Preface

This Guide introduces the design and principles behind global processing functionality. It is intended to provide guidance for Finastra consultants and banks on how banks should approach implementing global processing.

# Introduction

This chapter provides an introduction to global processing functionality and introduces some of the terminology used in this Guide.

## What is Global Processing?

Banks providing trade service are very much focused on how to maintain and grow their trade service business whilst containing costs, maximising use of their skills, and reacting to changes in the market in which they operate.

Many banks offering trade services have grown to a size where they have market share and global reach, and are looking to maximise their assets by adopting a processing model that removes the constraints otherwise imposed by such factors as:

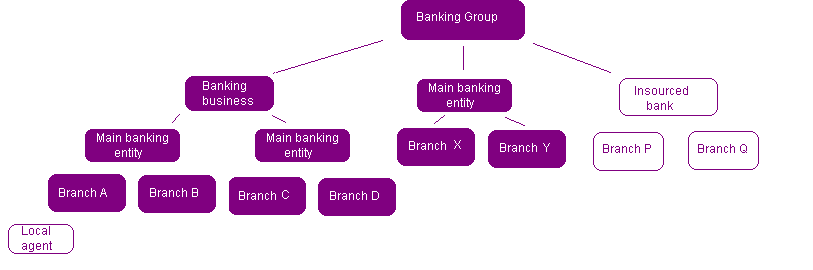
* The location of their infrastructure
* The location of their staff, and the skills mix at any particular location
* The banking entities they support (for example brands and subsidiaries)
* Time zones
* Currency
* Access to operational or management information

Global processing is a way of operating that allows internationally active banks to reduce their operational cost by automating cross-country operations, transactions and services; and helps them expand their business across different banking entities, regions and countries, thereby increasing their revenue and client base. In addition, it allows them to offer as a service the initiation of new transactions and the progression of existing ones through extended hours processing, regardless of where a transaction originated or of the banking entities involved.

## The Approach to Global Processing

The global processing solution offered by the system is tailored towards banks who are already operating in multiple regions or who are in a growing/emerging market and want to take advantage of the opportunity to expand their business. It is also easily adaptable to single-region banks and banks that need to operate some of their business in a separate unit (for example, banks that operate insourced banking services or that have multiple brands).

Trade Innovation’s approach to global processing involves dividing a bank's business into a hierarchy of banking entities that reflects its organisational and legal structure.

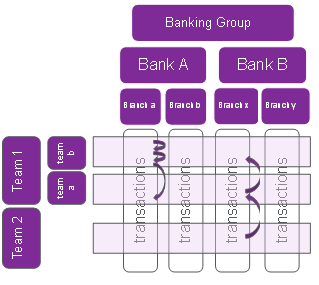


These banking entities may include:

* Banking groups (each consisting of a number of banking businesses)
* Banking businesses (typically the head offices of each banking business in a group)
* Main banking entities (a key part of the banking hierarchy that corresponds to a single general ledger that covers a number of transaction branches)
* Branches within a banking business where transactions are recorded
* Local agents associated with the bank's branches
* Insourced banking units (separate banking businesses run on behalf of other organisations)

Each of these banking entities is referred to in this Guide by the generic term branch.

Each branch is associated with one or more teams. Transactions belonging to a particular branch can be worked on only by users who belong to a team linked to that branch (or to a branch above it in the branch hierarchy). The teams associated with a branch do not have to be physically located at the branch, or in the same time zone as the branch, but can be located world-wide, allowing work on a transaction to progress even when the actual branch that owns it is closed for transaction processing.



If the bank has several discrete operations, each handled separately with their own general ledger, then it can designate certain branches within its hierarchy as main banking entities, e.g. Bank A and Bank B above. Each branch and the branches below them in the bank's branch hierarchy are then treated as a separate banking entity in the system that is associated with a general ledger for each.

Users within a team can be restricted so that they are allowed to input or work on transactions for one particular branch, or for that branch and branches beneath it in the team hierarchy, but will be permitted to enquire on transactions for the main banking entity in which they work.

The term scope is used in this Guide to refer to the transactions that a user can access:

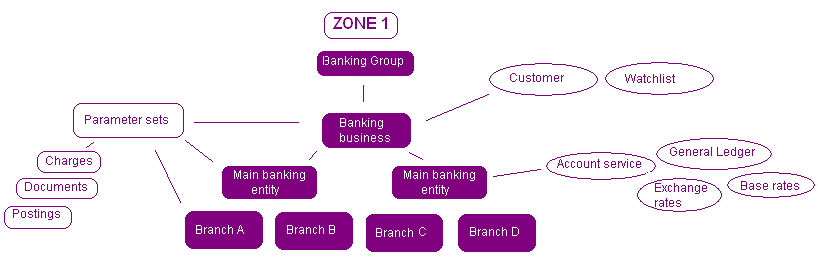
* Their transaction scope refers to the transactions they are able to input or work on
* Their enquiry scope refers to the transactions that they will be able to view, but not work on or that they can initiate but not complete

## Global Processing and the Bank's Structure

The approach to global processing involves dividing the bank up into its individual legal and/or organisational components, and then arranging them into a branch hierarchy that reflects that bank's structure.

As each branch is set up the bank defines, among other things:

* The zone to which it belongs
* The back office systems and other external systems (for example, for limit checking and watch list checking) with which the branch interfaces
* The type of branch it is (this is done using an entity type, a static data type with values set up by the banks themselves to identify all the different types of banking entities in their organisation)
* The parameters used to govern how the system behaves, how transactions are processed and what items are generated upon release of events can be set up at individual branch level using *parameter sets*. This allows different branches to process products and events in different ways, if required
* The reference structure and sequence numbers to be used by each main banking entity and associated branches



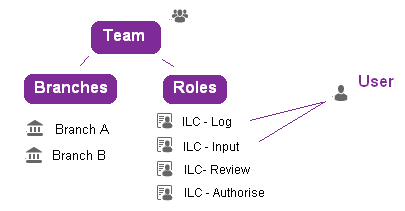
Branches are organised into zones. Each zone has its own database of transactions, static data, teams and users; and the branches within a zone share that database. This structuring allows for different local currencies by branch and different base currencies by main banking entity.

Zones can be set up geographically (corresponding, for example, to time zones or national boundaries); to reflect the bank's own business regions; or to meet a legal requirement for data to be located within a specific country (as, for example, in Singapore). The bank may also choose to operate its business out of a single zone.

Whilst most of functionality runs at the level of the zone, it also provides an overarching global layer of administration and control functionality and functionality to allow users to monitor work levels and workflow across the entire organisation. A Dashboard feature allows users - input clerks, their supervisors and managers - to monitor and manage the work allocated to them and the teams to which they belong or for which they are responsible.

## Teams, Users and Branches

For each branch the bank decides what actual processing functions can be carried out there. It then puts together one or more teams to handle the work. Each team is set up in the system as a collection of user roles, with each user role corresponding to an actual real-work role that a member of the bank's staff carries out in that team. Users are assigned user roles as they are assigned to teams. Teams are then linked to the branches whose transactions they will process.



The benefit of the approach taken by the system is that, recognising that users leave a bank but their role remains, it allows the bank to set access rights to the transaction processing functionality by user role rather than by individual user. The bank does not have to set up access rights separately for each new user it recruits; it simply assigns them the appropriate user role within the team(s) to which they are assigned.

Each transaction entered belongs to a particular branch - the Behalf Of branch - and can be worked on only by users in teams that are linked to the Behalf Of branch. This allows the bank to limit what transactions a particular user can work on by branch. (The bank can go further and restrict individual users so that they can work only on transactions belonging to the branch that they themselves belong to.)

Each team can also be assigned an accounting branch per main banking entity, which is used as the input branch for each transaction for the purpose of splitting fees and posting them to the relevant team.

Each user can be assigned to one or more teams, and can have different roles in different teams. This allows the bank flexibility in how it deploys its staff.

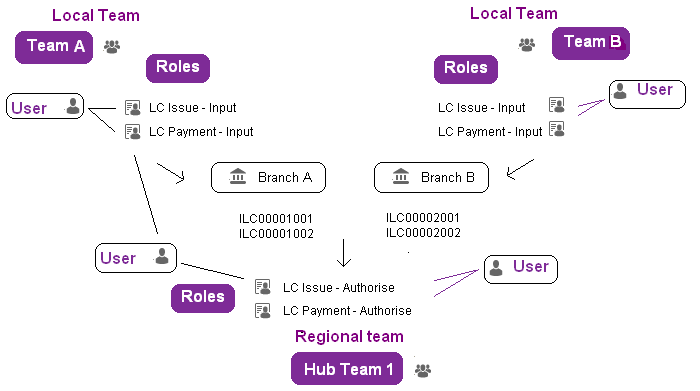
If a user is assigned to more than one team, one of those teams can be designated their default team. When they log on to the system, they are logged on using their default team. They can subsequently change their team in which they are working during transaction processing.

The bank can set up users to be team supervisors, so that they can access information about the workload of the teams to which they belong. This allows them to monitor workflow using the Dashboard and reassign work from one team or user to another.

The system allows the bank to configure events so that they are automatically routed to particular teams at particular points in their life-cycle. See the System Tailoring User Guide – Trade Innovation for details.

### Teams, Users and User Roles - Example

In this example the bank is organised so that it has a local branch acting as a service centre where customers can make an initial application for a letter of credit, and present documents received. Staff here initiates events, which are then passed to a regional centre, where they are checked and then authorised. (This model is commonly termed the 'hub and spoke' model.)



The bank sets up a team for that local branch containing the following user roles:

|  |  |
| --- | --- |
| User Role | What it Allows Users to Do |
| LC Issue Input | Bank staff assigned this user role can log and input Pre Advise, Advise, Issue and Amend events for letters of credit. |
| LC Payment Input | Bank staff assigned this user role can log and input Claim Received, Outstanding Claim, Documents Presented and Outstanding Presentation events for letters of credit. |

The team is linked to the local branch, and as actual users are allocated to the team they are assigned one or both of the user roles shown above.

The bank sets up a second team for the regional centre to which events are routed from the local branch.

|  |  |
| --- | --- |
| User Role | What it Allows Users to Do |
| LC Issue Authorise | Bank staff assigned this user role can authorise Pre Advise, Advise, Issue and Amend events for letters of credit. |
| LC Payment Authorise | Bank staff assigned this user role can authorise Claim Received, Outstanding Claim, Documents Presented and Outstanding Presentation events for letters of credit. |

1. To support this set-up the bank can configure the system so that events entered by the local branch team are automatically routed to the regional branch team once an Input step has completed.

To allow for flexibility when workloads are high or for periods when it is short-staffed the bank sets up one of its regional centre employees so that they belong to the local branch team and the regional branch team. Most of the while they work as a member of the regional centre team (their default team) authorising events received from the local branch; but they can also switch to working for the local team when required, logging and inputting events and/or reallocating work to other teams or team members.

If a bank has a number or branches that all operate in the same way (that is, they have the same staff roles in every branch) then the bank can set up a single team for all branches and have all their branch staff within the same team performing the same type of role such as logging requests to issue a letter of credit. They can then 'lock' users to their transaction branch (see page 19) so that they can work only with transactions for their home branch.

## Controlling What Transaction Processing a User Can Perform

A user's access to transaction processing functionality is controlled by the user roles they are given when they are added to teams. A user role gives a user access to each of the events they process in the course of their job, at the relevant step in each event's life-cycle.

User roles are constructed using event groups. Each user role is linked to a single event group, which brings together all the events (within or across products) that a user assigned that user role will be able to work on. As a user role is set up the bank selects the relevant event group, and then specifies the step or steps at which users assigned that role can work on the events in that group. So, for each user role:

* The event group provides the events users a can work on
* The user role settings provide the steps at which they can work on them

Whilst each user role has a single event group, event groups can be used by more than one user role.

Following on from our previous example, the bank can set up a single 'LC Issue' event group containing the following events:

* ILC Pre Advise
* ILC Issue
* ILC Amend
* ELC Pre Advise
* ELC Advise
* ELC Amend

When this event group is linked to the 'LC Issue Input' user role, the following steps are specified:

* Log
* Input

When this event group is linked to the 'LC Issue Authorise' user role, the following step is specified:

* Authorise

When setting up user roles, in addition to defining the steps at which users with that user role can work with events, the bank can also specify which master-level tasks they can perform. Master-level tasks involve functionality that is not event-specific, such as working with notes against a transaction and diary actions.

When a user logs on to the system and opens the transaction processing application, they are logged on using their default team (which they can change). The team in which they are currently working determines:

* Which transactions they can process. They can process only transactions belonging to branches linked to their current team and transactions belonging to branches below them in the branch hierarchy (see page 25). (In addition to this team-related restriction, individual users may be prevented from working with transactions for branches other than their own; this is done by assigning them a transaction branch and locking it (see page 23))

Their user roles within their current team control:

* Which products and events they can process. A user can process only the products and events that are specified in the event groups associated with the user role(s) to which they are assigned in their current team
* The step(s) at which they can work with events
* What master-level tasks they can perform

The bank can also apply additional, user-level restrictions on what transactions a user can work with (see page 23).

In addition, users may be assigned an enquiry branch. If a user has an enquiry branch set, then they will be able to view transactions where their enquiry branch matches the transaction's Behalf of branch, and transactions for branches below it in the branch hierarchy. They will also be able to work on these transactions, but will not be able to complete steps on them (they will need to pend them). If a user has no enquiry branch set they will be able to see transactions for all branches within the branch hierarchy linked to, and below, their team (or within the main banking entity, where one is set).

Appendix B provides examples of how transaction branch and enquiry branch control what transactions a user can access.

## Controlling Access to Other Trade Innovation Functionality

User roles control access to transaction processing functionality. Access to other applications and functionality is controlled by:

* A user's class - whether they are a normal user or supervisor, a security officer, a system administrator, an operator or a help desk user
* Capabilities - which determine what applications and static and tailoring functions they can access

### User Classes

Each user is set up with an indicator showing what class of user they are. Users can be defined as belonging to one of the following classes:

|  |  |
| --- | --- |
| Help Desk | This level of user exists to allow your bank to re-enable user profiles that have been disabled because the allowed period of inactivity has been exceeded or because the permitted number of failed login attempts has been exceeded. (Profiles that have been disabled because they have reached their expiry date cannot be re-enabled by help desk users.) |
| Normal users | Normal users run the applications to set up and maintain system tailoring data and static data and to process transactions. |
| Operators | This level of user exists as a security feature to allow system administrators to log on with restricted access to the system functionality in order to run overnight processing and to suspend and back up a zone. |
| Security officers | Security officers have access to the full functionality of the system and to all applications. This includes functions used to register new users and to set up and maintain security parameters. |
| Supervisors | Supervisors have the same access level as normal users, but in addition can view the workload of all users in the teams (and their child teams) they supervise. |
| System administrators | System administrators can define zones and set the global parameters governing the security. |
| User administrators | User administrators can create and maintain the following classes of users in the Global Application: Helpdesk, Normal users, Operators and other User administrators. This class of users exists so that a specific user or group of users can be set up to oversee user on-boarding. |

A user's class determines the extent to which a user can access functionality used to set up and administer the system, including defining zones and users, starting and stopping the system, and monitoring user activities. Normal users and supervisors have no access to such functionality, and are limited to working with the functionality provided by the applications.

### Capabilities

Access to the applications other than the transaction processing application is controlled by capabilities. Each capability controls:

* What application the user can open. Each application has an application-level capability that must be assigned to a user for them to be able to access it when they log in
* What they can then use that application to do - certain tasks have an associated capability, and access to some menu options and buttons are also capability controlled

See the Security Guide – Trade Innovation for information on capabilities and instructions on assigning them to users.

## Workflow and the Dashboard

A bank's users can access the transaction processing application in one of two ways:

* Via the Masters window, which can be used to list all the transactions that the user can work with within the selected zone
* Via the Dashboard window, which provides a team-oriented view of work in progress on transactions that the user can work with across all zones the team is authorised to work in

Both windows allow users to find, open and work with (or just view) transactions to which they have access.

Users can be assigned the Dashboard window, the Masters window or both via capabilities.

When a user opens the transaction processing application, the system determines which transactions that user can work with using the user roles to which they are assigned in their default team, and their transaction and enquiry branches, if these are set. (If the user subsequently changes team, the system determines which transactions they can work with using that team.)

In the Masters window the system displays the zone transactions a user can process. If the user has an enquiry branch set, then it also displays transactions the user can view, but not process. Filter fields allow the user to display a sub-set of the transactions found for them, and to determine whether the Masters window lists master records or events.

For supervisors, the Masters window lists all transactions for all the teams to which they belong. Whether they can open those transactions to continue work on them (as opposed to just viewing them) depends on the user roles assigned to them within their current team (plus the usual other restrictions).

If master records are listed, they are ordered by master record reference. If events are displayed, they are listed by elapsed time since they were initiated.

In the Dashboard, the system determines which transactions a user can access in the same way, but the display is different, and there is additional functionality for users who are supervisors. Chapter 4 provides a full description of the Dashboard window and what it displays.

The list of transactions displayed uses colours to indicate where a transaction is in relation to its completion time. Each transaction is displayed as having one of three statuses - low, medium or high - depending on how close to their target completion time they are. The bank defines when an event’s status will change to medium and then to high, relative to its target completion time. The target times are calculated using service level agreement information entered by the bank using the system tailoring application, with off-set times calculated from the time an event was initiated.

Each of the three statuses has its own bank-definable colour code (the default is low = green, medium = amber, high = red). This visual colour coding allows users to see at a glance which transactions are the most urgent.

The visual displays in the Dashboard allow users to monitor their workload. They show the following information:

* Overall totals for the team
* All transaction active steps by red/amber/green status
* All transaction active steps according to their current phase
* Team and user total figures, including:
* Figures that allow users to see the number of transaction active steps assigned to a team. The figures can by shown by different categories - by branch, product, customer, key customer, step and team and user
* A list of individual transactions for the team and user showing the estimated time to completion (the number of transactions can be configured in the Dashboard profile)

These graphic displays are capability-controlled and so can be set at individual user level.

For supervisors, the Dashboard shows figures for each team in which they are flagged as a supervisor and allows them to identify under- and over-capacity. Supervisors are able to move work from one team or user to another.

Target times for transaction completion are defined against an Input step for events, with default target times for those events for which no specific target times have been defined. The Dashboard shows the status of each transaction using the colour coding set up by the bank.

## Multiple End of Days

Each zone can have multiple end of day processing cycles. Branches are assigned to a particular cycle through their main banking entities and any transactions for that branch will have end of day actions such as interest accruals and diary actions applied during that cycle’s end of day process.

During this time other branches not using that cycle operate as usual during their day time processing. A processing cycle can run its end of day at any time. Other processing cycles may be in business hours or end of day.

## Extended Hours

In order to provide as long a processing window as possible the system enables extended hours processing of transactions whilst a branch is performing an end of day cycle.

Transactions can be started or work can continue on already entered ones. The processing is carried out as if the user was now processing on the next processing date. In other words, processing for the next date may commence, prior to completion of the current end of day cycle.

1. For those transactions that are subject to Foreign Exchange Rate Fixing, processing may be continued following completion of Rate Reconciliation.

See the *Business Operations Guide* – Trade Innovation for details of rate fixing functionality. Transactions cannot be fully released until the end of day processing cycle is complete. There may also be some services for items such as account balance checks that may not be available if the corresponding back office is off line.

# Implementing Global Processing

This chapter discusses how a bank should go about implementing global processing functionality. It covers the design considerations involved and the decisions the bank will need to make; and outlines the steps involved in configuring the system to perform global processing.

It is relevant to banks implementing the system for the first time and to banks upgrading to global processing release from earlier releases.

## Overview

The global processing functionality requires careful thought and planning on the part of the bank before it is implemented, since it involves setting up layers of interlinked parameters governing how the system will behave and who can access it to do what. These parameters need to be designed carefully in relation to each other to ensure that the system works the way the bank intended. Also, if they are implemented in a planned way, the bank can save a lot of time by re-using blocks of parameters, rather than having to set them up repeatedly.

The process of designing a bank's implementation of the global processing, known as the target operating model (TOM) has five aspects:

* Designing the bank's organisational hierarchy
* Setting up access to the system for bank staff
* Configuring the system to allow for differences in how a bank's constituent banking entities operate commercially and legally
* Establishing the preferred workflow model
* Identifying all the target interfaces and the points within the organisational hierarchy where these services are required. This in turn may establish what end of day cycles are required

Setting up the system to perform global processing involves setting up values for a large number of parameters. Many of the parameters used are linked to other parameters. Branches, for example, are linked to several types of parameter sets, and to parameters defining, among other things:

* External systems and services they use
* Daily processing cycles
* Teams

The process of linking parameters is facilitated by browsers and involves filtering for the actual value needed and then selecting it from a returned list. The bank can save much time, and avoid potential errors, by using meaningful names for parameter values or a naming convention that makes it easy to identify the correct parameter value to use.

## Upgrade Considerations

This chapter explains how to implement the global processing functionality from a clean start. The same considerations apply when the bank is upgrading from a previous version of the system.

However, a bank that is upgrading will already have much of its architecture in place. It will be necessary to analyse the current operating model in order to see where changes are required to operate globally, especially around the parameters used and the security model. Additional layers within the branch setup will also be required to reflect the organisational hierarchy.

## The Stages in Implementing Global Processing

This section provides a brief overview of the process of implementing the global processing functionality and outlines the order in which to do things.

The first task is to understand the bank's structure. This involves identifying:

* All the branches owned by the bank (with branch being used in the generic sense established earlier in this guide on page 1)
* The geographical locations in which they operate
* Any legal restrictions affecting where their databases need to be located. This allows you to define the bank's branch hierarchy and zone structure
* The interfaces required to support, for example, general ledger processing, customer limit checking and access to customer information files
* Which branches constitute main banking entities (see page 14)

Once a bank's hierarchy of branches has been established:

* Define the reference structure and sequence numbers to be used by each main banking entity and its associated branches
* Decide what parameters and parameter sets to set for each branch. Specifically, define at root level base parameter sets that can be refined or expanded downwards to meet the requirements of specific branches lower down the hierarchy
* Establish what work is carried out at each branch. This allows you to:
* Identify the user roles the bank needs
* Identify the required teams (and the user roles each must contain)
* Assign branches to the teams responsible for them
* Assign the actual bank users to each team, with the appropriate user role(s)
* Assign transaction branches and enquiry branches to individual users, as required
* Assign accounting branches to teams, as required
* Identify supervisors

1. Finastra recommend that you keep the overall design as simple as possible, using as few teams and user roles as possible. This will help improve visibility as you set up the system to use global processing, and will make the task of maintaining the system easier.

## Designing the Bank's Organisational Hierarchy

The bank first needs to identify discrete banking entities within its organisation that can be considered as branches for the purposes of using the system for global processing. These branches need to be organised into a hierarchy, taking in to account their legal, geographical and organisations relationships and the way in which the bank conducts trade finance (for example, the levels of responsibility their branches have with regard to processing transactions).

A branch can be, for example:

* A main bank
* A subsidiary bank
* An insourced (white label) bank
* A head office
* A regional centre
* A local branch
* A local agency

All trade finance transactions belong (in a legal sense) to a specific business entity within its legal structure, and the way in which transactions are processed - in particular their accounting treatment - reflects this ownership. The transaction to which a branch belongs in this legal sense is termed the Behalf Of branch in the system, and is typically determined by the primary customer (the bank's corporate customer).

In addition, any banking entity that requires parameter set(s) tailored to its own requirements should be set up as a branch. These might include, for example, subsidiary banks or an insourced bank.

### The Importance of Main Banking Entities (MBE)

When defining branches within the system there are various flags that can be set against a branch to denote different processing behaviour.

The most important of these is the field for the *main banking entity* (MBE).

The bank should identify one or more main banking entities, below which all the child branches use the same general ledger system.

In essence the MBE is used to create a ‘mini zone’ within the main zone.

Any MBE can have a base currency defined; this would mirror the base currency of the associated back office system that the branches below it would be using. It can also have its own set of major processing options and trade finance options. These are used to define what information goes on postings, whether to send out FX deals or just FX postings, whether to use a facilities service and so on.

* A main banking entity is the point at which the bank defines the reference structure and sequence numbers to be used when entering transactions. Sequence numbers can be defined at the main banking entity level only, or optionally separate ranges can be associated with individual transaction branches.
* When defining the services to be used some of these also can only exist at the level of an MBE in the branch hierarchy. See section Services for details.

Other services can exist at a level above the MBE.

### Branch Hierarchies

The bank can set up a single hierarchy of branches, or separate ones to represent different legal entities within its organisation. It can have independent branches that are not mapped to any parent branch, to allow for in-sourced banks whose business is self-contained and carried out all within a single banking entity (an autonomous banking entity).

It can also have non-transaction branches used, for example, to cater for parts of its organisation where no transaction processing is carried out or to group together a number of related branches that might share a set of parameters. See Appendix C for an example of a non-transaction branch.

Branches can be flagged, at any level in the hierarchy, as main banking entities; that branch and the branches below it in the bank's branch hierarchy are then treated as a separate business entity in the system. Users can then be assigned a main banking entity as their enquiry branch; this will allow them to access at view-only level all transactions for that branch and for all branches below it in the branch hierarchy.

A starting point for a bank in defining its branch hierarchy would be to identify all those banking entities that can function as a Behalf of branch (that is, branches that have customer accounts). This is typically the structure in place at a bank at account level. Further branches can then be identified above and below the Behalf Of branches in the bank's hierarchy. The branches below a Behalf Of branch may consist of, for example, local offices used as service centres for scanning in documents or performing basic logging services for requests which are then passed to the Behalf Of branch. The branches above may be ‘containers’ for allowing a group of branches to share a common back office (an MBE) or a common set of parameters (a *non transaction branch*) or services.

The hierarchy a bank sets up should reflect its business model.

1. Take care not to confuse branches and teams (for example, a service centre might actually be a team, rather than a branch).

### Before Beginning to Set Up Branches

Before the bank begins to set up branches the system it needs to identify:

* Which external systems each branch will use. External system include general ledger systems, watch list checking systems, limit checking facilities, customer information systems, and corporate access systems such as Trade Portal
* The overnight processing sequence of actions each branch will follow

It then needs to use the system to set up static data values for each of the external systems and overnight processing cycles required. These values are then available when linking branches to them.

### Parameter Sets

The system is a parameter-driven system, with banks able to access a wide range of parameters to govern system behaviour and how transactions are processed and to set up at event level the outputs (for example, customer documents and postings) to be generated when an event is released.

To support global processing banks are able to set up parameter sets for:

* System options, including product related options
* Product / event level documents (customer documents)
* Step level documents (advanced workflow)
* Postings
* Charges
* Tracers
* Clauses
* Error messages and warning messages
* Interest types
* Product types
* SLA details
* Workflow orchestrations
* Payment actions

1. See the Workflow Implementation Guide for details on maintaining Step level documents and the advanced capabilities of Workflow orchestration parameter sets.

For each of the parameter types listed above the bank can set up a single parameter set or a number of parameter sets.

Parameter sets can have a flat structure (that is, a number of parameter sets unrelated to each other), be organised into hierarchies, or have a mixture of flat structure and hierarchy.

Each parameter set is given a parameter set ID, which is used to fix its position in the hierarchy of parameter sets and to link it to the branch or branches that will use it.

For banks where branches have little commonality across a particular type of parameter, a flat structure allows them to set up separate parameter sets for each branch. For banks whose branches have complete commonality across a particular parameter set, they can set up a single parameter set for all branches.

For banks that have many branches across different time zones and/or countries, a hierarchical structure allows them to set up a standard parameter set at the top level of a hierarchy that can then be refined or expanded downwards to meet the requirements of specific branches lower down the hierarchy. This standard parameter set might consist of, for example, 100 standard clauses. A child parameter set created at the next level down in the hierarchy inherits the entire parameter set of 100 clauses; but the bank may decide, for example, to modify the wording of some of the clauses and to delete others that are not relevant at that level. These modifications to the inherited parameter set create the new child parameter set.

Each child parameter set in a hierarchy inherits the settings of its parent, which can be further modified.

If, at a future date, a change is needed to a particular item (such as a clause definition or a posting definition) within a parameter set, it need only be made in one place, and is then applied automatically at all levels below in the hierarchy where that item has not already been modified or deleted.

Once set up, parameter sets are linked to the branches that will use them via parameter set IDs. If a branch does not have a specific parameter set linked to it, it uses instead the parameter set linked to its parent branch (or the next branch up the hierarchy that has a parameter set linked to it).

1. Whilst parameter set hierarchies may mirror a bank's branch hierarchies, no exact correlation is required.

#### Before You Begin Setting up Parameter Sets

When using hierarchies it is expected that banks will define different parameter sets for each time zone or country, and will start with a standard parameter set for each data type for each time zone/country.

Before it begins to define parameter sets the bank should, therefore, identify what values are common across all (or most) branches and define top-tier parameter sets accordingly. It should work out which variations are required, and for which branch or branches each variant is needed.

To facilitate the linking of branches to the parameter sets they will use, banks should devise a naming convention for parameter set IDs that will allow them to quickly and accurately identify which ones are to be applied to which branches. Finastra suggests the following convention:

branch\_type

where:

branch is the unique ID for the branch and its children that will use the parameter set and

type is the type of parameter set

For example:

London\_Charges

would identify the charges parameter set for the London branch.

(Each parameter set ID can have up to thirty characters in its unique ID field, and the underscore character can be placed at any point.)

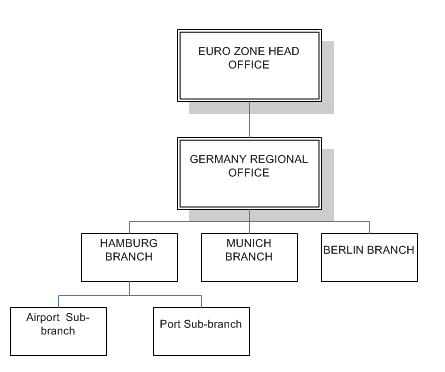
The parameter set IDs delivered with the system are given the name:

DEFAULT-aaaaaaa

where aaaaaaa is the type (for example documents or charges).

#### Example of Simple Branch Hierarchy/Parameter Set Relationship

This section provides a simple example of a branch hierarchy/parameter set hierarchy relationship. The example uses postings sets:



The posting sets defined at each level are as follows:

|  |  |  |
| --- | --- | --- |
|  |  | The EURO ZONE BRANCH has foreign exchange deal postings set up for it. |
|  |  | The GERMANY REGIONAL OFFICE inherits FX postings from the EURO ZONE BRANCH.  Customer liability postings are added. |
|  |  | The HAMBURG BRANCH inherits a parameter set from the GERMANY REGIONAL OFFICE. The inherited postings are amended to create two new postings. |
|  |  | The sub-branches have no parameter set defined for them, so they inherit the postings parameter set from the HAMBURG BRANCH. Only the new postings (and not the amended ones) are inherited. |

#### Product References and Sequence Numbers

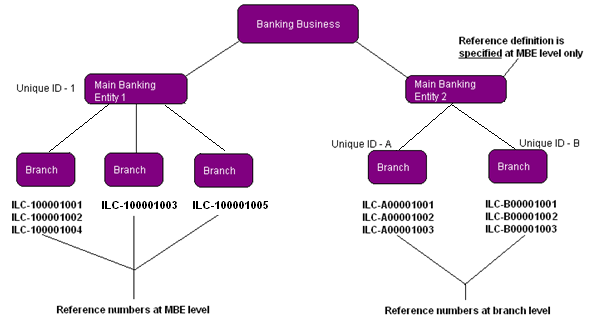
The system is delivered with initial settings for prefixes and reference numbers for products and their associated events to allow the bank to define separate product reference formats for the different main banking entities in their banking. Reference number sequence numbers can then be defined at either main banking entity to be used across a set of transaction branches and optionally at transaction branch level.

Finastra recommend that reference definitions are set up before the bank starts to use the system; and that the sequence number used for master records are not changed, once it has begun using the system to process transactions.

The key points relating to the construction of reference numbers are as follows:

* Reference definition is limited to 16 characters; this is essentially determined by SWIFT tag20 which is used to identify a transaction
* References may or may not include a product prefix (e.g. ‘ILC’ – Import Letter of Credit)
* References must uniquely identify a transaction within a Main banking entity or branch. As a result, in Global processing it is necessary to create a reference definition which includes a “Unique identifier” for each specific level of a bank’s business that has its own reference number range.
* The scope of reference numbers available to the bank is significantly increased by allowing them to be defined at branch level.

For example:



Main Banking Entity 1

The reference definition for the main banking entity is in the format ILCUnnnnnnnn where:

* ILC –Product prefix
* U – is a unique identifier
* nnnnnnnn – is the reference number range

Reference number ranges for Main Banking Entity 1:

* are defined at main banking entity level only
* The Unique Identifier is ‘1’

Branches below that main banking entity issuing ILCs include unique identifier ‘1’ in their reference numbers. As each L/C is issued by a branch, the next reference number is allocated. These branches are assigned the next available reference number from the range defined at the main banking entity level.

Main Banking Entity 2

The reference definition for the main banking entity is in the format ILCUnnnnnnnn where:

* ILC –Product prefix
* U – is a unique identifier
* nnnnnnnn – is the reference number range

Reference number ranges for Main Banking Entity 2:

* are defined at branch level. Each branch has its own distinct reference sequence which includes the unique identifier
* Branch 1 – has Unique Identifier is ‘A’
* Branch 2 – has Unique Identifier is ‘B’

## Defining the Bank's Workflow and Security

Once the bank has identified the branches within its organisation it can start to look at the work carried out there in order to design user roles and teams.

### Users and User Roles

User roles correspond to the actual work individual bank staff carry out during transaction processing. They are used in conjunction with event groups to define what actions users can perform.

To determine which user roles are required at a particular branch the bank needs to ascertain for each of its employees:

* What products they work with
* Which events within each product they are allowed to process
* The step(s) at which they are allowed to process each event
* What master-level tasks they perform

The bank should then be able to produce a matrix showing which users work with which events.

| Product | Event | User A | User B | User C | User D |
| --- | --- | --- | --- | --- | --- |
| ILC | Pre Advise |  |  |  |  |
|  | Issue |  |  |  |  |
|  | Amend |  |  |  |  |
|  | Claim Received |  |  |  |  |
|  | Outstanding Claim |  |  |  |  |
| ELC | Pre Advise |  |  |  |  |
|  | Advise |  |  |  |  |
|  | Amend |  |  |  |  |
|  | Documents Presented |  |  |  |  |
|  | Outstanding Presentation |  |  |  |  |

This allows the bank to determine which event groups it needs to set up. The table above identifies the need for two event groups. The first event group (let's call it LC Issuance) contains the following events:

* ILC Pre Advise
* ILC Issue
* ILC Amend
* ELC Pre Advise
* ELC Issue
* ELC Amend

The second event group (let's call it LC Payment) contains the following events:

* ILC Claim Received
* ILC Outstanding Claim
* ELC Documents Presented
* ELC Outstanding Presentation

Having identified and created the event groups it needs the bank can then create the actual user roles. This involves taking into account the steps at which each individual user can work with the events in each event group.

A separate user role can then be set up for every permutation of steps that can be carried out against the events in a particular event group.

| Product/event | Step | User A | User B | User C | User D |
| --- | --- | --- | --- | --- | --- |
| ILC |  |  |  |  |  |
| Pre Advise | Log step |  |  |  |  |
| Pre Advise | Input step |  |  |  |  |
| Pre Advise | Review step |  |  |  |  |
| Pre Advise | Authorise step |  |  |  |  |
| Issue | Log step |  |  |  |  |
| Issue | Input step |  |  |  |  |
| Issue | Review step |  |  |  |  |
| Issue | Authorise step |  |  |  |  |
| Amend | Log step |  |  |  |  |
| Amend | Input step |  |  |  |  |
| Amend | Review step |  |  |  |  |
| Amend | Authorise step |  |  |  |  |
| Claim Received | Log step |  |  |  |  |
| Claim Received | Input step |  |  |  |  |
| Claim Received | Review step |  |  |  |  |
| Claim Received | Authorise step |  |  |  |  |
| Outstanding Claim | Log step |  |  |  |  |
| Outstanding Claim | Input step |  |  |  |  |
| Outstanding Claim | Review step |  |  |  |  |
| Outstanding Claim | Authorise step |  |  |  |  |
| ELC |  |  |  |  |  |
| Pre Advise | Log step |  |  |  |  |
| Pre Advise | Input step |  |  |  |  |
| Pre Advise | Review step |  |  |  |  |
| Pre Advise | Authorise step |  |  |  |  |
| Advise | Log step |  |  |  |  |
| Advise | Input step |  |  |  |  |
| Advise | Review step |  |  |  |  |
| Advise | Authorise step |  |  |  |  |
| Amend | Log step |  |  |  |  |
| Amend | Input step |  |  |  |  |
| Amend | Review step |  |  |  |  |
| Amend | Authorise step |  |  |  |  |
| Documents Presented | Log step |  |  |  |  |
| Documents Presented | Input step |  |  |  |  |
| Documents Presented | Review step |  |  |  |  |
| Documents Presented | Authorise step |  |  |  |  |
| Outstanding Presentation | Log step |  |  |  |  |
| Outstanding | Input step |  |  |  |  |
| Outstanding | Review step |  |  |  |  |
| Outstanding | Authorise step |  |  |  |  |

In the table above, there are two permutations of steps against each of the two event groups, requiring the creation of four separate user roles:

|  |  |  |  |
| --- | --- | --- | --- |
| User Role | Event Grouping | Steps | What the User Can Do |
| LC Issue Input | LC Issuance | Log step  Input step  Final print | Log and input Pre Advise, Issue and Amend events for import letters of credit.  Log and input Pre Advise, Advise and Amend events for export letters of credit. |
| LC Issue Authorise | LC Issuance | Review step  Authorise step | Review and authorise Pre Advise, Issue and Amend events for import letters of credit.  Review and authorise Pre Advise, Advise and Amend events for export letters of credit. |
| LC Payment Input | LC Payment | Log step  Input step | Log and input Claim Received and Outstanding Claim events for import letters of credit.  Log and input Documents Presented and Outstanding Presentation events for export letters of credit. |
| LC Payment Authorise | LC Payment | Review step  Authorise step | Review and authorise Claim Received and Outstanding Claim events for import letters of credit.  Review and authorise Documents Presented and Outstanding Presentation events for export letters of credit. |

Once user roles have been set up they are used to build teams of users for each branch. Each team is assigned one or more user roles. Actual bank staff are then assigned to each team, and are assigned one or more user roles within that team.

|  |  |  |  |
| --- | --- | --- | --- |
| Branch |  | User | User Role |
|  |  | Supervisor | LC Issuance Authorise  LC Payment Authorise |
| HAMBURG BRANCH | Input Clerk (LC Issuance) | LC Issuance Input |
|  | Input Clerk (LC Payment) | LC Payment Input |

1. User roles are also used to determine which master-level tasks a user can perform.

By mapping to HAMBURG BRANCH each user can do work for the two sub-branches AIRPORT SUB-BRANCH and PORT SUB-BRANCH (unless further restricted at individual user level).

#### User-level Settings

Access to some aspects of the transaction processing functionality is controlled at user level.

Banks can use the security application to:

* Restrict users' access to individual events at the Input, Review or Authorise step based on the event value, using staff bands and associated limit amounts. See the Security Guide – Trade Innovation.

Users can be assigned to:

* A transaction branch. This is used as the user's default Input branch and determines the branches for which that user can enter and complete work (their transaction scope).
* An enquiry branch. This is used to allow the user to view transactions for branches other than their transaction branch for enquiry purposes; and to initiate, but not complete, events for transactions belonging to those branches (their enquiry scope).

Your bank can set up different transaction and enquiry branches for each user in each of the teams to which they are assigned. This enables them to perform different roles in different teams.

The user's transaction branch and the enquiry branch are matched to a transaction's Input branch and Behalf Of branch in the following way:

* If a user has a transaction branch set but not locked then this branch is used as the default Input branch for a transaction and the Behalf Of branch can be set to any branch mapped to the user's current team. The user can also change the Input branch to be any branch mapped to the team, as required.
* If a user has a transaction branch set and locked, then this branch is used as the default Input branch for a transaction; the Behalf Of branch can be selected only from those branches below the user's transaction branch in the branch hierarchy. The user can change the Input branch to be any other branch below their transaction branch within the branch hierarchy, if required. Events can only be completed by this user where the Input branch and the Behalf Of branch are both within the user's transaction scope. Depending on the user's enquiry scope they may be able to create events for branches outside their own transaction scope but will not be able to complete them.
* If a user has a transaction branch set and locked and that branch is at the lowest level of the branch hierarchy (that is, there are no branches below it) then the Input branch and the Behalf Of branch are both set to the user's transaction branch. Depending on the user's enquiry scope they may be able to create events for branches outside their own transaction scope but will not be able to complete them.
* If a user has an enquiry branch set, then they will be able to view transactions belonging to the enquiry branch and to all branches below it in the branch hierarchy. They will also be able to work on transactions for those branches, but will need to pend events rather than save them. If a user has no enquiry branch set they will be able to see transactions for all branches within the branch hierarchy (or within the main banking entity, where one is set).

The enquiry branch is provided to allow staff in one location to provide assistance to a customer of another branch by, for example, enquiring on the status of a transaction or initiating a new one (which would then be pended and subsequently assigned by a supervisor to the correct team for completion).

#### Supervisors

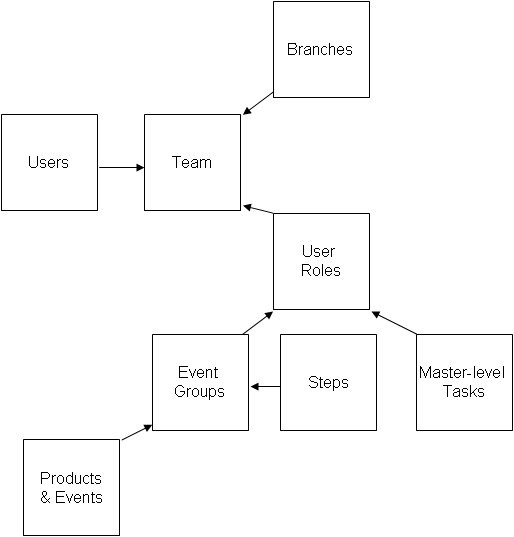
Users who are flagged as supervisors have their access to the transaction processing controlled in the same way as for normal users, via teams, user roles and their transaction branch and enquiry branch. In addition they can view the workload of all users in the teams to which they belong and their child teams.

If they have the capability ReassignTxns assigned to them, then they can also reassign transactions from one user or team to another.

When assigning supervisors to a team you do not need to assign them a user role. This allows you to set up users whose sole role is that of supervisor; they will be able to see the workloads of their teams, but not work on transactions.

### Teams

Teams provide the mechanism in the global processing functionality by which bank staff are linked to the branches on whose transactions they will work.



They are also an important element of the security system, and can be used to control workflow.

As part of the bank's implementation of the global processing functionality it needs to identify what work is carried out at a particular branch and then design the corresponding teams containing the required user roles.

The bank can design its teams in any way that it requires. For example, it might decide to set up:

* A single team for each branch, containing all the roles needed for users to carry out the work at that branch
* Several teams, each with users with the same skills set, which is then linked to several branches
* Or combinations of the above

If the bank wishes to allow extended hours processing on transactions belonging to a branch it can do this in one of two ways:

* It can set up a team for that branch that includes bank staff from each of the separate time zones necessary to ensure extended coverage
* It can assign the branch to different teams in different time zones

The bank needs to assign at least one team to each branch.

Once teams are set up the bank:

* Assigns the required user roles to that team
* Assigns actual bank staff to teams. If a user is assigned to more than one team, one of those teams can be designated their default team
* Assigns each bank staff member at least one user role within the team (except for supervisors, who do not need a user role unless they are also doing transaction processing)

The team hierarchy a bank sets up should reflect its business model. In particular:

* Supervisors need to be able to see the complete workload for the teams they are responsible for
* Managers need to be able to see the work for the entire business they are responsible for

#### Teams and Security

Team structures may be flat or hierarchical, or a combination of both. They are used in conjunction with branch hierarchies to control what transactions a user can access.

Each team is linked to one or more branches in the branch hierarchy.

Access at transaction level is controlled by branch/team linkages in the following way:

* If a user is logged on as a member of a team that has no other teams below it in the bank's team hierarchy, they will be able to process only transactions belonging to the branch(es) linked to that team
* If a user logs on using a team that has other teams below it in the bank's team hierarchy they will be able to process transactions belonging to the branch(es) linked to that team. In addition they will be able process transactions belonging to branches linked to child teams below the one they are currently using. Further:
* If they are a member of any of these child teams, their user role within each child team determines how they can process the relevant transactions
* If they are not members of a child team, they will be able to open the relevant transactions in view-only mode

Users may also be restricted to processing only transactions belonging to their own branch; or may be prevented from processing events beyond a certain amount. These restrictions apply in addition to the access control provided by branch/team assignments described above. Additional user-level security can be implemented by assigning users a transaction branch and an enquiry branch (see page 19).

Appendix B contains examples of the security models possible.

1. The bank needs to take into account the interrelationships between branch hierarchies and team hierarchies in their implementation design.

#### Teams and Workflow

The system provides a way of routing transactions to the teams that will work on them at each stage in their life-cycle.

A transaction received via gateway, such as an LC application can have a team defined on it allowing the transaction to be routed to that team for the next relevant step in the processing of that transaction.

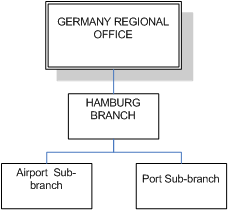
Event/team mappings allow the bank to route particular types of transactions to specific teams based on the details of the transaction. For each product/event combination the bank can determine which team or user an event should be routed to at each point in its life-cycle. See the System Tailoring User Guide – Trade Innovation for instructions.

Where there are no Event/Team mappings, the system will determine the next suitable team to be used for the transaction on a ‘round robin’ basis using the roles and branches associated with different teams to determine whether a team can process the transaction. The system assigns transactions to teams identified as auto-allocate teams in preference to standard teams.

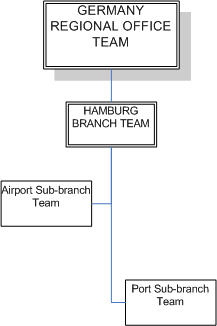
#### Example of a Simple Branch Hierarchy/Team Hierarchy Relationship

This section provides a simple example of a branch hierarchy/team hierarchy relationship.

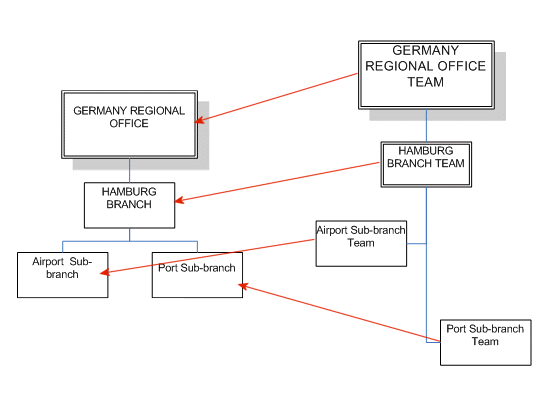
The branch hierarchy is designed to reflect a small, single-zone bank with a simple branch hierarchy. Referring back to the earlier example:



The complexity of work done at each branch diminishes down the hierarchy. Such a model supports a one-to-one relationship of branch to team.



The mapping of teams to branches is simple.



Members of the GERMANY REGIONAL OFFICE team are able to view and process transactions\* belonging to the PORT and AIRPORT sub-branches. These are the only transactional branches shown in this hierarchy. HAMBURG BRANCH is purely a ‘container’ for that part of the business. They can also view and process for other city based teams (not shown here) such as MUNICH TEAM (covering the MUNICH BRANCH business).

Members of the HAMBURG BRANCH TEAM are also able to view and process transactions\* belonging to the PORT and AIRPORT sub-branches.

Members of the PORT SUB-BRANCH TEAM are able to process transactions\* belonging to the PORT sub-branch only.

Members of the AIRPORT SUB-BRANCH TEAM are able to process transactions\* belonging to the AIRPORT sub-branch only.

Teams lower than the GERMANY REGIONAL OFFICE team have no access to transactions\* for branches in other parts of the branch hierarchy, for example the MUNICH branches.

1. \*user roles permitting. Also, an individual user's transaction branch (and whether it is locked) and their enquiry branch also affect what they are able to see and which transactions they can process.

(An alternative to the above model would have all users at the HAMBURG BRANCH and at the AIRPORT and PORT sub-branches belonging to the HAMBURG BRANCH TEAM, with no separate teams for the AIRPORT and PORT branches. Users at the AIRPORT and PORT branches could then be assigned their own branch as their transaction branch, and the transaction branch then locked, so that they would not be able to work with transactions for other branches.)

Appendix B provides further examples.

### Zones, Branches and Teams

Zones are the top-most level at which the system is administered, and each zone brings together branches related for geographical, legal reasons or others.

Branches may be designated as belonging to a particular zone as a result of regulatory constraints requiring data to be held in separate locations or within specific national boundaries. Other reasons for having separate zones include a significant partitioning of business style, such as Islamic versus conventional banking, or in-sourced versus core business, or to ensure there are sufficiently wide maintenance windows to allow upgrades zone by zone.

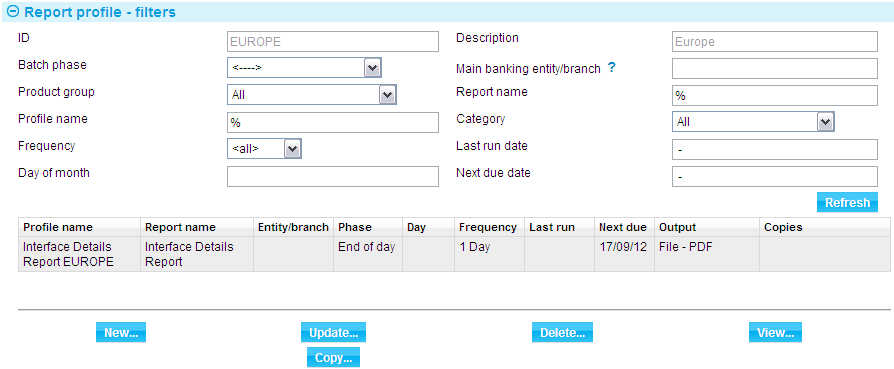
Zones are set up as part of the global processing system. An overarching cross zone dashboard allows the bank to monitor work at the global level, providing information for in progress transactions for all zones.

Each zone has its own database, shared by all the branches belonging to that zone (although the database can be logically partitioned using the security settings assigned to team structures).

1. This means that the users, user roles, event groups, teams and other parameters that support global processing described in this Guide need to be set up separately for each zone if they are common across zones.

## Crystal Reporting within Global Processing

Crystal reports can be incorporated into Processing Cycle end of day phases using Report profiles. Each report can be configured to run within the cycle or cycles required; in this example - Europe.



Pre delivered and user defined Crystal reports can be included in a processing cycle phase. The following capabilities are provided:

* Reports which select by branch, and no branch is selected, will spool once for all Main Banking Entities in the Processing cycle. All branches within each entity are included.
* Reports which select by branch, and a Main Banking Entity is selected, will spool once for that Main Banking Entity in the Processing cycle. All branches within the entity are included.
* Reports which select by branch, and a branch is selected, will spool for the selected branch. Lower branches in the hierarchy are not included.
* Reports which select by customer are defined by the source banking business, to ensure a unique reference is identified.
* Where reports use base currency, this is the Main Banking Entity base currency.
* Where reports use local currency, this is the Branch local currency.
* Data configurable by parameter set is available for reporting.
* Where account details are shown, the back office account number is provided.

Pre delivered and user defined crystal reports can be run ad-hoc during business hours. The following capabilities are provided:

* Report content is restricted to the requesting user’s teams/roles enquiry scope.
* Where a report selects by branch, only branches within the user’s enquiry scope are included.
* Where product, event and branch information are included within a report, only the products, events and branches available to the user requesting the report are included.
* Where a report is available by Main Banking Entity, and one is not selected, all entities in the zone are included.

## Defining the Bank’s Interfacing

Global processing enables the bank to support multiple interfaces out of one or more zones. The system provides the tools to manage the links to services at various levels of the branch hierarchy.

### Services

External systems can support one or more services required by the system. For example it is possible that one external system supports services for accounting, customer information, FX rates and limits. Or these could reside in separate systems.

The system uses an external system identifier to aid routing of requests in/out of the relevant external systems.

Each service supported by a system is then mapped to the branches that will use those systems.

The bank can define services against the Main Banking Entity or higher for certain services.

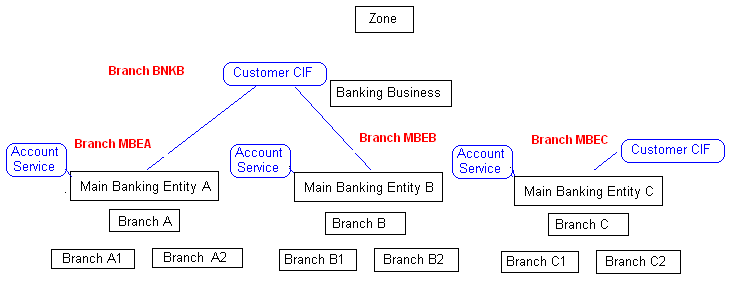
The following services can only exist at the MBE level:

* Account
* Account balance
* General ledger
* Standing settlement instructions
* FX
* Financing
* Limits
* Facilities

Other services such as customer files may exist at a higher level than the MBE branch (but not lower than it).

For example it may be that a bank has three general ledgers that it uses for account based services and postings but one single customer information file (CIF) shared across the organisation.

#### Service Mapping Example



The example above shows that there are three account services defined at the levels of Main Banking Entity A (branch MBEA) and B (branch MBEB) and C (branch MBEC).

### Customer Information Files

The system can link to one or more customer information files in the bank.

It is possible for the same customer mnemonic to arise from multiple CIFs. The system uniquely identifies the source of the customer details by including the source of the record (the branch).

The example above shows that all the branches in Main Banking Entities A (branch MBEA) and B (branch MBEB) share a central CIF defined at the Banking Business Level (BNKB), while Main Banking Entity C (branch MBEC) has a separate customer source. The source is identified by the branch mnemonic at the mapped service level and is held in the database together with the customer mnemonic to uniquely identify the customer:

|  |  |  |
| --- | --- | --- |
| Source | Mnemonic | Name |
| BNKB | ABC | ABC Limited |
| BNKB | DARWIN | Darwin industries |
| BNKB | SMITH | Smith Limited |
| MBEC | MULTICORP | Multicorp Inc |
| MBEC | ABC | ABC Limited |

In this example customer ABC exists in two CIFs for the bank. When searching for the customer ABC within the user will be given the choice of two (if the user can work for a team that includes both BNKB or lower and MBEC or lower branches). The system clearly shows the source of the customer on any screens.

In many cases users will only be aware of a single customer ABC as they may only be working in one discrete area of the bank. It may only become apparent where a hub team is covering the work of multiple locations covering multiple banking entities and CIFs.

Note – Customer groups are also defined per source banking business.

# The Parameters that Support Global Processing

This chapter covers the parameters that need to be set up to support global processing. It explains how each is set up, and the dependencies between them.

1. This chapter is intended as a reference section and provides information on these parameters in alphabetical order.

## Overview

This chapter covers (in alphabetical order) the following parameters relevant to global processing:

* Branches
* Corporate access systems
* Entity types
* Event groups
* External systems
* Parameter sets
* Processing cycles
* Product references and sequence numbers
* Services
* System options
* Teams & User roles
* Workflow orchestrations
* Zones

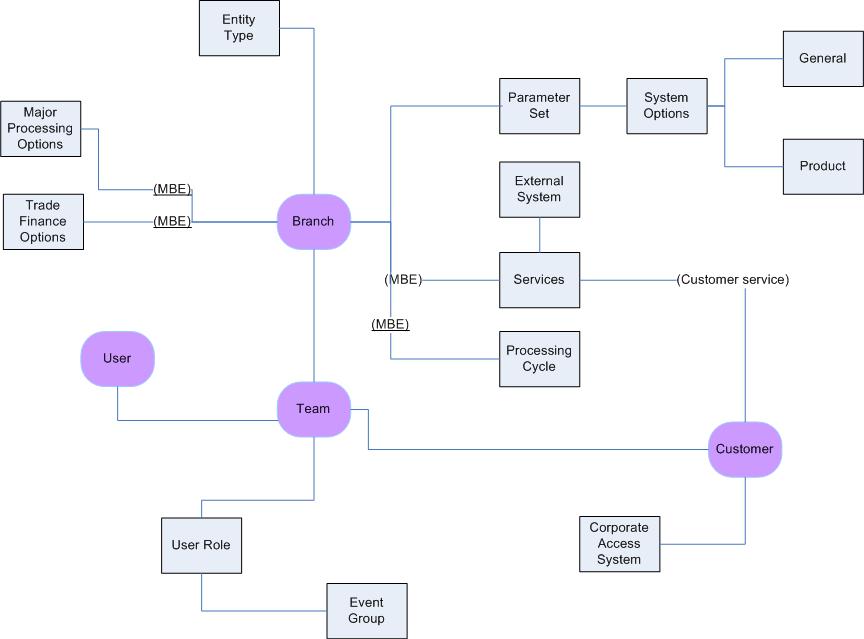
For each parameter it explains where they are set up and how they relate to other parameters, specifically whether there are any dependencies.

The following diagram shows some of the key parameters supporting global processing and the relationships between them.

(MBE) – this denotes that the parameter can relate to a branch of type Main Banking Entity or higher.

(MBE) - this denotes that the parameter can relate to a branch of type Main Banking Entity ONLY.

(Customer Service) – this denotes that the relationship with the customer is only for the service of type ‘Customer’.



## Branches

Within the system, each discrete legal and/or organisational component of a bank is considered a branch, and those branches are expected to be arranged into a hierarchy that reflects the bank's structure.

In addition some branches are used for groupings and for specifying the point at which services are available to all the child branches.

Branches can be assigned a time zone and an entity type, either as they are defined or at a later date. Both of these are optional. The time zone may be different from the geographic zone to which the branch belongs; it provides the time to be recorded against work on transactions for that branch.

Branches can be assigned to parent branches as they are defined or at a later date.

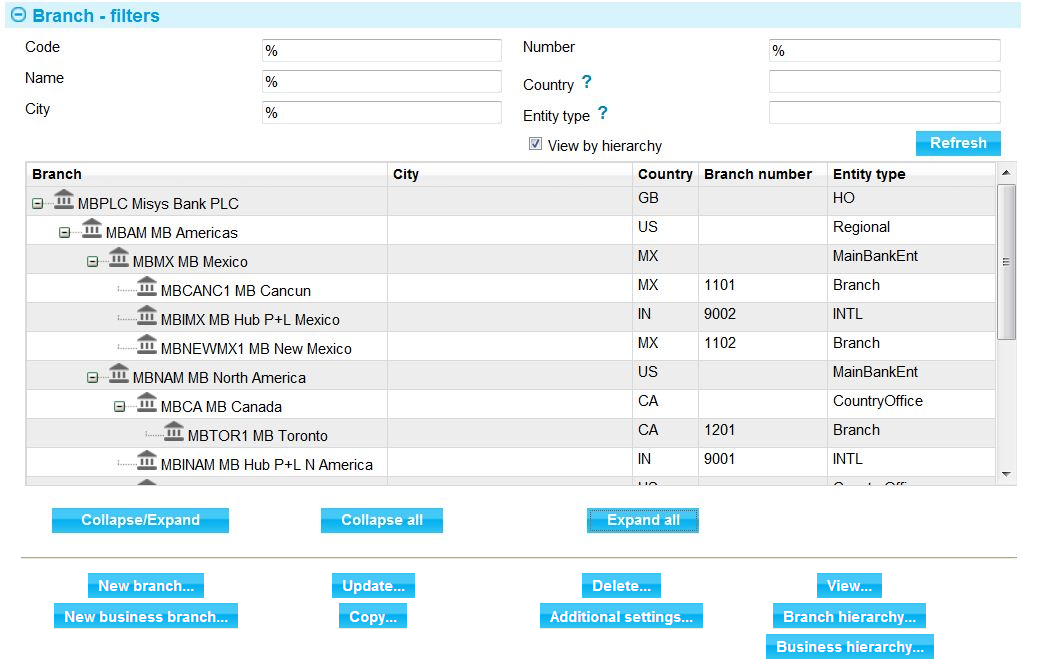
The typical order of creating the branch hierarchy is as follows:

* Define transactional branches. This comes first as these may be replicated in the database from an external system so they will already be present
* Identify and define the parent MBE. This is dictated by the general ledgers of the bank
* Identify and define any non transaction branches below the MBE and link to the MBE. These may be groupings required to allow areas of the business to share common parameters or to enable the security model within the teams
* Link each transaction branch to a non transaction branch or directly to an MBE
* Define any groupings above MBE such as banking business and banking group

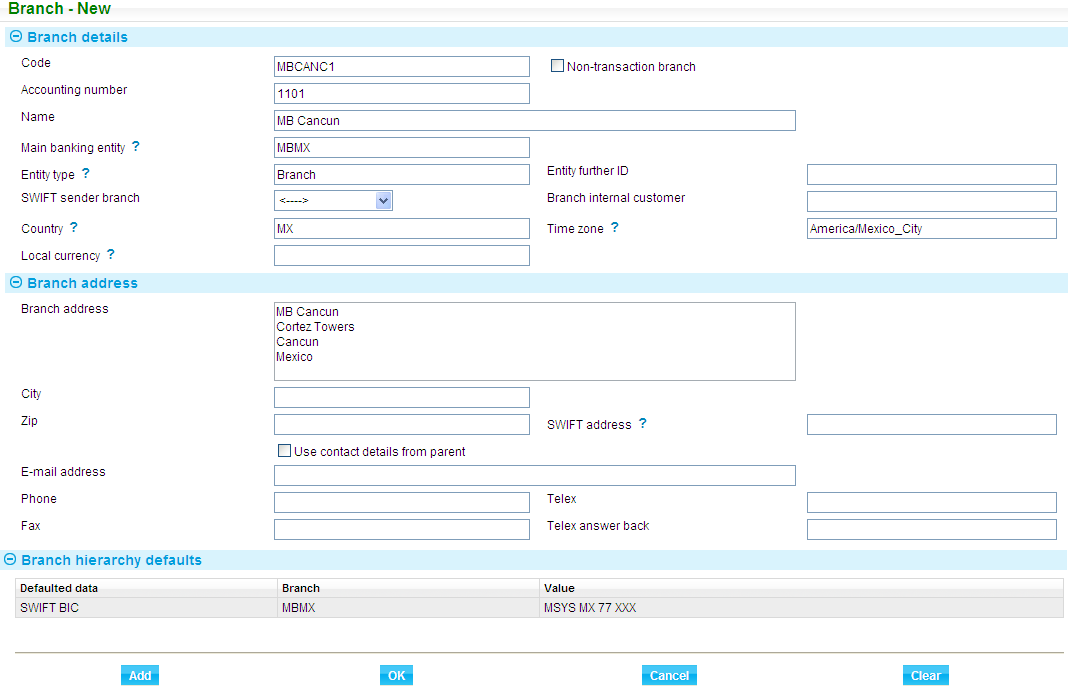
The flexibility within the system means that the model can be altered once set up. For example to introduce more non transaction branches to aid in system configuration.

Once set up, branches are assigned parameter sets and linked to teams, and may also be linked to users as transaction or enquiry branches (see page 23). Additionally for the MBE or higher there may be mappings to services and processing cycles.

Branches are set up using the static data application's Branches|Branches menu option. When you select this option the system opens a browser window that allows you to view the branches already set up. If the View by Hierarchy flag is checked the system shows the branches within their hierarchy; otherwise they are shown in alphanumerical order.



The window used to set up a new branch is illustrated below:



It is not necessary to set up all the details for all branches. Some values can be inherited from the branch parent as shown in the example above where the SWIFT BIC is taken from the parent branch MBMX. The following details can be inherited from the parent branch:

* Accounting number
* Country
* Time zone
* Local currency
* SWIFT address
* Telephone, fax, telex, email

### Main Banking Entity

The following fields become available if a branch is denoted as a Main banking entity:

* Base currency
* Local currency
* Exchange rate format
* Exchange rate tolerance
* A MBE is automatically set as a non transaction branch and a banking business

Other branches can be set as banking businesses and banking groups providing that they sit higher in the hierarchy than the MBE.

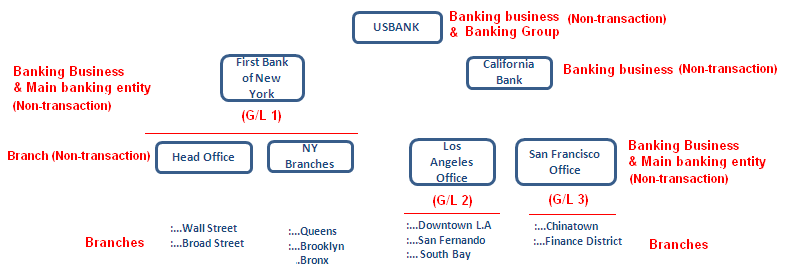
The following matrix illustrates the processing behaviour controlled by setting the various flags on the branch:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Banking Group | Banking Business | Main Banking Entity | Non Transaction | Autonomous | Branch | Agency |
| A/c No |  |  |  |  |  |  | O |
| Services | D | D |  |  |  |  |  |
| Major processing options | D | D |  |  |  |  |  |
| Base ccy & FX rates |  |  |  |  |  |  |  |
| Trade finance options |  |  |  |  |  |  |  |
| Transactions allowed |  |  |  |  |  |  |  |

O = Optional

D = Certain services can be set up higher than MBE

Example:



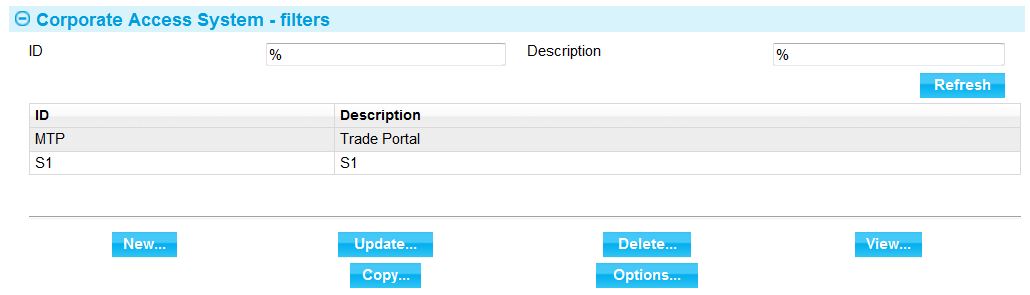
See the Static Data Maintenance User Guide – Trade Innovation for full instructions on setting up branches.

## Corporate Access Systems

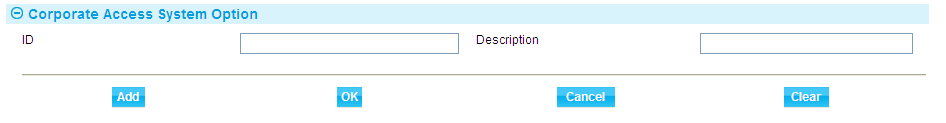
Corporate access systems are a special type of external system (Trade Portal, for example) used by banks to allow their customers to communicate with the system. A bank can interface with any number of corporate access systems.

Once a corporate access system has been defined, it can be associated with the corporate customers that will use it (using the static data application's Customer Maintenance|Customer Details menu option). Messages are then sent to the relevant corporate access system for corporate customers as part of transaction processing.

Corporate access systems are set up using the system tailoring application's General System Definition|Corporate Access Systems Options menu option. The window displayed when you select this menu option lists all the corporate access systems already set up on your system.



The window used to create a new corporate access system allows you to enter a unique code for the corporate access system and a description.



Both fields are mandatory.

Each system can have its own set of processing options. For example they may support a different range of products.

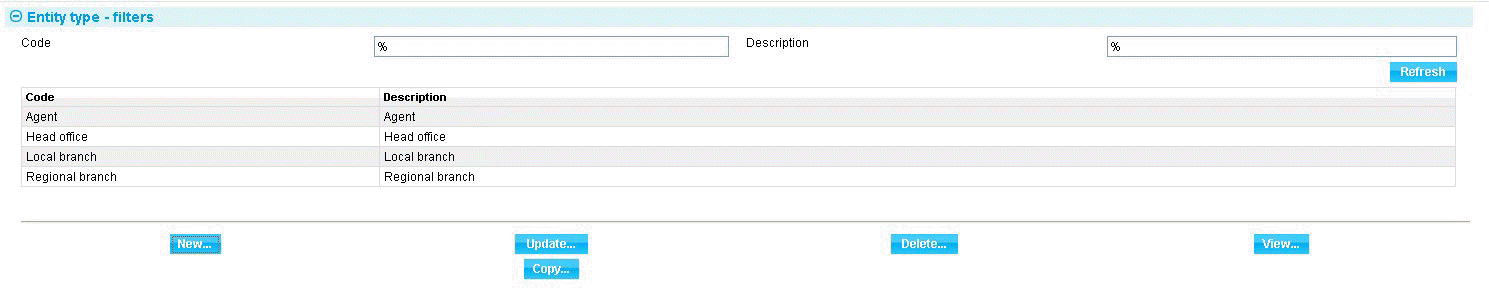
See the System Tailoring User Guide – Trade Innovation for full instructions on setting up corporate access systems.

## Entity Types

Entity types are used to categorise branches. The bank sets up its own values for entity types, giving each a unique ID and a description. The relevant entity type can be assigned as branches are set up or maintained.

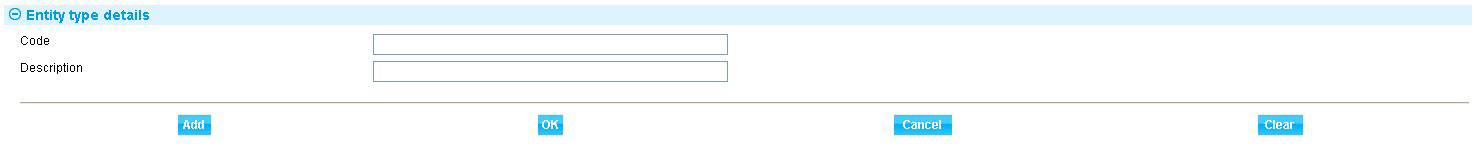
Entity types are not mandatory, and the entity type setting is not inherited by child branches. Entity types are used as a filter field on the Branch Browser; and the bank can create reports using this value (for example, to report by type of branch or to exclude insourced banks from reporting).

Entity types are set up using the static data application's General|Branch Entity Types menu option.



The window displayed when you select this menu option lists all the entity types already set up on your system.

The window used to create a new entity type allows you to enter a unique code for the entity type and a description. Both fields are mandatory.



See the Static Data Maintenance User Guide – Trade Innovation for full instructions on setting up entity types.

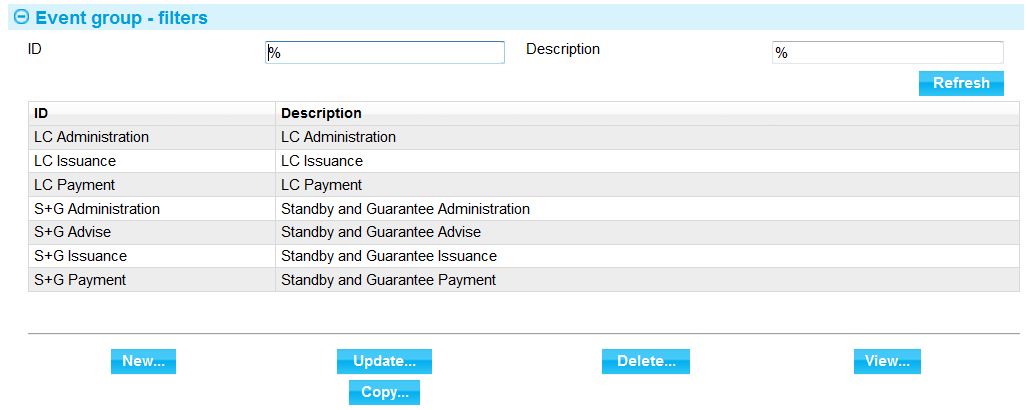
## Event Groups

Event groups are used to control what events users performing a particular user role can work with. Event groups bring together all the events (within or across products) that actual users performing a particular role within your bank might be expected to process.

Typical groupings might be a group for all events to do with issuing and amending a product or group of products; a group for all events to do with payments; a group to do with general administration. The groupings will probably match the way that the bank already organises their staff roles, although a consideration when moving to a hub and spoke based global processing model is the opportunity to re-engineer the business processes to create different groupings.

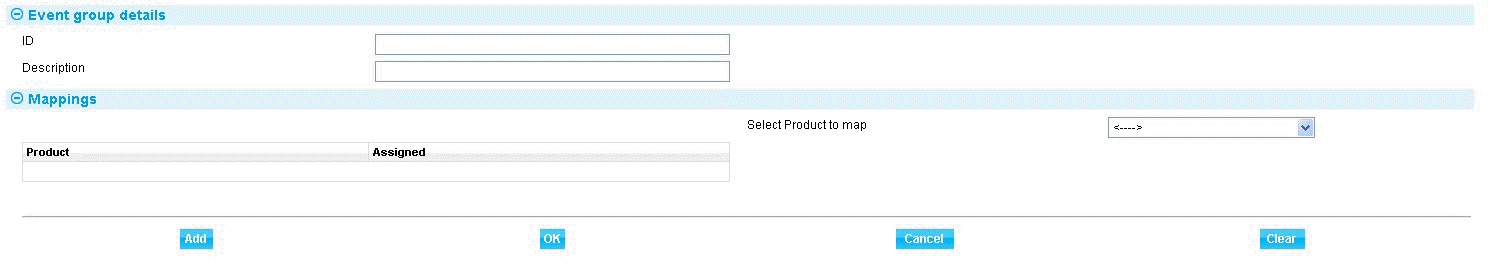
Event groups are used in conjunction with user roles, and must be set up before the user roles that will use them can be created.

To set up an event group, in the security application select the Security|Event Groups menu option.

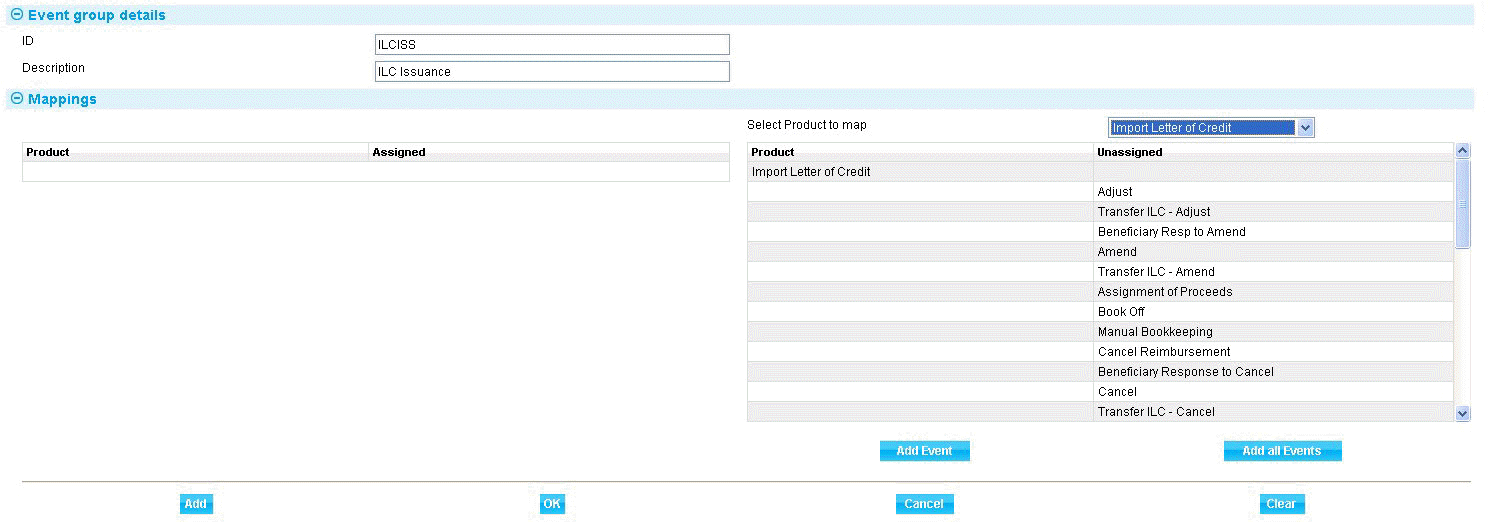


The window displayed when you select this menu option lists all the event groups already set up on your system.

The window used to create a new event group allows you to enter a unique ID and description for the event group. Both fields are mandatory.



You can then select each product, and the events from it, in turn to add to the event group.



See the Security Guide – Trade Innovation for full instructions on setting up event groups.

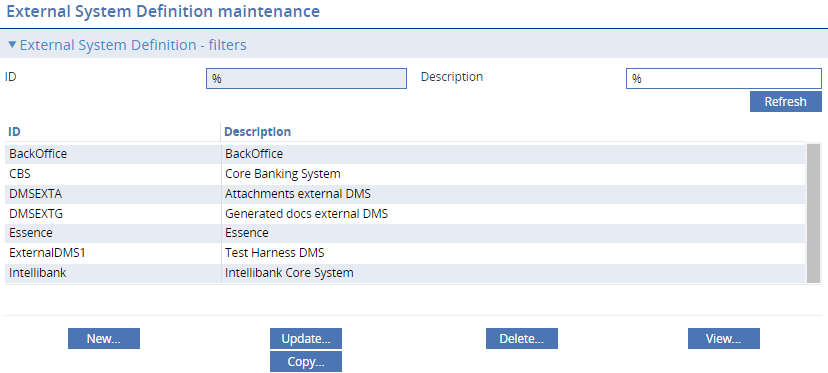
## External Systems and Services

External systems define each of the systems external to Trade Innovation that has services used by the system. Services are then assigned to those systems as some systems may cover multiple services such as general ledger, customer information and limit checking. Services may be, for example:

* General ledger systems
* Limit checking systems
* Facility systems
* Watch list checking
* Foreign exchange processing systems
* SWIFT
* Financing
* Customer
* Exchange data and External review step query responses systems
* Systems receiving release-notification of transaction data
* Attachments and Generated documents document management systems (DMS)

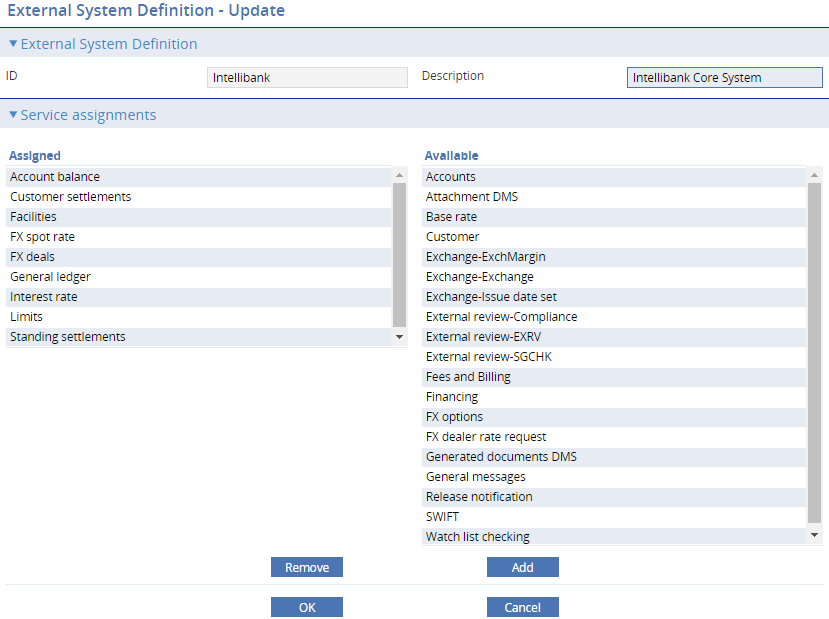
External systems are defined within system option parameter sets, and thus to the branches that use those actual external systems. This allows a zone to interface to different external systems of the same type - for example, to multiple general ledger systems - as required at branch level. When the system generates messages such as postings or FX deals upon the release of an event, the external system identifier is included in the message as routing information to help the bank's routing infrastructure to pass it to the correct external system.

External systems are set up using the system tailoring application's General System Definition|External System Definition menu option. The window displayed when you select this menu option lists all the external systems already set up on your system.



The window used to create a new external system allows you to enter a unique code for the external system and a description. Services are then assigned to the external system.

All fields are mandatory.



### Assigning Services to Branches

Once the external systems and the services they can support have been defined these must be linked to the branch that will use these services.

This is done using the system tailoring application’s General Branch Definition|Branch Options/Services Mapping menu option. The window displayed when you select this menu option lists all the branches and shows which services/external systems have been applied at this level.

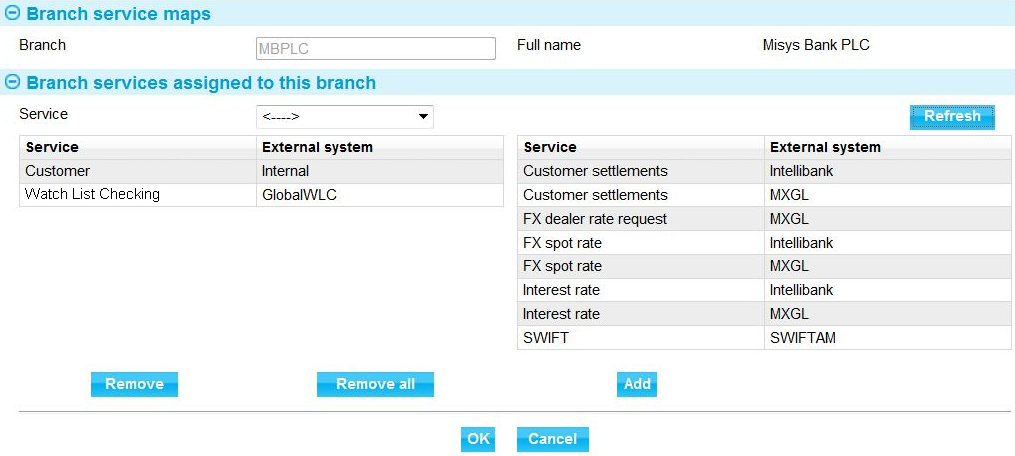
In this example the customer service for the bank has been linked to the highest level banking group of MBPLC branch. This means that ALL branches within the hierarchy use one service for the lookup of customer details.

For Mexico a specific external system MXGL supports services for the Mexico operations (branch MBMX and its child branches). Each service mapped shows which external system it will use to supply the service.



The Assign Services button is used to assign services from the available list of services to the branch.

All the services by external system that have not yet been assigned to a branch are shown as available for mapping.



Some services can be satisfied in two ways:

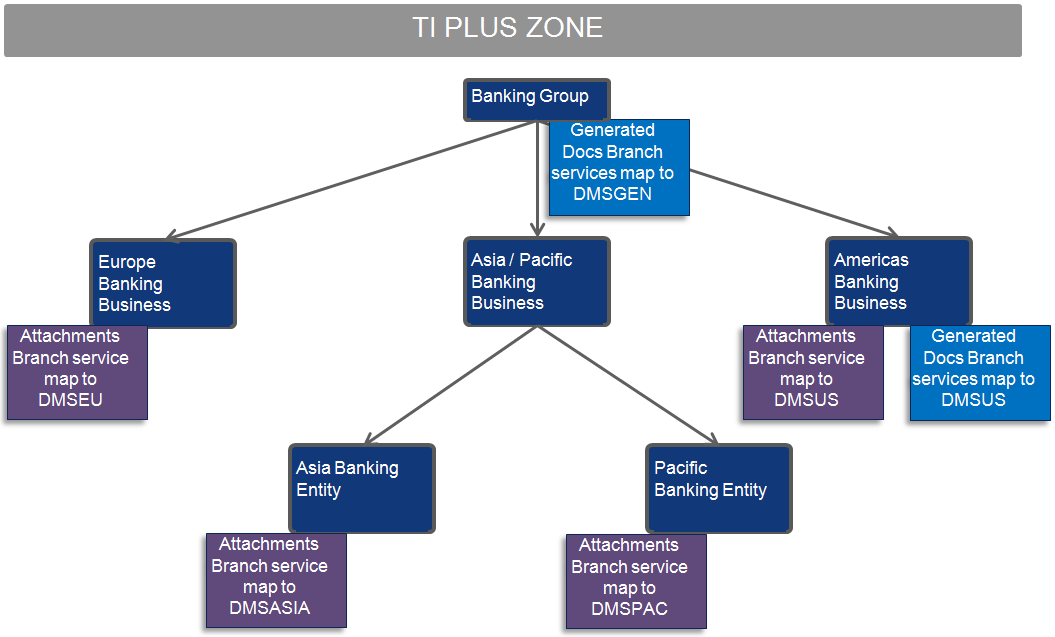
* A named external service – requests are routed to a service external to the system in real time. In the example above customer settlements payment routing service are available via two external systems
* An internal service – requests retrieve the information directly from the database. In the example above customers are available from the system’s database

Internal services are supported for the following items:

* Customers
* Standing settlement instructions
* Accounts
* Base rates
* Limits – if you are using the departmental limits for limit checking
* Attachments and Generated documents document management system (DMS)

This means that details will have been replicated (or manually entered) to the database as there is no external service call to retrieve these details.

The DMS systems associated within the zone branch hierarchy can be different for Attachments and Generated documents. See the following diagram:



This example has the Attachments (scanned images, spreadsheets etc.) and Generated documents (Trade Innovation generated PDFs) configured to be stored in different external DMS’s across the zone.

* Separate Attachments DMS’s are utilised to segregate document into separate groups for searching to be restricted to the relevant regions.
* Generated documents are collected into a single DMS to manage efficient deletion of expired/replaced generated documents.
* Americas utilises a separate DMS to other regions for data security reasons.

See the SDK Application Extension Guide and the SDK System Interfacing Guide for details on implementing the external DMS incoming and outgoing web services to manage metadata and enable housekeeping of removed documents.

1. The DMS used for document templates and generated reports can be set only for the zone through zone general system options that define the ID of the system used for that purpose.

See the System Tailoring User Guide – Trade Innovation for full instructions on setting up external systems and zone options.

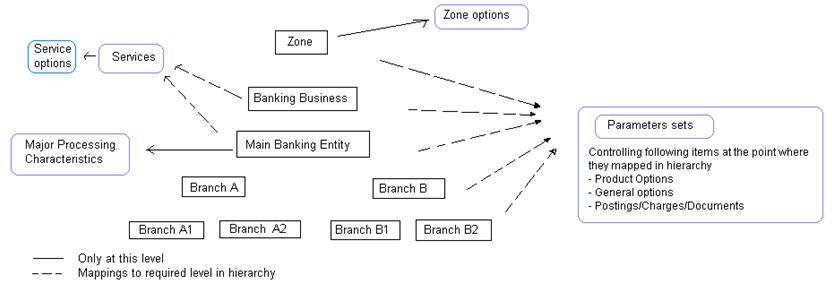
## Parameter Sets

Parameter sets are used to define the processing characteristics that will govern how the system behaves; and the data sets the system will use during transaction processing. They can be set at any level in the branch hierarchy. These sets are defined independently of the branch hierarchy and then mapped subsequently. They can be linked into hierarchical sets with general system settings at the top and more specific regional or local sets further down the hierarchy.

Separate parameter sets are defined for the following data items:

* Bank-defined error messages and warning messages
* Charges
* Clauses
* Customer documents (within Product / event level documents parameter set)
* Advanced workflow documents (within Step level documents parameter set)
* Postings
* System options – general and product related
* Tracers
* Interest types
* Product types
* SLA details
* Orchestrations
* Payment actions

The following diagram illustrates the various levels within a branch hierarchy that the parameters used within the system can be deployed. Some options are only valid for a branch flagged as a main banking entity, others can be mapped to any type of branch.



The relevant parameter sets, parameter set IDs and hierarchical relationships and structures must be defined before any of the data listed above is set up. After the initial design has been done, the process of setting up parameter sets has three stages:

* Set up parameter set IDs and organise them into the required hierarchical structure
* Map each parameter set ID to each of the branches that will use it
* Set up the actual parameter sets, assigning each the correct parameter set ID as you do so. Start with the top-most parameter set for each type and work down the hierarchy to allow each level of parameter sets to inherit the settings from the one above it

Finastra recommend that you define a naming convention for parameter set IDs that will make them easy to identify when mapping them to branches and assigning them to actual parameter sets (see page 15).

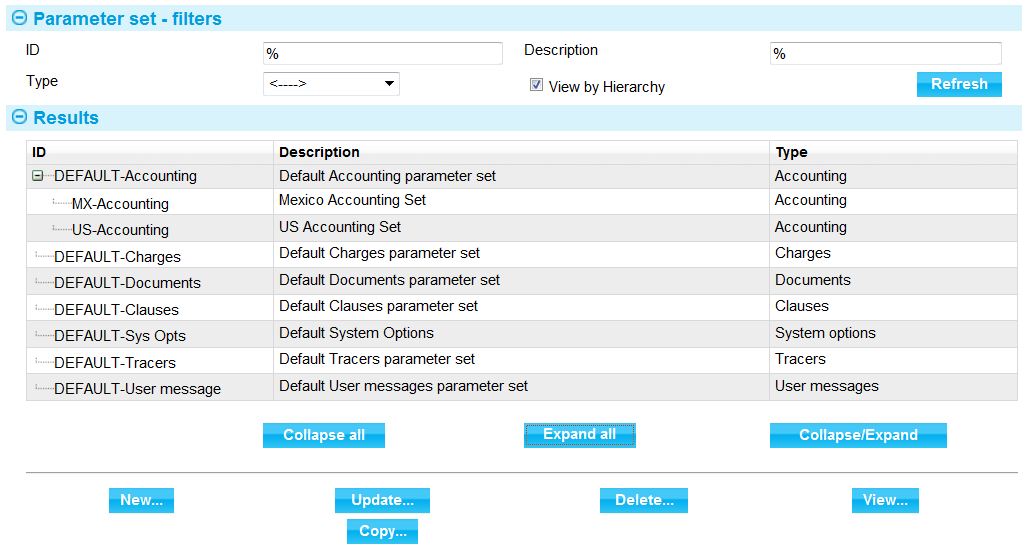
The parameter set IDs delivered with the system are given the name:

DEFAULT-aaaaaaa

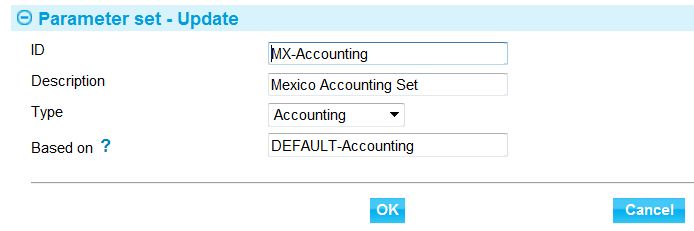
where aaaaaaa is the type (for example documents or charges).

### Setting Up Parameter Set IDs

Parameter sets are initiated using the system tailoring application's Parameter Sets|Parameter Set Definition menu option. The window displayed when you select this menu option shows all the parameter sets already defined. If the View by Hierarchy flag is checked the system shows the parameter set IDs within their hierarchy; otherwise they are shown in alphanumerical order.



The window used to create a new parameter set is illustrated below. All fields are mandatory.

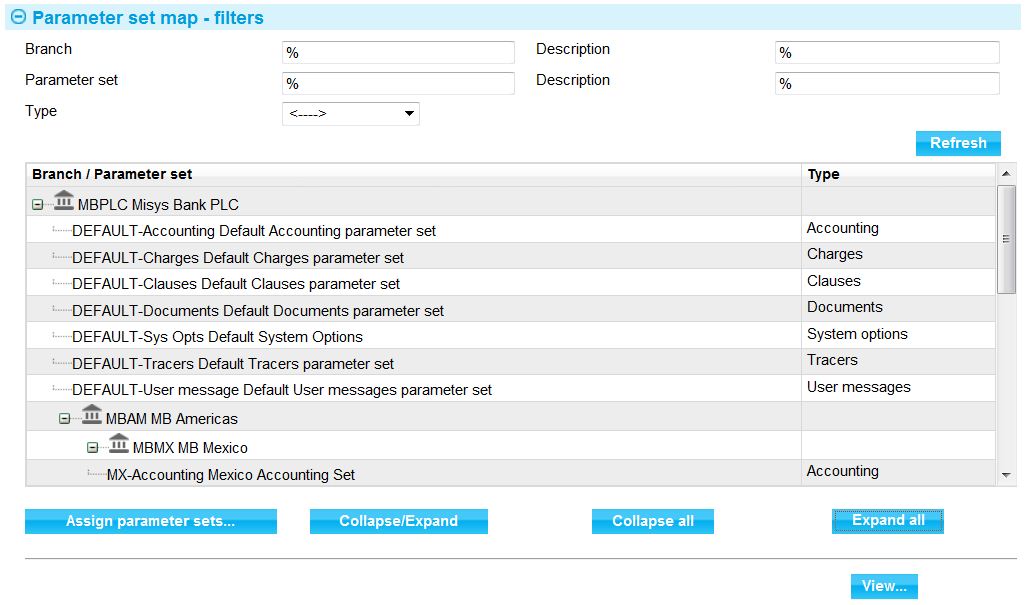


The Type field allows you to select the type of parameter set this parameter set ID will be used with.

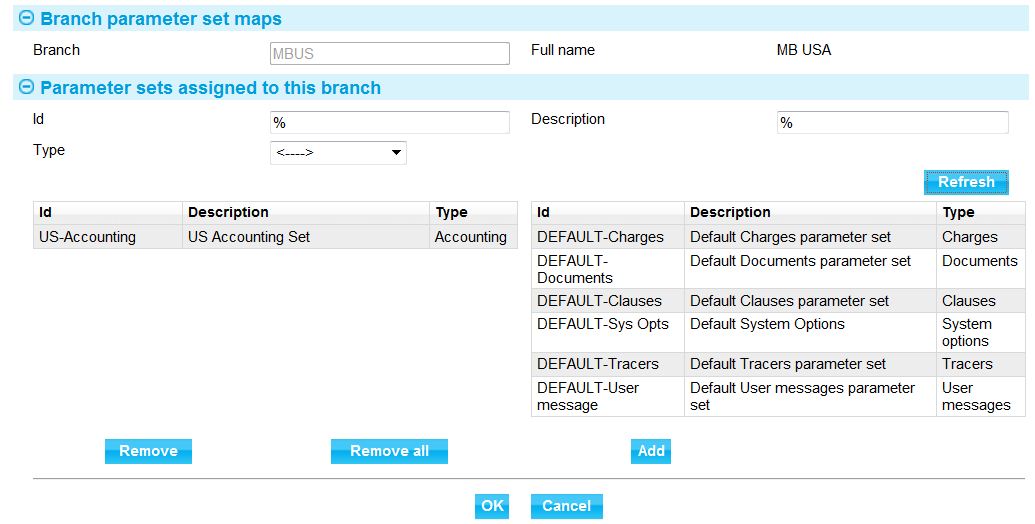
The Based On field allows you to select the parent parameter set ID, if one is being used.

### Mapping Parameter Set IDs to Branches

Parameter sets are mapped to the branches that will use them using the system tailoring application's Parameter Sets|Parameter Set Mapping menu option. The window that is displayed lists all the branches set up, and shows any parameter sets (together with an indicator of their type) already mapped to them.



The window displayed when you create a new mapping is illustrated below:



For each branch you can map a single parameter set ID of each type. If you do not map a parameter set ID of a particular type to a branch, it inherits the parameter set of that type from its parent in the bank's branch hierarchy.

See the System Tailoring User Guide – Trade Innovation for instructions on setting up parameter sets and mapping them to branches. For parameter sets related to workflow orchestrations, see the *Workflow Tailoring User Guide* – Trade Innovation.

### Setting Up Parameter Sets

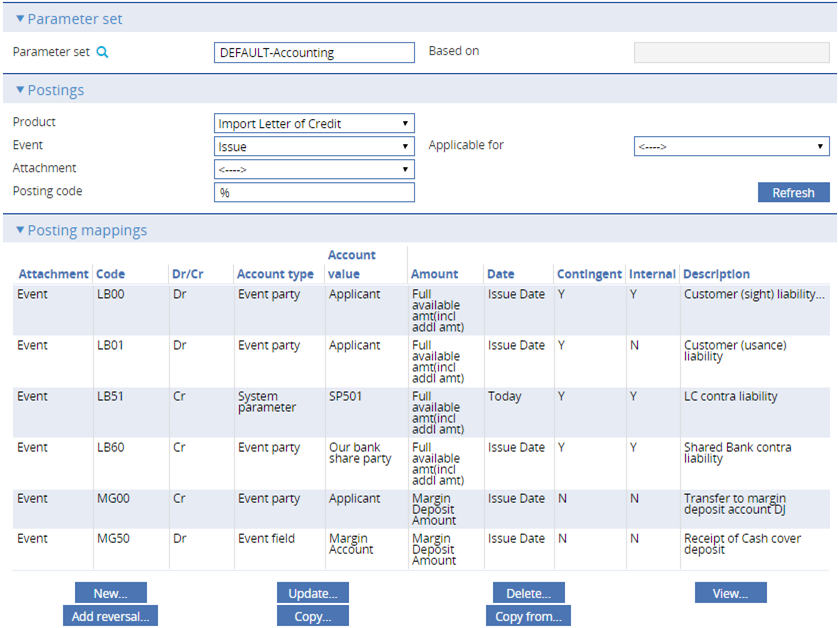
The actual parameter sets consist of sets of (for example) charges, postings and tracers; and these are set up using the appropriate menu option from within the system tailoring application. This section will use postings as an example, but all other types of parameter are handled in the same way.

Select the relevant system tailoring menu option (in this case Parameter Sets|Postings).

In the window that is displayed, to show information for an existing set of postings enter the parameter set ID, the product and event and press the Refresh button.

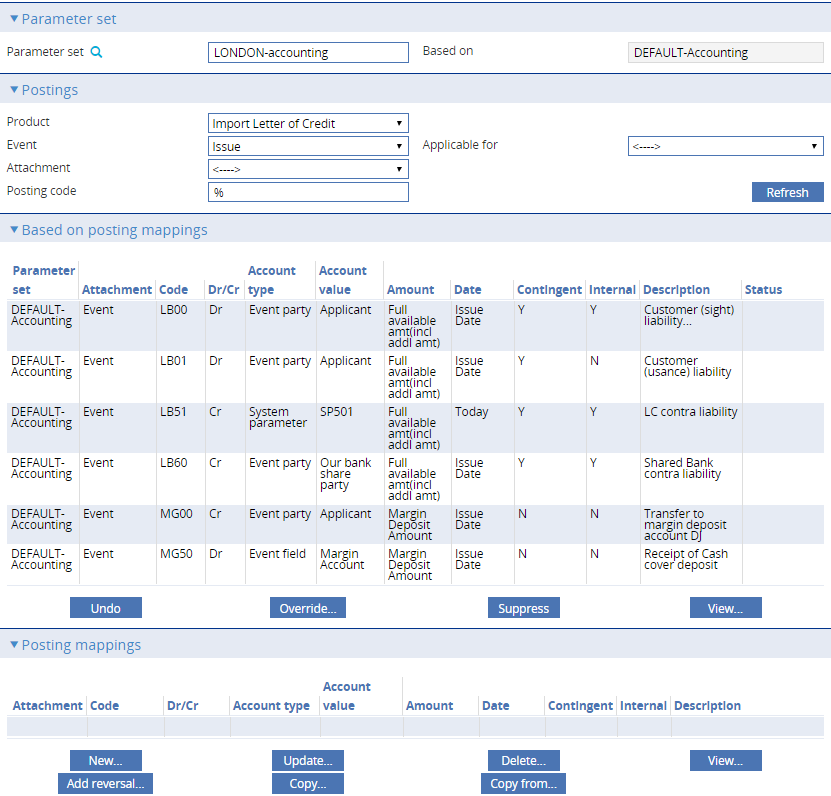
The system displays postings relevant for the level of parameter in the hierarchy. The screen shows the parameter set that this set is based on. In this case this set has no parent set.

1. Postings rules include a parameter to manage generation on provisional and/or final events. The filter ‘Applicable for’ allows selection of postings by Final and or Provisional category.



To define a new parameter set, select the parameter set ID in the Parameter Set field, then select the product and event (and attachment level, too, if relevant) for which you wish to define the new posting set.

In this case the new set to be defined is the LONDON set and this is based on the DEFAULT set.

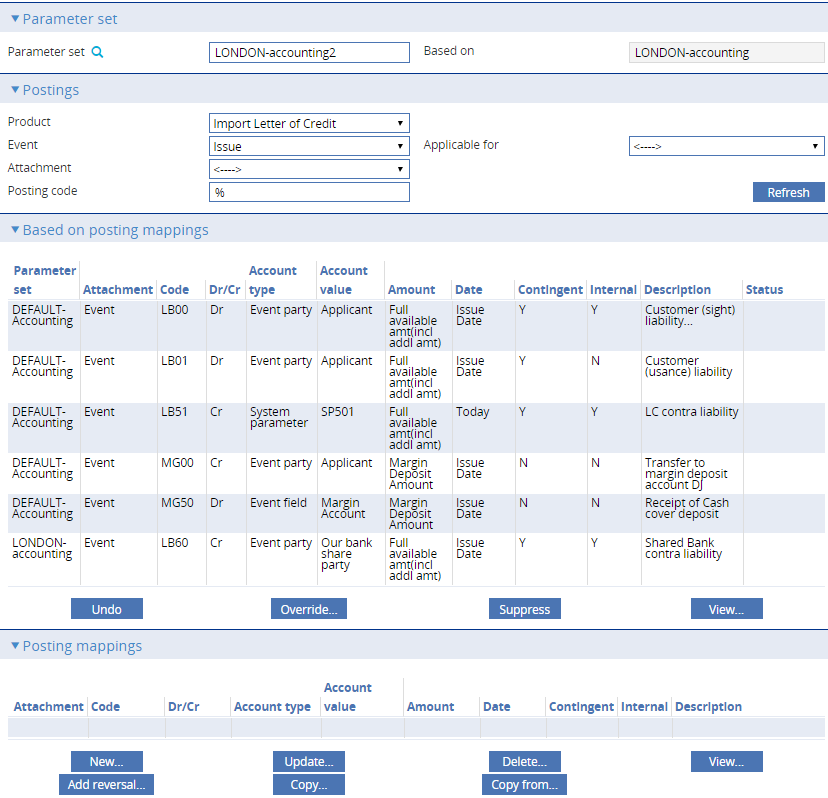


By default the postings set for this parameter set ID includes any postings inherited from the parent parameter set ID (shown in the Based On Postings Mappings pane) and any new ones entered for the child parameter set (shown in the Posting Mappings pane).

As well as adding new posting definitions, the bank can amend or delete any inherited ones. The Status column in the Based On Postings Mappings pane indicates whether a posting definition in the inherited set has been amended or deleted. Where a posting definition is amended, an entry is created for the new values in the Posting Mappings pane.



If you create a further set of posting definitions for the next parameter set ID down in the hierarchy from the one shown above, the postings shown in the two panes are consolidated; the Based On Postings Mappings shows any posting definitions that were not changed or deleted, plus the new ones.

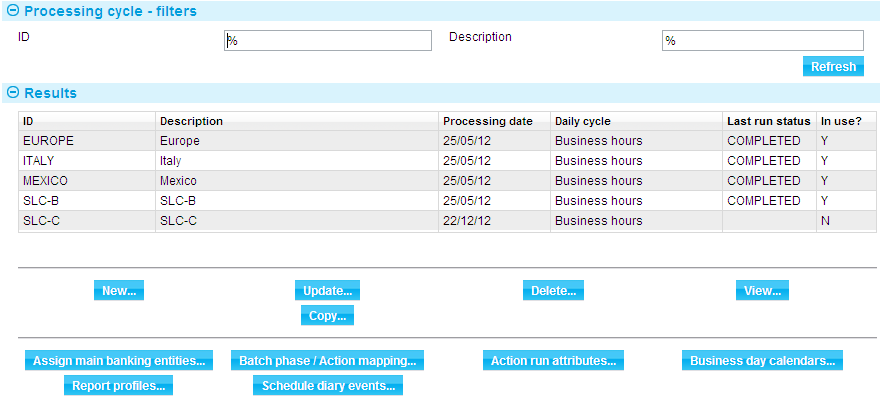


## Processing Cycles

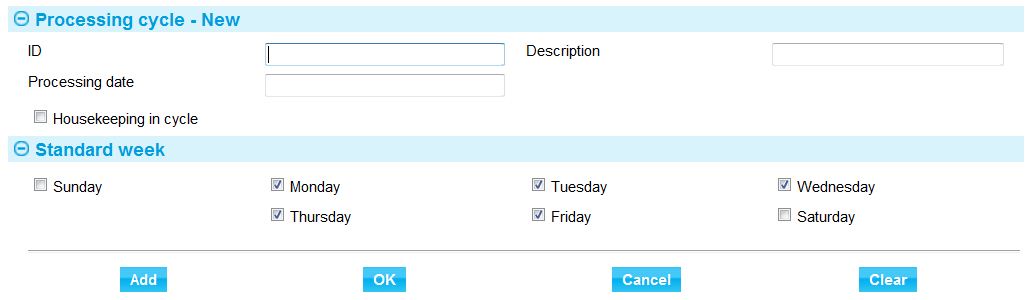
Before the bank can define what actions overnight processing will perform, and in what order, it needs to set up the necessary overnight processing cycle parameters.

Overnight processing cycle parameters are set up using the batch setup application's Phases|Processing cycles maintenance menu option.

The window displayed when this menu option is selected lists all the processing cycle parameters already set up on your system.



The window used to create a new processing cycle allows you to enter a unique ID code for the processing cycle parameter and a description. Both fields are mandatory.



The start processing date and standard business days must also be specified as these are used when determining the next business date for all the branches included in that cycle.

1. You must ensure that all branches included within a cycle operate on the same business days. For example do not include branches in France with branches in Dubai as the weekend days differ.

Some end of day tasks relate to housekeeping and are not branch specific. One of the cycles must be designated as the cycle to handle those tasks. The list includes the following actions:

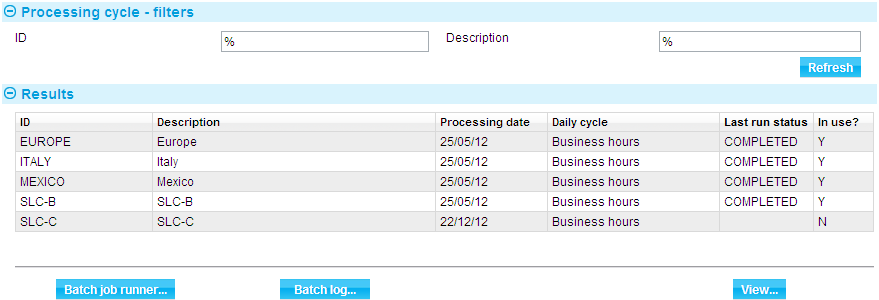
|  |
| --- |
| Archive deleted SWIFT in messages |
| Customer reorganisation |
| Delete expired reports |
| Delete expired reporting view data files |
| Delete downloaded static messages |
| Journal delete |
| Delete sent SWIFT messages |
| Delete sent Telex messages |

See the Business Operations Guide – Trade Innovation for full instructions on setting up processing cycles, including for full information on setting up daily processing cycle data for the following:

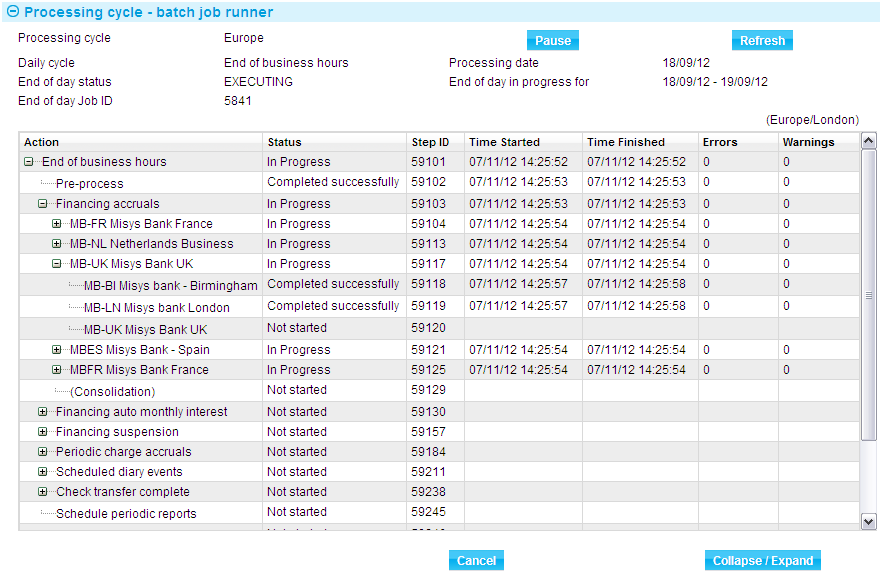
* Assigning main banking entities to cycles
* Assigning actions to run each phase in cycles
* Assigning run attributes to actions run in cycles
* Assigning business day calendars to cycles
* Assigning report profiles to cycles
* Assigning diary event schedules to cycles

### Processing Cycle Batch Job Runner

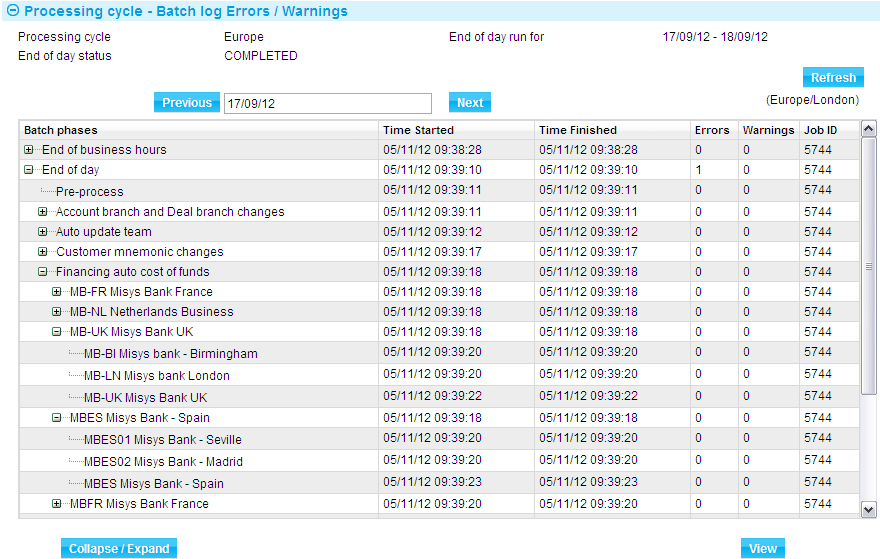
A processing cycle end of day can be submitted and monitored from the batch jobs application’s Processing cycle filters Batch Job Runner button.



This provides information down to individual branch level on the progress of an action within a batch phase for each branch.

 Processing Cycle Diagnostics

A log is maintained for each end of day cycle. The log is available from the batch jobs application’s Processing cycle filters Batch log button. This provides information down to individual branch level on the success or failure of an action within a batch phase for each branch.



See the Business Operations Guide – Trade Innovation for full instructions on running processing cycle end of days and viewing cycle logs.

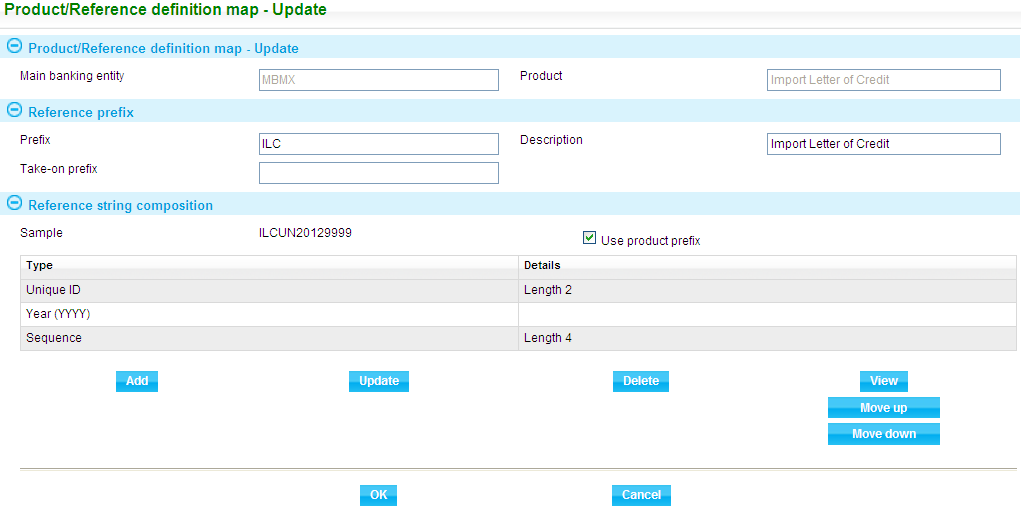
## Product References and Sequence Numbers

The system allows your bank to define the product references to be used within the banking hierarchy. This involves defining:

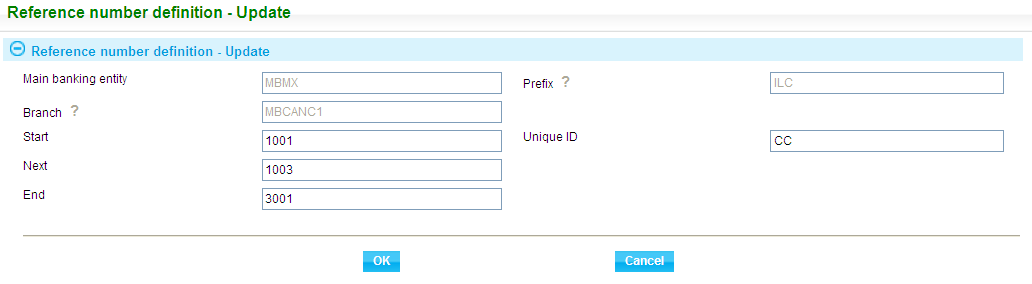
* separate product reference formats for the different main banking entities
* reference number sequence numbers can then be defined at either main banking entity to be used across a set of transaction branches and optionally at transaction branch level.

Select Product|Reference definition to define the required reference structure for each product in each main banking entity as follows:

This allows you to enter details of the Product Prefix, Unique ID, Sequence number and optional fields to be used in constructing the transaction reference.



The next step involves defining the reference sequence numbers to be used by the main banking entity or associated transaction branch.



You can enter details of the main banking entity, branch (where reference numbers are defined at branch level) Unique ID for the branch and sequence number.

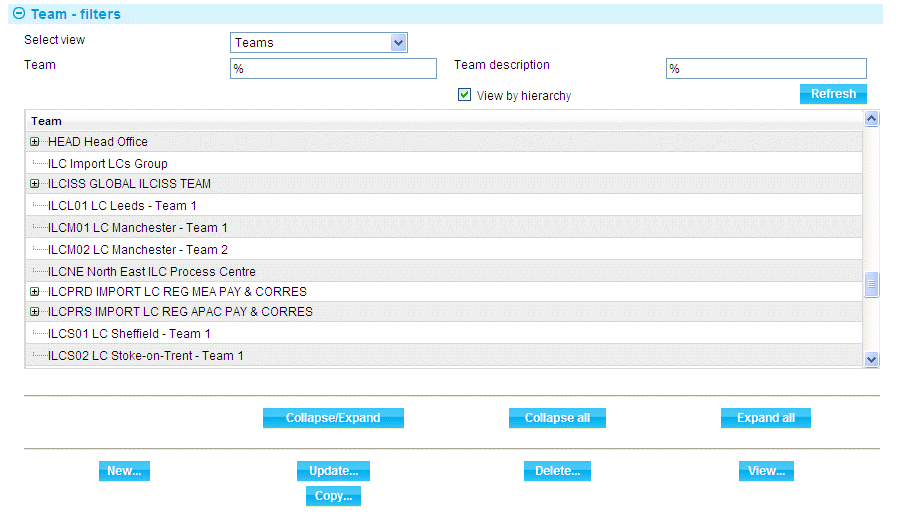
See the System Tailoring User Guide – Trade Innovation for full instructions on setting up references.

## Teams

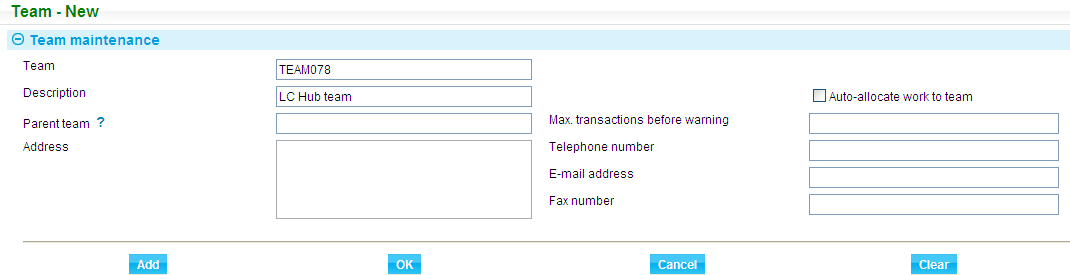
Teams provide the link between branches, users, user roles and customers.

They can be set up using a hierarchical structure, a flat structure, or both. Teams are set up independent of any other parameters and can be assigned to parent teams as they are defined or at a later date.

Teams are set up using the security application's Security|Teams menu option. When you select this option the system opens a browser window that lists the teams already set up.



The window used to create a new team is illustrated below. The Team Name and Team Description fields are mandatory.



A team can be flagged as being available for auto-allocation of work. All teams that allow this can be included in a round robin allocation of work to that team.

From within the Teams Browser window, the drop-down on the Select View field, used in conjunction with the buttons displayed, allows you to assign:

* Users to a team
* Branches to a team
* Accounting branches to teams
* User roles to a team
* Key customers to a team
* Assign transaction branches and enquiry branches to users
* Users to roles within a team

See the Security Guide – Trade Innovation for full instructions on setting up teams.

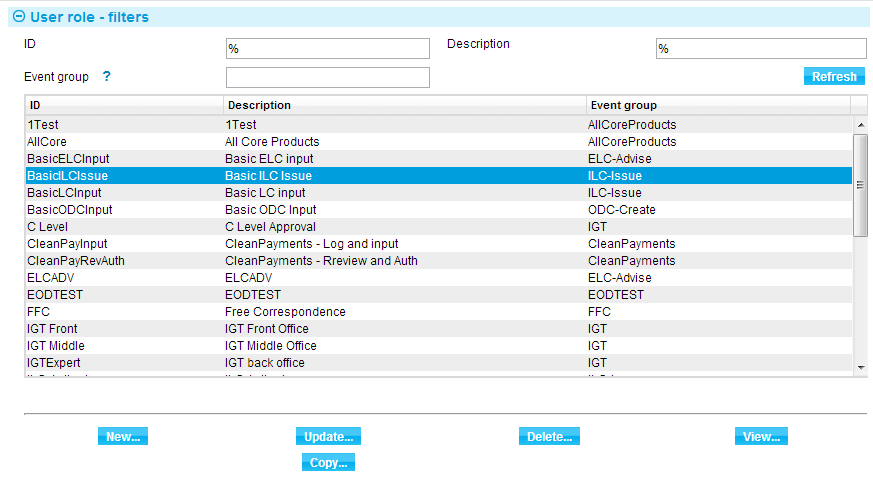
## User Roles

User roles correspond to the actual work members of a team perform during transaction processing. They are used in conjunction with event groups to define what actions users can perform.

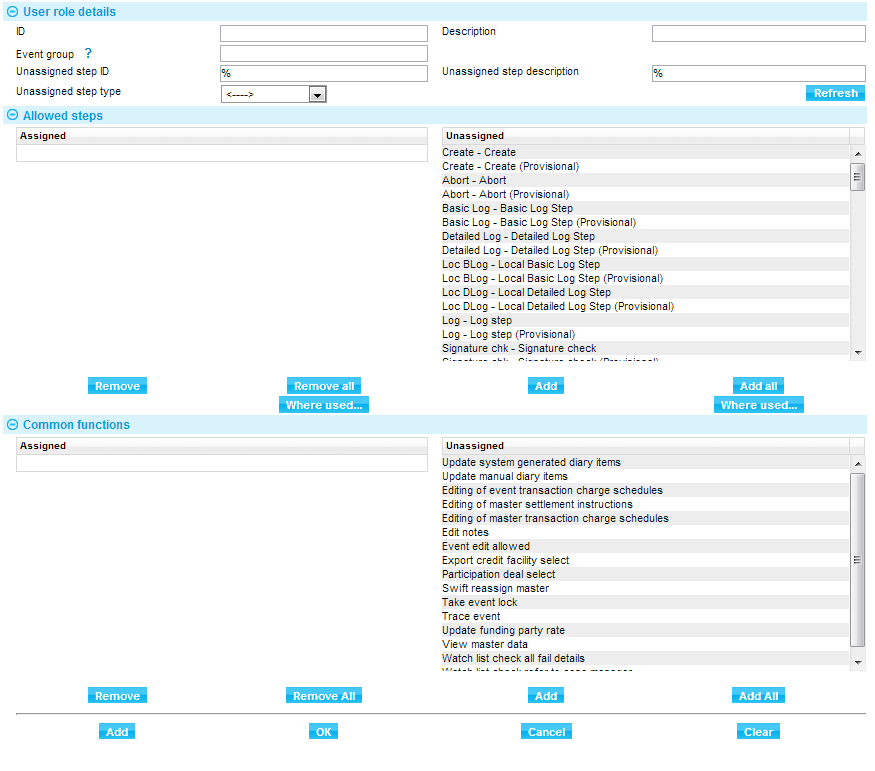
A user role uses a single event group, which must have been set up before the user role.

User roles are assigned to teams, and then to actual users as those users are added to a team.

User roles are set up using the security application's Security|User Roles menu option. When you select this menu option the system lists all the user roles currently set up on your system.



The window used to create a new user role is illustrated below. The ID and Description fields are mandatory.



The window allows you to select an event group and then the steps at which users assigned to the user role will be able to work with events in that event group. The Common Functions pane is used to assign the ability to perform master-level tasks to the user role.

See the Security Guide – Trade Innovation for full instructions on setting up user roles.

### Workflow Orchestration

The system is designed to support the complex workflow requirements of a trade finance bank. In particular it enables a bank to operate either regionally or globally using a spoke and hub type workflow set up.

In such a setup a bank can offer a local customer facing services in the spoke part of the workflow, for example receiving and scanning documents received from a customer, negotiating FX contracts and checking limits and printing out documents for the customer. The hub part of the workflow can provide a centralised centre of excellence for further data entry and for checking the workability of the customers’ requests. The hub would typically complete the data entry and any additional checks such as for watch list compliance. The transaction would then proceed through one or more review steps before items such as SWIFT messages and accounting entries are released. The system comes with a delivered set of workflow steps and sequences that can be configured according to your bank’s needs. This is known as a *workflow orchestration*.

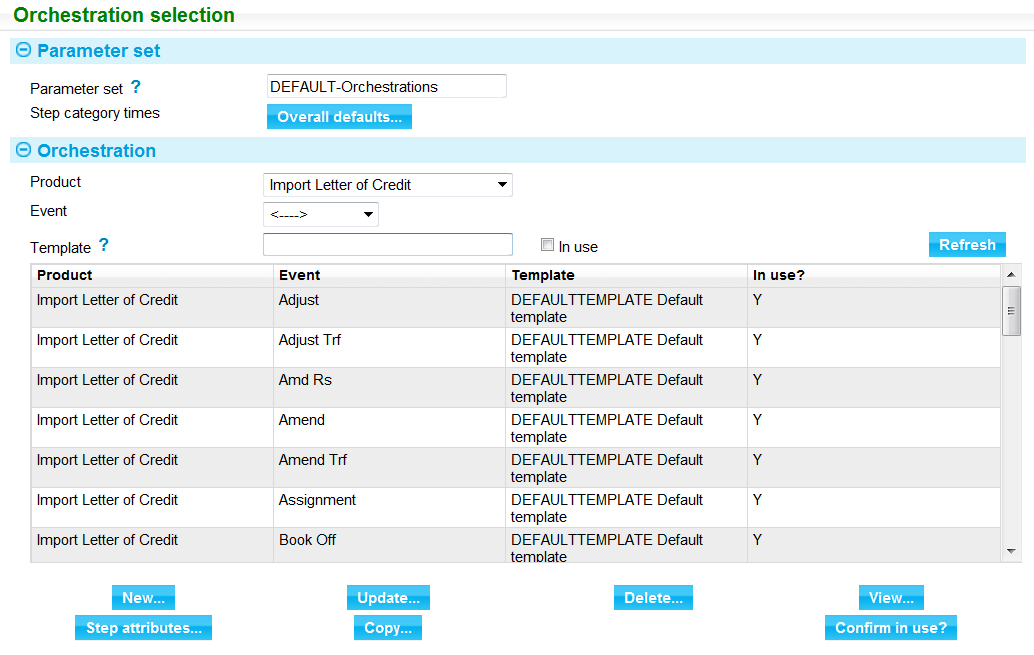
If your bank has purchased the Advanced workflow features your bank can create custom workflow steps, templates and orchestrations.

Workflow orchestrations are set up using the menu option Parameter Sets|Workflow Orchestrations.

There are several other menu options associated with creating a workflow orchestration:

* General business functions|Workflow orchestration steps – used to define processing steps. Each step has an underlying type that governs its behaviour and where in the workflow it can be used (advanced workflow feature)
* General business functions|Workflow orchestration templates – used to link steps into a sequence. Templates can be used to define the typical sequence of steps required. Different events may require different flows. For example it is highly likely that the steps required to issue a letter of credit will be more than the steps required to send correspondence to a customer (advanced workflow feature)
* Parameter sets|Parameter set definition – used to create a set defining a particular type of workflow. These sets are used by a workflow orchestration and linked to the branches that will use them

A workflow orchestration uses a particular orchestration template and is linked to a parameter set. Within the orchestration you can define the default processing times for each step, the conditions under which the step is invoked, for example when an amount is above a certain value and other workflow related behaviour. The initial and rejection target steps are defined within orchestrations.



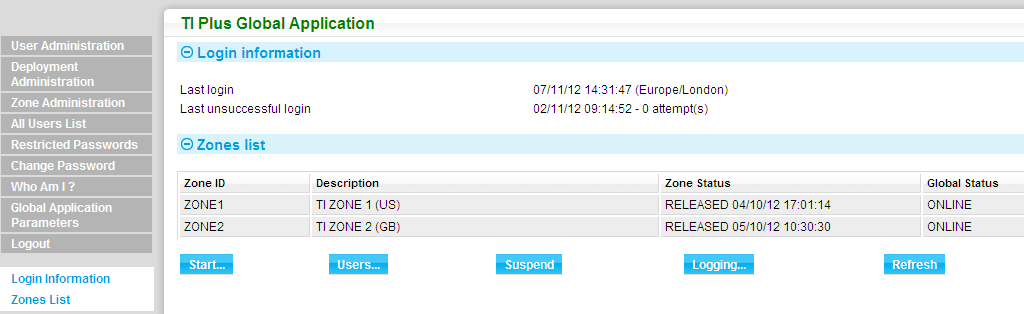
See the *System Tailoring User Guide* – Trade Innovation for full instructions on setting up workflow orchestrations.

For advanced workflow see the Workflow Tailoring User Guide – Trade Innovation for full instructions.

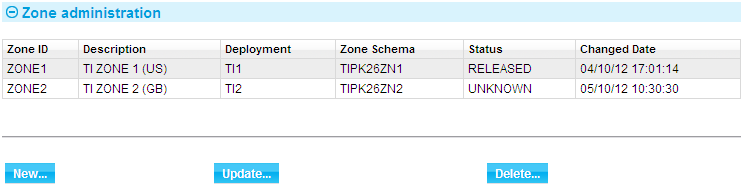
## Zones

Zones must be decided upon and set up before any other configuration begins, since the databases are set up within zones.

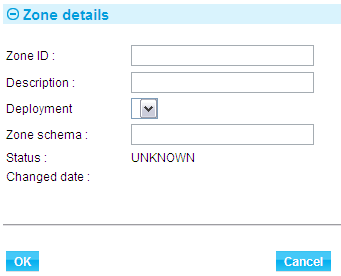
Appendix C provides some guidance on the reasons why more than one zone may be required for a bank.



Zones are managed using the Zone Administration link, which displays a list of zones already set up on your system.



The New button displays the window used to set up a new zone.



The Zone ID and Description fields are mandatory.

See the Business Operations Guide – Trade Innovation for instructions on setting up zones and zone management.

# Monitoring Work Loads

This chapter explains how the Dashboard allows users to monitor their workloads and, if supervisors, the workloads of their teams.

## Overview

The system provides various means for banks to monitor and manage their workload:

* Service level agreements (SLAs)
* Automatic routing of transactions to specific teams or users at certain points in their life-cycle
* The Dashboard - for monitoring workloads and progress against SLAs
* All step start and finish timestamps are presented in the user’s location time zone. The step times as at the behalf of branch are also available for hub processing.

## Service Level Agreements

The service level agreement functionality allows the bank to set up SLAs involving target times for completing time-sensitive events. SLAs are allocated to customers either individually or by customer groups or customer type. Target times can be made dependent on factors such as the value of the transaction or its principal party's country.

During transaction processing, events are displayed in the Dashboard as having one of three SLA statuses - low, medium or high - depending on how close to their target completion time they are. Each status has a bank-definable colour so that those approaching their target time can be easily identified visually. The bank can configure, for each event/SLA combination, when the event's status should change to medium and then to high, relative to the event's target completion time. SLAs can optionally operate within branch business hours and/or initiate at the start of the next business day if received after a cut-off. An SLA can initiate on an orchestration step transition.

For information on setting up SLAs see the System Tailoring User Guide – Trade Innovation.

Average processing times can be set up for each step in an event. These are used during transaction processing to calculate and display in graphical formats the workloads of different teams and users. For events not covered by a specific SLA agreement, the bank can set up average figures for the completion of each of the steps in an event's life-cycle, and a default completion time through the remaining steps. Advanced workflow includes a total remaining time including parallel steps. These are used in the same way as SLA details when calculating workloads during transaction processing.

These are configured within orchestrations parameter sets to apply to the steps in each workflow. For information on setting up SLA step times see the Workflow Tailoring User Guide – Trade Innovation.

## Automatic Routing

The system provides a way of routing transactions to the users who will work on them at each stage in their life-cycle. A key aspect of global processing is the ability to define teams and their associated branches and roles. This information can then be used to automatically determine relevant teams that can process an event based on the behalf of branch and transaction type and current event step.

When a user creates a new event from the dashboard or master browser, they specify the team and behalf of branch for which the transaction is being created, and the event is flagged as ‘manually assigned’ and the transaction is associated with the input user. As the transaction moves to the next step the system automatically assigns it to a team or team/user who can perform that step as follows:

* Team or Team/User based on the event team map – this allows rules to be set up to route work to specialist teams
* The current team if the team has the role to be able to complete the step
* The next available team on a round robin basis. The system will attempt to assign the transaction to teams flagged as suitable for auto-allocation of work first. If no auto-allocate teams are found then the standard teams are used
* If there are no teams available then the system will assign the transaction to the default team for the product

For every transaction received from SWIFT or Gateway, the behalf of branch is either set in the transaction or determined from the SWIFT header. This branch is then used to determine the work allocation as detailed above. If a team cannot be determined, the transaction will be assigned to the SWIFT repair or Gateway repair team.

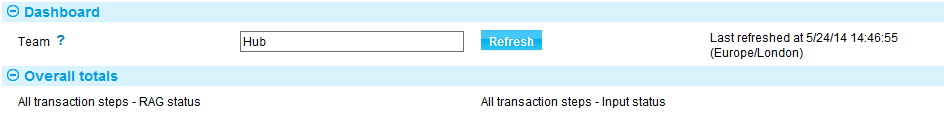
A team can be specified on an incoming gateway message. Once the event is created, the system will attempt to move it to the step defined when creating from Gateway defined in System Tailoring Event Steps. If the Team cannot perform the step then the system uses workflow allocation as detailed above to determine an available team.

For information on setting up Event/Team mappings see the System Tailoring User Guide – Trade Innovation.

## The Dashboard

The Dashboard provides an enhanced user interface for users to monitor work in progress and manage workflow so that target times are met. The colour coding associated with SLA statuses allows the users to see at a glance which events are at risk of failing to meet their target completion times. In addition, the Dashboard allows users to view the workload of each of the teams to which they belong. This workload is calculated using the average times entered for the completion of events.

The Dashboard is illustrated below:





How the Dashboard behaves depends on whether the user is a normal user or a supervisor, as explained in the following sections. The charts showing statistical information are displayed only if the user has the capability ViewSLACharts assigned to them.

1. Access to the dashboard is controlled by the Security Capability, “ViewDashboard”. If a user is assigned this Capability, they have access to view the Cross Zone Dashboard.

The Dashboard is presented for the user’s default team if defined. Otherwise, a team must be entered before the information is displayed.

Users can limit what part of their workload is displayed for them to continue working on (for example to work for a specific team or for a specific user role assigned to them).

The user can enter a specific team to see the work outstanding for the user in that team or filter by a specific role. If the user changes team, then it loads the roles for the new team. The Work in Progress window displays all outstanding work that matches the selected roles and team.

Note that work deadline dates and times are shown in the user’s location time zone.

When a work in progress item is selected, the event is opened in the appropriate mode for the user’s role. For example:

* The user may have no roles assigned within the team currently assigned to the event so they can only View the event
* The user may have a role that allows them to Authorise but the event is currently at the Input step so they can only View the event
* The user may have a role that allows them to Input and the event is currently at the Input step so they can fully carry out the input stage of the event

### Displaying Transactions Cross Zone

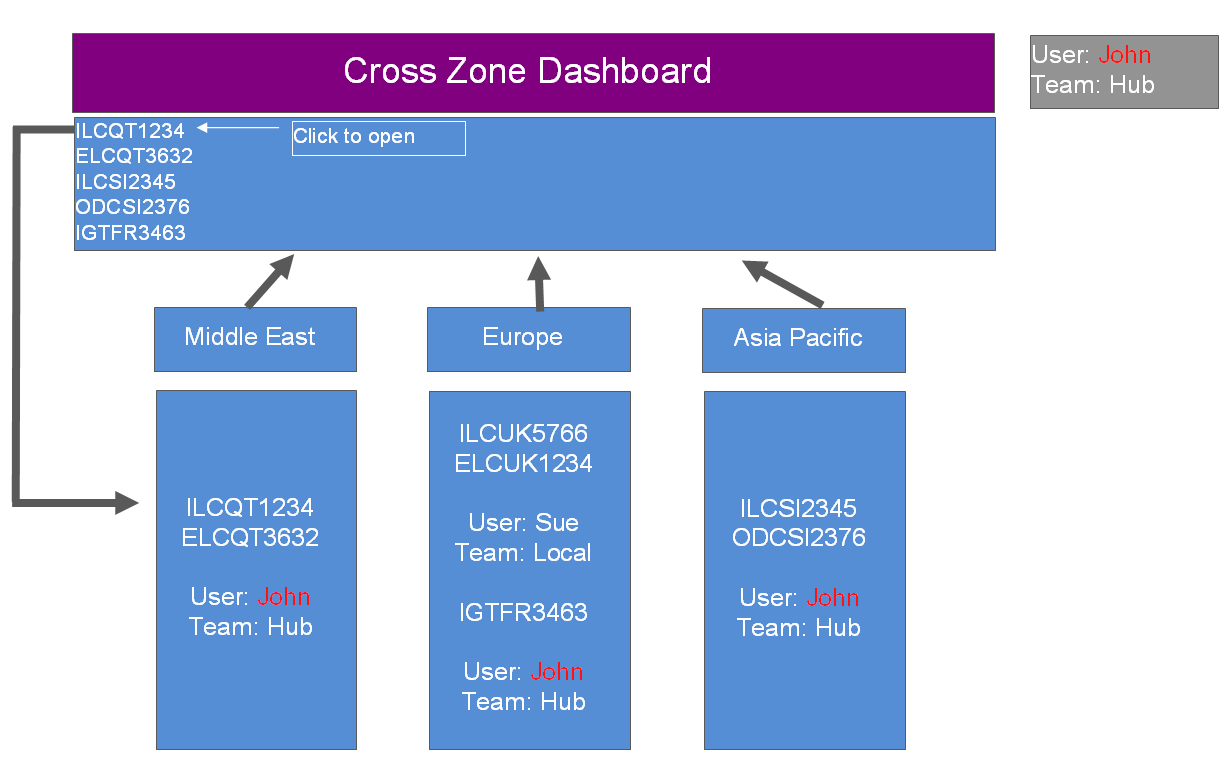
Where the bank has deployed the system across more than one zone it is possible for the dashboard to display work for teams that cover multiple zones, for example in a centralised processing hub.

If a user is only permitted to access one zone then they do not have the capability to view work across zones. The field ‘Local Zone Only’ is not shown on the Work In Progress section of the dashboard. Where a user has access across zones then this field is available so that they can restrict the work in progress view to the zone they are currently in. The default is to select work that meets the selection criteria across all the zones they have access to up to the maximum row limit of transactions that can be displayed in the dashboard.

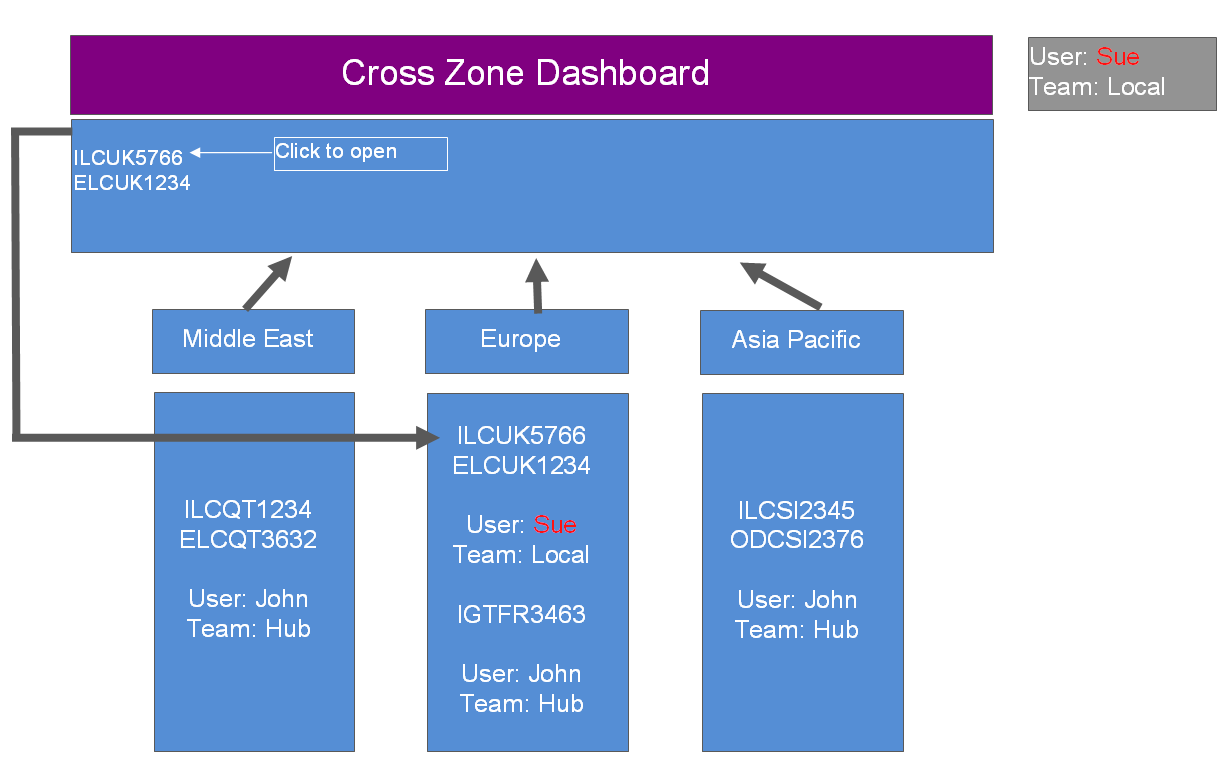
Transaction event step start and finish times are shown in the user’s location time zone. Where the user is a hub team member continuing an event on behalf of another team, the times as at the behalf of branch can be shown.

The following diagrams illustrate the different modes of working:

The first diagram shows the work that a supervisor that has multiple zone access would see on their dashboard as work assigned to them.



The second diagram shows the work that a user that only has single zone access would see on the dashboard as work assigned to them.



If a transaction selected to work on is in a different zone then the user is automatically transferred to that zone to work on that event.

1. The master browser only operates in zone and does not have cross zone capability.

For a full description of the Dashboard see the SLA Dashboard User Guide – Trade Innovation.

### Normal Users

When a normal user opens the Dashboard their default team, if they have one, is used to retrieve those transactions on which they can work in the usual way (see page 6).

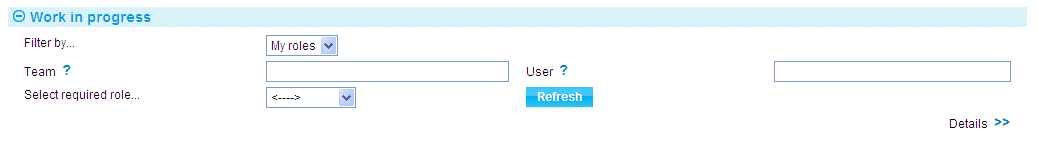
If they have no default team, they must choose one in the Dashboard pane.

The team selected here governs what transactions are included in the charts in the Dashboard and, initially, also what is shown in the Work in Progress pane.

If they change their team, the charts and the Work In Progress panes also change to reflect the new data set.

The Work In Progress pane lists all the transactions assigned to the user’s teams and to the relevant sub-teams that are within the user’s transaction and enquiry scope. This pane provides indicators - the colour, and the value in the Status column - showing where each transaction is in relation to its target completion time.

Normal users can limit what transactions are included in the Work in Progress pane in various ways, using the fields in that pane.



* They can apply a team profile or a personal profile. These profiles have pre-set filters for (for example) branch, product and business area
* They can select to list only those transactions assigned to a specific team or user (the drop-down field governing which team they can select is limited to the teams to which they belong and their sub-teams; and the drop-down field governing which users they can select is limited to users assigned to those teams)
* They can select a specific user role from those assigned to them across the relevant teams; the Work In Progress pane then lists only those transactions they can process using that user role

The settings of these filter fields changes what is displayed in the Work In Progress pane, but do not affect what is shown in the chart displays. The chart displays always show data for the team (and relevant sub-teams) for the team displayed in the Team field in the Dashboard window.

### Supervisors

For users who are flagged as supervisors, the Dashboard behaves in the same way as for normal users but with the following exceptions:

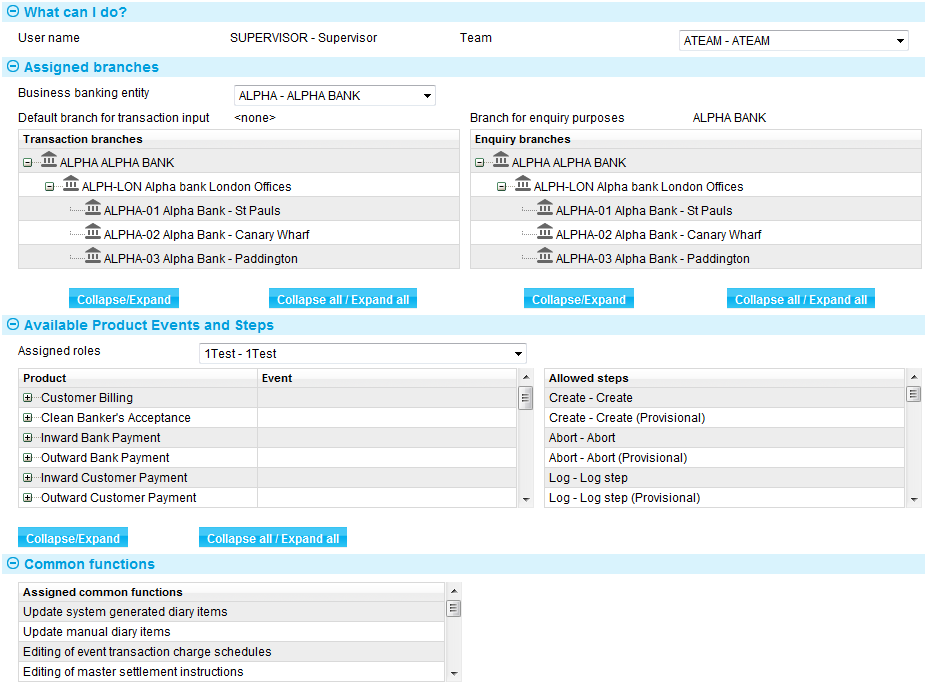
* Supervisors can use the filter fields in the Work in Progress pane to show transactions for all user roles, not just the user roles assigned to them. This allows them to see the workload of all users across all their teams
* Supervisors can reassign transactions to a different user. (They need the capability ReassignTxns to be able to do this; and the team or user that they assign a transaction to must have the appropriate user role to be able to continue work on the transaction)

The charts allow supervisors to see the current levels of work assigned to their teams, and to reassign work to balance out loads. See the SLA Dashboard User Guide – Trade Innovation for information on how they do this.

## What Can I Do?

The system's What Can I Do? link displays a window that shows which branches a user is assigned to, and allows them to see which teams they belong to and what user roles they have within them.

The window that opens shows the user name the user used to log in. The team defaults to that assigned to the user’s profile. If the user has no default team assigned to their profile, there will be no information to display. The user has to select a team to see what he can do in that team context.



The Teams drop-down field lists all the teams the user is assigned to. Once the user selects a team the Business Banking Entity drop-down list is populated with the business/banking entities that the team is associated with. Once the user selects a team and business banking entity the columns in the Assigned Branches pane show information for the selected combination:

* The Transaction Branches column lists all the branches whose transactions the user can work on
* The Enquiry Branches column lists all the branches whose transactions the user can view but not work on

The Available Products Events and Steps pane includes a drop-down list of the user roles available to the user for the selected combination. When the user selects a user role, the Available Product Events and Steps pane shows information on the products and events that user role allows them to work with, and at what step. The Common Functions pane lists the master-level tasks the user role allows them to perform.

This same functionality is available to security officers, but is extended to allow them to view information for all users.

The What Can I Do? Feature can be useful during initial implementation to ensure that the team based security model has been implemented in line with the bank’s expectations.

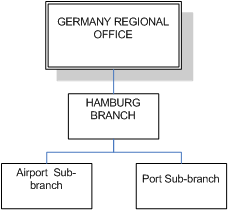
# Appendix A Global Processing and Different Banking Models

The approach to global processing is flexible enough to support many banking models. This appendix provides examples of its use with three different banking models:

* A small single-zone bank
* A larger single-zone bank
* A multi-zone, multi-country bank

## Small Single-zone Banks

For banks that operate in a single zone and have no need for cross-zone global processing the system nevertheless offers functional advantages.

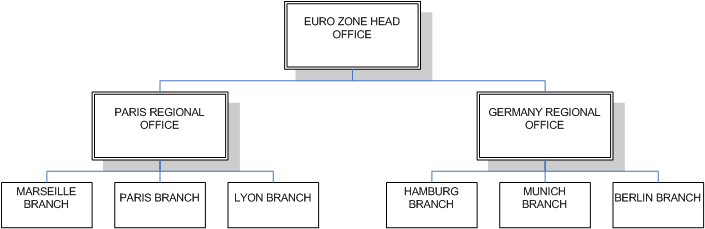


Where all of a bank's branches are in the same time zone, the hierarchical model can be employed to set up common processing parameters and common postings sets and customer documentation sets in one place, and then cascade them down to branches.

Any regional differences (for example, additional legal clauses where a branch operates under a different system of law, or different branding on customer documents) can be catered for, as can any branches used for outsourced services. Different opening hours can also be set up where there are regional differences, with overnight processing run separately for individual branches.

## Large Single-zone Bank

A larger single-zone bank may have additional tiers consisting of regional processing centres.

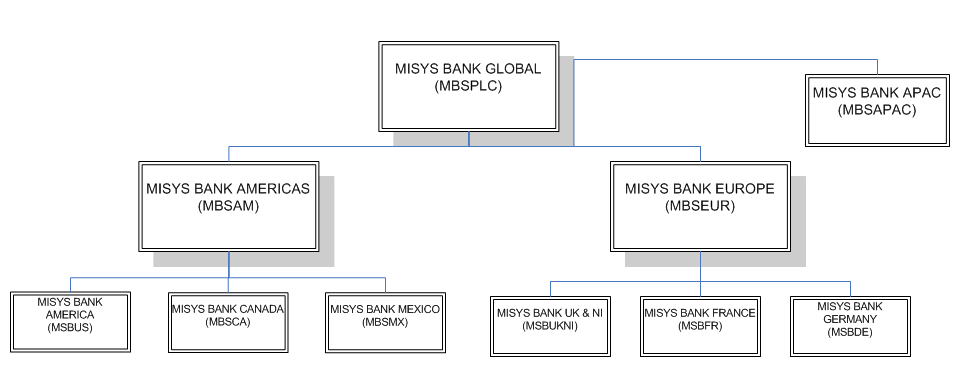


In the larger single-zone bank model the branches might be used as service centres, for example for scanning in documents or performing basic logging services for requests that are then passed on to the regional centre.

The regional centre would house the bank's trade services specialists, and these would carry out the detailed processing of requests, including the creation of charges, postings and customer documents.

In both models, the head office is also likely to be able to carry out transaction processing, and would need to have an overall view of the bank's workload and productivity. A head office management team would allow its users to see the workload of the entire bank, whilst additional teams set up at regional level would allow their users to see the workload of the regional office and all the branches below it.

## Multi-zone, Multi-country Banks

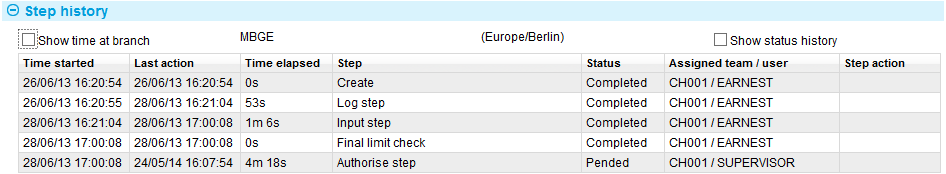


Multi-zone, multi-country banks benefit fully from the flexibility of the global processing functionality. They have the same ability as smaller banks to distribute work across branches and regional offices, and to allow the head office and regional offices to gain an overall view of workloads.

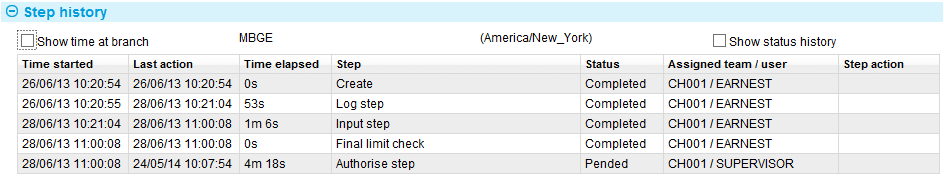
For banks operating across time zones, the system has the advantage of allowing teams to be created in one time zone and linked to branches in another. For example, a team geographically located in MISYS BANK AMERICA can be linked to the bank's MISYS BANK GERMANY branch; transactions belonging to the German branch can then continue to be progressed outside of Germany's branch's working day, supporting 'follow-the-sun' workflow.

1. All timestamps are shown according to the user’s time-zone location. A user located in America will see the step actions made in Germany in their own local US time.

Transaction as seen by German team:



Same transaction as seen by US team:

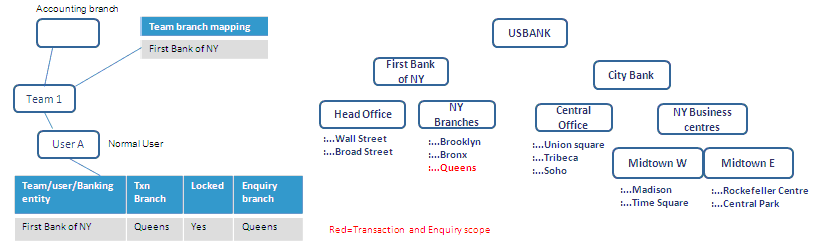


The ability of banks to set up parameters governing (for example) how products are processed at branch level supports cross-country operations, as does their ability to set base currency per main banking entity (general ledger). This ensures that a transaction is processed in the same way, using the same parameter set, wherever it is worked on.

# Appendix B How Transaction Branch and Enquiry Branch Control What Transactions a User can Access

This appendix provides examples showing how transaction branch and enquiry branch affect what transactions (their scope) a user can access within a team hierarchy.

## Example 1 User locked to transaction branch and transaction branch = enquiry branch



The team has all branches for First Bank of NY mapped to it; however, the user is locked to Queens for transaction and enquiries. The user can only enter or select Queens as the Behalf Of branch.

## Example 2 User locked to transaction branch and enquiry branch > transaction branch

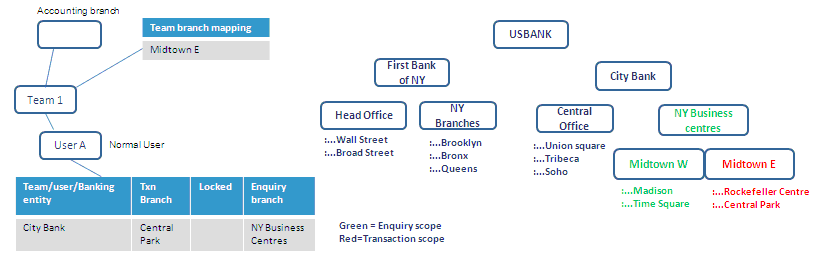


The team has all branches for First Bank of NY mapped to it.

The user is locked to Queens for transaction input. The user can enter and complete transactions for Queens.

Browsing on Behalf Of branch will show branches in the NY Branches hierarchy within the user's enquiry scope. The user can enter transactions but not complete for branches - NY Branches, Brooklyn, Bronx.

## Example 3 Transaction branch entered but not locked. Enquiry branch entered



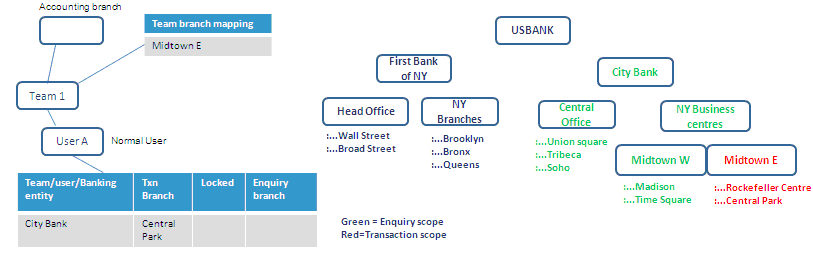
The user has a default transaction branch of Central Park.

Browsing on Behalf Of branch will show branches in the NY Business centres hierarchy within the user's enquiry scope.

The user can enter transactions and complete transactions for any branch in Midtown East.

The user can enter transaction but not complete them for any other branch in the NY Business centres hierarchy.

## Example 4 Transaction branch entered but not locked. Enquiry branch not entered



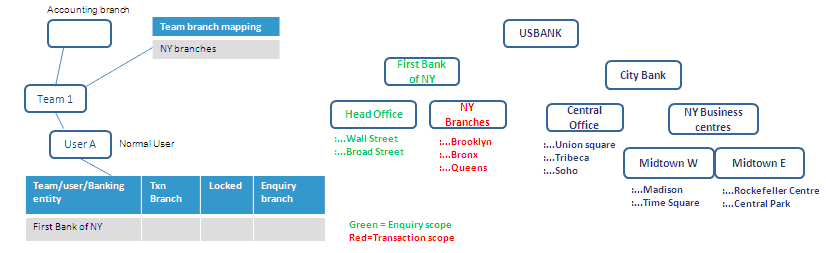
The user has a default transaction branch of Central Park.

Browsing on Behalf Of branch will only show branches in the City Bank hierarchy.

The user can enter transactions and complete transactions for any branch in Midtown East.

The user can enter transactions but not complete them for any other branch in the City Bank hierarchy.

## Example 5 No transaction branch entered and no enquiry branch entered



The user has no default transaction branch. The user can select any branch within the NY Branches hierarchy (that is, branches associated with the team).

Browsing on Behalf Of branch will only show branches in the First Bank of NY hierarchy.

The user can enter transactions and complete transactions for any branch in NY Branches.

The user can enter transactions but not complete them for any other branch in the First Bank of NY hierarchy.

# Appendix C Global Processing Bank - Example Set-up

This appendix is intended to illustrate a model global bank set up using the global processing functionality. It implements the following key elements:

* Users
* Branch structure
* Event groupings
* User roles
* Processing teams
* Roles, users and branches assigned to teams
* Parameter settings
* Links to external services (for example limit checking and general ledger)
* Links to customer information files
* Daily processing cycles

This example can be used:

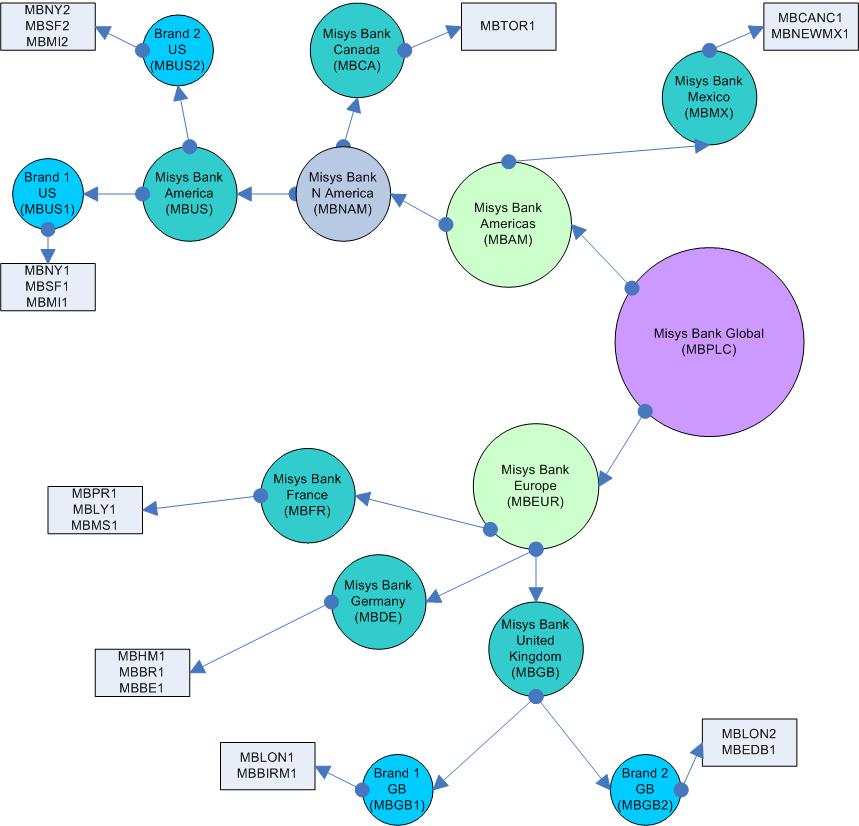
* To illustrate the key concepts under pinning global processing
* As a model that can be used during implementation to cover the main areas that require analysis and build

## Scope

This example illustrates a hub and spoke model across the Americas region. Although a Europe hub is illustrated in some places, the detailed worked example concentrates on showing how the Americas business is set up for Letters of Credit.

## Geographical Bank Structure

* One bank operating in multiple geographies
* Finastra Bank brand plus other in country brands in certain geographies



The identifiers in brackets equate to Branch IDs and will be illustrated in more detail later in this Appendix.

## Zone Structure

For this example the zone structure to support this would be:

* US in one zone
* Europe in another zone

However, they could equally operate within one zone. The choice depends on many factors and would typically be discussed with the bank when establishing their target operating model (TOM). Determining factors could be:

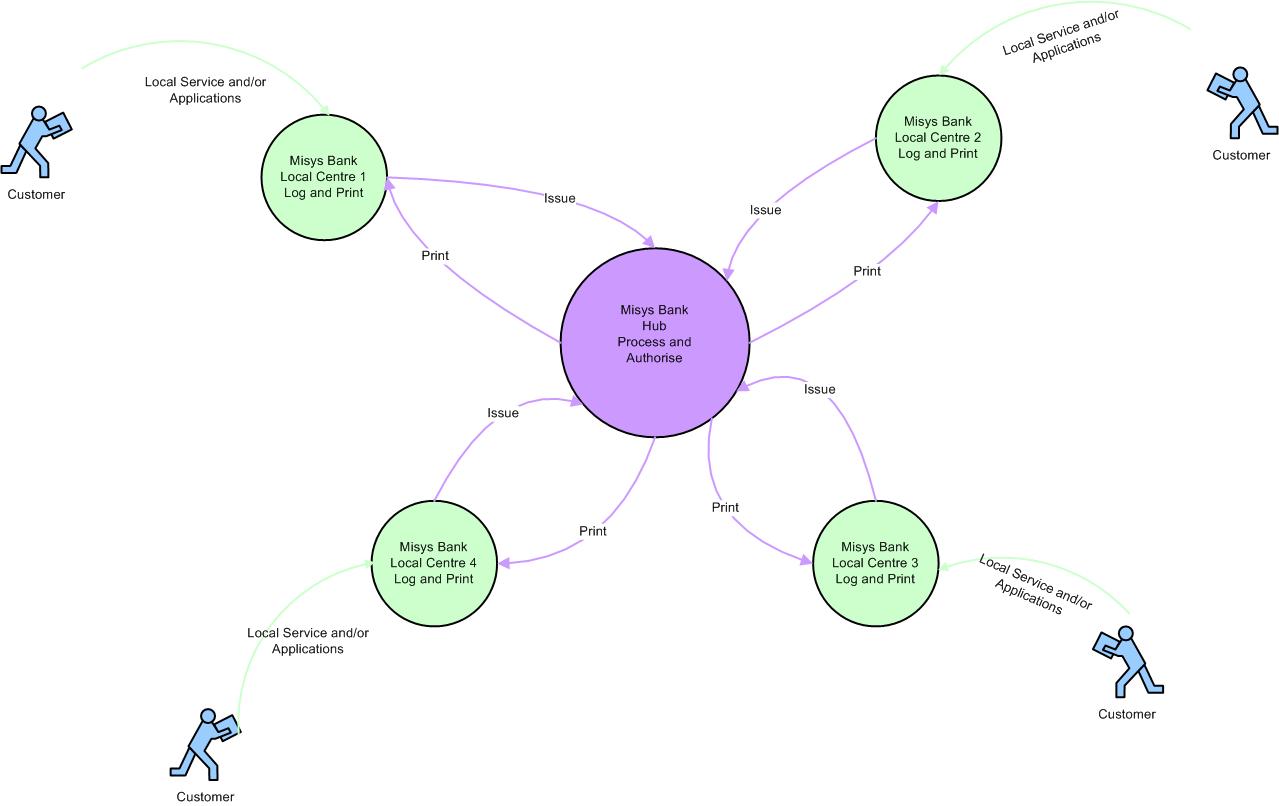
* Legal constraints about segregating a country’s data from others
* Ease of rollout
* Commonality of operating models
* Complexity of interfacing (managing all interfaces out of one zone may be harder to manage)
* Approach to rolling out the implementation
* Volumes and scalability
* Resilience (single point of failure, single point of upgrade)
* Other factors may also be a consideration and this list may be expanded in later versions of this guide based on further global processing implementation experience

The example in this appendix illustrates:

* Bank structure
* Different base currencies
* Different external system interfacing
* Customer information files
* Shared and local parameters
* Transactional security
* The capabilities for hub and spoke operation through the use of teams
* Different end of day cycles

## Hub and Spoke Operation

### Two Tier Hub and Spoke



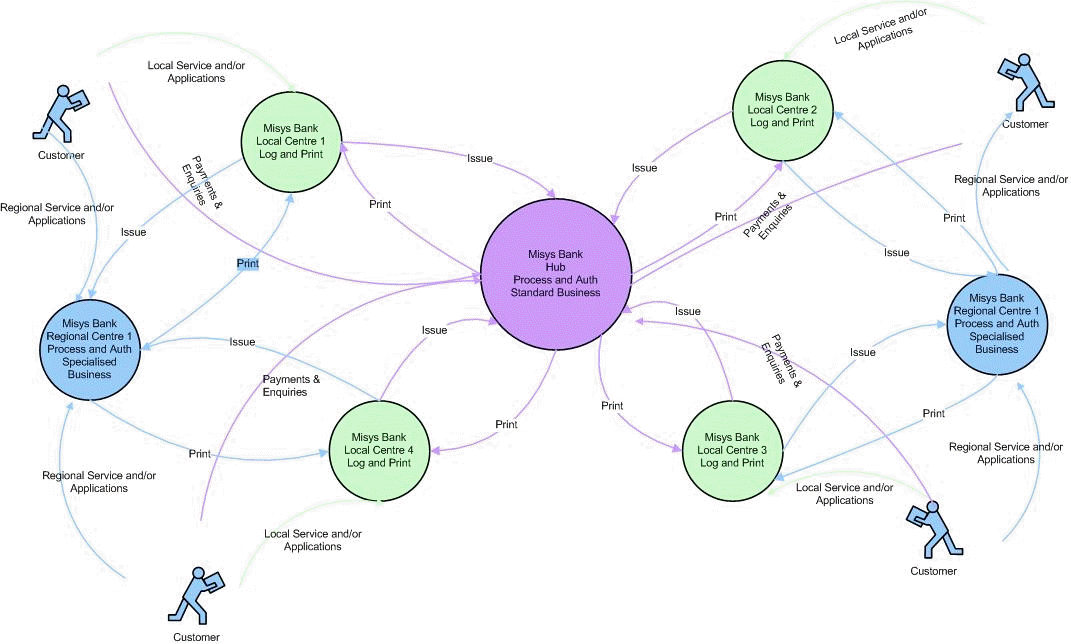
* Two levels of customer service centre:
* Spoke (green) – a local walk in branch that would provide a logging and printing function. This is typically the branch that will actually maintain or have responsibility for the client accounts and / or will maintain the ‘client relationship.’
* Central /Hub (purple) – Typically will undertake the actual processing of ‘standard or business as usual’ transactions on behalf of the spokes. Port of call for any follow on enquiries or payments. May however also be a ‘specialist’ centre focusing on a particular business stream such as oil credits.

Work would be routed to available teams at the appropriate time in the workflow based on either:

* Specific teams defined in Event team mappings
* ‘Round robin’ basis to teams capable of performing the next step (based on branches and roles assigned to the team )

### Alternative Three Tier Hub and Spoke

Whilst not covered in this Appendix it is possible to have tiered service levels as shown below:



* Three levels of customer service centre:
* Spoke (green) – a local walk in branch that would provide a logging and printing function. This is typically the branch that will actually maintain or have responsibility for the client accounts and / or will maintain the ‘client relationship.’
* Regional (blue) – Typically the ‘controlling’ branch for the spokes. Will usually make the decisions around pricing structures, facilities and limits and service levels. May also provide specialist / dedicated services for premium clients, or handle ‘high value’ transactions.
* Central /Hub (purple) – Typically will undertake the actual processing of ‘standard or business as usual’ transactions on behalf of the spokes. Port of call for any follow on enquiries or payments. May however also be a ‘specialist’ centre focusing on a particular business stream such as oil credits.

## Users

The users listed in the tables below are to support processing at the local logging level. These are the spoke centres where a customer may walk in for service. The tables show the user ID, whether the user acts as a supervisor (for local users none are supervisors) and also the default team they are associated with.

#### Americas – Local Users

| User | Supervisor | Team |
| --- | --- | --- |
| CAUSER1 |  | Local |
| MXUSER1 |  | Local |
| USUSER1 |  | Local |
| US1USER1 |  | Local |
| NY1USER1 |  | Local |
| SF1USER1 |  | Local |
| MI1USER1 |  | Local |
| US2USER1 |  | Local |
| NY2USER1 |  | Local |
| SF2USER1 |  | Local |
| MI2USER1 |  | Local |

The following users are to support processing at the central hub level. As they can operate across multiple teams no default is given.

#### Overall Hub

|  |  |  |
| --- | --- | --- |
| User | Supervisor | Team |
| SUPERVISOR | Yes |  |
| HUBSUPER1 | Yes |  |
| HUBSUPER2 | Yes |  |

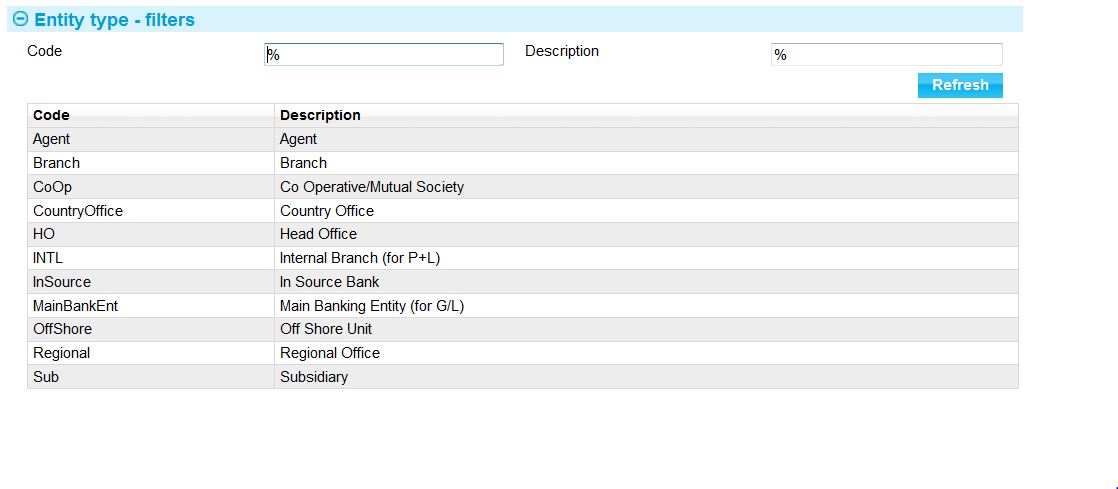
#### LCs

|  |  |  |
| --- | --- | --- |
| User | Supervisor | Team |
| HUBUSER1 |  |  |
| HUBUSER2 |  |  |
| HUBSUPER3 | Yes |  |
| HUBUSER3 |  |  |
| HUBUSER4 |  |  |
| HUBSUPER4 | Yes |  |
| HUBUSER5 |  |  |
| HUBUSER6 |  |  |
| HUBSUPER5 | Yes |  |

## Branch Entity Types

These represent the typical entity types that may (optionally) be associated with a branch. You can tailor these to meet your own requirements.

| ID | Description |
| --- | --- |
| HO | Head Office |
| Branch | Branch |
| Agent | Agent |
| InSource | In Source/White Label Bank |
| Regional | Regional Office |
| Sub | Subsidiary |
| OffShore | Off Shore Unit |
| CountryOffice | Country Office |
| CoOp | Co Operative/Mutual Society |
| MainBankEnt | Main Banking Entity (for G/L Purposes) |
| INTL | Internal Branch (for P+L) |



## Branches

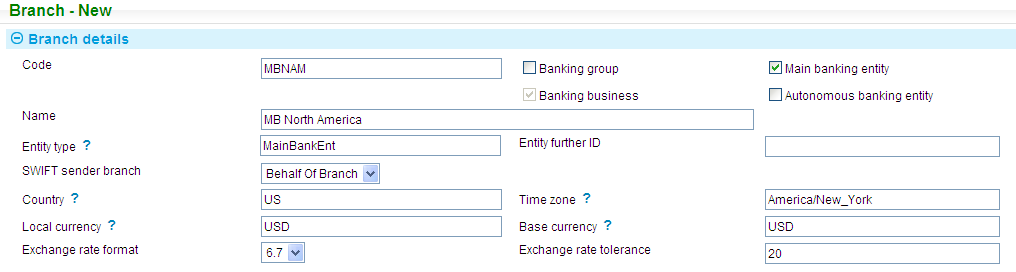
The branch IDs and relationship to parents are shown in the Geographical Bank Structure diagram (see page 76).

## Geographical Branch Structure

Within the Americas there are two back office systems used. One supports the business of Mexico (branches MBMX and below) the other supports the US and Canada (branches MBNAM and below).

Each branch declared within the system may have certain characteristics (flags) associated with it. These denote any special behaviour where the branch is not purely a transactional branch. These are described in detail in Chapter 3 and in the Static Data Guide.

The following are important flags when deciding on the processing model for Global Processing:



The diagram below shows the flags overlaid on the branch model on page 7676. It also shows that there is a single customer information file accessible by all areas of the bank:

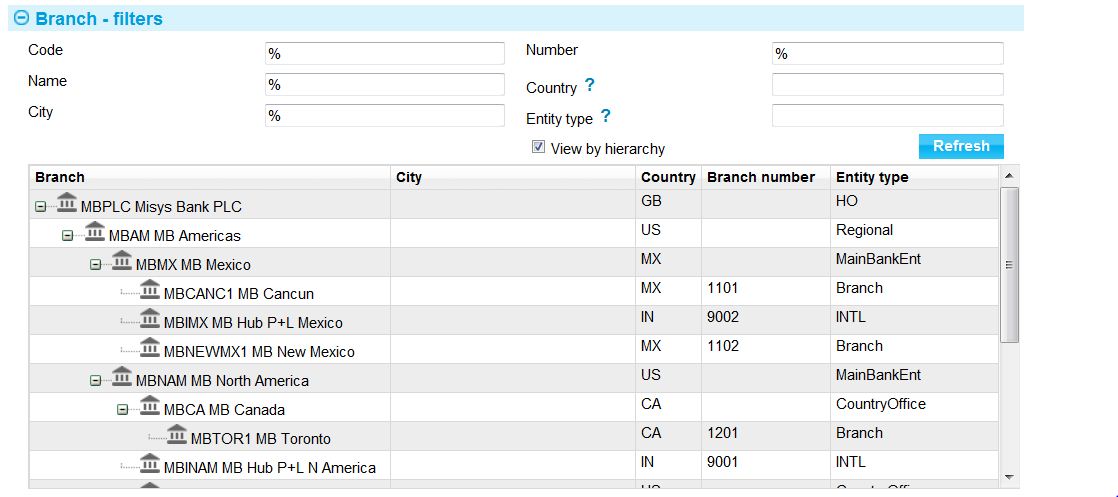


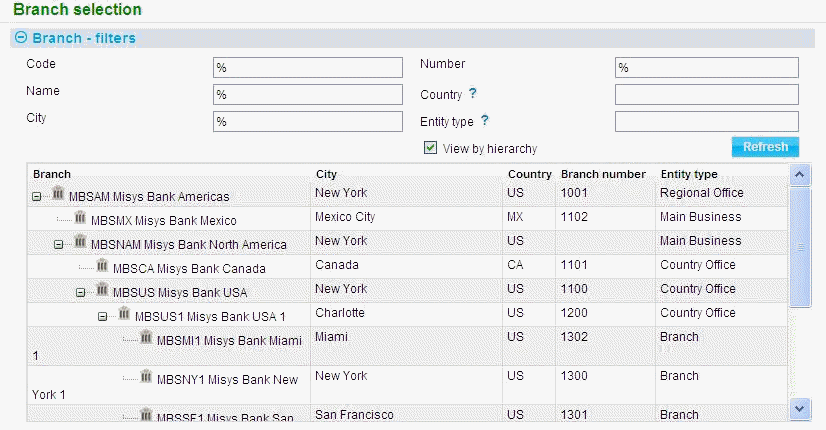
Other settings related to a branch are shown in the table below. The colours follow those shown in the diagram above in order to illustrate the hierarchical levels (purple being the highest level (zone) and grey being the lowest (transactional branch)).

MBE stands for Main Banking Entity.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Branch ID | Parent | Banking Group | Banking Business | MBE | Non Txn | Base Currency | Entity Type |
| MBPLC |  | Yes | Yes |  | Yes |  | Head Office |
| MBAM | MBPLC |  | Yes |  | Yes |  | Regional |
| MBMX | MBAM |  | Yes | Yes | Yes | MXN | Main Bank Ent |
| MBCANC1 | MBMX |  |  |  |  |  | Branch |
| MBIMX | MBMX |  |  |  |  |  | InternalBranch |
| MBNEWMX1 | MBMX |  |  |  |  |  | Branch |
| MBNAM | MBAM |  | Yes | Yes | Yes | USD | Main Bank Ent |
| MBCA | MBNAM |  |  |  | Yes |  | CountryOffice |
| MBTOR1 | MBCA |  |  |  |  |  | Branch |
| MBINAM | MBNAM |  |  |  |  |  | InternalBranch |
| MBUS | MBNAM |  |  |  | Yes |  | CountryOffice |
| MBUS1 | MBUS |  |  |  | Yes |  | Sub |
| MBMI1 | MBUS1 |  |  |  |  |  | Branch |
| MBNY1 | MBUS1 |  |  |  |  |  | Branch |
| MBSF1 | MBUS1 |  |  |  |  |  | Branch |
| MBUS2 | MBUS |  |  |  | Yes |  | Sub |
| MBMI2 | MBUS2 |  |  |  |  |  | Branch |
| MBNY2 | MBUS2 |  |  |  |  |  | Branch |
| MBSF2 | MBUS2 |  |  |  |  |  | Branch |

The branches of type ‘InternalBranch’ i.e. MBINAM and MBIMX are internal branches which can be used for the sharing of charges and interest between a processing hub and a local team. When setting the accounting branch on a team it is this that is selected (see later under Teams). This is only necessary if your bank apportions some of the processing income between the centres doing the work. Otherwise all income would be attributed to the ‘Behalf of branch’.



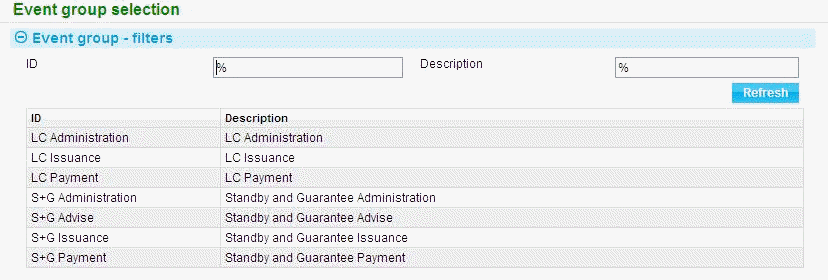


## Event Groupings

The following groupings will be used to set up the user roles for members of various teams handling import letters of credit:

| Group Name | Product | Event |
| --- | --- | --- |
| LC Issuance | Import LC | Pre Advise |
|  |  | Issue |
|  |  | Receive Acknowledgement |
|  |  | Amend |
|  |  | Beneficiary Response to Amend |
|  |  | Assignment of Proceeds |
|  |  | Transfer |
|  |  | Transfer Receive Acknowledgement |
|  |  | Amend Transfer |
|  |  | Beneficiary Response to Amend Transfer |
|  |  | Issue Shipping Guarantee |
|  |  | Revolve |
|  |  | Revolve Notice |
|  |  | Correspondence |
|  |  | Transfer Correspondence |
|  |  |  |
| LC Payment | Import LC | Claim Received |
|  |  | Outstanding Claim |
|  |  | Discount |
|  |  | Repay Financing |
|  |  | Pay Charge |
|  |  |  |
| LC Administration | Import LC | Adjust |
|  |  | Expire |
|  |  | Book Off |
|  |  | Book Keeping |
|  |  | Cancel |
|  |  | Beneficiary Response to Cancel |
|  |  | Correspondence |
|  |  | Link Correspondence |
|  |  | Maintain Liability |
|  |  | Maintain Charge |
|  |  | Adjust Transfer |
|  |  | Cancel Transfer |
|  |  | Beneficiary Response to Transfer Cancel |
|  |  | Transfer Correspondence |
|  |  | Transfer Expire |
|  |  | Cancel Reimbursement |

Other similar groupings would be required for other product areas.

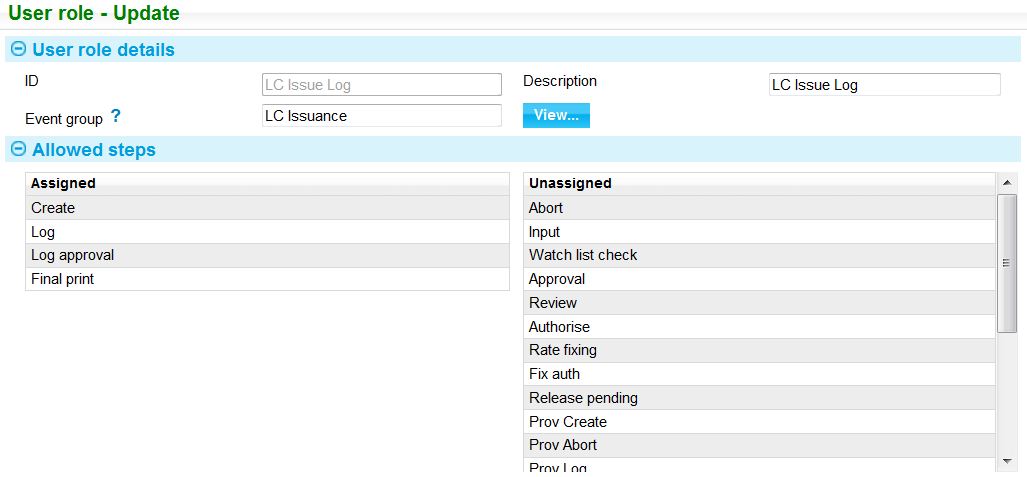


## User Roles

For the purposes of simplicity the roles only cover import letters of credit.

There is a role for users in branches that enables them to do logging and printing, with other roles in the process being given to the central users/teams.

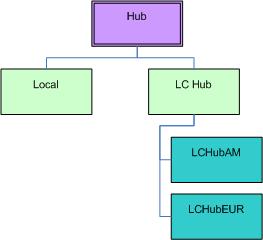
Each role includes every common master level task (for example ‘Edit Notes’) unless stated otherwise.

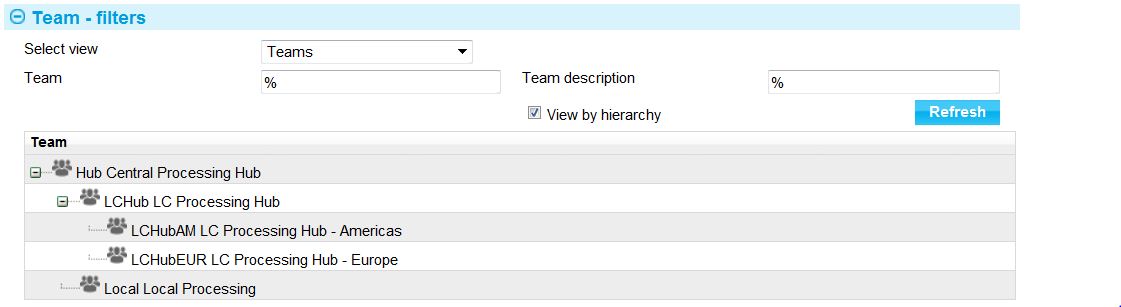


|  |  |  |
| --- | --- | --- |
| Role ID | Event Grouping | Step |
| LC Issue Log | LC Issuance | Log  Log Approval  Final Print |
| LC Issue Input | LC Issuance | Input |
| LC Issue Auth | LC Issuance | All bar Log, Log Approval and Input |
| LC Payment Log | LC Payment | Log  Log Approval  Final Print |
| LC Payment Input | LC Payment | Log  Log Approval  Input |
| LC Payment Auth | LC Payment | All bar Log, Log Approval and Input |
| LC Admin Log | LC Administration | Log  Log Approval  Final Print |
| LC Admin Input | LC Administration | Input |
| LC Admin Auth | LC Administration | All bar Log, Log Approval and Input |

## Team Hierarchy

The Hub team at the top of the hierarchy enables certain users to have an overall view of the bank’s business at both a spoke and a hub level. Below that, users can be allocated to a local processing (spoke) team or to various sub teams within the processing hub (or to both). This model has a hub handling LCs which is further split between the Americas and Europe.





1. Where a team is intended to support cross zone work it is important to note that you must declare that Team ID in all zones where global visibility is required. Within each zone the users and branches visible to that zone can then be mapped into the team. The cross zone dashboard will then consolidate and report on all of the outstanding work for the team/users across all zones where that team is declared.

### Local Teams

At branch/agent level users are only able to receive documents, log them and then print out documents locally at the end of each transaction process. It is possible to create one team covering all the local branches/agents. Where a user is only allowed to work on items for their own branch then the security within the team can lock the user down to that branch and its transactions. This means it is not necessary to create multiple teams at this level. Of course if, required this can be done as well, perhaps to reflect geographical regions or countries or particular time zones.

|  |  |
| --- | --- |
| Team ID | Parent |
| Local | Hub |

The local team has the Hub team as a parent. This enables a user such as a supervisor or manager in the Hub team to have visibility over the work of all the teams locally and within the central hub.

### Central Teams

A centralised team provides a processing hub providing a service across the globe.

For this example the global hub will have one team covering Import LCs with two sub teams to cover each region. There is also an overall hub team that monitors all the workloads of the sub-teams.

|  |  |
| --- | --- |
| Team ID | Parent |
| Hub |  |
| LCHub | Hub |
| LCHubAM | LCHub |
| LCHubEUR | LCHub |

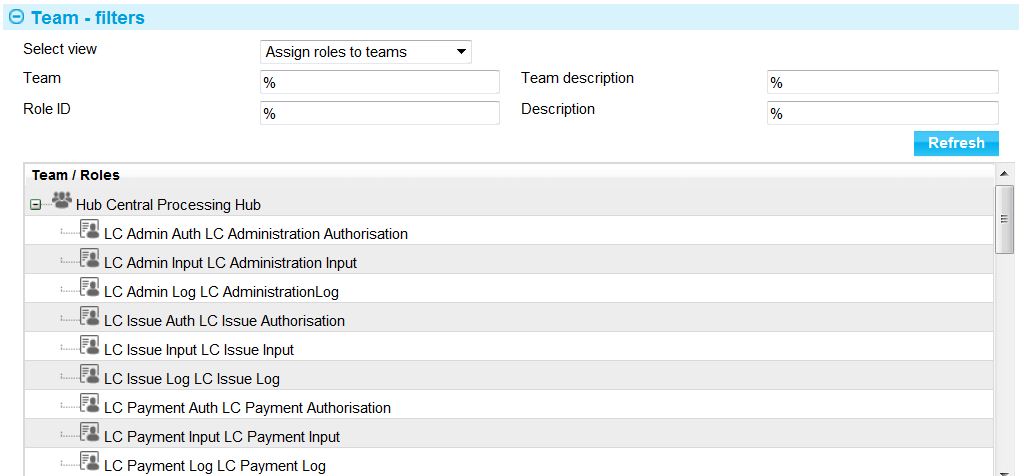
## Team Security

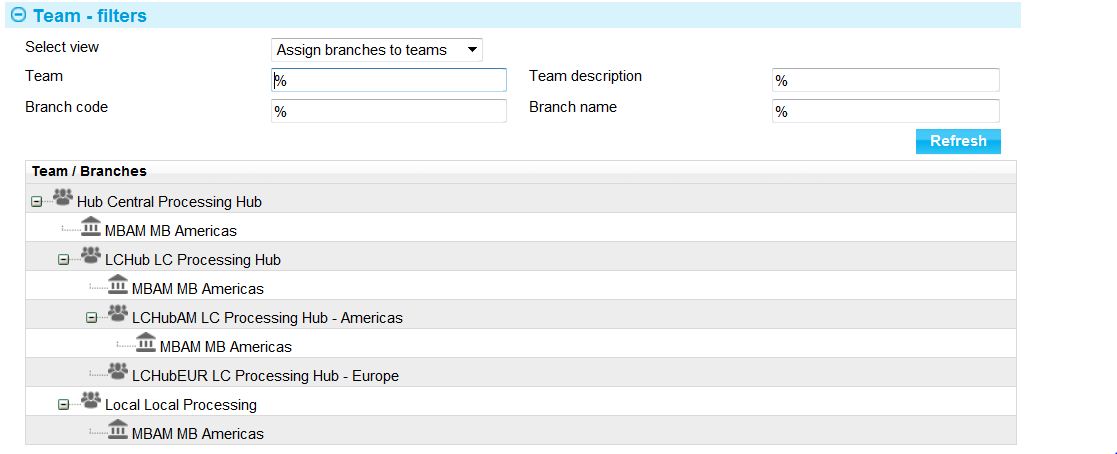
### Roles and Branch Assignments

Part of the team security assigns the valid roles to each team and also the branches covered by the team.

1. Note that any child branches below the branches identified are automatically deemed to be covered by that team.
2. So, in this example all teams only have the MBAM branch listed which is the highest level branch so they can access and process for all branches in the hierarchy for the Americas (subject to individual user restrictions on which branches they can see and work with).

This example only covers the America and Europe parts of the hierarchy:





An accounting branch can be assigned for each Main Banking Entity identified within the branch scope mapped to the team. So in this example there are two branches marked as Main Banking Entities under the overall MBAM branch. The branches used for accounting are the internal branches used to record any fee splitting between the processing team and the spoke where required i.e. MBIMX (for the Mexico general ledger) and MBINAM (for the North America general ledger).

|  |  |  |  |
| --- | --- | --- | --- |
| Team | Roles | Branches | Accounting Branches |
| Local | LC Issue Log  LC Payment Log | MBAM | MBIMX  MBINAM |
| Hub | All Events/Steps  (so they can see everything) | MBAM | MBIMX  MBINAM |
| LCHub | LC Issue Log  LC Issue Input  LC Issue Auth  LC Payment Log  LC Payment Input  LC Payment Auth  LC Admin Input  LC Admin Auth | MBAM | MBIMX  MBINAM |
| LCHubAM | LC Issue Log  LC Issue Input  LC Issue Auth  LC Payment Log  LC Payment Input  LC Payment Auth  LC Admin Input  LC Admin Auth | MBAM | MBIMX  MBINAM |
| LCHubEUR | LC Issue Log  LC Issue Input  LC Issue Auth  LC Payment Log  LC Payment Input  LC Payment Auth  LC Admin Input  LC Admin Auth |  | MBIMX  MBINAM |

### User Role Assignments

Users are allocated to teams and then have roles assigned to them. This defines what products, events and steps they can work with.

Users in the local team can only do the logging tasks.

Users in the other hub teams can do all tasks.

In this example the user SUPERVISOR is a member of every team with appropriate roles. In reality this is unlikely to be the case but is used here in this example as it enables a single sign on user to be able to work with all transactions in all teams:

|  |  |  |
| --- | --- | --- |
| Team | User | Roles |
| Local | A user is required per branch. All have the same roles i.e. they are able to log, log approve and print.  CAUSER1  TORUSER1  MXUSER1  USUSER1  US1USER1  NY1USER1  SF1USER1  MI1USER1  US2USER1  NY2USER1  SF2USER1  MI2USER1  SUPERVISOR | LC Issue Log  LC Payment Log |
| Hub | HUBSUPER1 | All roles |
|  | HUBSUPER2 |  |
|  | SUPERVISOR |  |
|  |  |  |
| LCHub | HUBUSER1  HUBUSER2  HUBSUPER3  SUPERVISOR | All roles |
|  |  |  |
| LCHubAM | HUBUSER3  HUBUSER4  HUBSUPER4  SUPERVISOR | All roles |
| LCHubEUR | HUBUSER5  HUBUSER6  HUBSUPER5  SUPERVISOR | All roles |

### User Branch Security

This defines what branches users can work with and enquire on. For example user MXUSER1 is locked to logging work only in the Cancun branch (MBCANC1) but can enquire on the work throughout Mexico (MBMX).

#### Local Team

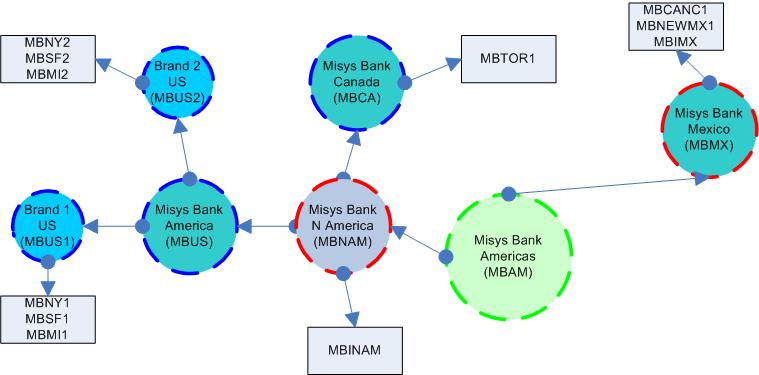
Contains two main banking entities enabling local logging functions across all the bank’s businesses. Some users are locked to a single branch for transaction entry and in this way many users in different branches can share one team but still have limited visibility/capability within the bank’s organisation. The enquiry branch enables them to have a wider branch visibility (scope) where required, for example to see the work of another branch within the same country but only for enquiry purposes.

The Supervisor can enter transactions across both the main banking entities and hence has two transaction branches defined, one per MBE. Other users are confined to one or other MBE by use of the ‘Exclude’ flag to completely block them out of access to branches within that MBE.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Team | Main Banking Entity | User | Transaction Branch | Locked? | Enquiry Branch |
| Local | MBNAM | CAUSER1  TORUSER1  USUSER1  US1USER1  NY1USER1  SF1USER1  MI1USER1  US2USER1  NY2USER1  SF2USER1  MI2USER1  SUPERVISOR | MBTOR1  MBTOR1  MBNY1  MBSF1  MBMI1  MBNY2  MBSF2  MBMI2  MBMI1 | Yes  Yes  No  No  Yes  Yes  Yes  No  Yes  Yes  Yes  No | MBCA  MBTOR1  MBNAM  MBUS1  MBUS1  MBUS1  MBMI1  MBUS2  MBUS  MBUS2  MBMI2 |
|  | MBMX | MXUSER1  SUPERVISOR | MBCANC1  MBCANC1 | Yes  No | MBMX |

#### Example of Branch Defaulting/Enquiry Scope

Please refer to the geographical bank structure (see page 76) for details of branch hierarchy. The Americas piece of the hierarchy is repeated below for ease of reference:



CAUSER1 is locked to MBTOR1 so is only allowed to select MBTOR1 branch as the behalf of branch. Can only enquire on transactions for MBCA branch and below.

NY1USER1 is locked to MBNY1 so is only allowed to select MBNY1 branch as the behalf of branch. Can enquire on transactions for MBUS1 and below.

They can enter transaction details but not complete for MBSF1 and MBMI1.

USUSER1 is not locked to a branch so is allowed to select any of the following as behalf of branch (from the main banking entity MBNAM):

* MBTOR1
* MBINAM,
* MBNY1
* MBSF1
* MBMI1
* MBNY2
* MBSF2
* MBMI2

1. MBNAM, MBUS and MBCA branches are flagged as ‘Non transaction branch’ so are not valid as branches on the transaction. Their purpose is to aggregate all the branches that share the same back office infrastructure or to share a set of operating parameters. A fuller explanation is given later see "[External Systems](#O_57508)" on page 95.

Can enquire on transactions of any branches below MBNAM.

## Defining Main Banking Entity Characteristics

A branch flagged as a Main Banking Entity (MBE) has special characteristics within the model. It is, in effect a ‘mini zone’. This means that all the settings that the system previously supported only at zone level (such as base currency, rate formats, SWIFT terminal ID and many others) can now be set per MBE. Similarly for the trade finance system options.

These items can ONLY be set per MBE and not at any other level. So in this example these can be set up for MBNAM and MBMX. The base currency and FX rate formats are specified directly when defining the branch. The other characteristics are set via sub options associated only with a branch flagged as an MBE (button ‘Branch Options’ on the ‘Branch Service Map’ or ‘Additional Settings’ on the ‘Branch’ menu option).

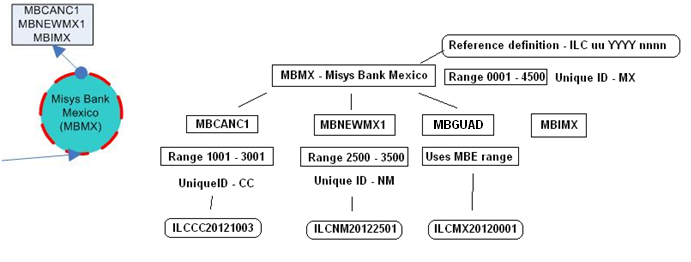
## Defining Product References and Sequence Numbers

Each time a new transaction is initiated, the system creates a master record to hold information about the transaction and its history. The information for each of the events associated with the transaction is linked to this master record. Master records are created with a reference structure, defined by the bank and made up of:

* A prefix, up to three characters long but not necessarily unique. Two or more products may share the same prefix
* A Unique Identifier for the main banking entity or branch
* A reference number, up to eight digits long
* A selection of optional data which the bank may choose to add

In the following example:

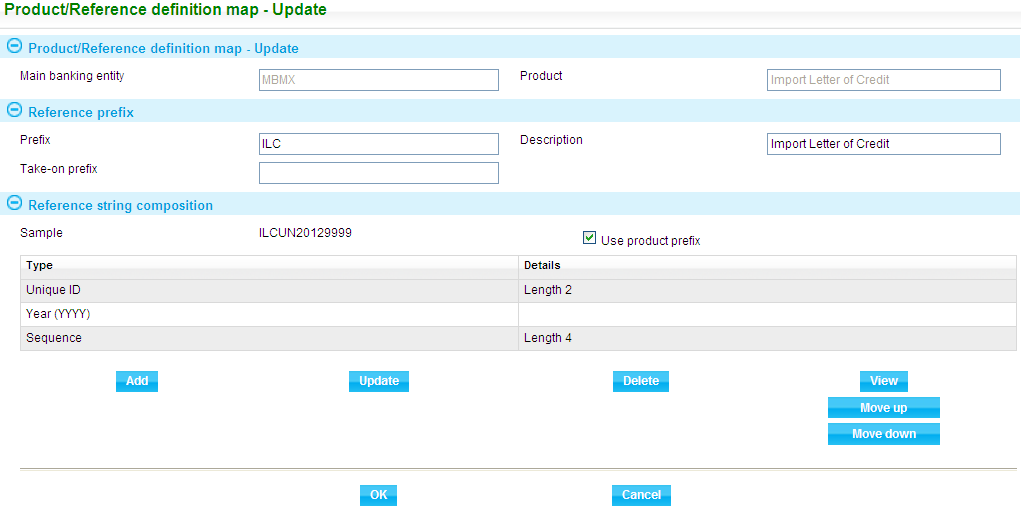
* The main banking entity Misys Bank Mexico defines the reference definition format ILCuuYYYYnnnn where:
* ILC - Product
* uu - Unique id
* YYYY - Year
* nnnn - Sequence number
* With number ranges assigned as follows:
* Cancun branch - Specific range 1001 – 3001 Unique ID - CC
* New Mexico branch – Specific range 2500 – 3500 Unique ID - NM
* Other branches, such as Guadalajara branch are assigned reference numbers at Main Banking Entity level. Specific range 1 – 4500 Unique ID - MX



Product/Reference definition for MBMX - Main banking entity.

Product / Reference definition for the ILC product is constructed as follows:

* Product Prefix
* Unique ID
* Year
* nnnn - Sequence

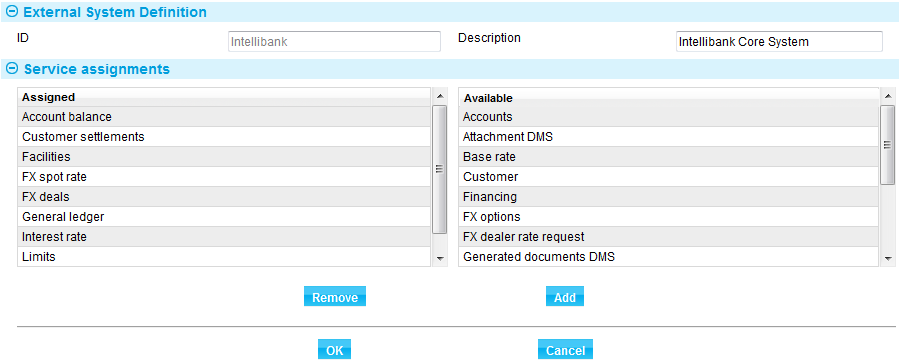


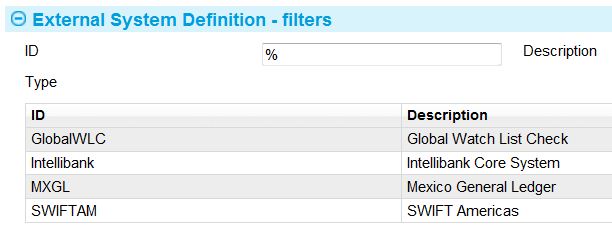
## External Systems and Services

Different areas of the organisation may link to different in house systems. In particular in this example there are two different general ledger systems in use. The system will need to route accounting entries to the correct G/L.

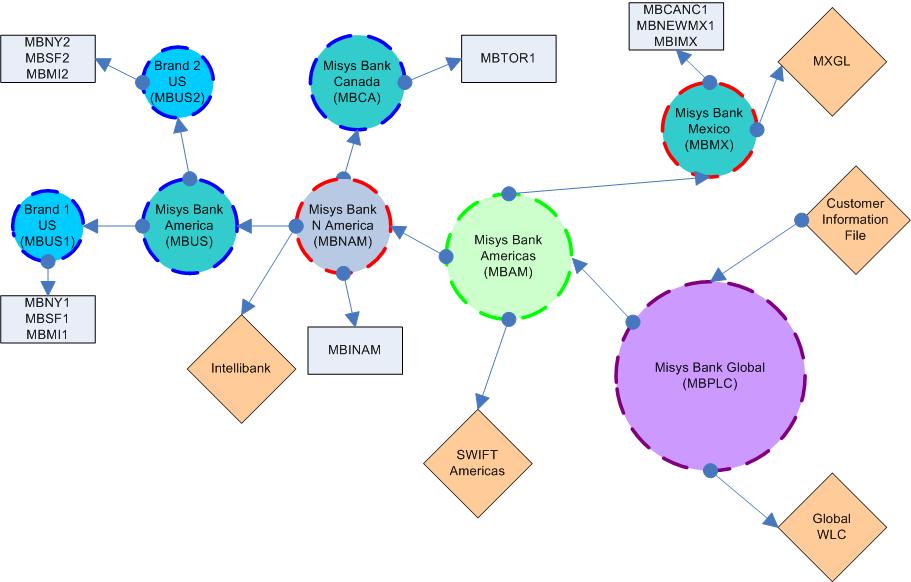
An important thing to note in this example is for the North American business. Canada and America share a G/L and Limits using the Intellibank G/L system. In order to support this model a ‘non transactional’ branch (MBNAM) was introduced to the model and this was flagged as a main banking entity (i.e. the start level for looking for a G/L). In the bank structure model this is indicated with a red dotted line around that branch.

External systems can support a range of sub services as shown here for Intellibank:

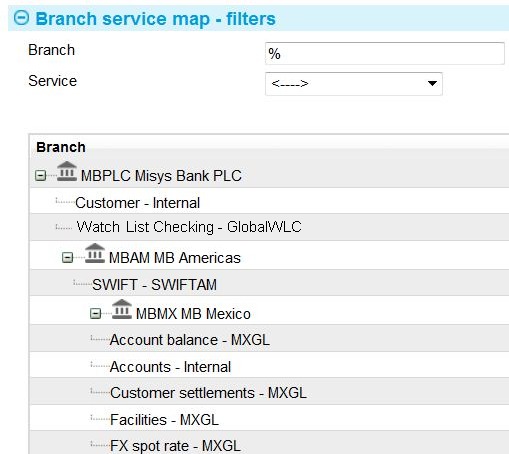




The following diagram shows the point in the branch hierarchy these systems are linked to (diamond shapes):



The Branch Service Map option links the services associated with each external system to the branches that will use the services:



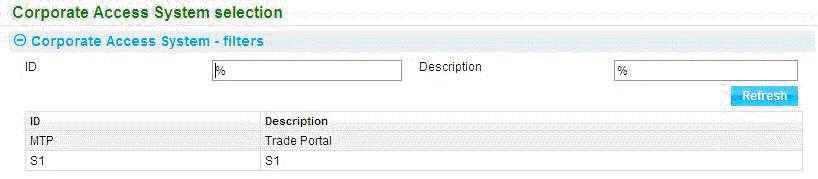
In the example any account balance checks for Mexico (MBMX) are supplied by the external system MXGL.

Any customer records are supplied by a higher level service for the whole organisation which is mapped to the branch MBPLC (which is a parent of MBMX).

Similarly there is one global watch list checking system (GlobalWLC) used by the whole organisation.

## Corporate Access Systems

The system allows multiple corporate access systems to be defined as different systems may be used in different parts of the bank by different customers. The corporate access system that a customer uses is recorded against them, so when a transaction is released the system automatically determines the system to which the gateway messages should be sent. In this example there are two corporate access systems in use:



## Customer Information Files

This example has a single global customer information file (CIF). This is mapped to the highest branch in the hierarchy (MBPLC).

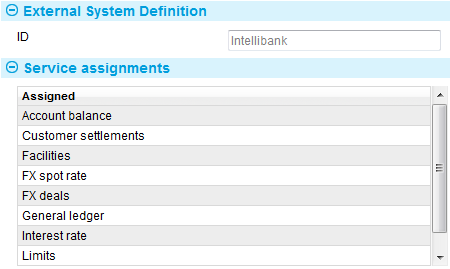
It is also possible to have different customer files in use in different parts of the global operation. For example to have a customer file linked to the Mexico business (MBMX) and another for Canada (MBCA).

When looking at customers within the system it will show where it got the customer from in brackets after the customer name e.g.:

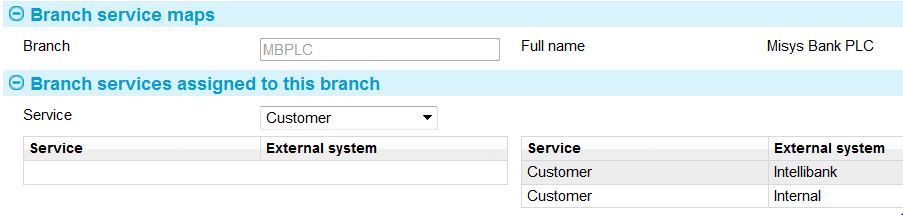
Customer with source CIF

This enables duplicate customer identifiers but still uniquely identifiable by their source CIF.

Customers may be read directly from the database (a service of this kind is called an ‘Internal’ service) or the details can be requested from an external system. Where they are required from an external system via a service call then they need to be mapped across as a service supported by an external system. For example if we were to add a Customer service to Intellibank:



Then this service can be selected for mapping to a branch:

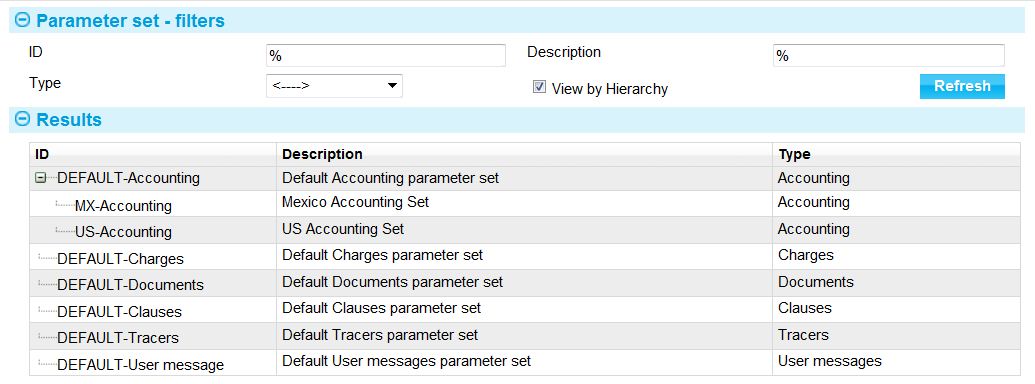


Here the bank now has the option of using either an internal customer source for MBPLC or a service link to a customer file externally via Intellibank.

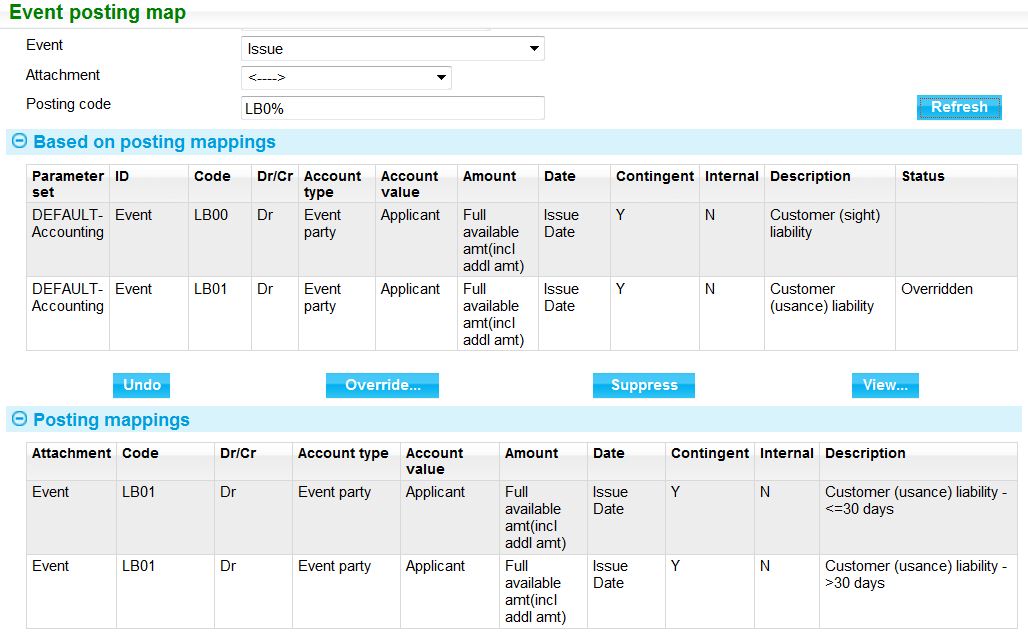
## Parameter Sets

In this example the US and Mexico have differences to the standard accounting parameter set:

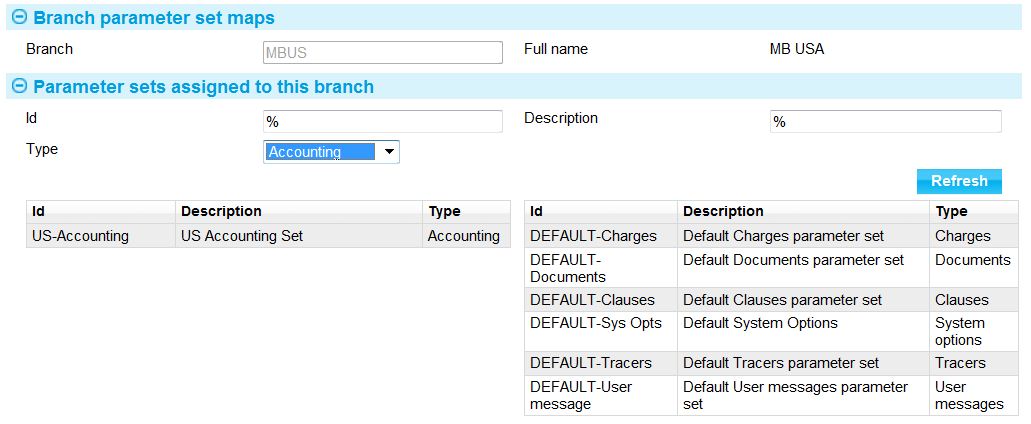
|  |  |  |
| --- | --- | --- |
| Set Type | Set ID | Based On |
| Accounting | US-Accounting | DEFAULT-Accounting` |
| Accounting | MX-Accounting | DEFAULT-Accounting` |



For the US the Import LC Postings vary the posting of Acceptance Liability (Usance). Instead of being to one account the account type posted to varies by the tenor of the LC. The posting in the DEFAULT set is marked as ‘Overridden’.



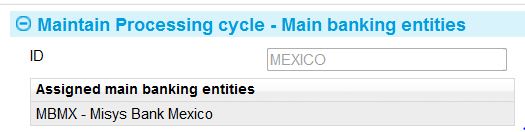
The set is then associated with a branch in the hierarchy. In this case MBUS for use by all branches in the USA. Canada would use the DEFAULT set and Mexico the Mexico set:



## Processing Cycles

In this model two processing cycles are defined, one to cover the MBNAM MBE and its branches and one to cover the MBMX MBE and its branches. The housekeeping tasks that are zone level (such as journal deletion and deletion of expired reports) are assigned to the NORTHAM processing cycle.

|  |  |  |  |
| --- | --- | --- | --- |
| Cycle ID | Description | Housekeeping? | MBEs Included |
| NORTHAM | North America | Y | MBNAM |
| MEXICO | Mexico | N | MBMX |



This enables two separate end of day cycles to be run covering all the branches within the scope of the MBEs included in the cycle.

# Glossary of Terms

The following table provides a list of special terms used in this Guide and explains what each means:

|  |  |
| --- | --- |
| Accounting branch | The branch assigned to a team for the purposes of apportioning transaction-related fees and charges. |
| Advanced workflow | Advanced workflow allows for multiple instances of each basic type of step to be defined. These can be linked into parallel sequences for banks to cater for complex workflow scenarios. |
| Autonomous banking entity | A branch that is not part of a branch hierarchy, for example an insourced bank whose business is self-contained and carried out all within a single banking entity. |
| Back office | One or systems within the bank for handling accounting, customer limits etc. |
| Banking entity | A discrete element of a bank’s organisation structure. |
| Banking group | A number of banking businesses that together constitutes a banking business. |
| Banking business | A discrete, autonomous banking organisation. |
| Behalf of branch | The branch that has legal ownership of a customer’s transaction. |
| Branch | Generic term used in this Guide and in the software for banking entities. |
| Capabilities | A method used in the system to control access to specific functionality. |
| Customer Information File (CIF) | One or more systems within the bank that holds information about the customer, including address details. |
| Dashboard | A display screen used in the system to provide information in graphic form on, among other things, teams’ workloads and the progress of transactions against service level agreements. |
| Enquiry scope | The transactions on which a user is allowed to enquire, and on which they can initiate, but not, complete, work. |
| Entity type | A parameter used for categorising branches according to the type of banking entity they are. |
| Event groups | A type of parameter used to define the events that users given a particular user role will be able to access. |
| Input branch | The branch entered against a transaction for the purpose of apportioning transaction-related fees and charges. |
| Main banking entity (MBE) | A branch in the bank's branch hierarchy which, together with all the branches below it, constitutes a discrete banking or business entity sharing the same general ledger and other services. |
| Master-level task | An action that can be carried out during transaction processing that is not specific to one particular product. For example, entering a note against a transaction, or special instructions against a party. |
| Master record | For each transaction, the system creates a master record to hold critical processing information about that transaction and its history. The information for each of the events associated with the transaction is linked to this master record. |
| Non-transaction branch | Parts of a bank's organisation where no transaction processing is carried out (typically used to group together related branches). |
| Parameter set | A set of parameters (such as postings definitions, charge definitions and clause definitions) used to create release items once an event is completed. Each branch is linked to the parameter sets it uses to create the release items (outputs). |
| Parameter set ID | A type of parameter used to provide a unique ID for sets of other parameters such as postings definitions, charge definitions and clause definitions. |
| Processing cycle | An end of day processing cycle that covers the transactions of one or more main banking entities. |
| Responsible team/Responsible user | The team and/or user that owns a transaction. These values are held against the master record and are intended to provide a point of contact for information regarding the transaction. |
| SLA | Service level agreement. |
| Team | A collection of users and user roles working for a defined set of branches. |
| Transaction scope | The transactions on which a user is allowed to initiate and complete work. |
| Coordinated Universal Time (UTC) | The primary time standard by which the world regulates clocks and time. Used in the system to calculate offsets for each time zone. |
| User roles | A type of parameter used to define a user’s access to the functionality, based on what their actual job involves. |
| Zone | A zone can be used to define a particular market segment, legal entity, team of users, and so on. A bank may choose to operate all business out of one zone or segregate into different zones. Zones can handle multiple banking entities and base currencies.  The major consideration for a zone is that it has its own database containing transactions and configuration data. |