N/A	
23	NUMERIC (15,5)	N/A	
24	NUMERIC (15,5)	N/A	
25	NUMERIC (8,2)	N/A	
26	NUMERIC (15,5)	N/A	
27	NUMERIC (15,5)	N/A	
28	CHAR (1)	N/A	

Field	Data Type	Description	Notes
29	CHAR (1)	N/A	
30	CHAR (1)	N/A	
31	CHAR (1)	N/A	
32	CHAR (1)	N/A	
33	CHAR (1)	N/A	
34	CHAR (1)	N/A	
35	CHAR (1)	N/A	
36	CHAR (1)	N/A	
37	CHAR (4)	N/A	
38	CHAR (254)	N/A	
39	CHAR (1)	Active Flag	"Y" or "N"

Version 1

The following table contains the layout of the Charge Tree ASCII file, CHARGE_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add B — Batch C — Change D — Delete
3	CHAR (4)	Import Type	Indicates the type of record being imported. The valid values are: C — Charge CB — Charge Branch CBC — Charge Branch Charge CEG — Charge Empl Groups CBEG — Charge Branch Empl Group CBCE — Charge Branch Charge Empl
4	CHAR (50)	Tree Code	
5	CHAR (30)	Tree Description	
6	CHAR (50)	Branch Code	

Field	Data Type	Description	Notes
7	CHAR (30)	Branch Description	
8	CHAR (50)	Parent Branch Code	
9	CHAR (100)	Charge Code	
10	CHAR (25)	Employee Group Code	
11	CHAR (20)	Employee ID	
12	CHAR (50)	Default UDT01	
13	CHAR (50)	Default UDT02	
14	CHAR (20)	Default UDT03	
15	CHAR (20)	Default UDT04	
16	CHAR (20)	Default UDT05	
17	CHAR (20)	Default UDT06	
18	CHAR (20)	Default UDT07	
19	CHAR (20)	Default UDT08	
20	CHAR (20)	Default UDT09	
21	CHAR (20)	Default UDT10	
22	CHAR (1)	Restrict Group Flag	"Y" or "N"
23	CHAR (1)	Restrict Employee Flag	"Y" or "N"
24	CHAR (1)	Active Flag	"Y" or "N"

Version 2

The following table contains the layout of the Charge Tree ASCII file, CHARGE_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add B — Batch C — Change D — Delete

Field	Data Type	Description	Notes
3	CHAR (4)	Import Type	Indicates the type of record being imported. The valid values are: C — Charge CB — Charge Branch CBC — Charge Branch Charge CEG — Charge Empl Groups CBEG — Charge Branch Empl Group CBCE — Charge Branch Charge Empl
4	CHAR (50)	Tree Code	
5	CHAR (30)	Tree Description	
6	CHAR (50)	Branch Code	
7	CHAR (30)	Branch Description	
8	CHAR (50)	Parent Branch Code	
9	CHAR (100)	Charge Code	
10	CHAR (25)	Employee Group Code	
11	CHAR (20)	Employee ID	
12	CHAR (50)	Default UDT01	
13	CHAR (50)	Default UDT02	
14	CHAR (20)	Default UDT03	
15	CHAR (20)	Default UDT04	
16	CHAR (20)	Default UDT05	
17	CHAR (20)	Default UDT06	
18	CHAR (20)	Default UDT07	
19	CHAR (20)	Default UDT08	
20	CHAR (20)	Default UDT09	
21	CHAR (20)	Default UDT10	
22	CHAR (1)	Restrict Group Flag	"Y" or "N"
23	CHAR (1)	Restrict Employee Flag	"Y" or "N"
24	CHAR (1)	Active Flag	"Y" or "N"
25	CHAR (10)	UDT06 Costpoint Company Code	

Field	Data Type	Description	Notes
26	CHAR (10)	UDT07 Costpoint Company Code	

Version 3

The following table contains the layout of the Charge Tree ASCII file, CHARGE_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "3."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add B — Batch C — Change D — Delete
3	CHAR (4)	Import Type	Indicates the type of record being imported. The valid values are: C — Charge CB — Charge Branch CBC — Charge Branch Charge CEG — Charge Empl Groups CBEG — Charge Branch Empl Group CBCE — Charge Branch Charge Empl
4	CHAR (50)	Tree Code	
5	CHAR (30)	Tree Description	
6	CHAR (50)	Branch Code	
7	CHAR (30)	Branch Description	
8	CHAR (50)	Parent Branch Code	
9	CHAR (100)	Charge Code	
10	CHAR (25)	Employee Group Code	
11	CHAR (20)	Employee ID	
12	CHAR (50)	Default UDT01	
13	CHAR (50)	Default UDT02	
14	CHAR (20)	Default UDT03	

Field	Data Type	Description	Notes
15	CHAR (20)	Default UDT04	
16	CHAR (20)	Default UDT05	
17	CHAR (20)	Default UDT06	
18	CHAR (20)	Default UDT07	
19	CHAR (20)	Default UDT08	
20	CHAR (20)	Default UDT09	
21	CHAR (20)	Default UDT10	
22	CHAR (1)	Restrict Group Flag	"Y" or "N"
23	CHAR (1)	Restrict Employee Flag	"Y" or "N"
24	CHAR (1)	Active Flag	"Y" or "N"
25	CHAR (10)	UDT06 Costpoint Company Code	
26	CHAR (10)	UDT07 Costpoint Company Code	
27	CHAR (1)	Comment Required Flag	"Y" or "N"

Version 4

The following table contains the layout of the Charge Tree ASCII file, CHARGE_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "4."
2	CHAR (1)	Import Code	Indicates the type of transaction. The valid values are: A — Add B — Batch C — Change D — Delete

Field	Data Type	Description	Notes
3	CHAR (4)	Import Type	Indicates the type of record being imported. The valid values are: C — Charge CB — Charge Branch CBC — Charge Branch Charge CEG — Charge Empl Groups CBEG — Charge Branch Empl Group CBCE — Charge Branch Charge Empl
4	CHAR (50)	Tree Code	
5	CHAR (30)	Tree Description	
6	CHAR (50)	Branch Code	
7	CHAR (30)	Branch Description	
8	CHAR (50)	Parent Branch Code	
9	CHAR (100)	Charge Code	
10	CHAR (25)	Employee Group Code	
11	CHAR (20)	Employee ID	
12	CHAR (50)	Default UDT01	
13	CHAR (50)	Default UDT02	
14	CHAR (20)	Default UDT03	
15	CHAR (20)	Default UDT04	
16	CHAR (20)	Default UDT05	
17	CHAR (20)	Default UDT06	
18	CHAR (20)	Default UDT07	
19	CHAR (20)	Default UDT08	
20	CHAR (20)	Default UDT09	
21	CHAR (20)	Default UDT10	
22	CHAR (1)	Restrict Group Flag	"Y" or "N"
23	CHAR (1)	Restrict Employee Flag	"Y" or "N"
24	CHAR (1)	Active Flag	"Y" or "N"
25	CHAR (10)	UDT06 Costpoint Company Code	

Field	Data Type	Description	Notes
26	CHAR (10)	UDT07 Costpoint Company Code	
27	CHAR (1)	Comment Required Flag	"Y" or "N"
28	CHAR(20)	Default UDT11	
29	CHAR(20)	Default UDT12	
30	CHAR(20)	Default UDT13	
31	CHAR(20)	Default UDT14	
32	CHAR(20)	Default UDT15	

Notes

For charge trees, you can import six different types of records. Depending on the type, certain fields are applicable. They are as follows:

- Import Type = "C"
  - Tree Code
  - Tree Description
  - Restrict Group Flag
- Import Type = "CB"
  - Tree Code
  - Branch Code
  - Branch Description
  - Parent Branch Code (if not top-level branch)
  - Restrict Group Flag
- Import Type = "CBC"
  - Tree Code
  - Branch Code
  - Charge Code
  - Default UDT01 ID
  - Default UDT02 ID
  - Default UDT03 ID
  - Default UDT04 ID
  - Default UDT05 ID
  - Default UDT06 ID
  - Default UDT07 ID
  - Default UDT08 ID
  - Default UDT09 ID

    - Default UDT10 ID
    - Default UDT11 ID
    - Default UDT12 ID
    - Default UDT13 ID
    - Default UDT14 ID
    - Default UDT15 ID
    - Active Flag
- Import Type = "CEG"
    - Tree Code
    - Employee Group Code
- Import Type = "CBEG"
    - Tree Code
    - Branch Code
    - Employee Group Code
- Import Type = "CBCE"
    - Tree Code
    - Branch Code
    - Charge Code
    - Employee ID

For Add Transactions, the following fields are required:

- Import Type = "C"
    - Tree Code
- Import Type = "CB"
    - Tree Code
    - Branch Code
- Import Type = "CBC"
    - Tree Code
    - Branch Code
    - Charge Code
    - UDT01 ID or UDT02 ID
- Import Type = "CEG"
    - Tree Code
    - Employee Group Code
- Import Type = "CBEG"
    - Tree Code
    - Branch Code

    - Employee Group Code
- Import Type = "CBCE"
    - Tree Code
    - Branch Code
    - Charge Code
    - Employee ID

For Add Transactions, the system will use the following defaulting logic if the column is blank:

- Import Type = "C"
    - Tree Description — Defaulted to Tree Code
    - Restrict Group Flag — "N"
- Import Type = "CB"
    - Branch Description — Defaulted to Branch Code
    - Restrict Group Flag — "N"
- Import Type = "CBC"
    - Restrict Employee Flag — "N"
    - Active Flag — "Y"
    - UDT06 Costpoint Company
        - Set to "1" if not you are not using the Costpoint multicompany feature.
        - If you are using the Costpoint multicompany feature and a default UDT06 exists, set to the associated UDT06 Costpoint company.
        - If you are using the Costpoint multicompany feature and a default UDT06 does not exist, set to null.
    - UDT07 Costpoint Company
        - Set to "1" if you are not using the Costpoint multicompany feature.
        - If you are using the Costpoint multicompany feature and a default UDT07 exists, set to the associated UDT07 Costpoint company.
        - If you are using the Costpoint multicompany feature and a default UDT07 does not exist, set to nulls.
- Import Type = "CEG"
    - None
- Import Type = "CBEG"
    - None
- Import Type = "CBCE"
    - None

For Add/Change transactions:

- Import Type = CBC
    - UDT09 Abbreviation ID — If you are using UDT09 Abbreviations and the default UDT09 ID is used or changed, set to the associated UDT09 Abbreviation.

Timesheet Invoices Layout

The following table contains the layout of the Timesheet Invoices ASCII file, TS_INVOICES_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (20)	Employee ID	Employee ID of timesheet on invoice.
4	CHAR (15)	Invoice Number	
5	CHAR (50)	Invoice Charge	
6	Date	Timesheet Date	Date of timesheet

Notes

For all transactions, the following fields are required:

- Employee ID
- Invoice Number
- Invoice Charge
- Timesheet Date

Timesheet Rejections Layout

The following sections contains the layouts for the Timesheet Rejections ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the Timesheet Rejection file. The file name must end with a ".CFM" extension. Please note that the file is a flat ASCII file without comma delimitation.

Field	Data Type	Description	Notes
1	1 — 10	Batch ID	
2	11 — 22	Employee ID	
3	23 — 26	Year	
4	27 — 29	Period	
5	30 — 39	Timesheet Schedule	
6	40 — 42	Line Number	
7	43 — 48	Spaces	

Field	Data Type	Description	Notes
8	49	Rejection Indicator	"R"
9	50 — 100	Rejection Reason	

Version 1

The following table contains the layout of the Timesheet Rejections ASCII file, TS_REJECTION_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	C — Change
3	CHAR (1)	Rejection Level	Indicates the type of record being imported. The valid values are: T — Timesheet L — Line C — Cell
4	CHAR (10)	Batch ID	
5	CHAR (20)	Employee ID	
6	CHAR (10)	Timesheet Schedule	
7	CHAR (4)	Year	
8	CHAR (3)	Period	
9	INTEGER	Line	
10	DATE	Column Date	
11	CHAR (1)	Cell Status	"R" or "O"
12	CHAR (254)	Rejection Reason	

Notes

Rejection information is provided in the timesheet export layout. If a record is rejected, the key information is returned along with the rejection reason. Please see "Export Time Data" in the “Importing and Exporting Data” chapter of this document for more details.

- If Rejection Level = "T," the following fields are required:
  - Batch ID
  - Employee ID
  - Timesheet Schedule
  - Year
  - Period

    - Cell Status
- If Rejection Level = "L," the following fields are required:
    - Batch ID
    - Employee ID
    - Timesheet Schedule
    - Year
    - Period
    - Line
    - Cell Status Code
- If Rejection Level = "C" the following fields are required:
    - Batch ID
    - Employee ID
    - Timesheet Schedule
    - Year
    - Period
    - Line Number
    - Column Date
    - Cell Status Code

Costpoint Company Layout

The following sections contains the layouts for the Costpoint Company ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 1

The following table contains the layout of the Costpoint Company ASCII file, CP_COMPANY_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (10)	Costpoint Company Code	
4	CHAR (30)	Costpoint Company Name	
5	CHAR (1)	Active Flag	"Y" or "N"

Notes

For all transactions, the following fields are required:

- Costpoint Company Code
- Costpoint Company Name

For Add/Change transactions, the system will use the following defaulting logic if the column is blank:

- Active Flag — "Y"

For Add transactions:

- Costpoint Company logo name directory will be set to the logo name found in the General Configuration table.

Current Schedules Layout

The following sections contains the layouts for the Currency Schedules ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the Currency Schedule Rates ASCII file, where additional information is stored in the file name.

CURRENCY_SCHEDULE_YYY_XXXXXXXXXX_YYYYMMDD.TXT

- YYY — The Base Currency
- XXXXXXXXXX — The Currency Schedule Code. The length of the code varies from 1 to 10 positions.
- YYYYMMDD — The effective date of the schedule.

The file is not comma-separated. Instead, it uses filler to match the file received from the Currency Update Service at www.XE.com.

Field	Data Type	Description	Notes
1-32	CHAR (32)	Transactional Currency	
33-35	CHAR (3)	Filler	
36-55	CHAR (19)	Transactional to Base Currency	
56-58	CHAR (3)	Filler	
59-76	CHAR (19)	Base Currency to Transactional	

Following is an example of data from www.XE.com:

Add Header	Add Header	Add Header	Add Header
USD	United States Dollars	1.00000	1.00000
EUR	Euro	1.10953	0.901282

Add Header	Add Header	Add Header	Add Header
GBP	United Kingdom Pounds	1.58781	0.629798
CAD	Canada Dollars	0.730600	1.36874
JPY	Japan Yen	0.00852199	117.343
CHF	Switzerland Francs	0.722864	1.38339
AFA	Afghanistan Afghanis	0.0233727	42.7850
ALL	Albania Leke	0.00814332	122.800
DZD	Algeria Dinars	0.0128254	77.9700

The import process creates two records for each record in the file. The first record is for the base currency to transactional amount. The second record is for the transactional to base currency amount.

Version 1

The following table contains the layout of the Currency Schedule rates ASCII file, Currency_Schedule_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (10)	Currency Schedule Code	
4	DATE	Effective Date	
5	CHAR (3)	From Currency	
6	CHAR (3)	To Currency	
7	NUMERIC (20,10)	Exchange Rate	

Notes For Version 1

For all transactions, the following fields are required:

- Currency Schedule Code
- Effective Date
- From Currency
- To Currency

For Add transactions, the system will use the following defaulting logic if the column is blank:

- Exchange Rate — "0.00"

Expense Report Payment and Advances Layout

The following sections contains the layouts for the Expense Report Payment and Advances ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 1

The following table contains the layout of the Expense Report Payment ASCII file, EXPENSE_REPORT_PAYMENT_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (10)	Expense Report ID	
4	CHAR (30)	Payment Reference Code	The reference number for the payment. Usually, it is the check number.
5	DATE	Payment Date	The date the payment occurred.
6	NUMERIC (15,5)	Pay Currency Amount	The amount of the payment in the employee's pay currency.
7	NUMERIC (15,5)	Base Currency Amount	The amount of the payment in the system's base currency.
8	CHAR (2)	Payment Method	AP — A/P Check/EFT PR — Payroll Check/EFT CR — Cash Receipt

Notes

For Add/Batch transactions, the following fields are required:

- Expense Report ID
- Payment Reference Code
- Payment Date
- Pay Currency Amount
- Base Currency Amount
- Payment Method

For Change/Delete transactions, the following fields are required:

- Expense Report ID
- Payment Reference Code

Version 2

The following table contains the layout of the Expense Report Payment ASCII file, EXPENSE_REPORT_PAYMENT_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (1)	Import Type	A – Advance P — Payment
4	CHAR (20)	Employee ID	
5	CHAR (20)	Reference Code	Check/Voucher ID
6	DATE	Payment/Advance Date	The date on which the payment/advance occurred.
7	CHAR (2)	Payment Method	AP — A/P Check/EFT PR — Payroll Check/EFT CR — Cash Receipt
8	CHAR (3)	Pay Currency Code	
9	CHAR (3)	Base Currency Code	
10	NUMERIC (15,5)	Pay Currency Amount	The amount of the payment in the employee's pay currency.
11	NUMERIC (15,5)	Base Currency Amount	The amount of the payment in the system's base currency.
12	CHAR (10)	Expense Report ID	Only provided if payment

Notes

For Add/Batch transactions, the following fields are required:

- Payment

    - Expense Report ID
    - Payment Reference Code
    - Payment Date
    - Pay Currency Code
    - Base Currency Code
    - Pay Currency Amount
    - Base Currency Amount
    - Payment Method
  - Advance
    - Employee ID
    - Payment Reference Code
    - Payment Date
    - Pay Currency Code
    - Base Currency Code
    - Pay Currency Amount
    - Base Currency Amount
    - Payment Method

For Change/Delete transactions, the following fields are required:

- Expense Report ID or Employee ID
- Payment Reference Code

Expense Rejections Layout

The following sections contains the layouts for the Expense Rejections ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 1

The following table contains the layout of the Expense Rejections ASCII file, EXP_REJECTION_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	C — Change
3	CHAR (10)	Expense Report ID	
4	CHAR (10)	Expense ID	
5	CHAR (254)	Rejection Reason	

Notes

For Change transactions, the following fields are required:

- Expense Report ID
- Expense ID
- Rejection Reason

Leave Types Layout

The following sections contains the layouts for the Leave Types ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 1

The following table contains the layout of the Leave Types ASCII file, LEAVE_TYPE_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (10)	Leave Type Code	
4	CHAR (30)	Leave Type Desc	
5	CHAR (10)	Costpoint Company Code	
6	CHAR (1)	Vacation Flag	"Y" or "N"
7	CHAR (1)	Holiday Flag	"Y" or "N"

Notes

For all transactions, the following fields are required:

- Leave Type Code
- Leave Type Desc
- Costpoint Company Code

For Add transactions, the system will use the following defaulting logic if the column is blank:

- Vacation Flag — "N"
- Holiday Flag — "N"

Per Diem Schedules Layout

The following sections contains the layouts for the Per Diem Schedules ASCII files. The files are comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 0

The following table contains the layout of the Government's Continental United States (CONUS) Per Diem schedule ASCII file. The name of the file can be any one of the following:

- CONNOW.TXT — Military and Civilian
- CONUSNM.TXT — Civilian
- CONUSMIL.TXT — Military

The fields are separated by a ";".

Field	Data Type	Description	Notes
1	CHAR (32)	State	State Description
2	CHAR (32)	Locality/City	Locality/City Description
3	CHAR (32)	County	County Description
4	CHAR (5)	Season Begin	Season Start MM/DD
5	CHAR (5)	Season End	Season End MM/DD
6	NUMERIC (3)	Max Lodging	Lodging Amount
7	NUMERIC (3)	Meals	Meal and Incidental Amount
8	NUMERIC (3)	Proportional Meals	Proportional Meal Amount
9	NUMERIC (3)	Max Per Diem Rate	Max Per Diem Amount
10	CHAR (10)	Effective Date	MM/DD/YYYY

Notes for Version 0

For Add transactions:

- Country Name — "United States"
- Source — "C"
- Location Type
  - If using CONUSNM.TXT, "C"
  - If using CONUSML.TXT, "M"
  - If using CONNOW.TXT, "B"
- Meals and Incidentals Amount — MEALS + "3"

Version 1

The following table contains the layout of the Government's Outside Continental United States (OCONUS) Per Diem schedule ASCII file. The name of the file can be any one of the following:

- OCONUS.TXT — Military and Civilian
- OCONUSNM.TXT — Civilian

The fields are separated by a ";".

Field	Data Type	Description	Notes
1	CHAR (42)	Country	Country Description
2	CHAR (32)	Locality/City	Locality/City Description
3	CHAR (5)	Season Begin	Season Start MM/DD
4	CHAR (5)	Season End	Season End MM/DD
5	NUMERIC (3)	Max Lodging	Lodging Amount
6	NUMERIC (3)	Local Meals	Meal Amount
7	NUMERIC (3)	Proportional Meals	Proportional Meal Amount
8	NUMERIC (3)	Off Base Incidentals	Off Base Incidental Amount
9	NUMERIC (3)	Footnote	Not Used
10	NUMERIC (3)	Footnote	Not Used
11	NUMERIC (3)	Max Per Diem Rate	Max Per Diem Amount
12	CHAR (10)	Effective Date	MM/DD/YYYY

Notes for Version 0

For Add transactions:

- County Name — "N/A"
- Source — "O"
- Location Type
  - If using CONUSNM.TXT, "C"
  - If using CONNOW.TXT , "B"
- Meals and Incidentals Amount -OFF BASE
  - Incidentals + Local Meals
- If Country = "Alaska"
  - State = "Alaska"
  - Country — "United States"
- If Country = "Hawaii"

- State = "Hawaii"
- Country — "United States"

Version 2

The following table contains the layout of the Per Diem Schedule Rates ASCII file, PER_DIEM_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "2."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (1)	Source	C — CONUS O — OCONUS U — User
4	CHAR (1)	Location Type	C — Civilian M — Military B — Both U — User
5	CHAR (45)	Country Name	
6	CHAR (35)	State Name	
7	CHAR (35)	City Name	
8	CHAR (35)	County Name	
9	CHAR (10)	Effective Date	MM/DD/YYYY
10	CHAR (5)	Season Start	MM/DD
11	CHAR (5)	Season End	MM/DD
12	NUMERIC (15,5)	Lodging Amount	
13	NUMERIC (15,5)	Meals and Incidentals Amount	
14	NUMERIC (15,5)	Proportional Meal Amount	

Notes for Version 2

For all transactions, the following fields are required:

- Source Code
- Location Code
- Country Name
- State Name
- City Name
- County Name

- Effective Date
- Season Start
- Season End

For Add transactions, the system will use the following defaulting logic if the column is blank:

- Lodging Amount — "0.00"
- Meals and Incidentals Amount — "0.00"
- Proportional Meal Amount — "0.00"

Tax Codes Layout

The following sections contains the layouts for the Tax Codes ASCII file. The file is comma-separated in accordance with the guidelines discussed earlier. Different versions of the file may exist for backwards compatibility.

Version 1

The following table contains the layout of the Tax Codes ASCII file, TAX_CODE_YYYYMMDDHHMMSS.CSV.

Field	Data Type	Description	Notes
1	SMALLINT	Version	The default is "1."
2	CHAR (1)	Import Code	A — Add B — Batch C — Change D — Delete
3	CHAR (10)	Tax Code	
4	CHAR (30)	Tax Code Desc	
5	CHAR (1)	Type	S – Sales U – Use V — VAT
6	CHAR (1)	Tier	1 – One 2 — Two
7	DECIMAL (5,4)	Tier 1 Rate	
8	DECIMAL (5,4)	Tier 2 Rate	
9	DECIMAL (3,2)	Tolerance Percentage	
10	CHAR (1)	Location Flag	"Y" or "N"
11	DECIMAL (3,2)	Tier 1 Rate Recovery Percentage	
12	DECIMAL (3,2)	Tier 2 Rate Recovery Percentage	
13	CHAR (1)	Tax ID Flag	"Y" or "N"

Notes

For all transactions, the following fields are required:

- Tax Code
- Tax Code Desc

For Add transactions, the system will use the following defaulting logic if the column is blank:

- Tier — "1"
- Type — "S"
- Tier 1 Rate — 0.00
- Tier 2 Rate — 0.00
- Tolerance Percentage — 0.00
- Tier 1 Rate Recovery Percentage — 0.00
- Tier 2 Rate Recovery Percentage — 0.00
- Location Flag — "N"
- Tax ID Flag — "Y"

FAX Attachments

Version 0

You can import attachments saved in .PDF format. When imported, the .PDF is moved to a storage location, which you select on the Domain Details screen of the Technical Console, and the expense report is updated to record the receipt of the attachment.

Notes

- For all transactions, the PDF size limit has been increased to 20 meg.

Import Staging Table Layouts

Overview

This section contains the table layouts for each of the import tables.

Table Access

A specific database user ID, "TCIMPORT," is created during system installation. This login account has "Insert Only" rights to the import tables. The password for the account should have been assigned during installation. You will use this account to insert records into the import tables.

Control Columns

You must provide several control columns in each import table. They are as follows:

- User Company ID - This column is required. You must populate it with the "Domain" name for the database that you wish to update.
- External User - Use this field to identify where the transaction came from. You can insert any value into this column.
- Date/Time Stamp - You must provide the date/time that the transaction was inserted into the table. Use this field to ensure that the transactions are processed in the order they are received.

Employee

This staging table contains the Employee Master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_EMPL

Description: Employee Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add C - Change D - Delete B - Batch

Field	Domain	FK	Null	Key	Notes
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
EMPL_ID	VARCHAR (20)		No	S5	Unique identifier for the employee.
LAST_NAME	VARCHAR (30)		Yes		Employee's last name.
FIRST_NAME	VARCHAR (30)		Yes		Employee's first name.
INITIAL_NAME	VARCHAR (1)		Yes		Employee's middle initial.
GOVT_EMPL_ID	VARCHAR (20)		Yes		Employee's government employee ID, e.g., social security number.
VENDOR_ID	VARCHAR (12)		Yes		The identifying vendor ID for this employee.
PAYROLL_ID	VARCHAR (20)		Yes		The payroll ID for the employee. For use with the Payroll interface.
PAYROLL_EMPL_ID	VARCHAR (20)		Yes		The payroll employee ID for the employee. For use with the Payroll interface.
HIRE_DT	DATE		Yes		The employee's hire date.
TERMINATE_DT	DATE		Yes		Date on which the employee was terminated.
HOME_PHONE_NO_S	VARCHAR (30)		Yes		Employee's home telephone number.
WORK_PHONE_NO_S	VARCHAR (30)		Yes		Employee work telephone number.
FAX_NO_S	VARCHAR (30)		Yes		Employee fax number.

Field	Domain	FK	Null	Key	Notes
CELL_PHONE_NO_S	VARCHAR (30)		Yes		Employee cell phone number.
PAGER_NO_S	VARCHAR (30)		Yes		Employee pager number.
MAIL_ADDRESS_S	VARCHAR (254)		Yes		The employee's e-mail address. Multiple e-mail addresses are delimited by a comma.
LOGIN_EMPL_ID	VARCHAR (20)		Yes		The ID that the employee uses to log on to the system.
ROLE_CD	VARCHAR (10)		Yes		The employee's role for system rights.
CODE1_CD	VARCHAR (20)		Yes		First user-defined alphanumeric field.
CODE2_CD	VARCHAR (20)		Yes		Second user-defined alphanumeric field.
CODE3_CD	VARCHAR (20)		Yes		Third user-defined alphanumeric field.
DATE1_DT	DATE		Yes		First user-defined date field.
DATE2_DT	DATE		Yes		Second user-defined date field.
DATE3_DT	DATE		Yes		Third user-defined date field.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether the employee is active.
CURRENCY_CD	VARCHAR (3)		Yes		The employee's base currency code.
LOCATION_DESC	VARCHAR(30)		Yes		The employee’s location.
TITLE_DESC	VARCHAR(30)		Yes		The Employee’s title

Notes

For Add/Batch transactions, the following fields are required:

- EMPL_ID
- LAST_NAME
- FIRST_NAME
- GOVT_EMPL_ID

For Change/Delete transactions, the following field is required:

- EMPL_ID

For Add transactions, the system will use the following defaulting logic if the column is null:

- LOGIN_EMPL_ID - Default EMPL_ID
- ROLE_CD - Default depends on system configuration.
- ACTIVE_FL - Default "Y"
- CURRENCY_CD - Default depends on system configuration.

Employee History

This staging table contains employee historical information that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_EMPL_HISTORY

Description: Employee History Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.

Field	Domain	FK	Null	Key	Notes
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
EMPL_ID	VARCHAR (20)		No	S5	Employee ID for history record.
EFFECTIVE_DT	DATE		No	S6	The start date for using this history record.
WORK_SCHEDULE_ CD	VARCHAR (10)		Yes		The current work schedule for the employee.
CLASS_CD	VARCHAR (10)		Yes		The current employee class for the employee.
TS_SCHEDULE_CD	VARCHAR (10)		Yes		The current timesheet frequency for the employee.
DFLT_UDT01_ID	VARCHAR (50)		Yes		The current default UDT01 for the employee.
DFLT_UDT02_ID	VARCHAR (50)		Yes		The current default UDT02 for the employee.
DFLT_UDT03_ID	VARCHAR (20)		Yes		The current default UDT03 for the employee.
DFLT_UDT04_ID	VARCHAR (20)		Yes		The current default UDT04 for the employee.
DFLT_UDT05_ID	VARCHAR (20)		Yes		The current default UDT05 for the employee.
DFLT_UDT06_ID	VARCHAR (20)		Yes		The current default UDT06 for the employee.

Field	Domain	FK	Null	Key	Notes
DFLT_UDT07_ID	VARCHAR (20)		Yes		The current default UDT07 for the employee.
DFLT_UDT08_ID	VARCHAR (20)		Yes		The current default UDT08 for the employee.
DFLT_UDT09_ID	VARCHAR (20)		Yes		The current default UDT09 for the employee.
DFLT_UDT10_ID	VARCHAR (20)		Yes		The current default UDT10 for the employee.
DFLT_UDT11_ID	VARCHAR (20)		Yes		The current default udt11 for the employee
DFLT_UDT12_ID	VARCHAR (20)		Yes		The current default udt12 or the employee
DFLT_UDT13ID	VARCHAR (20)		Yes		The current default udt13 or the employee
DFLT_UDT14ID	VARCHAR (20)		Yes		The current default udt14 or the employee
DFLT_UDT15ID	VARCHAR (20)		Yes		The current default udt15 or the employee
CP_COMPANY_CD	VARCHAR (10)		No		Employee's Costpoint Company code.
RATE1_RATE	DECIMAL (15,5)		No		Employee's first user-defined rate.
RATE2_RATE	DECIMAL (15,5)		No		Employee's second user-defined rate.
FRINGE_REDUCTION_RATE	DECIMAL (15,5)		No		Employee's Fringe Reduction Rate.

Field	Domain	FK	Null	Key	Notes
UDT06_CP_COMPANY_CD	VARCHAR (10)		Yes		The UDT06 Costpoint Company code from the associated default UDT06.
UDT07_CP_COMPANY_CD	VARCHAR (10)		Yes		The UDT07 Costpoint Company code from the associated default UDT07.
EXP_CLASS_CD	VARCHAR (20)		Yes		The default expense class for the employee at this point in history.

Notes

For all transactions, the following fields are required:

- EMPL_ID
- EFFECTIVE_DT

For Add transactions, the system will use the following defaulting logic if the column is blank:

- WORK_SCHEDULE_CD - Default depends on Deltek Time Collection configuration.
- TS_SCHEDULE_CD - Default depends on Time Collection configuration.
- CLASS_CD - Default depends on Time Collection configuration.
- CP_COMPANY_CD - Set to "1" if you are not using the Costpoint Multicompany feature.
- UDT06 Costpoint Company
  - Set to "1" if you are not using the Costpoint Multicompany feature.
  - If you are using the Costpoint Multicompany feature and a default UDT06 exists, set to the associated UDT06 Costpoint company.
  - If you are using the Costpoint Multicompany feature and a default UDT06 does not exist, set to null.
- UDT07 Costpoint Company
  - Set to "1" if you are not using the Costpoint Multicompany feature.
  - If you are using the Costpoint Multicompany feature and a default UDT07 exists, set to the associated UDT07 Costpoint company.
  - If you are using the Costpoint Multicompany feature and a default UDT07 does not exist, set to null.
- EXP_CLASS_CD - Default depends on Deltek Expense configuration.

Employee Leave

This staging table contains leave balances that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_EMPL_LEAVE

Description: Employee Leave Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
EMPL_ID	VARCHAR (20)		No	S5	Unique identifier for the employee.
LEAVE_TYPE_CD	VARCHAR (10)		No	S6	Unique identifier for the type of leave.
TRANS_DT	DATE		No	S7	Date of transaction.
S_TRANS_TYPE_CD	VARCHAR (10)		No	S8	The type of transaction. The valid values are: BEG - Beginning Balance ACCRUED - Accrued ADJUSTMENT - Adjustment
LEAVE_HRS	DECIMAL (10,4)		Yes		The number of hours.

Field	Domain	FK	Null	Key	Notes
ADJUSTMENT_TEXT	VARCHAR (60)		Yes		The reason for an adjustment record.

Notes

For all transactions, the following fields are required:

- EMPL_ID
- LEAVE_TYPE_CD
- TRANS_DT
- S_TRANS_TYPE_CD

For Adjustment transactions, the following field is required:

- ADJUSTMENT_TEXT

Employee Groups

This staging table contains employee group information (employees and supervisors) that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_EMPL_GROUP

Description: Employee Group Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.

Field	Domain	FK	Null	Key	Notes
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
EMPL_GROUP_CD	VARCHAR (25)		No	S5	Identifying code for the employee group.
S_IMPORT_TYPE_CD	VARCHAR (1)		No		Indicates the type of record being imported. The valid values are: G - Group M - Employee O - Supervisor
EMPL_GROUP_DESC	VARCHAR (30)		Yes		Description of the employee group. This value must be unique within the EMPL_GROUP and is enforced via a unique constraint. This is in addition to the primary key constraint.
EMPL_GROUP_TYPE_CD	VARCHAR (10)		Yes		The type of the employee group.
EMPL_ID	VARCHAR (20)		Yes		Employee ID of a member of the employee group.
SPVSR_EMPL_ID	VARCHAR (20)		Yes		Employee ID of a supervisor of the employee group.
FUNCTIONAL_ROLE_CD	VARCHAR (10)		Yes		The type of functional role the supervisor holds.

Notes

For Add/Batch transactions, the following fields are required:

- S_IMPORT_TYPE_CD = "G"
  - EMPL_GROUP_CD
  - EMPL_GROUP_DESC
  - EMPL_GROUP_TYPE_CD

- S_IMPORT_TYPE_CD = "M"
  - EMPL_GROUP_CD
  - EMPL_ID
- S_IMPORT_TYPE_CD = "O"
  - EMPL_GROUP_CD
  - SPVSR_EMPL_ID

For Change transactions, the following fields are required:

- S_IMPORT_TYPE_CD = "G"
  - EMPL_GROUP_CD
- S_IMPORT_TYPE_CD = "M"
  - EMPL_GROUP_CD
  - EMPL_ID
- S_IMPORT_TYPE_CD = "O"
  - EMPL_GROUP_CD
  - SPVSR_EMPL_ID

For Delete transactions, the following fields are required:

- S_IMPORT_TYPE_CD = "G"
  - EMPL_GROUP_CD
- S_IMPORT_TYPE_CD = "M"
  - EMPL_GROUP_CD
  - EMPL_ID
- S_IMPORT_TYPE_CD = "O"
  - EMPL_GROUP_CD
  - SPVSR_EMPL_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- S_IMPORT_TYPE_CD = "G"
  - None
- S_IMPORT_TYPE_CD = "M"
  - None
- S_IMPORT_TYPE_CD = "O"
  - FUNCTIONAL_ROLE_CD - "PSPVSR"

UDT01

This staging table contains UDT01 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT01

Description: UDT01 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT01_ID	VARCHAR (50)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT01_NAME	VARCHAR (120)		Yes		Description of user-defined ID.
UDT01_TYPE_CD	VARCHAR (10)		Yes		The type code for this UDT01.
CODE1_CD	VARCHAR (20)		Yes		First user-defined alphabetic field.
CODE2_CD	VARCHAR (20)		Yes		Second user-defined alphabetic field.
CODE3_CD	VARCHAR (20)		Yes		Third user-defined alphabetic field.
DATE1_DT	DATE		Yes		First user-defined date field.
DATE2_DT	DATE		Yes		Second user-defined date field

Field	Domain	FK	Null	Key	Notes
DATE3_DT	DATE		Yes		Third user-defined date field
UDT02_LINK_FL	VARCHAR (1)		Yes		Logical as to whether UDT02 codes are to be linked with this UDT01 item.
UDT07_LINK_FL	VARCHAR (1)		Yes		Logical as to whether UDT07 codes are to be linked with this UDT01 item.
UDT09_LINK_FL	VARCHAR (1)		Yes		Logical as to whether UDT09 codes are to be linked with this UDT01 item.
UDT07_REQUIRED_FL	VARCHAR (1)		Yes		Logical as to whether a UDT07 code is required for this UDT01 item.
START_DT	DATE		Yes		Start date for this UDT01 ID.
END_DT	DATE		Yes		End date for this UDT01 ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT01 ID is active.
ALLOW_CHARGE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT01 ID allows charges.
INACTIVE_DT	DATE		Yes		The date on which the UDT01 ID became inactive.
S_DATE_EDIT_CD	VARCHAR (1)		Yes		The types of edit that will be performed on start and end dates. The valid options are: H - Hard edit S - Soft edit N - No Edit
LINE_APPROVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT01 is line approved.

Field	Domain	FK	Null	Key	Notes
UDT01_ABBRV_ID	VARCHAR (20)		Yes		Abbreviation for UDT01.

Notes

For Add/Batch transactions, the following fields are required:

- UDT01_ID
- UDT01_Name

For Change/Delete transactions, the following field is required:

- UDT01_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- UDT01_TYPE_CD - Default depends on system configuration.
- S_DATE_EDIT_CD - Default depends on system configuration.
- UDT02_LINK_FL - "N"
- UDT07_LINK_FL - "N"
- UDT07_REQUIRED_FL - "N"
- ACTIVE_FL - "Y"
- ALLOW_CHARGE_FL - "Y"
- LINE_APPROVE_FL - "N"

For Add/Batch/Change transactions:

- If UDT07_LINK_FL = "Y," UDT07_REQUIRED_FL will be set to "Y."

UDT02

This staging table contains UDT02 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT02

Description: UDT02 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.

Field	Domain	FK	Null	Key	Notes
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT02_ID	VARCHAR (50)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT02_NAME	VARCHAR (120)		Yes		Description of user-defined ID.
CODE1_CD	VARCHAR (20)		Yes		First user-defined alphabetic field.
CODE2_CD	VARCHAR (20)		Yes		Second user-defined alphabetic field.
CODE3_CD	VARCHAR (20)		Yes		Third user-defined alphabetic field.
DATE1_DT	DATE		Yes		First user-defined date field.
DATE2_DT	DATE		Yes		Second user-defined date field.
DATE3_DT	DATE		Yes		Third user-defined date field.
UDT01_LINK_FL	VARCHAR (1)		Yes		Logical as to whether UDT01 codes are to be linked with this UDT02 item.
UDT07_LINK_FL	VARCHAR (1)		Yes		Logical as to whether UDT07 codes are to be linked with this UDT02 item.

Field	Domain	FK	Null	Key	Notes
UDT09_LINK_FL	VARCHAR (1)		Yes		Logical as to whether UDT09 codes are to be linked with this UDT02 item.
UDT07_REQUIRED_FL	VARCHAR (1)		Yes		Logical as to whether a UDT07 code is required for this UDT02 item.
START_DT	DATE		Yes		Start date for this UDT02 ID.
END_DT	DATE		Yes		End date for this UDT02 ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT02 ID is active.
ALLOW_CHARGE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT02 ID allows charges.
INACTIVE_DT	DATE		Yes		The date on which the UDT02 ID became inactive.
S_DATE_EDIT_CD	VARCHAR (1)		Yes		The types of edit that will be performed on start and end dates. The valid options are: H - Hard edit S - Soft edit N- No Edit
LINE_APPROVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT02 is line approved.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company Code for this UDT02.
UDT02_ABBRV_ID	VARCHAR (20)		Yes		Abbreviation for UDT02.

Field	Domain	FK	Null	Key	Notes
S_TYPE_CD	VARCHAR (1)		No		Indicates which system can be used for this UDT02. Valid options are: T - Time E - Expense B - Both

Notes

For Add/Batch transactions, the following fields are required:

- UDT02_ID
- UDT02_NAME

For Change/Delete transactions, the following field is required:

- UDT02_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- S_DATE_EDIT_CD - Default depends on system configuration.
- UDT01_LINK_FL - "N"
- UDT07_LINK_FL - "N"
- UDT07_REQUIRED_FL - "N"
- ACTIVE_FL - "Y"
- ALLOW_CHARGE_FL - "Y"
- LINE_APPROVE_FL - "N"
- CP_COMPANY_CD - Set to "1" if you are not using the Costpoint multicompany feature.
- S_TYPE_CD - "B"

For Add/Batch/Change transactions:

- If UDT07_LINK_FL = "Y," UDT07_REQUIRED_FL will be set to "Y."

UDT03

This staging table contains UDT03 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT03

Description: UDT03 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT03_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT03_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT03 ID is active.

Notes

For Add/Batch transactions, the following fields are required:

- UDT03_ID
- UDT03_NAME

For Change/Delete transactions, the following field is required:

- UDT03_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"

UDT04

This staging table contains UDT04 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT04

Description: UDT04 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT04_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT04_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT04 ID is active.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company for the UDT04.
UDT04_RATE	DECIMAL (15,5)		No		The rate associated with the UDT04 value.

Notes

For Add/Batch transactions, the following fields are required:

- UDT04_ID
- UDT04_NAME

For Change/Delete transactions, the following field is required:

- UDT04_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"
- CP_COMPANY_CD - Set to "1" if you are not using the Costpoint multicompany feature.
- UDT04_RATE - "0.00"

UDT05

This staging table contains UDT05 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT05

Description: UDT05 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT05_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.

Field	Domain	FK	Null	Key	Notes
UDT05_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT05 ID is active.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company code for the UDT05.

Notes

For Add/Batch transactions, the following fields are required:

- UDT05_ID
- UDT05_Name

For Change/Delete transactions, the following field is required:

- UDT05_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"
- CP_COMPANY_CD - Set to "1" if you are not using the Costpoint multicompany feature.

UDT06

This staging table contains UDT06 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT06

Description: UDT06 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.

Field	Domain	FK	Null	Key	Notes
UDT06_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT06_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT06 ID is active.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company code for the UDT06.

Notes

For Add/Batch transactions, the following fields are required:

- UDT06_ID
- UDT06_NAME
- CP_COMPANY_CD

For Change/Delete transactions, the following fields are required:

- UDT06_ID
- CP_COMPANY_CD

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"

UDT07

This staging table contains UDT07 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT07

Description: UDT07 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.

Field	Domain	FK	Null	Key	Notes
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT07_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT07_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT07 ID is active.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company code for the UDT07.
RATE1_RATE	DECIMAL (15,5)		No		User-defined rate1 for UDT07.
RATE2_RATE	DECIMAL (15,5)		No		User-defined rate2 for UDT07.

Notes

For Add/Batch transactions, the following fields are required:

- UDT07_ID
- UDT07_NAME
- CP_COMPANY_CD

For Change/Delete transactions, the following fields are required:

- UDT07_ID
- CP_COMPANY_CD

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"
- RATE1_RATE - "0.00"
- RATE2_RATE - "0.00"

UDT08

This staging table contains UDT08 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT08

Description: UDT08 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT08_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT08_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT08 ID is active.

Notes

For Add/Batch transactions, the following fields are required:

- UDT08_ID
- UDT08_NAME

For Change/Delete transactions, the following field is required:

- UDT08_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"

UDT09

This staging table contains UDT09 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT09

Description: UDT09 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT09_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.

Field	Domain	FK	Null	Key	Notes
UDT09_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT09 ID is active.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company code for the UDT09.
UDT09_ABBRV_ID	VARCHAR (20)		Yes		Abbreviation for UDT09.

Notes

For Add/Batch transactions, the following fields are required:

- UDT09_ID
- UDT09_NAME

For Change/Delete transactions, the following field is required:

- UDT09_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"
- CP_COMPANY_CD - Set to "1" if you are not using the Costpoint multicompany feature.

UDT10

This staging table contains UDT10 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT10

Description: UDT10 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.

Field	Domain	FK	Null	Key	Notes
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT10_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT10_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
OT_FL	VARCHAR (1)		Yes		Logical as to whether this UDT10 ID is for overtime.
OT_UDT10_ID	VARCHAR (20)		Yes		The UDT10 ID to which overtime premiums are charged for this UDT10 ID.
COST_FL	VARCHAR (1)		Yes		Logical as to whether this UDT10 ID is for "cost only" hours.
TEXT_FL	VARCHAR (1)		Yes		Logical as to whether comments are required when this UDT10 ID is used.

Field	Domain	FK	Null	Key	Notes
MISC_LABOR_CD	VARCHAR (30)		Yes		Miscellaneous field contains codes used to feed the labor distribution file during export.
MISC_PAYROLL_CD	VARCHAR (30)		Yes		Miscellaneous field contains codes used to feed the payroll file during export.
PAYROLL_EARNINGS_CD	VARCHAR (30)		Yes		Earnings code for this UDT10 ID.
S_PAYROLL_MAP_CD	VARCHAR (10)		Yes		This code determines how the Earnings code is mapped during export. The valid options are as follows: N - None R - Regular O - Overtime 3 - Earnings 3 4 - Earnings 4 5 - Earnings 5 H3 - Hours 3 H4 - Hours 4
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT10 ID is active.
RATE1_FACTOR_AMT	DECIMAL (15,5)		No		Multiply factor for user-defined rate 1 amount.
RATE1_FIXED_AMT	DECIMAL (15,5)		No		Fixed lump sum amount for user-defined rate 1 amount.

Field	Domain	FK	Null	Key	Notes
S_RATE1_CALC_CD	VARCHAR (5)		No		The method by which user-defined rate 1 amount is calculated. The valid values are as follows: CALC1 - (Hours x Rate x Factor) = Amount CALC2 - (Hours x Rate x Factor) + (Hours x Amount) CALC3 - (Hours x Rate x Factor) + (Hours x Amount x Factor)
RATE2_FACTOR_AMT	DECIMAL (15,5)		No		Multiply factor for user-defined rate 2 amount.
RATE2_FIXED_AMT	DECIMAL (15,5)		No		Fixed lump sum amount for user-defined rate 2 amount.
S_RATE2_CALC_CD	VARCHAR (5)		No		The method by which user-defined rate 2 amount is calculated. The valid values are as follows: CALC1 - (Hours x Rate x Factor) = Amount CALC2 - (Hours x Rate x Factor) + (Hours x Amount) CALC3 - (Hours x Rate x Factor) + (Hours x Amount x Factor)
RECAST_LABOR_ UDT10_ID	VARCHAR (20)		Yes		The UDT10 code used for recasting labor exports.
RECAST_PAYROLL_ UDT10_ID	VARCHAR (20)		Yes		The UDT10 code used for recasting payroll exports.

Field	Domain	FK	Null	Key	Notes
PRORATE_FL	VARCHAR (1)		No		Logical as to whether timesheet hours assigned to this UDT10 will be included in the hour's proration logic.
FUTURE_FL	VARCHAR(1)		No		Logical as to whether this UDT10 can be used into the future.

Notes

For Add/Batch transactions, the following fields are required:

- UDT10_ID
- UDT10_NAME

For Change/Delete transactions, the following field is required:

- UDT10_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- OT_FL - "N"
- COST_FL - "N"
- TEXT_FL - "N"
- S_PAYROLL_MAP_CD - "N"
- ACTIVE_FL - "Y"
- RATE1_FACTOR_AMT - "0.00"
- RATE1_FIXED_AMT - "0.00"
- S_RATE1_CALC_CD - "CALC1"
- RATE2_FACTOR_AMT - "0.00"
- RATE2_FIXED_AMT - "0.00"
- S_RATE2_CALC_CD - "CALC1"
- PRORATE_FL - "Y"

UDT11

This staging table contains UDT11 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT11

Description: UDT11 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT11_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT11_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT08 ID is active.

Notes

For Add/Batch transactions, the following fields are required:

- UDT11_ID
- UDT11_NAME

For Change/Delete transactions, the following field is required:

- UDT11_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"

UDT12

This staging table contains UDT12 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT12

Description: UDT12 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT12_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT12_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT08 ID is active.

Notes

For Add/Batch transactions, the following fields are required:

- UDT12_ID
- UDT12_NAME

For Change/Delete transactions, the following field is required:

- UDT12_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"

UDT13

This staging table contains UDT13 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT13

Description: UDT13 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT13_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT13_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT08 ID is active.

Notes

For Add/Batch transactions, the following fields are required:

- UDT13_ID
- UDT13_NAME

For Change/Delete transactions, the following field is required:

- UDT13_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"

UDT14

This staging table contains UDT14 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT14

Description: UDT14 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT14_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.

Field	Domain	FK	Null	Key	Notes
UDT14_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT08 ID is active.

Notes

For Add/Batch transactions, the following fields are required:

- UDT14_ID
- UDT14_NAME

For Change/Delete transactions, the following field is required:

- UDT14_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"

UDT15

This staging table contains UDT15 master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT15

Description: UDT15 Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.

Field	Domain	FK	Null	Key	Notes
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT15_ID	VARCHAR (20)		No	S5	Unique user-provided identifier ID for this user-defined table record.
UDT15_NAME	VARCHAR (30)		Yes		Description of user-defined ID.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this UDT08 ID is active.

Notes

For Add/Batch transactions, the following fields are required:

- UDT15_ID
- UDT15_NAME

For Change/Delete transactions, the following field is required:

- UDT15_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"

UDT01 Supervisors

This staging table contains UDT01 supervisor data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT01_SPVSR

Description: UDT01 Supervisor Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid values are:

Field	Domain	FK	Null	Key	Notes
					A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (18)		No		User ID of the non-TC user initiating the insert into the Import table.
UDT01_ID	VARCHAR (50)		No		UDT01 that is being assigned a supervisor.
SPVSR_EMPL_ID	VARCHAR (20)		No		Employee ID of supervisor being assigned.
FUNCTIONAL_ROLE_CD	VARCHAR (10)		No		The type of supervisor. Must be a valid functional role.

Notes

For all transactions, the following fields are required:

- UDT01_ID
- SPVSR_EMPL_ID
- FUNCTIONAL_ROLE_CD

UDT02 Supervisors

This staging table contains UDT02 supervisor data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_UDT02_SPVSR

Description: UDT02 Supervisor Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for database schema that shall be updated.
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into table.
EXTERNAL_USER	VARCHAR (18)		No		User ID of the non-TC user initiating the insert into the Import table.
UDT02_ID	VARCHAR (50)		No		UDT02 that is being assigned supervisor.
SPVSR_EMPL_ID	VARCHAR (20)		No		Employee ID of supervisor being assigned.
FUNCTIONAL_ROLE_CD	VARCHAR (10)		No		The type of supervisor. Must be a valid functional role.

Notes

For all transactions, the following fields are required:

- UDT01_ID
- SPVSR_EMPL_ID
- FUNCTIONAL_ROLE_CD

UDT01/UDT02 Link

This staging table contains valid combinations of UDT01 and UDT02 that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_LINK12

Description: UDT01/UDT02 Link Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema being updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT01_ID	VARCHAR (50)		No	S5	UDT01 code involved in the link.
UDT02_ID	VARCHAR (50)		No	S6	UDT02 code involved in the link.

Notes

For all transactions, the following fields are required:

- UDT01_ID
- UDT02_ID

UDT01/UDT07 Link

This staging table contains valid combinations of UDT01 and UDT07 that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_LINK17

Description: UDT01/UDT07 Link Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT01_ID	VARCHAR (50)		No	S5	UDT01 code involved in the link.
UDT07_ID	VARCHAR (20)		No	S6	UDT07 code involved in the link.
UDT07_NAME	VARCHAR (30)		Yes		Description of the link.
RATE1_RATE	DECIMAL (15,5)		No		User-defined rate 1 for UDT01/UDT07.
RATE2_RATE	DECIMAL (15,5)		No		User-defined rate 2 for UDT01/UDT07.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company code for the UDT07.

Notes

For Add/Delete transactions, the following fields are required:

- UDT01_ID
- UDT07_ID

For Change/Batch transactions, the following fields are required:

- UDT01_ID
- UDT07_ID
- UDT07_NAME

For Add Transactions, the system will use the following defaulting logic if column is blank:

- UDT07_NAME - Value from UDT07.UDT07_NAME
- RATE1_RATE - "0.00"
- RATE2_RATE - "0.00"
- CP_COMPANY_CD - Set to "1" if you are not using the Costpoint multicompany feature.

UDT01/UDT09 Link

This staging table contains valid combinations of UDT01 and UDT09 that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_LINK19

Description: UDT01/UDT09 Link Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT01_ID	VARCHAR (50)		No	S5	UDT01 code involved in the link.

Field	Domain	FK	Null	Key	Notes
UDT09_ID	VARCHAR (20)		No	S6	UDT09 code involved in the link.

Notes

For all transactions, the following fields are required:

- UDT01_ID
- UDT09_ID

UDT02/UDT07 Link

This table represents a staging table that contains valid combinations of UDT02 and UDT07 that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_LINK27

Description: UDT02/UDT07 Link Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT02_ID	VARCHAR (50)		No	S5	UDT02 code involved in the link.

Field	Domain	FK	Null	Key	Notes
UDT07_ID	VARCHAR (20)		No	S6	UDT07 code involved in the link.
UDT07_NAME	VARCHAR (30)		Yes		Description of the link.
RATE1_RATE	DECIMAL (15,5)		No		User-defined rate 1 for UDT02/UDT07.
RATE2_RATE	DECIMAL (15,5)		No		User-defined rate 2 for UDT02/UDT07.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company code for the UDT07.

Notes

For Add/Delete transactions, the following fields are required:

- UDT02_ID
- UDT07_ID

For Change/Batch transactions, the following fields are required:

- UDT02_ID
- UDT07_ID
- UDT07_NAME

For Add transactions, the system will use the following defaulting logic if the column is blank:

- UDT07_NAME - Value from UDT07.UDT07_NAME
- RATE1_RATE - "0.00"
- RATE2_RATE - "0.00"
- CP_COMPANY_CD - Set to "1" if you are not using the Costpoint multicompany feature.

UDT02/UDT09 Link

This staging table contains valid combinations of UDT02 and UDT09 that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_LINK29

Description: UDT02/UDT09 Link Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT02_ID	VARCHAR (50)		No	S5	UDT02 code involved in the link.
UDT09_ID	VARCHAR (20)		No	S6	UDT09 code involved in the link.

Notes

For all transactions, the following fields are required:

- UDT02_ID
- UDT09_ID

UDT09/UDT03 Link

This staging table contains valid combinations of UDT09 and UDT03 that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_LINK93

Description: UDT09/UDT03 Link Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
UDT09_ID	VARCHAR (20)		No	S5	UDT09 code involved in the link.
UDT03_ID	VARCHAR (20)		No	S6	UDT03 code involved in the link.
UDT03_NAME	VARCHAR (30)		Yes		Description of the link.

Notes

For Add/Delete transactions, the following fields are required:

- UDT09_ID
- UDT03_ID

For Change/Batch transactions, the following fields are required:

- UDT09_ID
- UDT03_ID
- UDT03_NAME

For Add transactions, the system will use the following defaulting logic if the column is blank:

- UDT03_NAME - Value from UDT03.UDT03_NAME

Charge Trees

This staging table contains Charge Tree/Branch/charge master table data that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_CHARGE

Description: Charge Tree/Branch

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No	S1	Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No	S2	Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No	S3	A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No	S4	User ID of the non-TC user inserting data into the Import table.
S_IMPORT_TYPE_ CD	VARCHAR (4)		No	S5	Indicates the type of record being imported. The valid values are: C - Charge CB - Charge Branch CBC - Charge Branch Charge CEG - Charge Empl Groups CBEG - Charge Branch Empl Group CBCE - Charge Branch Charge Empl
CHARGE_TREE_CD	VARCHAR (50)		No	S6	The top-level identifier of the charge tree.

Field	Domain	FK	Null	Key	Notes
CHARGE_TREE_ DESC	VARCHAR (30)		Yes	S7	The name of charge tree.
CHARGE_BRANCH_ CD	VARCHAR (50)		Yes	S8	The identifier of the branch in the charge tree.
CHARGE_BRANCH_ DESC	VARCHAR (30)		Yes	S9	The description of the charge branch.
PARENT_CHARGE_ BRANCH_CD	VARCHAR (50)		Yes	S10	This field identifies the parent branch, if there is one.
CHARGE_CD	VARCHAR (100)		Yes	S11	The identifier of the charge in the charge tree/branch.
EMPL_GROUP_CD	VARCHAR (25)		Yes	S12	Employee group authorized for charge or branch.
EMPL_ID	VARCHAR (20)		Yes	S13	Employee ID authorized for charge branch charge.
DFLT_UDT01_ID	VARCHAR (50)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT02_ID	VARCHAR (50)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT03_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT04_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT05_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.

Field	Domain	FK	Null	Key	Notes
DFLT_UDT06_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT07_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT08_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT09_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT10_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT11_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT12_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT13_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT14_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.
DFLT_UDT15_ID	VARCHAR (20)		Yes		If present, this value provides an overriding default for this UDT.

Field	Domain	FK	Null	Key	Notes
RESTRICT_EMPL_ GROUP_FL	VARCHAR (1)		Yes		Logical as to whether this tree should be restricted to certain employee groups.
RESTRICT_EMPL_FL	VARCHAR (1)		Yes		Logical as to whether this charge should be restricted to certain employees.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this charge is active.
UDT06_CP_COMPANY_CD	VARCHAR (10)		Yes		Costpoint Company code for the default UDT06.
UDT07_CP_COMPANY_CD	VARCHAR (10)		Yes		Costpoint Company code for the default UDT07.
COMMENT_REQUIRED_FL	VARCHAR(1)		Yes		Logical as to whether comment is required when this charge is used on timesheet.

Notes

You can import six different types of records for charge trees. Depending on the type, certain fields are applicable. They are as follows:

- S_IMPORT_TYPE_CD = "C"
  - CHARGE_TREE_CD
  - CHARGE_TREE_DESC
  - RESTRICT_EMPL_GROUP_FL
- S_IMPORT_TYPE_CD = "CB"
  - CHARGE_TREE_CD
  - CHARGE_BRANCH_CD
  - CHARGE_BRANCH_DESC
  - PARENT_CHARGE_BRANCH_CD (if not top level branch)
  - RESTRICT_EMPL_GROUP_FL

- S_IMPORT_TYPE_CD = "CBC"
  - CHARGE_TREE_CD
  - CHARGE_BRANCH_CD
  - CHARGE_CD
  - DFLT_UDT01_ID
  - DFLT_UDT02_ID
  - DFLT_UDT03_ID
  - DFLT_UDT04_ID
  - DFLT_UDT05_ID
  - DFLT_UDT06_ID
  - DFLT_UDT07_ID
  - DFLT_UDT08_ID
  - DFLT_UDT09_ID
  - DFLT_UDT10_ID
  - DFLT_UDT11_ID
  - DFLT_UDT12_ID
  - DFLT_UDT13_ID
  - DFLT_UDT14_ID
  - DFLT_UDT15_ID
  - ACTIVE_FL
- S_IMPORT_TYPE_CD = "CEG"
  - CHARGE_TREE_CD
  - EMPL_GROUP_CD
- S_IMPORT_TYPE_CD = "CBEG"
  - CHARGE_TREE_CD
  - CHARGE_BRANCH_CD
  - EMPL_GROUP_CD
- S_IMPORT_TYPE_CD = "CBCE"
  - CHARGE_TREE_CD
  - CHARGE_BRANCH_CD
  - CHARGE_CD
  - EMPL_ID

For Add Transactions, the following fields are required:

- S_IMPORT_TYPE_CD = "C"
  - CHARGE_TREE_CD
- S_IMPORT_TYPE_CD = "CB"

    - CHARGE_TREE_CD
    - CHARGE_BRANCH_CD
- S_IMPORT_TYPE_CD = "CBC"
    - CHARGE_TREE_CD
    - CHARGE_BRANCH_CD
    - CHARGE_CD
    - UDT01 ID or UDT02 ID
- S_IMPORT_TYPE_CD = "CEG"
    - CHARGE_TREE_CD
    - EMPL_GROUP_CD
- S_IMPORT_TYPE_CD = "CBEG"
    - CHARGE_TREE_CD
    - CHARGE_BRANCH_CD
    - EMPL_GROUP_CD
- S_IMPORT_TYPE_CD = "CBCE"
    - CHARGE_TREE_CD
    - CHARGE_BRANCH_CD
    - CHARGE_CD
    - EMPL_ID

For Add transactions, the system will use the following defaulting logic if the column is blank:

- S_IMPORT_TYPE_CD = "C"
    - CHARGE_TREE_DESC - Default to CHARGE_TREE_CD
    - RESTRICT_EMPL_GROUP_FL – "N"
- S_IMPORT_TYPE_CD = "CB"
    - CHARGE_TREE_DESC - Default to CHARGE_BRANCH_CD
    - RESTRICT_EMPL_GROUP_FL - "N"
- S_IMPORT_TYPE_CD = "CBC"
    - RESTRICT_EMPL_FL - "N"
    - ACTIVE_FL - "Y"
    - UDT06_CP_COMPANY_CD
        - Set to "1" if you are not using the Costpoint multicompany feature.
        - If you are using the Costpoint multicompany feature and a DFLT_UDT06_ID exists, set to the associated UDT06.cp_company_cd value.
        - If you are using the Costpoint Multicurrency feature and a DFLT_UDT06_ID does not exist, set to Null.
    - UDT07_CP_COMPANY_CD

        - Set to "1" if you are not using the Costpoint multicompany feature.
        - If you are using the Costpoint multicompany feature and a DFLT_UDT07_ID exists, set to the associated UDT07.cp_company_cd value.
        - If you are using the Costpoint Multicurrency feature and a DFLT_UDT06_ID does not exist, set to Null.
- S_IMPORT_TYPE_CD = "CEG"
    - None
- S_IMPORT_TYPE_CD = "CBEG"
    - None
- S_IMPORT_TYPE_CD = "CBCE"
    - None

For Add/Change transactions:

- S_IMPORT_TYPE_CD = "CBC"
    - UDT09_ABBREV_ID
        - If you are using UDT09 abbreviations and the DFLT_UDT09_ID is used or changed, set to the associated UDT09.UDT09_ABBREV_ID.

Timesheet Invoices

This staging table contains timesheet invoices that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_TS_INVOICE

Description: Timesheet Invoices

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for database schema that shall be updated.
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid values are: A - Add C - Change D - Delete B - Batch

Field	Domain	FK	Null	Key	Notes
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into table.
EXTERNAL_USER	VARCHAR (18)		No		User ID of the non-TC user initiating the insert into the Import table.
EMPL_ID	VARCHAR (20)		No		Employee ID of timesheet contained on invoice.
INVOICE_NO_CD	VARCHAR (15)		No		Invoice number.
INVOICE_UDT_ID	VARCHAR (50)		No		Invoice charge.
TIMESHEET_DT	DATE		No		Date of timesheet.

Notes

For all transactions, the following fields are required:

- EMPL_ID
- INVOICE_NO_CD
- INVOICE_UDT_ID
- TIMESHEET_DT

Timesheet Rejections

This staging table contains timesheet rejections that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_REJECTION

Description: Timesheet Rejections

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for the database schema that will be updated.

Field	Domain	FK	Null	Key	Notes
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid value is: C - Change
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No		User ID of the non-TC user inserting data into the Import table.
S_REJECT_TYPE_CD	VARCHAR (1)		No		Indicates the type of record being imported. The valid values are: T - Timesheet L - Line C - Cell
BATCH_ID	VARCHAR (10)		No		The ID of the batch to which the rejection belongs.
EMPL_ID	VARCHAR (20)		No		Employee ID identifier.
TS_SCHEDULE_CD	VARCHAR (10)		No		Timesheet frequency identifier.
YEAR_NO_CD	VARCHAR (4)		No		The timesheet frequency's year code.
PERIOD_NO_CD	VARCHAR (3)		No		Period number for timesheet frequency.
LINE_NO	INTEGER		Yes		The timesheet line number to which this rejection belongs.
HRS_DT	DATE		Yes		The column date to which this rejection belongs.

Field	Domain	FK	Null	Key	Notes
S_CELL_STATUS_CD	VARCHAR (1)		No		The status that should be set for lines and cells. The valid values are: R - Reject O - Open
REJECTION_TEXT	VARCHAR (254)		Yes		The reason for the rejection.

Notes

Rejection information is provided in the timesheet export layout. If a record is rejected, the key information is returned along with the rejection reason. Please see "Export Time Data" in the “Importing and Exporting Data” chapter of this document for more details.

For Change transactions, the following fields are required:

- S_REJECT_TYPE_CD = "T"
  - BATCH_ID
  - EMPL_ID
  - TS_SCHEDULE_CD
  - YEAR_NO_CD
  - PERIOD_NO_CD
  - S_CELL_STATUS_CD
- S_REJECT_TYPE_CD = "L"
  - BATCH_ID
  - EMPL_ID
  - TS_SCHEDULE_CD
  - YEAR_NO_CD
  - PERIOD_NO_CD
  - LINE_NO
  - S_CELL_STATUS_CD
- S_REJECT_TYPE_CD = "C"
  - BATCH_ID
  - EMPL_ID
  - TS_SCHEDULE_CD
  - YEAR_NO_CD
  - PERIOD_NO_CD
  - LINE_NO
  - HRS_DT

- S_CELL_STATUS_CD

Costpoint Company

This staging table contains Costpoint companies that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_CP_COMPANY

Description: Costpoint Company Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid values are: A - Add C - Change D - Delete B - Batch
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No		User ID of the non-TC user inserting data into the Import table.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company Code
CP_COMPANY_NAME	VARCHAR (30)		No		Description of Company Code.
ACTIVE_FL	VARCHAR (1)		Yes		Logical as to whether this Company is active.

Notes

For All transactions, the following fields are required:

- CP_COMPANY_CD
- CP_COMPANY_NAME

For Add transactions, the system will use the following defaulting logic if the column is blank:

- ACTIVE_FL - "Y"

For Add transactions:

- CP_COMPANY_LOGO_NAME_DIR_S will be set to GENERAL_CONFIG.LOGO_NAME_DIR_S.

Currency Schedules

This staging table contains Currency schedules that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_CURRENCY_SCHEDULE

Description: Currency Schedule Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid values are: A - Add C - Change D - Delete B - Batch
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No		User ID of the non-TC user inserting data into the Import table.
CURRENCY_SCHEDULE_CD	VARCHAR (10)		No		The identifier for the currency schedule.
EFFECTIVE_DT	DATE		No		The date the exchange rate is effective.

Field	Domain	FK	Null	Key	Notes
FROM_CURRENCY_CD	VARCHAR (3)		No		The currency code of the transaction that is being converted.
TO_CURRENCY_CD	VARCHAR (3)		No		The currency code being converted to.
EXCHANGE_RATE	DECIMAL (20,10)		Yes		The exchange rate.

Notes

For all transactions, the following fields are required:

- CURRENCY_SCHEDULE_CD
- EFFECTIVE_DT
- FROM_CURRENCY_CD
- TO_CURRENCY_CD

Expense Report Payment & Advances

This staging table contains expense report payments that will be imported to Deltek Time & Expense if validations are passed.

Note: This staging table contains valid expense report advance and payment records that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_EXP_RPT_ADVANCE_ PAYMENT

Description: Expense Report Advances and Payments Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for database schema that will be updated.

Field	Domain	FK	Null	Key	Notes
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into table.
EXTERNAL_USER	VARCHAR (20)		No		User ID of the non-TC user initiating the insert into the Import table.
EMPL_ID	VARCHAR (20)		Yes		The unique identifier of the employee for the advance.
REFERENCE_CD	VARCHAR (20)		No		The reference number for the advance or payment. Usually it is the number of the check for the advance or payment.
S_METHOD_TYPE_CD	VARCHAR (5)		No		The type of advance or method of payment. The valid values for an advance type are: CHECK - Check EFT - Electronic Funds Transfer OTHER - Other The valid values for a payment method are: AP - A/P Check/EFT PR - Payroll Check/EFT CR - Cash Receipt

Field	Domain	FK	Null	Key	Notes
PAY_CURRENCY_CD	VARCHAR (3)		Yes		The currency code the advance was paid in.
BASE_CURRENCY_CD	VARCHAR (3)		Yes		The base currency code of the system.
PAY_AMT	DECIMAL (15,5)		No		The amount of the advance or payment in the employee's pay currency.
BASE_AMT	DECIMAL (15,5)		No		The amount of the advance or payment in the system's base currency.
S_IMPORT_TYPE_CD	VARCHAR (1)		No		Indicates the type of record being imported. The valid values are: A - Advance P - Payment
PAYMENT_DT	DATE		No		Date of the advance or payment.
EXP_RPT_ID	VARCHAR (10)		Yes		The unique identifier of either the expense report or the expense authorization for an advance.

Notes

For Add/Batch transactions, the following fields are required:

- EXP_RPT_ID (if Payment)
- EMPL_ID (if Advance)
- PAYMENT_REFERENCE_CD
- PAYMENT_DT
- PAY_PAYMENT_AMT
- BASE_PAYMENT_AMT
- S_PAYMENT_METHOD_CD

For Change/Delete transactions, the following fields are required:

- EXP_RPT_ID (if Payment)
- EMPL_ID (if Advance)

- PAYMENT_REFERENCE_CD

Expense Rejections

This staging table contains expense rejections that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_EXPENSE_REJECTION

Description: Expense Rejections Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid value is: C - Change
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No		User ID of the non-TC user inserting data into the Import table.
EXP_RPT_ID	VARCHAR (10)		No		The unique identifier for the expense report.
EXPENSE_ID	INTEGER		No		The unique identifier for the expense.
REJECTION_TEXT	VARCHAR (254)		No		The reason for the expense rejection.

Notes

For Change transactions, the following fields are required:

- EXP_RPT_ID
- EXPENSE_ID
- REJECTION_TEXT

Leave Types

This staging table contains leave types that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_LEAVE_TYPE

Description: Leave Types

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid values are: A - Add C - Change D - Delete B - Batch
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No		User ID of the non-TC user inserting data into the Import table.
LEAVE_TYPE_CD	VARCHAR (10)		No		Unique user-provided type code.
LEAVE_TYPE_DESC	VARCHAR (30)		No		Unique description for the leave type.
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company code for the leave type.
VACATION_FL	VARCHAR (1)		Yes		Logical as to whether this leave type is a vacation type.

Field	Domain	FK	Null	Key	Notes
HOLIDAY_FL	VARCHAR (1)		Yes		Logical as to whether this leave type is a holiday type.

Notes

For all transactions, the following fields are required:

- LEAVE_TYPE_CD
- LEAVE_TYPE_DESC
- CP_COMPANY_CD

For Add transactions, the system will use the following defaulting logic if the column is blank:

- VACATION_FL - "N"
- HOLIDAY_FL - "N"

Per Diem Schedules

This staging table contains Per Diem data that will be imported to Deltek Time & Expense if validations are passed.

Table**:** IMPORT_PER_DIEM

Description: Import Per Diem Data

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for the database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid values are: A - Add C - Change D - Delete B - Batch

Field	Domain	FK	Null	Key	Notes
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into the table.
EXTERNAL_USER	VARCHAR (20)		No		User ID of the non-TC user inserting data into the Import table.
COUNTRY_NAME	VARCHAR (45)		No		The name of the country for the Per Diem schedule location.
STATE_NAME	VARCHAR (35)		No		The name of the state or providence for the Per Diem schedule location.
CITY_NAME	VARCHAR (35)		No		The name of the city for the Per Diem schedule location.
COUNTY_NAME	VARCHAR (35)		No		The name of the county for the Per Diem schedule location.
EFFECTIVE_DT	DATE		No		The date on which the schedule is effective.
SEASON_START_DATE	VARCHAR (5)		No		The month and day on which the season starts for the schedule. Format must be MM/DD. For non-seasonal schedules, this field should be 01/01.

Field	Domain	FK	Null	Key	Notes
SEASON_END_DATE	VARCHAR (5)		No		The month and day on which the season ends for the schedule. Format must be MM/DD. For non-seasonal schedules, this field should be 12/31.
LODGING_AMT	DECIMAL (15,5)		No		The amount allocated for Lodging for this schedule (exclusive of taxes).
MEAL_INCIDENTAL_AMT	DECIMAL (15,5)		No		The amount allocated for Meals and Incidentals for this schedule.
S_SOURCE_CD	VARCHAR (1)		No		Indicates the source of the schedule. Valid values are: C - CONUS O - OCONUS U - User
S_LOCATION_TYPE_CD	VARCHAR (1)		No		Indicates the type of schedule location. Valid values are: C - Civilian M - Military B - Both U - User
PROPORTIONAL_MEALS_AMT	DECIMAL (15,5)		No		The proportional meal allowance for this schedule.

Notes

For all transactions, the following fields are required:

- COUNTRY_NAME

- STATE_NAME
- CITY_NAME
- COUNTY_NAME
- EFFECTIVE_DT
- SEASON_START_DT
- SEASON_END_DT
- S_SOURCE_CD
- LOCATION_TYPE_CD

For Add transactions, the system will use the following defaulting logic if the column is blank:

- LODGING_AMT - "0.00"
- MEAL_INCIDENTAL_AMT - "0.00"
- PROPORTIONAL_MEALS_AMT - "0.00"

Tax Codes

This staging table contains tax codes that will be imported to Deltek Time & Expense if validations are passed.

Note: This staging table contains tax codes that will be imported to Deltek Time & Expense if validations are passed.

Table: IMPORT_TAX_CODE

Description: Tax Code Import

Type: Import

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
USER_COMPANY_ID	VARCHAR (30)		No		Alias for database schema that will be updated.
S_IMPORT_CD	VARCHAR (1)		No		Indicates the type of transaction. The valid values are: A - Add B - Batch C - Change D - Delete

Field	Domain	FK	Null	Key	Notes
TIME_STAMP	DATETIME		No		A time/date stamp that indicates when the record was originally inserted into table.
EXTERNAL_USER	VARCHAR (20)		No		User ID of the non-TC user initiating the insert into the Import table.
TAX_SCHEDULE_CD	VARCHAR (10)		No		The tax schedule unique identifier.
TAX_SCHEDULE _DESC	VARCHAR (30)		No		The description for the tax schedule.
S_TYPE_CD	VARCHAR (1)		No		The type of tax for the schedule. The valid values are as follows: S - Sales U - Use V - Value Added
S_RATE_TIER_CD	VARCHAR (1)		Yes		Determines the tax rate tier structure tax. The valid values are as follows: 1 - One 2 - Two
DFLT_TAX_RATE1_PCT	DECIMAL (5,4)		Yes		The default tax rate percentage for Rate1 for new schedules.
DFLT_TAX_RATE2_PCT	DECIMAL (5,4)		Yes		The default tax rate percentage for Rate2 for new schedules.
DFLT_TOLERANCE_PCT	DECIMAL (3,2)		Yes		The default amount by which the user-entered tax rate can deviate from system-calculated rate.
LOCATION_FL	VARCHAR (1)		Yes		Logical as to whether schedule should include location.

Field	Domain	FK	Null	Key	Notes
DFLT_TAX_RECOVERY_RATE1_PCT	DECIMAL (3,2)		No		The default rate 1 tax recovery percentage.
DFLT_TAX_RECOVERY_RATE2_PCT	DECIMAL (3,2)		No		The default rate 2 tax recovery percentage.
TAX_ID_FL	VARCHAR (1)		No		Logical as to whether the schedule should include the tax ID.

Notes

For all transactions, the following fields are required:

- TAX_SCHEDULE_CD
- TAX_SCHEDULE_DESC

For Add transactions, the system will use the following defaulting logic if the column is blank:

- S_RATE_TIER_CD - "1"
- S_TYPE_CD - "S"
- DFLT_TAX_RATE1_PCT - 0.00
- DFLT_TAX_RATE2_PCT - 0.00
- DFLT_TOLERANCE_PCT - 0.00
- DFLT_TAX_RECOVERY_RATE1_PCT - 0.00
- DFLT_TAX_RECOVERY_RATE2_PCT - 0.00
- LOCATION_FL - "N"
- TAX_ID_FL - "Y"

Import XML Schemas

Overview

Deltek Time & Expense supports a simple XML format for data import. The information set is similar to the ASCII and RDBMS table layouts described in this document. The details of the supported XML format are fully described in the XSD schema definition later in this section. Text versions of these files are also located on the CD. More information about XML schema can be found at http://www.w3.org/XML/Schema. An introductory primer to XML Schema is http://www.w3.org/TR/xmlschema-0. The data structure is fully specified by the document found at http://www.w3.org/TR/xmlschema-1. The data types used are specified by the document at http://www.w3.org/TR/xmlschema-2.

The Deltek Time & Expense XSD schema is divided up into a number of sub-schema documents. Together, these sub-schema documents form the supported schema. One of the sub-schema documents is named "common.xsd" and contains global definitions of elements used in more than one of the other sub-schema documents. There is a top-level document named "tcdata.xsd" that includes all of the other sub-schema documents and provides the top-level structure of the allowable XML instance documents/messages. There are then a number of other sub-schema documents, each one defining the allowable format for a data packet element.

Structure

You can import files containing XML documents consisting of a top-level importTCData element containing any number of tcData import elements that follow the XML schema described in this section. Also see "File Name Convention" later in this section for the file naming conventions that must be used.

You can also post individual XML documents directly to a JMS queue in the form of text messages.

It is not necessary to include the schema location in these XML instance documents. The Deltek Time & Expense import process always uses the installed copy of the schema.

There are two high-level elements in each tcData top-level element. The first, "tcControl," is a control element that provides routing information. The second element is a data packet element consisting of one of several groups of data elements.

There is a required attribute on the tcData element that must contain an integer corresponding to the version of the schema that defines this instance of the document/message. For Time & Expense 6, the value of the version attribute must be 600.

Control Element

You must provide a control element (tcControl) within each (tcData) document/message. This element contains the following control elements:

- User Company ID (user_company_id) - This element is optional in Time & Expense 6.0. If used, must contain the Domain name for the database that you wish to update.
- Import Code (s_import_cd) - This code defines the type of transaction to perform using the attached data packet element. It can be an addition, update, deletion, or batch operation.
- External User (external_user) - Use this element to identify from where the transaction came. You can insert any value into this element.
- Date/Time Stamp (time_stamp) - You must provide the date/time that the document transaction was created. Use this element to ensure that the transactions are processed in the order they are received.

Data Packet Elements

Each data packet element contains of group of related data elements such as Employee (et_empl), Employee Leave (et_empll), User Defined Table (udt01), etc.

File Name Convention

The import file name must adhere to Windows and Unix file-system naming conventions, and must be readable by the import program (i.e., it must be located in a suitable directory and have appropriate permissions). The file must follow the following naming convention:

XXXXXXXXXX_YYYYMMDDHHMMSS.XML

XXXXXXXXXX is an acronym for the type of data elements being imported (for example, "EMPL" for employee). YYYYMMDDHHMMSS is a timestamp that indicates when the file was created. The format is Year/Month/Day/Hours/Minutes/Seconds (for example, "20031002125959").

Common

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
        elementFormDefault = "qualified">
        <xsd:element name = "tcControl">
                <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name = "user_company_id" type = "user_company_id"
                             minOccurs = "0"/>
                            <xsd:element name = "s_import_cd" type = "s_import_cd"/>
                            <xsd:element name = "time_stamp" type = "xsd:dateTime"/>
                            <xsd:element name = "external_user" type = "external_user"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "user_company_id">
                <xsd:restriction base = "xsd:string">
                        <xsd:maxLength value = "30"/>
                </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "s_import_cd">
                <xsd:restriction base = "xsd:string">
                        <xsd:enumeration value = "A"/>
                        <xsd:enumeration value = "C"/>
```

```
                <xsd:enumeration value = "D"/>
                <xsd:enumeration value = "B"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "external_user">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "empl_group_cd">
            <xsd:restriction base = "xsd:string">
                                <xsd:maxLength value = "25"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
    </xsd:simpleType>
        <xsd:simpleType name = "required_empl_group_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "empl_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>

        <xsd:simpleType name = "required_empl_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
    </xsd:simpleType>
```

```
<xsd:simpleType name = "role_cd">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "10"/>
                <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "code1_cd">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "code2_cd">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "code3_cd">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "date1_dt">
        <xsd:restriction base = "xsd:dateTime"/>
</xsd:simpleType>
<xsd:simpleType name = "date2_dt">
        <xsd:restriction base = "xsd:dateTime"/>
</xsd:simpleType>
<xsd:simpleType name = "date3_dt">
        <xsd:restriction base = "xsd:dateTime"/>
</xsd:simpleType>

<xsd:simpleType name = "active_fl">
        <xsd:restriction base = "xsd:boolean"/>
</xsd:simpleType>
<xsd:simpleType name = "ts_schedule_cd">
        <xsd:restriction base = "xsd:string">
```

```
                <xsd:maxLength value = "10"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "required_ts_schedule_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "10"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt01_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "50"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt02_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "50"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt03_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt03_name">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "30"/>
            </xsd:restriction>
```

```
</xsd:simpleType>
<xsd:simpleType name = "udt07_id">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "udt07_name">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "30"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "udt09_id">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "dflt_udt01_id">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "50"/>
                <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "dflt_udt02_id">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "50"/>
                <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "dflt_udt03_id">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
```

```
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt04_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt05_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt06_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt07_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt08_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt09_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
```

```
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt10_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt11_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt12_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt13_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt14_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "dflt_udt15_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
```

```
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt07_link_fl">
            <xsd:restriction base = "xsd:boolean"/>
        </xsd:simpleType>
        <xsd:simpleType name = "udt09_link_fl">
            <xsd:restriction base = "xsd:boolean"/>
        </xsd:simpleType>
        <xsd:simpleType name = "udt07_required_fl">
            <xsd:restriction base = "xsd:boolean"/>
        </xsd:simpleType>
        <xsd:simpleType name = "start_dt">
            <xsd:restriction base = "xsd:dateTime"/>
        </xsd:simpleType>
        <xsd:simpleType name = "end_dt">
            <xsd:restriction base = "xsd:dateTime"/>
        </xsd:simpleType>
        <xsd:simpleType name = "allow_charge_fl">
            <xsd:restriction base = "xsd:boolean"/>
        </xsd:simpleType>
        <xsd:simpleType name = "inactive_dt">
            <xsd:restriction base = "xsd:dateTime"/>
        </xsd:simpleType>
        <xsd:simpleType name = "s_date_edit_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:enumeration value = "H"/>
                <xsd:enumeration value = "S"/>
                <xsd:enumeration value = "N"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "line_approve_fl">
            <xsd:restriction base = "xsd:boolean"/>
        </xsd:simpleType>
        <xsd:simpleType name = "spvsr_empl_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
```

```
            </xsd:restriction>
        </xsd:simpleType>
         <xsd:simpleType name = "required_spvsr_empl_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType name = "functional_role_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "10"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
         <xsd:simpleType name = "required_functional_role_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "10"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType name = "cp_company_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "10"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType name = "required_cp_company_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "10"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
         </xsd:simpleType>
         <xsd:simpleType name = "udt06_cp_company_cd">
            <xsd:restriction base = "xsd:string">
```

```
                <xsd:maxLength value = "10"/>
                <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "udt07_cp_company_cd">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "10"/>
                <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "rate1_rate">
        <xsd:restriction base = "xsd:decimal">
          <xsd:totalDigits value="15" />
          <xsd:fractionDigits value="5" />
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "rate2_rate">
        <xsd:restriction base = "xsd:decimal">
          <xsd:totalDigits value="15" />
          <xsd:fractionDigits value="5" />
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "udt01_abbrv_id">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "udt02_abbrv_id">
        <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "udt09_abbrv_id">
        <xsd:restriction base = "xsd:string">
```

```
                <xsd:maxLength value = "20"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "leave_type_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "10"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "effective_dt">
            <xsd:restriction base = "xsd:dateTime"/>
    </xsd:simpleType>
    <xsd:simpleType name = "exp_rpt_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "10"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
</xsd:schema>
```

TCDATA

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
        elementFormDefault = "qualified">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:include schemaLocation = "import_ts_rejection_schema.xsd"/>
        <xsd:include schemaLocation = "import_empl_hist_schema.xsd"/>
        <xsd:include schemaLocation = "import_empl_leave_schema.xsd"/>
        <xsd:include schemaLocation = "import_empl_grp_schema.xsd"/>
        <xsd:include schemaLocation = "import_empl_schema.xsd"/>
        <xsd:include schemaLocation = "import_chrg_schema.xsd"/>
        <xsd:include schemaLocation = "import_link12_schema.xsd"/>
        <xsd:include schemaLocation = "import_link17_schema.xsd"/>
```

```
    <xsd:include schemaLocation = "import_link19_schema.xsd"/>
    <xsd:include schemaLocation = "import_link27_schema.xsd"/>
    <xsd:include schemaLocation = "import_link29_schema.xsd"/>
    <xsd:include schemaLocation = "import_link93_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt01_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt02_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt03_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt04_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt05_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt06_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt07_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt08_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt09_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt10_schema.xsd"/>
    <xsd:include schemaLocation = "import_ts_invoice_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt01_spvsr_schema.xsd"/>
    <xsd:include schemaLocation = "import_udt02_spvsr_schema.xsd"/>
      <xsd:include schemaLocation = "import_cp_company_schema.xsd"/>
    <xsd:include schemaLocation = "import_leave_type_schema.xsd"/>
    <xsd:include schemaLocation = "import_exp_rejection_schema.xsd"/>
    <xsd:include schemaLocation = "import_per_diem_schedule_schema.xsd"/>
    <xsd:include schemaLocation = "import_currency_schedule_schema.xsd"/>
<xsd:include schemaLocation = "import_expense_report_payment_schema.xsd"/>
    <xsd:element name = "importTCData">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref = "tcData" maxOccurs = "unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:element name = "tcData">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element ref = "tcControl"/>
                <xsd:choice>
                    <xsd:element ref = “ts_rejection”/>
```

```
<xsd:element ref = "ts"/>
<xsd:element ref = "ts_line"/>
<xsd:element ref = "ts_cell"/>
<xsd:element ref = "charge"/>
<xsd:element ref = "charge_branch"/>
<xsd:element ref = "charge_branch_charge"/>
<xsd:element ref = "charge_empl_group"/>
<xsd:element ref = "charge_branch_empl_group"/>
<xsd:element ref = "charge_branch_charge_empl"/>
<xsd:element ref = "empl"/>
<xsd:element ref = "empl_history"/>
<xsd:element ref = "empl_leave"/>
<xsd:element ref = "empl_group"/>
<xsd:element ref = "empl_group_empl"/>
<xsd:element ref = "empl_group_spvsr"/>
<xsd:element ref = "link12"/>
<xsd:element ref = "link17"/>
<xsd:element ref = "link19"/>
<xsd:element ref = "link27"/>
<xsd:element ref = "link29"/>
<xsd:element ref = "link93"/>
<xsd:element ref = "udt01"/>
<xsd:element ref = "udt02"/>
<xsd:element ref = "udt03"/>
<xsd:element ref = "udt04"/>
<xsd:element ref = "udt05"/>
<xsd:element ref = "udt06"/>
<xsd:element ref = "udt07"/>
<xsd:element ref = "udt08"/>
<xsd:element ref = "udt09"/>
<xsd:element ref = "udt10"/>
<xsd:element ref = "ts_invoice"/>
<xsd:element ref = "udt01_spvsr"/>
<xsd:element ref = "udt02_spvsr"/>
<xsd:element ref = "cp_company"/>
<xsd:element ref = "leave_type"/>
```

```
                                <xsd:element ref = "exp_rejection"/>
                                <xsd:element ref = "per_diem_schedule"/>
                                <xsd:element ref = "currency_schedule"/>
                                <xsd:element ref = "expense_report_payment"/></xsd:choice>
                        </xsd:sequence>
                        <xsd:attribute name = "version" use = "optional" type = "xsd:integer"/>
                </xsd:complexType>
        </xsd:element>
</xsd:schema>
```

Employee

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation = "import_common.xsd"/>
   <xsd:element name = "empl">
     <xsd:complexType>
        <xsd:sequence>
              <xsd:element name = "empl_id" type = "required_empl_id"/>
              <xsd:element name = "last_name" type = "last_name" minOccurs = “0”/>
              <xsd:element name = "first_name" type = "first_name" minOccurs = “0”/>
              <xsd:element name = "initial_name" type = "initial_name" minOccurs = "0"/>
              <xsd:element name = "govt_empl_id" type = "govt_empl_id" minOccurs = “0”/>
              <xsd:element name = "vendor_id" type = "vendor_id" minOccurs = "0"/>
              <xsd:element name = "payroll_id" type = "payroll_id" minOccurs = "0"/>
              <xsd:element name = "payroll_empl_id" type = "payroll_empl_id" minOccurs = "0"/>
              <xsd:element name = "hire_dt" type = "hire_dt" minOccurs = "0"/>
              <xsd:element name = "terminate_dt" type = "terminate_dt" minOccurs = "0"/>
              <xsd:element name = "home_phone_no_s" type = "home_phone_no_s" minOccurs = "0"/>
              <xsd:element name = "work_phone_no_s" type = "work_phone_no_s" minOccurs = "0"/>
              <xsd:element name = "fax_no_s" type = "fax_no_s" minOccurs = "0"/>
              <xsd:element name = "cell_phone_no_s" type = "cell_phone_no_s" minOccurs = "0"/>
              <xsd:element name = "pager_no_s" type = "pager_no_s" minOccurs = "0"/>
              <xsd:element name = "mail_address_s" type = "mail_address_s" minOccurs = "0"/>
              <xsd:element name = "login_empl_id" type = "login_empl_id" minOccurs = "0"/>
```

```
            <xsd:element name = "role_cd" type = "role_cd" minOccurs = "0"/>
            <xsd:element name = "code1_cd" type = "code1_cd" minOccurs = "0"/>
            <xsd:element name = "code2_cd" type = "code2_cd" minOccurs = "0"/>
            <xsd:element name = "code3_cd" type = "code3_cd" minOccurs = "0"/>
            <xsd:element name = "date1_dt" type = "date1_dt" minOccurs = "0"/>
            <xsd:element name = "date2_dt" type = "date2_dt" minOccurs = "0"/>
            <xsd:element name = "date3_dt" type = "date3_dt" minOccurs = "0"/>
            <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
            <xsd:element name = "currency_cd" type = "currency_cd" minOccurs = "0"/>
        </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name = "last_name">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "first_name">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "initial_name">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "1"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "govt_empl_id">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "20"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "vendor_id">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "12"/>
    </xsd:restriction>
```

```
</xsd:simpleType>
<xsd:simpleType name = "payroll_id">
  <xsd:restriction base = "xsd:string">
     <xsd:maxLength value = "20"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "payroll_empl_id">
  <xsd:restriction base = "xsd:string">
     <xsd:maxLength value = "20"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "hire_dt">
  <xsd:restriction base = "xsd:dateTime"/>
</xsd:simpleType>
<xsd:simpleType name = "terminate_dt">
  <xsd:restriction base = "xsd:dateTime"/>
</xsd:simpleType>
<xsd:simpleType name = "home_phone_no_s">
  <xsd:restriction base = "xsd:string">
     <xsd:maxLength value = "30"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "work_phone_no_s">
  <xsd:restriction base = "xsd:string">
     <xsd:maxLength value = "30"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "fax_no_s">
  <xsd:restriction base = "xsd:string">
     <xsd:maxLength value = "30"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "cell_phone_no_s">
  <xsd:restriction base = "xsd:string">
     <xsd:maxLength value = "30"/>
  </xsd:restriction>
```

```
    </xsd:simpleType>
    <xsd:simpleType name = "pager_no_s">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "mail_address_s">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "254"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "login_empl_id">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "20"/>
      </xsd:restriction>
    </xsd:simpleType>
  <xsd:simpleType name = "currency_cd">
            <xsd:restriction base = "xsd:string">
                    <xsd:maxLength value = "3"/>
            </xsd:restriction>
    </xsd:simpleType>
  </xsd:schema>
```

Employee History

```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
  <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "empl_history">
      <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "empl_id" type = "required_empl_id"/>
            <xsd:element name = "effective_dt" type = "effective_dt"/>
            <xsd:element name = "work_schedule_cd" type = "work_schedule_cd" minOccurs = "0"/>
            <xsd:element name = "class_cd" type = "class_cd" minOccurs = "0"/>
            <xsd:element name = "ts_schedule_cd" type = "ts_schedule_cd" minOccurs = "0"/>
```

```
            <xsd:element name = "dflt_udt01_id" type = "dflt_udt01_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt02_id" type = "dflt_udt02_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt03_id" type = "dflt_udt03_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt04_id" type = "dflt_udt04_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt05_id" type = "dflt_udt05_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt06_id" type = "dflt_udt06_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt07_id" type = "dflt_udt07_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt08_id" type = "dflt_udt08_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt09_id" type = "dflt_udt09_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt10_id" type = "dflt_udt10_id" minOccurs = "0"/>
            <xsd:element name = "udt06_cp_company_cd" type = "udt06_cp_company_cd" minOccurs
            = "0"/>
            <xsd:element name = "udt07_cp_company_cd" type = "udt07_cp_company_cd" minOccurs
            = "0"/>
            <xsd:element name = "rate1_rate" type = "rate1_rate" minOccurs = "0"/>
            <xsd:element name = "rate2_rate" type = "rate2_rate" minOccurs = "0"/>
            <xsd:element name = "fringe_reduction_rate" type = "fringe_reduction_rate" minOccurs =
            "0"/>
              <xsd:element name = "exp_class_cd" type = "exp_class_cd" minOccurs = "0"/>
              <xsd:element name = "dflt_udt11_id" type = "dflt_udt11_id" minOccurs = "0"/>
              <xsd:element name = "dflt_udt12_id" type = "dflt_udt12_id" minOccurs = "0"/>
              <xsd:element name = "dflt_udt13_id" type = "dflt_udt13_id" minOccurs = "0"/>
              <xsd:element name = "dflt_udt14_id" type = "dflt_udt14_id" minOccurs = "0"/>
              <xsd:element name = "dflt_udt15_id" type = "dflt_udt15_id" minOccurs = "0"/>
        </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
     <xsd:simpleType name = "work_schedule_cd">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "10"/>
        <xsd:pattern value = "[^a-z]*"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "class_cd">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "20"/>
```

```
        <xsd:pattern value = "[^a-z]*"/>
      </xsd:restriction>
  </xsd:simpleType>
<xsd:simpleType name = "fringe_reduction_rate">
                <xsd:restriction base = "xsd:decimal">
                  <xsd:totalDigits value="15" />
                  <xsd:fractionDigits value="5" />
                </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "exp_class_cd">
                <xsd:restriction base = "xsd:string">
                        <xsd:maxLength value = "20"/>
                        <xsd:pattern value = "[^a-z]*"/>
                </xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

Employee Leave

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation = "import_common.xsd"/>
   <xsd:element name = "empl_leave">
     <xsd:complexType>
        <xsd:sequence>
              <xsd:element name = "empl_id" type = "required_empl_id"/>
              <xsd:element name = "leave_type_cd" type = "leave_type_cd"/>
              <xsd:element name = "trans_dt" type = "trans_dt"/>
              <xsd:element name = "s_trans_type_cd" type = "s_trans_type_cd"/>
              <xsd:element name = "leave_hrs" type = "leave_hrs" minOcurs = “0”/>
              <xsd:element name = "adjustment_text" type = "adjustment_text" minOccurs = "0"/>
        </xsd:sequence>
     </xsd:complexType>
   </xsd:element> <xsd:simpleType name = "trans_dt">
     <xsd:restriction base = "xsd:dateTime"/>
   </xsd:simpleType>
```

```
    <xsd:simpleType name = "s_trans_type_cd">
      <xsd:restriction base = "xsd:string">
         <xsd:enumeration value = "BEG"/>
         <xsd:enumeration value = "ACCRUED"/>
         <xsd:enumeration value = "TAKEN"/>
         <xsd:enumeration value = "ADJUSTMENT"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "leave_hrs">
      <xsd:restriction base = "xsd:decimal"/>
          <xsd:totalDigits value=”10” />
          <xsd:fractionDigits value="4" />
       </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "adjustment_text">
      <xsd:restriction base = "xsd:string">
         <xsd:maxLength value = "60"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:schema>
```

Employee Groups

```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
  <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>

     <xsd:element name = "empl_group">
        <xsd:complexType>
           <xsd:sequence>
                        <xsd:element name = "empl_group_cd" type = "required_empl_group_cd"/>
              <xsd:element name = "s_import_type_cd" type = "s_import_type_cd"/>
              <xsd:element name = "empl_group_desc" type = "empl_group_desc" minOccurs = "0"/>
              <xsd:element name = "empl_group_type_cd" type = "empl_group_type_cd" minOccurs =
              "0"/>
              <xsd:element name = "empl_id" type = "empl_id" minOccurs = "0"/>
```

```
            <xsd:element name = "spvsr_empl_id" type = "spvsr_empl_id" minOccurs = "0"/>
            <xsd:element name = "functional_role_cd" type = "functional_role_cd" minOccurs = "0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name = "empl_group_empl">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "empl_group_cd" type = "required_empl_group_cd"/>
            <xsd:element name = "s_import_type_cd" type = "s_import_type_cd"/>
            <xsd:element name = "empl_group_desc" type = "empl_group_desc" minOccurs = "0"/>
            <xsd:element name = "empl_group_type_cd" type = "empl_group_type_cd" minOccurs =
            "0"/>
            <xsd:element name = "empl_id" type = "required_empl_id"/>
            <xsd:element name = "spvsr_empl_id" type = "spvsr_empl_id" minOccurs = "0"/>
            <xsd:element name = "functional_role_cd" type = "functional_role_cd" minOccurs = "0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

<xsd:element name = "empl_group_spvsr">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "empl_group_cd" type = "required_empl_group_cd"/>
            <xsd:element name = "s_import_type_cd" type = "s_import_type_cd"/>
            <xsd:element name = "empl_group_desc" type = "empl_group_desc" minOccurs = "0"/>
            <xsd:element name = "empl_group_type_cd" type = "empl_group_type_cd" minOccurs =
            "0"/>
            <xsd:element name = "empl_id" type = "empl_id" minOccurs = "0"/>
            <xsd:element name = "spvsr_empl_id" type = "required_spvsr_empl_id"/>
            <xsd:element name = "functional_role_cd" type = "required_functional_role_cd"
            minOccurs = "0"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
```

```
    <xsd:simpleType name = "s_import_type_cd">
        <xsd:restriction base = "xsd:string">
            <xsd:enumeration value = "G"/>
            <xsd:enumeration value = "M"/>
            <xsd:enumeration value = "O"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "empl_group_desc">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "30"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "empl_group_type_cd">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "10"/>
            <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

UDT01

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt01">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "udt01_id" type = "udt01_id"/>
                <xsd:element name = "udt01_name" type = "udt01_name" minOccurs = "0"/>
                <xsd:element name = "udt01_type_cd" type = "udt01_type_cd" minOccurs = "0"/>
                <xsd:element name = "code1_cd" type = "code1_cd" minOccurs = "0"/>
                <xsd:element name = "code2_cd" type = "code2_cd" minOccurs = "0"/>
                <xsd:element name = "code3_cd" type = "code3_cd" minOccurs = "0"/>
```

```
                <xsd:element name = "date1_dt" type = "date1_dt" minOccurs = "0"/>
                <xsd:element name = "date2_dt" type = "date2_dt" minOccurs = "0"/>
                <xsd:element name = "date3_dt" type = "date3_dt" minOccurs = "0"/>
                <xsd:element name = "udt02_link_fl" type = "udt02_link_fl" minOccurs = "0"/>
                <xsd:element name = "udt07_link_fl" type = "udt07_link_fl" minOccurs = "0"/>
                <xsd:element name = "udt09_link_fl" type = "udt09_link_fl" minOccurs = "0"/>
                <xsd:element name = "udt07_required_fl" type = "udt07_required_fl" minOccurs = "0"/>
                <xsd:element name = "start_dt" type = "start_dt" minOccurs = "0"/>
                <xsd:element name = "end_dt" type = "end_dt" minOccurs = "0"/>
                <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                <xsd:element name = "allow_charge_fl" type = "allow_charge_fl" minOccurs = "0"/>
                <xsd:element name = "inactive_dt" type = "inactive_dt" minOccurs = "0"/>
                <xsd:element name = "s_date_edit_cd" type = "s_date_edit_cd" minOccurs = "0"/>
                <xsd:element name = "line_approve_fl" type = "line_approve_fl" minOccurs = "0"/>
            <xsd:element name = "udt01_abbrv_id" type = "udt01_abbrv_id" minOccurs = "0"/>
        </xsd:sequence>
    </xsd:complexType>
      </xsd:element>
      <xsd:simpleType name = "udt01_name">
    <xsd:restriction base = "xsd:string">
          <xsd:maxLength value = "120"/>
    </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt01_type_cd">
<xsd:restriction base = "xsd:string">
          <xsd:maxLength value = "10"/>
          <xsd:pattern value = "[^a-z]*"/>
</xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "udt02_link_fl">
    <xsd:restriction base = "xsd:boolean"/>
    </xsd:simpleType>
</xsd:schema>
```

UDT02

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt02">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "udt02_id" type = "udt02_id"/>
                <xsd:element name = "udt02_name" type = "udt02_name" minOccurs = "0"/>
                <xsd:element name = "code1_cd" type = "code1_cd" minOccurs = "0"/>
                <xsd:element name = "code2_cd" type = "code2_cd" minOccurs = "0"/>
                <xsd:element name = "code3_cd" type = "code3_cd" minOccurs = "0"/>
                <xsd:element name = "date1_dt" type = "date1_dt" minOccurs = "0"/>
                <xsd:element name = "date2_dt" type = "date2_dt" minOccurs = "0"/>
                <xsd:element name = "date3_dt" type = "date3_dt" minOccurs = "0"/>
                <xsd:element name = "udt01_link_fl" type = "udt01_link_fl" minOccurs = "0"/>
                <xsd:element name = "udt07_link_fl" type = "udt07_link_fl" minOccurs = "0"/>
                <xsd:element name = "udt09_link_fl" type = "udt09_link_fl" minOccurs = "0"/>
                <xsd:element name = "udt07_required_fl" type = "udt07_required_fl" minOccurs = "0"/>
                <xsd:element name = "start_dt" type = "start_dt" minOccurs = "0"/>
                <xsd:element name = "end_dt" type = "end_dt" minOccurs = "0"/>
                <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                <xsd:element name = "allow_charge_fl" type = "allow_charge_fl" minOccurs = "0"/>
                <xsd:element name = "inactive_dt" type = "inactive_dt" minOccurs = "0"/>
                <xsd:element name = "s_date_edit_cd" type = "s_date_edit_cd" minOccurs = "0"/>
                <xsd:element name = "line_approve_fl" type = "line_approve_fl" minOccurs = "0"/>
                  <xsd:element name = "cp_company_cd" type = "cp_company_cd" minOccurs = "0"/>
                  <xsd:element name = "udt02_abbrv_id" type = "udt02_abbrv_id" minOccurs = "0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:simpleType name = "udt02_name">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "120"/>
```

```
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "udt01_link_fl">
        <xsd:restriction base = "xsd:boolean"/>
    </xsd:simpleType>
</xsd:schema>
```

UDT03

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation = "import_common.xsd"/>
   <xsd:element name = "udt03">
     <xsd:complexType>
        <xsd:sequence>
              <xsd:element name = "udt03_id" type = "udt03_id"/>
              <xsd:element name = "udt03_name" type = "udt03_name" minOccurs = “0”/>
              <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
        </xsd:sequence>
     </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

UDT04

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation = "import_common.xsd"/>
   <xsd:element name = "udt04">
     <xsd:complexType>
        <xsd:sequence>
              <xsd:element name = "udt04_id" type = "udt04_id"/>
              <xsd:element name = "udt04_name" type = "udt04_name" minOccurs = “0”/>
              <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
              <xsd:element name = "cp_company_cd" type = "cp_company_cd" minOccurs = "0"/>
              <xsd:element name = "udt04_rate" type = "udt04_rate" default = "0.00"/>
```

```
      </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:simpleType name = "udt04_id">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "20"/>
        <xsd:minLength value = "1"/>
        <xsd:pattern value = "[^a-z]*"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "udt04_name">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
      </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "udt04_rate">
                <xsd:restriction base = "xsd:decimal">
                  <xsd:totalDigits value="15" />
                  <xsd:fractionDigits value="5" />
                </xsd:restriction>          </xsd:simpleType>
  </xsd:schema>
```

UDT05

```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
  <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt05">
      <xsd:complexType>
        <xsd:sequence>
              <xsd:element name = "udt05_id" type = "udt05_id"/>
              <xsd:element name = "udt05_name" type = "udt05_name" minOccurs = “0”/>
              <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
              <xsd:element name = "cp_company_cd" type = "cp_company_cd" minOccurs = "0"/>
        </xsd:sequence>
      </xsd:complexType>
```

```
    </xsd:element>
    <xsd:simpleType name = "udt05_id">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "20"/>
        <xsd:minLength value = "1"/>
        <xsd:pattern value = "[^a-z]*"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "udt05_name">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
      </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

UDT06

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt06">
      <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "udt06_id" type = "udt06_id"/>
            <xsd:element name = "udt06_name" type = "udt06_name" minOccurs = “0”/>
            <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
            <xsd:element name = "cp_company_cd" type = "required_cp_company_cd"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:simpleType name = "udt06_id">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "20"/>
        <xsd:minLength value = "1"/>
        <xsd:pattern value = "[^a-z]*"/>
      </xsd:restriction>
```

```
    </xsd:simpleType>
    <xsd:simpleType name = "udt06_name">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:schema>
```

UDT07

```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
  <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt07">
      <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "udt07_id" type = "udt07_id"/>
            <xsd:element name = "udt07_name" type = "udt07_name" minOccurs = “0”/>
            <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
            <xsd:element name = "cp_company_cd" type = "required_cp_company_cd"/>
            <xsd:element name = "rate1_rate" type = "rate1_rate" minOccurs = "0"/>
            <xsd:element name = "rate2_rate" type = "rate2_rate" minOccurs = "0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
```

UDT08

```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
  <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt08">
      <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "udt08_id" type = "udt08_id"/>
```

```
            <xsd:element name = "udt08_name" type = "udt08_name" minOccurs = “0”/>
            <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:simpleType name = "udt08_id">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "20"/>
        <xsd:minLength value = "1"/>
        <xsd:pattern value = "[^a-z]*"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "udt08_name">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:schema>
```

UDT09

```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
  <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt09">
      <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "udt09_id" type = "udt09_id"/>
            <xsd:element name = "udt09_name" type = "udt09_name" minOccurs = “0”/>
            <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
            <xsd:element name = "cp_company_cd" type = "cp_company_cd" minOccurs = "0"/>
            <xsd:element name = "udt09_abbrv_id" type = "udt09_abbrv_id" minOccurs = "0"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
```

```
    <xsd:simpleType name = "udt09_name">
      <xsd:restriction base = "xsd:string">
         <xsd:maxLength value = "30"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:schema>
```

UDT10

```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
  <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt10">
      <xsd:complexType>
         <xsd:sequence>
               <xsd:element name = "udt10_id" type = "udt01_id"/>
               <xsd:element name = "udt10_name" type = "udt10_name" minOccurs = “0”/>
               <xsd:element name = "ot_fl" type = "ot_fl" minOccurs = "0"/>
               <xsd:element name = "ot_udt10_id" type = "ot_udt10_id" minOccurs = "0"/>
               <xsd:element name = "cost_fl" type = "cost_fl" minOccurs = "0"/>
               <xsd:element name = "text_fl" type = "text_fl" minOccurs = "0"/>
               <xsd:element name = "misc_labor_cd" type = "misc_labor_cd" minOccurs = "0"/>
               <xsd:element name = "misc_payroll_cd" type = "misc_payroll_cd" minOccurs = "0"/>
               <xsd:element name = "payroll_earnings_cd" type = "payroll_earnings_cd" minOccurs =
               "0"/>
               <xsd:element name = "s_payroll_map_cd" type = "s_payroll_map_cd" minOccurs = "0"/>
               <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
               <xsd:element name = "rate1_factor_amt" type = "rate1_factor_amt" minOccurs = "0"/>
     <xsd:element name = "rate1_fixed_amt" type = "rate1_fixed_amt" minOccurs = "0"/>
         <xsd:element name = "s_rate1_calc_cd" type = "s_rate1_calc_cd" minOccurs = "0"/>
               <xsd:element name = "rate2_factor_amt" type = "rate2_factor_amt" minOccurs = "0"/>
               <xsd:element name = "rate2_fixed_amt" type = "rate2_fixed_amt" minOccurs = "0"/>

               <xsd:element name = "s_rate2_calc_cd" type = "s_rate2_calc_cd" minOccurs = "0"/>
               <xsd:element name = "recast_labor_udt10_id" type = "recast_labor_udt10_id" minOccurs =
               "0"/>
```

```
        <xsd:element name = "recast_payroll_udt10_id" type = "recast_payroll_udt10_id" minOccurs =
        "0"/>
    <xsd:element name = "prorate_fl" type = "prorate_fl" minOccurs = "0"/>
        </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:simpleType name = "udt10_id">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "20"/>
        <xsd:minLength value = "1"/>
        <xsd:pattern value = "[^a-z]*"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "udt10_name">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "ot_fl">
    <xsd:restriction base = "xsd:boolean"/>
  </xsd:simpleType>
  <xsd:simpleType name = "ot_udt10_id">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "20"/>
        <xsd:pattern value = "[^a-z]*"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "cost_fl">
    <xsd:restriction base = "xsd:boolean"/>
  </xsd:simpleType>
  <xsd:simpleType name = "text_fl">
    <xsd:restriction base = "xsd:boolean"/>
  </xsd:simpleType>
  <xsd:simpleType name = "misc_labor_cd">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
```

```
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "misc_payroll_cd">
    <xsd:restriction base = "xsd:string">
      <xsd:maxLength value = "30"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "payroll_earnings_cd">
    <xsd:restriction base = "xsd:string">
      <xsd:maxLength value = "30"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "s_payroll_map_cd">
    <xsd:restriction base = "xsd:string">
      <xsd:enumeration value = "N"/>
      <xsd:enumeration value = "R"/>
      <xsd:enumeration value = "O"/>
      <xsd:enumeration value = "3"/>
      <xsd:enumeration value = "4"/>
      <xsd:enumeration value = "5"/>
      <xsd:enumeration value = "H3"/>
      <xsd:enumeration value = "H4"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name = "rate1_factor_amt">
            <xsd:restriction base = "xsd:decimal">
              <xsd:totalDigits value="15" />
              <xsd:fractionDigits value="5" />
            </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "rate1_fixed_amt">
            <xsd:restriction base = "xsd:decimal">
              <xsd:totalDigits value="15" />
              <xsd:fractionDigits value="5" />
            </xsd:restriction>
    </xsd:simpleType>
```

```
<xsd:simpleType name = "s_rate1_calc_cd">
<xsd:restriction base = "xsd:string">
        <xsd:enumeration value = "CALC1"/>
                <xsd:enumeration value = "CALC2"/>
                <xsd:enumeration value = "CALC3"/>
                <xsd:maxLength value = "5"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "rate2_factor_amt">
        <xsd:restriction base = "xsd:decimal">
          <xsd:totalDigits value="15" />
          <xsd:fractionDigits value="5" />
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "rate2_fixed_amt">
        <xsd:restriction base = "xsd:decimal">
          <xsd:totalDigits value="15" />
          <xsd:fractionDigits value="5" />
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "s_rate2_calc_cd">
<xsd:restriction base = "xsd:string">
        <xsd:enumeration value = "CALC1"/>
        <xsd:enumeration value = "CALC2"/>
        <xsd:enumeration value = "CALC3"/>
                <xsd:maxLength value = "5"/>
</xsd:restriction>
</xsd:simpleType>
  <xsd:simpleType name = "recast_payroll_udt10_id">
<xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "recast_labor_udt10_id">
<xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
```

```
        </xsd:restriction>
      </xsd:simpleType>
        <xsd:simpleType name = "prorate_fl">
                <xsd:restriction base = "xsd:boolean"/></xsd:simpleType>
</xsd:schema>
```

UDT11

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "udt11">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name = "udt11_id" type = "udt11_id"/>
                                <xsd:element name = "udt11_name" type = "udt11_name" minOccurs =
                                "0"/>
                                <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "udt11_id">
                <xsd:restriction base = "xsd:string">
                        <xsd:maxLength value = "20"/>
                        <xsd:minLength value = "1"/>
                        <xsd:pattern value = "[^a-z]*"/>
                </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt11_name">
                <xsd:restriction base = "xsd:string">
                        <xsd:maxLength value = "30"/>
                </xsd:restriction>
        </xsd:simpleType>
</xsd:schema>
```

UDT12

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt12">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "udt12_id" type = "udt12_id"/>
                <xsd:element name = "udt12_name" type = "udt12_name" minOccurs =
                "0"/>
                <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:simpleType name = "udt12_id">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "20"/>
            <xsd:minLength value = "1"/>
            <xsd:pattern value = "[^a-z]*"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "udt12_name">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "30"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

UDT13

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt13">
```

```
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name = "udt13_id" type = "udt13_id"/>
                    <xsd:element name = "udt13_name" type = "udt13_name" minOccurs =
                    "0"/>
                    <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "udt13_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt13_name">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "30"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:schema>
```

UDT14

```
    <?xml version = "1.0" encoding = "UTF-8"?>
    <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
    <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "udt14">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name = "udt14_id" type = "udt14_id"/>
                    <xsd:element name = "udt14_name" type = "udt14_name" minOccurs =
                    "0"/>
                    <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                </xsd:sequence>
```

```
            </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "udt14_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt14_name">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "30"/>
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:schema>
```

UDT15

```
    <?xml version = "1.0" encoding = "UTF-8"?>
    <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
    <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "udt15">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name = "udt15_id" type = "udt15_id"/>
                    <xsd:element name = "udt15_name" type = "udt15_name" minOccurs =
                    "0"/>
                    <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "udt15_id">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
                <xsd:minLength value = "1"/>
```

```
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "udt15_name">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "30"/>
            </xsd:restriction>
        </xsd:simpleType>
</xsd:schema>
```

UDT01 Supervisors

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "udt01_spvsr">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name = "udt01_id" type = "udt01_id"/>
                    <xsd:element name = "spvsr_empl_id" type = "required_spvsr_empl_id"/>
                    <xsd:element name = "functional_role_cd" type =
                     "required_functional_role_cd"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
</xsd:schema>
```

UDT02 Supervisors

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "udt02_spvsr">
        <xsd:complexType>
```

```
            <xsd:sequence>
                <xsd:element name = "udt02_id" type = "udt02_id"/>
                <xsd:element name = "spvsr_empl_id" type = "required_spvsr_empl_id"/>
                <xsd:element name = "functional_role_cd" type = "required_functional_role_cd"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

UDT01/ UDT02 Link

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation = "import_common.xsd"/>
   <xsd:element name = "link12">
     <xsd:complexType>
        <xsd:sequence>
              <xsd:element name = "udt01_id" type = "udt01_id"/>
              <xsd:element name = "udt02_id" type = "udt02_id"/>
        </xsd:sequence>
     </xsd:complexType>
   </xsd:element>
</xsd:schema>
```

UDT01/ UDT07 Link

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation = "import_common.xsd"/>
   <xsd:element name = "link17">
     <xsd:complexType>
        <xsd:sequence>
              <xsd:element name = "udt01_id" type = "udt01_id"/>
              <xsd:element name = "udt07_id" type = "udt07_id"/>
              <xsd:element name = "udt07_name" type = "udt07_name" minOccurs = "0"/>
```

```
            <xsd:element name = "rate1_rate" type = "rate1_rate" minOccurs = "0"/>
            <xsd:element name = "rate2_rate" type = "rate2_rate" minOccurs = "0"/>
            <xsd:element name = "cp_company_cd" type = "required_cp_company_cd"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
```

UDT01/ UDT09 Link

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation = "import_common.xsd"/>
  <xsd:element name = "link19">
    <xsd:complexType>
      <xsd:sequence>
          <xsd:element name = "udt01_id" type = "udt01_id"/>
          <xsd:element name = "udt09_id" type = "udt09_id"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

UDT02/ UDT07 Link

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation = "import_common.xsd"/>
  <xsd:element name = "link27">
    <xsd:complexType>
      <xsd:sequence>
          <xsd:element name = "udt02_id" type = "udt02_id"/>
          <xsd:element name = "udt07_id" type = "udt07_id"/>
          <xsd:element name = "udt07_name" type = "udt07_name" minOccurs = "0"/>
```

```
            <xsd:element name = "rate1_rate" type = "rate1_rate" minOccurs = "0"/>
            <xsd:element name = "rate2_rate" type = "rate2_rate" minOccurs = "0"/>
            <xsd:element name = "cp_company_cd" type = "required_cp_company_cd"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
```

UDT02/ UDT09 Link

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation = "import_common.xsd"/>
  <xsd:element name = "link29">
    <xsd:complexType>
      <xsd:sequence>
            <xsd:element name = "udt02_id" type = "udt02_id"/>
            <xsd:element name = "udt09_id" type = "udt09_id"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

UDT09/ UDT03 Link

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation = "import_common.xsd"/>
  <xsd:element name = "link93">
    <xsd:complexType>
      <xsd:sequence>
            <xsd:element name = "udt09_id" type = "udt09_id"/>
            <xsd:element name = "udt03_id" type = "udt03_id"/>
            <xsd:element name = "udt03_name" type = "udt03_name" minOccurs = "0"/>
      </xsd:sequence>
```

```
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
```

Charge Trees

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
  <xsd:include schemaLocation = "import_common.xsd"/>
  <xsd:element name = "charge">
    <xsd:complexType>
      <xsd:sequence>
          <xsd:element name = "s_import_type_cd" type = "s_chrg_import_type_cd"/>
          <xsd:element name = "charge_tree_cd" type = "charge_tree_cd"/>
          <xsd:element name = "charge_tree_desc" type = "charge_tree_desc" minOccurs = “0”/>
          <xsd:element name = "charge_branch_cd" type = "charge_branch_cd" minOccurs = "0"/>
          <xsd:element name = "parent_charge_branch_cd" type = "parent_charge_branch_cd"
          minOccurs = "0"/>
          <xsd:element name = "charge_branch_desc" type = "charge_branch_desc" minOccurs =
          "0"/>
          <xsd:element name = "charge_cd" type = "charge_cd" minOccurs = “0”/>
          <xsd:element name = "empl_group_cd" type = "empl_group_cd" minOccurs = "0"/>
          <xsd:element name = "empl_id" type = "empl_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt01_id" type = "dflt_udt01_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt02_id" type = "dflt_udt02_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt03_id" type = "dflt_udt03_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt04_id" type = "dflt_udt04_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt05_id" type = "dflt_udt05_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt06_id" type = "dflt_udt06_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt07_id" type = "dflt_udt07_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt08_id" type = "dflt_udt08_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt09_id" type = "dflt_udt09_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt10_id" type = "dflt_udt10_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt11_id" type = "dflt_udt11_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt12_id" type = "dflt_udt12_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt13_id" type = "dflt_udt13_id" minOccurs = "0"/>
          <xsd:element name = "dflt_udt14_id" type = "dflt_udt14_id" minOccurs = "0"/>
```

```
<xsd:element name = "dflt_udt15_id" type = "dflt_udt15_id" minOccurs = "0"/>

<xsd:element name = "restrict_empl_group_fl" type = "restrict_empl_group_fl" minOccurs =
 "0"/>
<xsd:element name = "restrict_empl_fl" type = "restrict_empl_fl" minOccurs = "0"/>
<xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
<xsd:element name = "udt09_abbrv_id" type = "udt09_abbrv_id" minOccurs = "0"/>
<xsd:element name = "udt06_cp_company_cd" type = "udt06_cp_company_cd" minOccurs
 = "0"/>
<xsd:element name = "udt07_cp_company_cd" type = "udt07_cp_company_cd" minOccurs
 = "0"/>
</xsd:sequence>
    </xsd:complexType>
  </xsd:element>
<xsd:element name = "charge_branch">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "s_import_type_cd" type = "s_chrg_import_type_cd"/>
                <xsd:element name = "charge_tree_cd" type = "charge_tree_cd"/>
                <xsd:element name = "charge_tree_desc" type = "charge_tree_desc"
                 minOccurs = "0"/>
                <xsd:element name = "charge_branch_cd" type =
                 "required_charge_branch_cd"/>
                <xsd:element name = "parent_charge_branch_cd" type =
                 "parent_charge_branch_cd" minOccurs = "0"/>
                <xsd:element name = "charge_branch_desc" type = "charge_branch_desc"
                 minOccurs = "0"/>
                <xsd:element name = "charge_cd" type = "charge_cd" minOccurs = "0"/>
                <xsd:element name = "empl_group_cd" type = "empl_group_cd" minOccurs =
                 "0"/>
                <xsd:element name = "empl_id" type = "empl_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt01_id" type = "dflt_udt01_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt02_id" type = "dflt_udt02_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt03_id" type = "dflt_udt03_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt04_id" type = "dflt_udt04_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt05_id" type = "dflt_udt05_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt06_id" type = "dflt_udt06_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt07_id" type = "dflt_udt07_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt08_id" type = "dflt_udt08_id" minOccurs = "0"/>
```

```
                <xsd:element name = "dflt_udt09_id" type = "dflt_udt09_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt10_id" type = "dflt_udt10_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt11_id" type = "dflt_udt11_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt12_id" type = "dflt_udt12_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt13_id" type = "dflt_udt13_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt14_id" type = "dflt_udt14_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt15_id" type = "dflt_udt15_id" minOccurs = "0"/>
                <xsd:element name = "restrict_empl_group_fl" type = "restrict_empl_group_fl"
                 minOccurs = "0"/>
                <xsd:element name = "restrict_empl_fl" type = "restrict_empl_fl" minOccurs =
                 "0"/>
                <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                <xsd:element name = "udt09_abbrv_id" type = "udt09_abbrv_id" minOccurs =
                 "0"/>
                <xsd:element name = "udt06_cp_company_cd" type =
                 "udt06_cp_company_cd" minOccurs = "0"/>
                <xsd:element name = "udt07_cp_company_cd" type =
                 "udt07_cp_company_cd" minOccurs = "0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name = "charge_branch_charge">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "s_import_type_cd" type = "s_chrg_import_type_cd"/>
                <xsd:element name = "charge_tree_cd" type = "charge_tree_cd"/>
                <xsd:element name = "charge_tree_desc" type = "charge_tree_desc"
                 minOccurs = "0"/>
                <xsd:element name = "charge_branch_cd" type =
                 "required_charge_branch_cd"/>
                <xsd:element name = "parent_charge_branch_cd" type =
                 "parent_charge_branch_cd" minOccurs = "0"/>
                <xsd:element name = "charge_branch_desc" type = "charge_branch_desc"
                 minOccurs = "0"/>
                <xsd:element name = "charge_cd" type = "required_charge_cd"/>
                <xsd:element name = "empl_group_cd" type = "empl_group_cd" minOccurs =
                 "0"/>
                <xsd:element name = "empl_id" type = "empl_id" minOccurs = "0"/>
```

```
                <xsd:element name = "dflt_udt01_id" type = "dflt_udt01_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt02_id" type = "dflt_udt02_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt03_id" type = "dflt_udt03_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt04_id" type = "dflt_udt04_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt05_id" type = "dflt_udt05_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt06_id" type = "dflt_udt06_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt07_id" type = "dflt_udt07_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt08_id" type = "dflt_udt08_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt09_id" type = "dflt_udt09_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt10_id" type = "dflt_udt10_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt11_id" type = "dflt_udt11_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt12_id" type = "dflt_udt12_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt13_id" type = "dflt_udt13_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt14_id" type = "dflt_udt14_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt15_id" type = "dflt_udt15_id" minOccurs = "0"/>
                <xsd:element name = "restrict_empl_group_fl" type = "restrict_empl_group_fl"
                 minOccurs = "0"/>
                <xsd:element name = "restrict_empl_fl" type = "restrict_empl_fl" minOccurs =
                 "0"/>
                <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                <xsd:element name = "udt09_abbrv_id" type = "udt09_abbrv_id" minOccurs =
                 "0"/>
                <xsd:element name = "udt06_cp_company_cd" type =
                 "udt06_cp_company_cd" minOccurs = "0"/>
                <xsd:element name = "udt07_cp_company_cd" type =
                 "udt07_cp_company_cd" minOccurs = "0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name = "charge_empl_group">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "s_import_type_cd" type = "s_chrg_import_type_cd"/>
                <xsd:element name = "charge_tree_cd" type = "charge_tree_cd"/>
                <xsd:element name = "charge_tree_desc" type = "charge_tree_desc"
                 minOccurs = "0"/>
```

```
<xsd:element name = "charge_branch_cd" type = "charge_branch_cd"
 minOccurs = "0"/>
<xsd:element name = "parent_charge_branch_cd" type =
 "parent_charge_branch_cd" minOccurs = "0"/>
<xsd:element name = "charge_branch_desc" type = "charge_branch_desc"
 minOccurs = "0"/>
<xsd:element name = "charge_cd" type = "charge_cd" minOccurs = "0"/>
<xsd:element name = "empl_group_cd" type = "required_empl_group_cd"/>
<xsd:element name = "empl_id" type = "empl_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt01_id" type = "dflt_udt01_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt02_id" type = "dflt_udt02_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt03_id" type = "dflt_udt03_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt04_id" type = "dflt_udt04_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt05_id" type = "dflt_udt05_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt06_id" type = "dflt_udt06_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt07_id" type = "dflt_udt07_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt08_id" type = "dflt_udt08_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt09_id" type = "dflt_udt09_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt10_id" type = "dflt_udt10_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt11_id" type = "dflt_udt11_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt12_id" type = "dflt_udt12_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt13_id" type = "dflt_udt13_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt14_id" type = "dflt_udt14_id" minOccurs = "0"/>
<xsd:element name = "dflt_udt15_id" type = "dflt_udt15_id" minOccurs = "0"/>
<xsd:element name = "restrict_empl_group_fl" type = "restrict_empl_group_fl"
 minOccurs = "0"/>
<xsd:element name = "restrict_empl_fl" type = "restrict_empl_fl" minOccurs =
 "0"/>
<xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
<xsd:element name = "udt09_abbrv_id" type = "udt09_abbrv_id" minOccurs =
 "0"/>
<xsd:element name = "udt06_cp_company_cd" type =
 "udt06_cp_company_cd" minOccurs = "0"/>
<xsd:element name = "udt07_cp_company_cd" type =
 "udt07_cp_company_cd" minOccurs = "0"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

```
<xsd:element name = "charge_branch_empl_group">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "s_import_type_cd" type =
            "s_chrg_import_type_cd"/>
            <xsd:element name = "charge_tree_cd" type = "charge_tree_cd"/>
            <xsd:element name = "charge_tree_desc" type = "charge_tree_desc"
            minOccurs = "0"/>
            <xsd:element name = "charge_branch_cd" type =
            "required_charge_branch_cd"/>
            <xsd:element name = "parent_charge_branch_cd" type =
            "parent_charge_branch_cd" minOccurs = "0"/>
            <xsd:element name = "charge_branch_desc" type =
            "charge_branch_desc" minOccurs = "0"/>
            <xsd:element name = "charge_cd" type = "charge_cd" minOccurs = "0"/>
            <xsd:element name = "empl_group_cd" type =
            "required_empl_group_cd"/>
            <xsd:element name = "empl_id" type = "empl_id" minOccurs = "0"/>
            <xsd:element name = "dflt_udt01_id" type = "dflt_udt01_id" minOccurs =
            "0"/>
            <xsd:element name = "dflt_udt02_id" type = "dflt_udt02_id" minOccurs =
            "0"/>
            <xsd:element name = "dflt_udt03_id" type = "dflt_udt03_id" minOccurs =
            "0"/>
            <xsd:element name = "dflt_udt04_id" type = "dflt_udt04_id" minOccurs =
            "0"/>
            <xsd:element name = "dflt_udt05_id" type = "dflt_udt05_id" minOccurs =
            "0"/>
            <xsd:element name = "dflt_udt06_id" type = "dflt_udt06_id" minOccurs =
            "0"/>
            <xsd:element name = "dflt_udt07_id" type = "dflt_udt07_id" minOccurs =
            "0"/>
            <xsd:element name = "dflt_udt08_id" type = "dflt_udt08_id" minOccurs =
            "0"/>
            <xsd:element name = "dflt_udt09_id" type = "dflt_udt09_id" minOccurs =
            "0"/>
            <xsd:element name = "dflt_udt10_id" type = "dflt_udt10_id" minOccurs =
            "0"/>
        <xsd:element name = "dflt_udt11_id" type = "dflt_udt11_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt12_id" type = "dflt_udt12_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt13_id" type = "dflt_udt13_id" minOccurs = "0"/>
```

```
        <xsd:element name = "dflt_udt14_id" type = "dflt_udt14_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt15_id" type = "dflt_udt15_id" minOccurs = "0"/>
                <xsd:element name = "restrict_empl_group_fl" type = "restrict_empl_group_fl"
                 minOccurs = "0"/>
                <xsd:element name = "restrict_empl_fl" type = "restrict_empl_fl" minOccurs =
                 "0"/>
                <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                <xsd:element name = "udt09_abbrv_id" type = "udt09_abbrv_id" minOccurs =
                 "0"/>
                <xsd:element name = "udt06_cp_company_cd" type =
                 "udt06_cp_company_cd" minOccurs = "0"/>
                <xsd:element name = "udt07_cp_company_cd" type =
                 "udt07_cp_company_cd" minOccurs = "0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>

    <xsd:element name = "charge_branch_charge_empl">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "s_import_type_cd" type = "s_chrg_import_type_cd"/>
                <xsd:element name = "charge_tree_cd" type = "charge_tree_cd"/>
                <xsd:element name = "charge_tree_desc" type = "charge_tree_desc"
                 minOccurs = "0"/>
                <xsd:element name = "charge_branch_cd" type =
                 "required_charge_branch_cd"/>
                <xsd:element name = "parent_charge_branch_cd" type =
                 "parent_charge_branch_cd" minOccurs = "0"/>
                <xsd:element name = "charge_branch_desc" type = "charge_branch_desc"
                 minOccurs = "0"/>
                <xsd:element name = "charge_cd" type = "required_charge_cd"/>
                <xsd:element name = "empl_group_cd" type = "empl_group_cd" minOccurs =
                 "0"/>
                <xsd:element name = "empl_id" type = "required_empl_id"/>
                <xsd:element name = "dflt_udt01_id" type = "dflt_udt01_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt02_id" type = "dflt_udt02_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt03_id" type = "dflt_udt03_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt04_id" type = "dflt_udt04_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt05_id" type = "dflt_udt05_id" minOccurs = "0"/>
```

```
                <xsd:element name = "dflt_udt06_id" type = "dflt_udt06_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt07_id" type = "dflt_udt07_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt08_id" type = "dflt_udt08_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt09_id" type = "dflt_udt09_id" minOccurs = "0"/>
                <xsd:element name = "dflt_udt10_id" type = "dflt_udt10_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt11_id" type = "dflt_udt11_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt12_id" type = "dflt_udt12_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt13_id" type = "dflt_udt13_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt14_id" type = "dflt_udt14_id" minOccurs = "0"/>
        <xsd:element name = "dflt_udt15_id" type = "dflt_udt15_id" minOccurs = "0"/>
                <xsd:element name = "restrict_empl_group_fl" type = "restrict_empl_group_fl"
                 minOccurs = "0"/>
                <xsd:element name = "restrict_empl_fl" type = "restrict_empl_fl" minOccurs =
                 "0"/>
                <xsd:element name = "active_fl" type = "active_fl" minOccurs = "0"/>
                <xsd:element name = "udt09_abbrv_id" type = "udt09_abbrv_id" minOccurs =
                 "0"/>
                <xsd:element name = "udt06_cp_company_cd" type =
                 "udt06_cp_company_cd" minOccurs = "0"/>
                <xsd:element name = "udt07_cp_company_cd" type =
                 "udt07_cp_company_cd" minOccurs = "0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
<xsd:simpleType name = "s_chrg_import_type_cd">
  <xsd:restriction base = "xsd:string">
    <xsd:enumeration value = "C"/>
    <xsd:enumeration value = "CB"/>
    <xsd:enumeration value = "CBC"/>
    <xsd:enumeration value = "CEG"/>
    <xsd:enumeration value = "CBEG"/>
    <xsd:enumeration value = "CBCE"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "charge_tree_cd">
  <xsd:restriction base = "xsd:string">
    <xsd:maxLength value = "50"/>
    <xsd:minLength value = "1"/>
```

```
        <xsd:pattern value = "[^a-z]*"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "charge_tree_desc">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "charge_branch_cd">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "50"/>
        <xsd:pattern value = "[^a-z]*"/>
      </xsd:restriction>
   </xsd:simpleType>
       <xsd:simpleType name = "required_charge_branch_cd">
                <xsd:restriction base = "xsd:string">
                        <xsd:maxLength value = "50"/>
                        <xsd:minLength value = "1"/>
                        <xsd:pattern value = "[^a-z]*"/>
                </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "charge_branch_desc">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "parent_charge_branch_cd">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "50"/>
        <xsd:pattern value = "[^a-z]*"/>
      </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "charge_cd">
      <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "100"/>
        <xsd:pattern value = "[^a-z]*"/>
```

```
        </xsd:restriction>
  </xsd:simpleType>
        <xsd:simpleType name = "required_charge_cd">
                <xsd:restriction base = "xsd:string">
                        <xsd:maxLength value = "100"/>
                        <xsd:minLength value = "1"/>
                        <xsd:pattern value = "[^a-z]*"/>
                </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "restrict_empl_group_fl">
      <xsd:restriction base = "xsd:boolean"/>
    </xsd:simpleType>
    <xsd:simpleType name = "restrict_empl_fl">
      <xsd:restriction base = "xsd:boolean"/>
    </xsd:simpleType>
  </xsd:schema>
```

Timesheet Invoices

```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
  <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "ts_invoice">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name = "empl_id" type = "required_empl_id"/>
                                <xsd:element name = "invoice_no_cd" type = "invoice_no_cd"/>
                                <xsd:element name = "invoice_udt_id" type = "invoice_udt_id"/>
                                <xsd:element name = "timesheet_dt" type = "timesheet_dt"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "invoice_no_cd">
                <xsd:restriction base = "xsd:string">
                        <xsd:maxLength value = "15"/>
```

```
<xsd:minLength value = "1"/>
                        <xsd:pattern value = "[^a-z]*"/>
                </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "invoice_udt_id">
                <xsd:restriction base = "xsd:string">
                        <xsd:maxLength value = "50"/>
<xsd:minLength value = "1"/>
                        <xsd:pattern value = "[^a-z]*"/>
                </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "timesheet_dt">
                <xsd:restriction base = "xsd:dateTime"/>
        </xsd:simpleType>
</xsd:schema>
```

Timesheet Rejections

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
   <xsd:include schemaLocation = "import_common.xsd"/>
<xsd:element name = "ts_rejection">
                <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name = "s_reject_type_cd" type = "s_reject_type_cd"/>
                                <xsd:element name = "batch_id" type = "batch_id"/>
                                <xsd:element name = "empl_id" type = "required_empl_id"/>
                                <xsd:element name = "ts_schedule_cd" type =
                                "required_ts_schedule_cd"/>
                                <xsd:element name = "year_no_cd" type = "year_no_cd"/>
                                <xsd:element name = "period_no_cd" type = "period_no_cd"/>
                                <xsd:element name = "line_no" type = "line_no" minOccurs = "0"/>
                                <xsd:element name = "hrs_dt" type = "hrs_dt" minOccurs = "0"/>
                                <xsd:element name = "s_cell_status_cd" type = "s_cell_status_cd"/>
                                <xsd:element name = "rejection_text" type = "rejection_text" minOccurs
                                = "0"/>
```

```
                    </xsd:sequence>
                </xsd:complexType>
            </xsd:element>
        <xsd:element name = "ts">
          <xsd:complexType>
            <xsd:sequence>
                <xsd:element name = "s_reject_type_cd" type = "s_reject_type_cd"/>
                <xsd:element name = "batch_id" type = "batch_id"/>
                <xsd:element name = "empl_id" type = "empl_id"/>
                <xsd:element name = "ts_schedule_cd" type = "ts_schedule_cd"/>
                <xsd:element name = "year_no_cd" type = "year_no_cd"/>
                <xsd:element name = "period_no_cd" type = "period_no_cd"/>
                <xsd:element name = "line_no" type = "xsd:integer" minOccurs = "0"/>
                <xsd:element name = "hrs_date" type = "xsd:datetime" minOccurs = "0"/>
                <xsd:element name = "s_cell_status_cd" type = "s_cell_status_cd"/>
                <xsd:element name = "rejection_text" type = "rejection_text" minOccurs = "0"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
    <xsd:element name = "ts_line">
                <xsd:complexType>
                    <xsd:sequence>
                        <xsd:element name = "s_reject_type_cd" type = "s_reject_type_cd"/>
                            <xsd:element name = "batch_id" type = "batch_id"/>
                            <xsd:element name = "empl_id" type = "required_empl_id"/>
                            <xsd:element name = "ts_schedule_cd" type =
                            "required_ts_schedule_cd"/>
                            <xsd:element name = "year_no_cd" type = "year_no_cd"/>
                            <xsd:element name = "period_no_cd" type = "period_no_cd"/>
                            <xsd:element name = "line_no" type = "line_no"/>
                            <xsd:element name = "hrs_dt" type = "hrs_dt" minOccurs = "0"/>
                            <xsd:element name = "s_cell_status_cd" type = "s_cell_status_cd"/>
                            <xsd:element name = "rejection_text" type = "rejection_text" minOccurs
                            = "0"/>
                    </xsd:sequence>
                </xsd:complexType>
```

```
        </xsd:element>
<xsd:element name = "ts_cell">
                <xsd:complexType>
                        <xsd:sequence>
                              <xsd:element name = "s_reject_type_cd" type = "s_reject_type_cd"/>
                                 <xsd:element name = "batch_id" type = "batch_id"/>
                                 <xsd:element name = "empl_id" type = "required_empl_id"/>
                                 <xsd:element name = "ts_schedule_cd" type =
                                 "required_ts_schedule_cd"/>
                                 <xsd:element name = "year_no_cd" type = "year_no_cd"/>
                                 <xsd:element name = "period_no_cd" type = "period_no_cd"/>
                                 <xsd:element name = "line_no" type = "line_no"/>
                                 <xsd:element name = "hrs_dt" type = "hrs_dt"/>
                                 <xsd:element name = "s_cell_status_cd" type = "s_cell_status_cd"/>
                                 <xsd:element name = "rejection_text" type = "rejection_text" minOccurs
                                 = "0"/>
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>

    <xsd:simpleType name = "s_reject_type_cd">
      <xsd:restriction base = "xsd:string">
         <xsd:enumeration value = "T"/>
         <xsd:enumeration value = "L"/>
         <xsd:enumeration value = "C"/>
      </xsd:restriction>
</xsd:simpleType>
        <xsd:simpleType name = "s_cell_status_cd">
                <xsd:restriction base = "xsd:string">
                        <xsd:enumeration value = "R"/>
                        <xsd:enumeration value = "O"/>
                        <xsd:enumeration value = "G"/>
                </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "batch_id">
      <xsd:restriction base = "xsd:string">
```

```
            <xsd:maxLength value = "10"/>
            <xsd:minLength value = "1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "year_no_cd">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "4"/>
            <xsd:minLength value = "1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "period_no_cd">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "3"/>
            <xsd:minLength value = "1"/>
        </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType name = "hrs_dt">
                <xsd:restriction base = "xsd:dateTime"/>
  </xsd:simpleType>
  <xsd:simpleType name = "line_no">
                <xsd:restriction base = "xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name = "rejection_text">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "254"/>
        </xsd:restriction>
    </xsd:simpleType>

  </xsd:schema>
```

Costpoint Company

```
  <?xml version = "1.0" encoding = "UTF-8"?>
  <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
  <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "cp_company">
```

```
            <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name = "cp_company_cd" type = "required_cp_company_cd"/>
                  <xsd:element name = "cp_company_name" type = "cp_company_name"
                  minOccurs="0"/>
                  <xsd:element name = "active_fl" type = "active_fl" minOccurs="0"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "cp_company_name">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "30"/>
            </xsd:restriction>
        </xsd:simpleType>

    </xsd:schema>
```

Currency Schedules

```
    <?xml version = "1.0" encoding = "UTF-8"?>
    <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
    <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "currency_schedule">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name = "currency_schedule_cd" type =
                    "currency_schedule_cd"/>
                    <xsd:element name = "effective_dt" type = "effective_dt"/>
                    <xsd:element name = "from_currency_cd" type = "currency"/>
                    <xsd:element name = "to_currency_cd" type = "currency"/>
                    <xsd:element name = "exchange_rate" type = "exchange_rate"
                    minOccurs="0"/>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "currency_schedule">
            <xsd:restriction base = "xsd:string">
```

```
                <xsd:maxLength value = "10"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "currency">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "3"/>
                <xsd:minLength value = "1"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "exchange_rate">
            <xsd:restriction base = "xsd:decimal">
              <xsd:totalDigits value="20" />
              <xsd:fractionDigits value="10" />
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:schema>
```

Expense Report Payments

```
    <?xml version = "1.0" encoding = "UTF-8"?>
    <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
    <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "expense_report_payment">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name = "exp_rpt_id" type = "exp_rpt_id"/>
                    <xsd:element name = "payment_reference_cd" type =
                    "payment_reference_cd"/>
                    <xsd:element name = "payment_dt" type = "payment_dt"/>
                    <xsd:element name = "pay_payment_amt" type = "currency_amt"/>
                    <xsd:element name = "base_payment_amt" type = "currency_amt"/>
                    <xsd:element name = "s_payment_method_cd" type =
                    "s_payment_method_cd"/>


                </xsd:sequence>
```

```
            </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "payment_reference_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "20"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "payment_dt">
            <xsd:restriction base = "xsd:dateTime"/>
        </xsd:simpleType>
        <xsd:simpleType name = "currency_amt">
            <xsd:restriction base = "xsd:decimal">
              <xsd:totalDigits value="15" />
              <xsd:fractionDigits value="5" />
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "s_payment_method_cd">
            <xsd:restriction base = "xsd:string">
                <xsd:enumeration value = "AP"/>
                <xsd:enumeration value = "PR"/>
                <xsd:enumeration value = "CR"/>
            </xsd:restriction>
        </xsd:simpleType
    </xsd:schema>
```

Expense Rejections

```
    <?xml version = "1.0" encoding = "UTF-8"?>
    <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
    <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "exp_rejection">
            <xsd:complexType>
                <xsd:sequence>
                    <xsd:element name = "exp_rpt_id" type = "exp_rpt_id"/>
                    <xsd:element name = "expense_id" type = "expense_id"/>
```

```
                <xsd:element name = "rejection_reason" type = "rejection_reason"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:simpleType name = "expense_id">
        <xsd:restriction base = "xsd:integer"/>
    </xsd:simpleType>
    <xsd:simpleType name = "rejection_reason">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "254"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

Leave Types

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
    <xsd:include schemaLocation = "import_common.xsd"/>
    <xsd:element name = "leave_type">
        <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "leave_type_cd" type = "leave_type_cd"/>
            <xsd:element name = "leave_type_desc" type = "leave_type_desc"
            minOccurs="0"/>
            <xsd:element name = "vacation_fl" type = "vacation_fl" minOccurs="0"/>
            <xsd:element name = "holiday_fl" type = "holiday_fl" minOccurs="0"/>
            <xsd:element name = "cp_company_cd" type = "cp_company_cd"
            minOccurs="0"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
    <xsd:simpleType name = "leave_type_desc">
        <xsd:restriction base = "xsd:string">
            <xsd:maxLength value = "30"/>
        </xsd:restriction>
    </xsd:simpleType>
```

```
        <xsd:simpleType name = "vacation_fl">
                <xsd:restriction base = "xsd:boolean"/>
        </xsd:simpleType>
        <xsd:simpleType name = "holiday_fl">
                <xsd:restriction base = "xsd:boolean"/>
        </xsd:simpleType>
</xsd:schema>
```

Per Diem Schedules

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
<xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
        <xsd:element name = "per_diem_schedule">
                <xsd:complexType>
                        <xsd:sequence>
                                <xsd:element name = "source" type = "source"/>
                                <xsd:element name = "location_type" type = "location_type"/>
                                <xsd:element name = "country_name" type = "country_name"/>
                                <xsd:element name = "state_name" type = "state_name"/>
                                <xsd:element name = "city_name" type = "city_name"/>
                                <xsd:element name = "county_name" type = "county_name"/>
                                <xsd:element name = "season_start_date" type = "season"/>
                                <xsd:element name = "season_end_date" type = "season"/>
                                <xsd:element name = "effective_dt" type = "effective_dt"/>
                                <xsd:element name = "lodging_amt" type = "lodging_amount"/>
                                <xsd:element name = "meals_incidental_amt" type =
                                "meals_and_incidentals_amount" />
                                <xsd:element name = "proportional_meals_amt" type =
                                "proportional_meal_amount" />
                        </xsd:sequence>
                </xsd:complexType>
        </xsd:element>
        <xsd:simpleType name = "source">
                <xsd:restriction base = "xsd:string">
                        <xsd:maxLength value = "1"/>
                        <xsd:minLength value = "1"/>
```

```
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "location_type">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "1"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "country_name">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "45"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "state_name">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "35"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "city_name">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "35"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "county_name">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "35"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
```

```
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "season">
            <xsd:restriction base = "xsd:string">
                <xsd:maxLength value = "5"/>
                <xsd:minLength value = "1"/>
                <xsd:pattern value = "[^a-z]*"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "lodging_amount">
            <restriction base="xsd:integer">
            <xsd:maxInclusive value="999"/>
            <xsd:minInclusive value="0"/>
            </restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "meals_and_incidentals_amount">
            <restriction base="xsd:integer">
            <xsd:maxInclusive value="999"/>
            <xsd:minInclusive value="0"/>
            </restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "proportional_meal_amount">
            <restriction base="xsd:integer">
            <xsd:maxInclusive value="999"/>
            <xsd:minInclusive value="0"/>
            </restriction>
        </xsd:simpleType>

    </xsd:schema>
```

Tax Code

```
    <?xml version = "1.0" encoding = "UTF-8"?>
    <!--Generated by XML Authority. Conforms to w3c http://www.w3.org/2001/XMLSchema-->
    <xsd:schema xmlns:xsd = "http://www.w3.org/2001/XMLSchema">
        <xsd:include schemaLocation = "import_common.xsd"/>
```

```
<xsd:element name = "tax_code">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name = "tax_schedule_cd" type = "tax_schedule_cd"/>
            <xsd:element name = "tax_schedule_desc" type =
            "tax_schedule_desc"/>
            <xsd:element name = "s_type_cd" type = "s_type_cd"/>
            <xsd:element name = "s_rate_tier_cd" type = "s_rate_tier_cd"/>
            <xsd:element name = "dflt_tax_rate1_pct" type = "dflt_tax_rate1_pct"/>
            <xsd:element name = "dflt_tax_rate2_pct" type = "dflt_tax_rate2_pct"/>
            <xsd:element name = "dflt_tolerance_pct" type = "dflt_tolerance_pct"/>
            <xsd:element name = "location_fl" type = "location_fl"/>
            <xsd:element name = "dflt_tax_recovery_rate1_pct" type =
            "dflt_tax_recovery_rate1_pct"/>
            <xsd:element name = "dflt_tax_recovery_rate2_pct" type =
            "dflt_tax_recovery_rate2_pct"/>
            <xsd:element name = "tax_id_fl" type = "tax_id_fl"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:simpleType name = "tax_schedule_cd">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "10"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "tax_schedule_desc">
    <xsd:restriction base = "xsd:string">
        <xsd:maxLength value = "30"/>
    </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "s_type_cd">
    <xsd:restriction base = "xsd:string">
        <xsd:enumeration value = "S"/>
        <xsd:enumeration value = "U"/>
        <xsd:enumeration value = "V"/>
        <xsd:length value = "1"/>
    </xsd:restriction>
```

```
</xsd:simpleType>
<xsd:simpleType name = "s_rate_tier_cd">
        <xsd:restriction base = "xsd:string">
                <xsd:enumeration value = "1"/>
                <xsd:enumeration value = "2"/>
                <xsd:length value = "1"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "dflt_tax_rate1_pct">
        <xsd:restriction base = "xsd:decimal">
                <xsd:totalDigits value = "5"/>
                <xsd:fractionDigits value = "4"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "dflt_tax_rate2_pct">
        <xsd:restriction base = "xsd:decimal">
                <xsd:totalDigits value = "5"/>
                <xsd:fractionDigits value = "4"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "dflt_tax_recovery_rate1_pct">
        <xsd:restriction base = "xsd:decimal">
                <xsd:totalDigits value = "3"/>
                <xsd:fractionDigits value = "2"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "dflt_tax_recovery_rate2_pct">
        <xsd:restriction base = "xsd:decimal">
                <xsd:totalDigits value = "3"/>
                <xsd:fractionDigits value = "2"/>
        </xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name = "dflt_tolerance_pct">
        <xsd:restriction base = "xsd:decimal">
                <xsd:totalDigits value = "3"/>
                <xsd:fractionDigits value = "2"/>
```

```
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name = "location_fl">
            <xsd:restriction base = "xsd:boolean"/>
        </xsd:simpleType>
        <xsd:simpleType name = "tax_id_fl">
            <xsd:restriction base = "xsd:boolean"/>
        </xsd:simpleType>
</xsd:schema>
```

Export Staging Table Layouts

Overview

This chapter contains the staging table layouts used when exporting timesheets and expense reports.

EXPORT_LABOR

Note: This table contains data staged to exported timesheet entries for labor distribution.

Table: EXPORT_LABOR

Description: Export labor distribution entries

Type: User

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
BATCH_ID	VARCHAR (10)		No	PK1	The ID of the batch of which this record is a member. From screen input.
TIMESHEET_DT	DATE		No	PK2	The column date for this cell. From screen input.
EMPL_ID	VARCHAR (20)		No	PK3	Employee ID identifier. It defaults from the TS table.
TS_SCHEDULE_CD	VARCHAR (10)		No	PK4	Timesheet frequency identifier. It defaults from the TS table.
YEAR_NO_CD	VARCHAR (4)		No	PK5	The timesheet frequency's year code. It defaults from the TS table.
PERIOD_NO_CD	VARCHAR (3)		No	PK6	Period number for timesheet frequency. It defaults from the TS table.

Field	Domain	FK	Null	Key	Notes
LINE_NO	INTEGER		No	PK7	The number of the timesheet line to which this timesheet cell belongs. It defaults from the TS_LINE table.
HRS_DT	DATE		No	PK8	The column date for this cell. It defaults from the TS_CELL table.
REVISION_NO	INTEGER		No		The current revision of the timesheet. It defaults from the TS table.
CLASS_CD	VARCHAR (20)		No		The employee class for the employee at the time of this timesheet. It defaults from the TS_LINE table.
WORK_SCHEDULE_ CD	VARCHAR (10)		No		The work schedule for the employee at the time of this timesheet. It defaults from the TS_LINE table.
LINE_DESC	VARCHAR (120)		No		Description of the charge on the line from the TS_LINE table.
CHARGE_TREE_CD	VARCHAR (50)		Yes		If you selected a charge from a charge tree, the charge code will display in this column. It defaults from the TS_LINE table.
CHARGE_BRANCH_ CD	VARCHAR (50)		Yes		If you selected a charge from a charge tree, this column will display the charge branch code from the TS_LINE table.

Field	Domain	FK	Null	Key	Notes
CHARGE_CD	VARCHAR (100)		Yes		If you selected a charge from a charge tree, this column will display the charge code for the tree/branch from the TS_LINE table.
UDT01_ID	VARCHAR (50)		No		The UDT01 identifier for the line. It defaults from the TS_LINE table.
UDT02_ID	VARCHAR (50)		Yes		The UDT02 identifier for the line. It defaults from the TS_LINE table.
UDT03_ID	VARCHAR (20)		Yes		The UDT03 identifier for the line. It defaults from the TS_LINE table.
UDT04_ID	VARCHAR (20)		Yes		The UDT04 identifier for the line. It defaults from the TS_LINE table.
UDT05_ID	VARCHAR (20)		Yes		The UDT05 identifier for the line. It defaults from the TS_LINE table.
UDT06_ID	VARCHAR (20)		Yes		The UDT06 identifier for the line. It defaults from the TS_LINE table.
UDT07_ID	VARCHAR (20)		Yes		The UDT07 identifier for the line. It defaults from the TS_LINE table.
UDT08_ID	VARCHAR (20)		Yes		The UDT08 identifier for the line. It defaults from the TS_LINE table.

Field	Domain	FK	Null	Key	Notes
UDT09_ID	VARCHAR (20)		Yes		The UDT09 identifier for the line. It defaults from the TS_LINE table.
UDT10_ID	VARCHAR (20)		No		The UDT10 identifier for the line. It defaults from the TS_LINE table.
ETC_HRS	DECIMAL (8,2)		No		The estimate-to-complete hours for this employee. This value defaults from the TS_LINE table.
BILLABLE_FL	VARCHAR (1)		No		Logical as to whether the hours are billable. It defaults from the TS_LINE table.
HRS	DECIMAL (8,2)		No		The exported labor hours for this cell. This value defaults from the TS_CELL table and is calculated as PRORATED_HRS minus EXPORTED_HRS
CORRECT_FL	VARCHAR (1)		No		Logical as to whether this cell is a correcting entry. This value defaults from the TS_CELL table.
ENTERED_DTT	DATETIME		No		The date and time when the hours were entered. This value defaults from the TS_CELL table.
OT_FL	VARCHAR (1)		No		Logical as to whether this UDT10 ID is for overtime. This value defaults from the UDT10 table.

Field	Domain	FK	Null	Key	Notes
COST_FL	VARCHAR (1)		No		Logical as to whether this UDT10 ID is for "cost only" hours. This value defaults from the UDT10 table.
MISC_LABOR_CD	VARCHAR (30)		Yes		The Miscellaneous field contains codes used to feed the labor distribution file during export. This value defaults from the UDT10 table.
SHORT_HEADER_ TEXT	VARCHAR (254)		Yes		Notes about the timesheet. This defaults from the TS_LINE table - first 254 characters.
SHORT_LINE_TEXT	VARCHAR (254)		Yes		User-provided notes about the line. They default from the TS_LINE table - first 254 characters.
SHORT_CELL_ TEXT	VARCHAR (254)		Yes		User-provided notes about the timesheet cell. They default from the TS_CELL table - first 254 characters.
EMPL_CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company Code for employee.
UDT06_CP_COMPANY_CD	VARCHAR (10)		Yes		Costpoint Company Code for UDT06 field.
UDT07_CP_COMPANY_CD	VARCHAR (10)		Yes		Costpoint Company Code for UDT07 field.
REVERSAL_FL	VARCHAR (1)		No		Logical as to whether entry is a reversing entry.

Field	Domain	FK	Null	Key	Notes
S_CP_TS_TYPE_CD	VARCHAR (1)		No		The Costpoint Timesheet Type. The valid values are: R - Regular C - Correcting N - Reversal D – Replacement
RATE1_AMT	DECIMAL (15,5)		No		The prorated amount for rate 1 minus the exported amount for rate 1.
RATE2_AMT	DECIMAL (15,5)		No		The prorated amount for rate 2 minus the exported amount for rate 2.
UDT11_ID	VARCHAR (20)		Yes		The UDT11 identifier for the line. It defaults from the TS_LINE table.
UDT12_ID	VARCHAR (20)		Yes		The UDT12 identifier for the line. It defaults from the TS_LINE table.
UDT13_ID	VARCHAR (20)		Yes		The UDT13 identifier for the line. It defaults from the TS_LINE table.
UDT14_ID	VARCHAR (20)		Yes		The UDT14 identifier for the line. It defaults from the TS_LINE table.
UDT15_ID	VARCHAR (20)		Yes		The UDT15 identifier for the line. It defaults from the TS_LINE table.

EXPORT_PAYROLL

Note: This table contains data staged to exported timesheet entries for payroll processing.

Table: EXPORT_PAYROLL

Description: Export payroll.

Type: User

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
BATCH_ID	VARCHAR (10)		No	PK1	The ID of the batch of which this record is a member. This value defaults from screen input.
TIMESHEET_DT	DATE		No	PK2	The column date for this cell. This value defaults from screen input.
EMPL_ID	VARCHAR (20)		No	PK3	Employee ID identifier. This value defaults from the TS table.
TS_SCHEDULE_CD	VARCHAR (10)		No	PK4	Timesheet frequency identifier. This value defaults from the TS table.
YEAR_NO_CD	VARCHAR (4)		No	PK5	The timesheet frequency's year code. This value defaults from the TS table.
PERIOD_NO_CD	VARCHAR (3)		No	PK6	Period number for timesheet frequency. This value defaults from the TS table.
LINE_NO	INTEGER		No	PK7	The number of the timesheet line to which this timesheet cell belongs. This value defaults from the TS_LINE table.
HRS_DT	DATE		No	PK8	The column date for this cell. This value defaults from the TS_CELL table.
PAYROLL_ID	VARCHAR (20)		Yes		The payroll ID for the employee. This value defaults from the EMPL table.
PAYROLL_EMPL_ID	VARCHAR (20)		Yes		The employee's Employee ID for payroll. This value defaults from the EMPL table.

Field	Domain	FK	Null	Key	Notes
UDT10_ID	VARCHAR (20)		No		The UDT10 identifier for the line. This value defaults from the TS_LINE table.
HRS	DECIMAL (8,2)		No		The exported labor hours for this cell. This value defaults from the TS_CELL table.
CORRECT_FL	VARCHAR (1)		No		Logical as to whether this cell is a correcting entry. This value defaults from the TS_CELL table.
OT_FL	VARCHAR (1)		No		Logical as to whether this UDT10 ID is for overtime. This value defaults from the UPD10 table.
COST_FL	VARCHAR (1)		No		Logical as to whether this UDT10 ID is for "cost only" hours. This value defaults from the UDT10 table.
MISC_PAYROLL_CD	VARCHAR (30)		Yes		The Miscellaneous field contains codes used to feed the payroll file during export. This value defaults from the UDT10 table.
PAYROLL_EARNINGS_CD	VARCHAR (30)		Yes		Earnings code for this UDT10 ID. This value defaults from the UDT10 table.
S_PAYROLL_MAP_ CD	VARCHAR (10)		Yes		This value determines how the Earnings Code is mapped during export. The valid options are as follows: N - None R - Regular O - Overtime 3 - Earnings 3 4 - Earnings 4 5 - Earnings 5 H3 - Hours 3 H4 - Hours 4
CLASS_CD	VARCHAR (20)		No		The employee class for the employee at the time the timesheet was entered. The code is from the TS table.

Field	Domain	FK	Null	Key	Notes
RATE1_AMT	DECIMAL (15,5)		No		TS_CELL.PRORATED_RATE1_AMT minus TS_CELL.EXPORTED_RATE1_AMT.
CODE1_CD	VARCHAR (20)		Yes		First user-defined alphanumeric field from the EMPL table.
CODE2_CD	VARCHAR (20)		Yes		Second user-defined alphanumeric field from the EMPL table.
CODE3_CD	VARCHAR (20)		Yes		Third user-defined alphanumeric field from the EMPL table.

EXPORT_EXPENSE

Note: This table contains data staged for exported expense reports.

Table: EXPENSE_EXPORT

Description: Expense Report entries

Type: User

Parent Table: None

Child Tables: None

Field	Domain	FK	Null	Key	Notes
BATCH_ID	VARCHAR (10)		No	PK1	The ID of the batch of which this record is a member. The Batch ID defaults from user screen entry during the export process.
EXP_RPT_ID	VARCHAR (10)		No	PK2	The unique identifier for the expense report.
PARENT_EXPENSE_ ID	INTEGER		No	PK3	The expense ID of the expense if this expense was generated from it.
EXPENSE_ID	INTEGER		No	PK4	The unique identifier for the expense.
ALLOCATION_ID	INTEGER		No	PK5	Synthetic.

Field	Domain	FK	Null	Key	Notes
S_TYPE_CD	VARCHAR (1)		No	PK6	The type of record. The valid values are E - Expense A - Advance C - Credit
S_CHARGE_TYPE_ CD	VARCHAR (2)		No	PK7	Values are: NA - Not Applicable UC - Under Ceiling OC - Over Ceiling U - Unallowable
VENDOR_ID	VARCHAR (20)		No		The identifying vendor ID for this employee.
EMPL_ID	VARCHAR (20)		No		The employee who originated the expense report.
LAST_NAME	VARCHAR (30)		No		From the EMPL table.
FIRST_NAME	VARCHAR (30)		No		From the EMPL table.
INITIAL_NAME	VARCHAR (1)		Yes		From the EMPL table.
EXP_RPT_DESC	VARCHAR (30)		No		Short description of the expense report.
INVOICE_DT	DATE		No		System-assigned invoice date.
EXP_RPT_DT	DATE		No		User-assigned date for expense report.
PURPOSE_TEXT	VARCHAR (4000)		No		The purpose of the expenses on the report.
EXP_RPT_TYPE_CD	VARCHAR (10)		No		The type of expense report.
EXP_CLASS_CD	VARCHAR (20)		No		The expense class of the employee at the time of the expense report.

Field	Domain	FK	Null	Key	Notes
CP_COMPANY_CD	VARCHAR (10)		No		Costpoint Company Code for the expense report. (Employee)
PAY_CURRENCY_ CD	VARCHAR (3)		No		The code of the currency in which the employee will be reimbursed.
BASE_CURRENCY_CD	VARCHAR (3)		No		The base currency code of the system.
AP_1099_FL	VARCHAR (1)		No		Logical as to whether this expense should be recorded as 1099 information.
AP_1099_TYPE_CD	VARCHAR (6)		Yes		Type of 1099 expense.
EXP_CATEGORY_ CD	VARCHAR (10)		No		The category of the expense.
EXP_TYPE_CD	VARCHAR (10)		No		The type of the expense.
EXPENSE_DT	DATE		No		The date on which the expense was incurred.
PROVIDER_CD	VARCHAR (20)		Yes		The provider of the service for the expense.
EXPENSE_TEXT	VARCHAR (120)		Yes		Description of expense.
PAY_METHOD_CD	VARCHAR (10)		No		User-specified unique identifier for the pay method
TRANS_CURRENCY_CD	VARCHAR (3)		No		The currency in which the transaction was incurred.
USER_EXCHANGE_RATE	DECIMAL (20,10)		No		The exchange rate the user provided.

Field	Domain	FK	Null	Key	Notes
TRANS_PAY_ EXCHANGE_RATE	DECIMAL (20,10)		No		The exchange rate provided by the system to take the transaction currency to the pay currency.
TRANS_BASE_ EXCHANGE_RATE	DECIMAL (20,10)		No		The exchange rate provided by the system to take the transaction currency to the pay currency.
UDT01_ID	VARCHAR (50)		Yes		The UDT01 identifier for the line.
UDT02_ID	VARCHAR (50)		Yes		The UDT02 identifier for the line.
UDT03_ID	VARCHAR (20)		Yes		The UDT03 identifier for the line.
UDT04_ID	VARCHAR (20)		Yes		The UDT04 identifier for the line.
UDT05_ID	VARCHAR (20)		Yes		The UDT05 identifier for the line.
UDT06_ID	VARCHAR (20)		Yes		The UDT06 identifier for the line.
UDT07_ID	VARCHAR (20)		Yes		The UDT07 identifier for the line.
UDT08_ID	VARCHAR (20)		Yes		The UDT08 identifier for the line.
UDT09_ID	VARCHAR (20)		Yes		The UDT09 identifier for the line.
UDT10_ID	VARCHAR (20)		Yes		The UDT10 identifier for the line.
UDT06_CP _COMPANY_CD	VARCHAR (10)		Yes		Costpoint Company Code for UDT06_ID field.

Field	Domain	FK	Null	Key	Notes
UDT07_CP _COMPANY_CD	VARCHAR (10)		Yes		Costpoint Company Code for UDT07_ID field.
EXPENSE_REF_CD	VARCHAR (20)		Yes		User-defined reference code.
TRANS_AMT	DECIMAL (15,5)		No		The amount in transaction currency.
PAY_AMT	DECIMAL (15,5)		No		The amount in pay currency.
BASE_AMT	DECIMAL (15,5)		No		The amount in system base currency.
PAY_VENDOR_ID	VARCHAR (20)		Yes		The identifying vendor ID for this employee.
CODE1_CD	VARCHAR (20)		Yes		The Code1 user-defined field. The values are based on the expense type.
CODE2_CD	VARCHAR (20)		Yes		The Code2 user-defined field. The values are based on the expense type.
CODE3_CD	VARCHAR (20)		Yes		The Code3 user-defined field. The values are based on the expense type.
COMMENTS_TEXT	VARCHAR (254)		Yes		The user comments about the expense.
TAX_SCHEDULE _CD	VARCHAR (10)		Yes		The tax schedule unique identifier
TAX_LOCATION_S	VARCHAR (20)		Yes		The location used in the tax schedule.
TAX_ID_S	VARCHAR (20)		Yes		The tax ID of the provider of the service or goods.

Field	Domain	FK	Null	Key	Notes
TRANS_TAX_AMT	DECIMAL (15,5)		No		The combined total of tax amounts in the transaction currency.
PAY_TAX_AMT	DECIMAL (15,5)		No		The combined total of tax amounts in the pay currency.
BASE_TAX_AMT	DECIMAL (15,5)		No		The combined total of tax amounts in the base currency.
TAX_RECOVERY_ RATE1_PCT	DECIMAL (3,2)		No		The recovery percentage for the rate1 tax.
TAX_RECOVERY_ RATE2_PCT	DECIMAL (3,2)		No		The recovery percentage for the rate 2 tax.
TRANS_TAX_ RECOVERY_RATE1_AMT	DECIMAL (15,5)		No		The amount of rate 1 tax that is recoverable in the transaction currency.
TRANS_TAX_ RECOVERY_RATE2_AMT	DECIMAL (15,5)		No		The amount of rate 2 tax that is recoverable in the transaction currency.
PAY_TAX_ RECOVERY_RATE1_AMT	DECIMAL (15,5)		No		The amount of rate 1 tax that is recoverable in the pay currency.
PAY_TAX_ RECOVERY_RATE2_AMT	DECIMAL (15,5)		No		The amount of rate 2 tax that is recoverable in the pay currency.
BASE_TAX_ RECOVERY_RATE1_AMT	DECIMAL (15,5)		No		The amount of rate 1 tax that is recoverable in the base currency.
BASE_TAX_ RECOVERY_RATE2_AMT	DECIMAL (15,5)		No		The amount of rate 2 tax that is recoverable in the base currency.

Field	Domain	FK	Null	Key	Notes
UDT11_ID	VARCHAR (20)		Yes		The UDT11 identifier for the line.
UDT12_ID	VARCHAR (20)		Yes		The UDT12 identifier for the line.
UDT13_ID	VARCHAR (20)		Yes		The UDT13 identifier for the line.
UDT14_ID	VARCHAR (20)		Yes		The UDT14 identifier for the line.
UDT15_ID	VARCHAR (20)		Yes		The UDT15 identifier for the line.

Transformed Export ASCII Layouts

About Deltek

Better software means better projects. Deltek delivers software and information solutions that enable superior levels of project intelligence, management and collaboration. Our industry-focused expertise makes your projects successful and helps you achieve performance that maximizes productivity and revenue. www.deltek.com