**User Guide**

**CALYPSO**

# Calypso Application and Migration (CAM) Tool

Calypso Application and Migration (CAM) Tool — User Guide — V2.2.0

# Document History

| Revision date | Edition number | Comment |
|---|---|---|
| 30 August 2007 | First Edition | First edition for version 9.2 (Unreleased) |
| 26 Septmber 2007 | 1.1 | Updated with additional query file info. |
| 15 April 2008 | 2.0 | CAM Tool 2.0 (GUI release) |
| 02 July 2009 | 2.1.0 | Cam Tool 2.1.0 |
| 30 October 2009 | 2.1.1 | Cam Tool 2.1.1 |
| 01 December 2009 | 2.1.1 | Added suggested data import order. |
| 05 March 2010 | 2.2.0 | Enhancement release. See "Release 2.2.0 — 05 March 2010" on page 5. |

# Contents

# Figures

# Changes

### Release 2.2.0 — 05 March 2010

Compatibility

CAM 2.2.0 no longer requires the use of the Query Service. CAM is now compatible with Calypso 9.3 and above.

Status Messages

CAM now displays status messages on the status bar while initializing, as well as progress indicators and the number of objects remaining during import and export.

Enhanced Object Display

When a user opens a file for import and CAM displays the contents, either the object's description (if available) or the identifier is listed. For example, when importing a file containing the Holidays object, the display names of its records are displayed rather than a only a concatenation of its identifiers. Because not all objects have plain text descriptions (e.g., Holiday Codes), some objects will continue to only display identifiers.

CAM's Import view now displays additional Identifiers as Dependencies.

CAM's Export view now uses a tree structure to displays objects.

New Object Support in the GUI

CAM now supports Economic PL Parameters and CDS Settlement Matrix in the GUI. Refer to Table 4-1, "CalypsoML to Calypso Objects Reference," on page 23.

Additional Object Support

- Credit Rate Derivative objects: CDS Index Definition, CDS Settlement Matrix, Correlation Matrix, CDS Basis Adjustment, etc.
- Risk Analysis Reports:Cross Asset PL, Forward Ladder, Composite, etc.
- Calypso Workstation Nodes: Risk Aggregation, Visokio, Folders, etc.

Exporting without Dependency

Added new functionality to export without dependency. It is now possible to export a object without its dependent data. Refer to Table 2.1.2, "Export Tab," on page 13 for the list of controls on the Export tab.

Enhanced Import Display

The Selection Panel only displays the selected objects. When using the query dialog, items not meeting the crieteria are filtered out.

### Configuration File Change

The CAM Tool's configuration file, `default-migrator-gui-config.xml` is now contained in the `calypsoml_impl.jar`.

## Release 2.1.1 — 30 October 2009

Enhancements to CAM Tool include:

- Replaced the Query Service with the New Persistence Service. CAM Tool 2.1.1 is compatible with Calypso 11.0 and above.
- CAM Tool is now compatible with WebStart environments.

## Release 2.1.0 — 02 July 2009

Enhancements to CAM Tool include:

- Support for FX and IRD configuration data.
- Improvements in Objects Selection.
- Expandable and collapsible tree view to that displays first-level dependents on the Import tab.
- Load and View Selection files (package definitions) and CML Documents (data files).
- Selective import.
- Better presentation of configuration data. The `default-migrator-gui-config.xml` file has been modified to enhance usability.
- Smart data filters.

# 1 Introduction

The Calypso Application and Migration (CAM) Tool provides a means of exporting and migrating information from one Calypso database to another. The version of each database must be the same. The tool will copy the designated objects from either Sybase or Oracle to an XML file. This pure XML file is RDMBS-independent and can easily be transported to another location and then imported into a Calypso database of the same version, even if that database is on a different RDBMS or operating system.

**Note:** The source and target data models must be identical. Further, access permission are not honored, nor does the application perform any sort of data comparison or validation during migration. The user is responsible for determining (via a configuration file) which objects to export. All objects in the designated file are imported.

## 1.1 Overview

The Calypso Application and Migration (CAM) Tool v2.2.0 implements a GUI interface and command-line version and is compatible with all versions of Calypso, V9.3 and above. The CAM Tool uses the Hibernate-based Persistence Service when operating in GUI mode. The Persistence Service is run as a background service.

To operate in the GUI mode, administrator must run the Persistence Service and ensure that the `persistence-service.jar` is in the DataServer classpath prior to launching the DataServer.

Additional required files are listed in Section 3.1, "Required Files," on page 20.

## 1.2 Object Coverage

The CAM Tool supports all CalypsoML objects for both import and export. However, only a subset of the most commonly required objects are available for selection from the GUI. Manually changes to a Selection Files created in the GUI provides access to all CalypsoML objects.

**Selection Files** indicate which objects to export. **Selection File** creation in the GUI is limited to the objects listed below:

| Category | | |
|---|---|---|
| | **Type** | |
| | | **Sub-Type** |
| **Startup Configuration** | | |
| | **Books and Bundles** | |
| | | • Trading Books |

| Category (Continued) | | |
| --- | --- | --- |
| | Type (Continued) | |
| | | Sub-Type (Continued) |

- Liquidation Configuration
- Position Configuration
- Position Aggregation

**Environment**

- Pricing Environment
- Pricer Configuration
- Pricer Parameter Set
- Quote Set
- Quote Value

**Fees**

- Fee Definition

**Filters**

- Static Data Filters
- Trade Filters
- Filter Set

**Legal Data**

- Legal Entities
- LE Contacts
- LE Agreements

**Reference Data**

- Currency Definitions
- Currency Pairs
- Holiday Codes
- Holiday Rules
- Date Rules
- Countries
- Product Code

**Settlements**

- Settlement Instructions
- Netting Config
- Netting Method

| Category (Continued) | | |
| --- | --- | --- |
| | Type (Continued) | |
| | | Sub-Type (Continued) |

**System**

- All Domain Values
- Domain Values

**User Access Control**

- Users
- Groups
- Task Station Defaults
- User Access Permission
- Curve FX

**Credit Derivatives**

- CDS Settlement Matrix
- CDS Settlement Matrix Configuration
- Reference Entity Basket
- CDS ABS Index Tranche Definition
- CDS ABS Index Definition
- CDS Index Definition

**Reporting and Risk**

- Report Template

**Accounting**

**Accounting**

- Accounts
- Accounting Books
- Accounting Events
- Accounting Rules

**Market Data**

**Feed**

- Feed Address

**Rates**

- FX Rates
- Rate Index Definitions
- Rate Index Defaults

**Curves**

- Curve CDS Basis Adjustment

| Category  (Continued) | | |
| --- | --- | --- |
| **Type  (Continued)** | | |
| | **Sub-Type  (Continued)** | |

- Curve FX
- Curve Probability
- Curve Recovery
- Curve Zero

**Surfaces**

- Volatility Surface
- FX Vol Surface

**Coorelation and Covariance**

- Basket Coorelation
- Coorelation Matrix
- Coorelation Formula

**Message Configuration**

**Messages and Matching**

- Message Config
- Sender Config

**Products**

**Listed Derivatives**

- Future Contracts
- Future Option Contracts

**Commodities**

- Commodities
- Commodity Reset

**Scheduled Tasks**

**Scheduled Tasks**

- Scheduled Tasks

**Workflows**

**Workflows**

- Workflows

**Configuration**

- Kick Off Cut Off Configuration

**Position and Risk**

**Risk Analysis**

- Composite Parameter

| Category  (Continued) | | |
| Type  (Continued) | | |
| Sub-Type  (Continued) | | |

- Cross Asset PL Parameters
- Economic PL Parameters
- Forward Ladder Parameters
- Pricing Parameter
- Scenario Parameter
- Sensitivity Parameter
- Simulation Parameter

**CWS Nodes**

- Folders
- Risk Aggregation Nodes
- Visokio Nodes
- Workspace Nodes

## 1.3   Document Layout

— This section.

. Operating instructions for end users.

. Installation and system configuration information for Administrators.

# 2      End User Instructions

## 2.1   CAM Tool — GUI Version

### 2.1.1     Launch the CAM Tool

Prior to launching the CAM Tool, ensure that the Data Server is running. The CAM Tool will not start unless the Data Server is available.

Launch the CAM Tool using the script or batch file provided by your administrator. The CAM Tool installation includes a Windows batch file to launch the application (`$CALYPSO_HOME\bin\launch-migrator.bat`).

Figure 2-1: CAM Tool Main Window (Export tab)
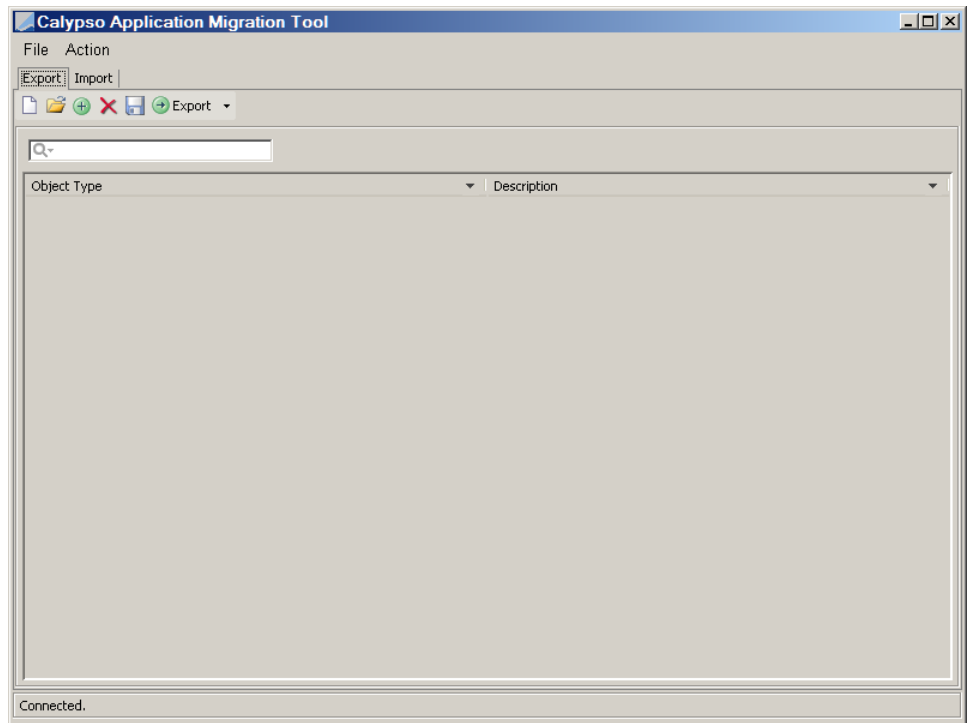


**Note:** File locations given in this manual are based on the installation instructions described in .

The CAM Tool status bar displays **Connected** when the connection to the DS is established.

### 2.1.2    Export Tab

Export tab actions include creating, loading, and saving Selection Files and exporting data.

Figure 2-2: CAM Tool Export Tab
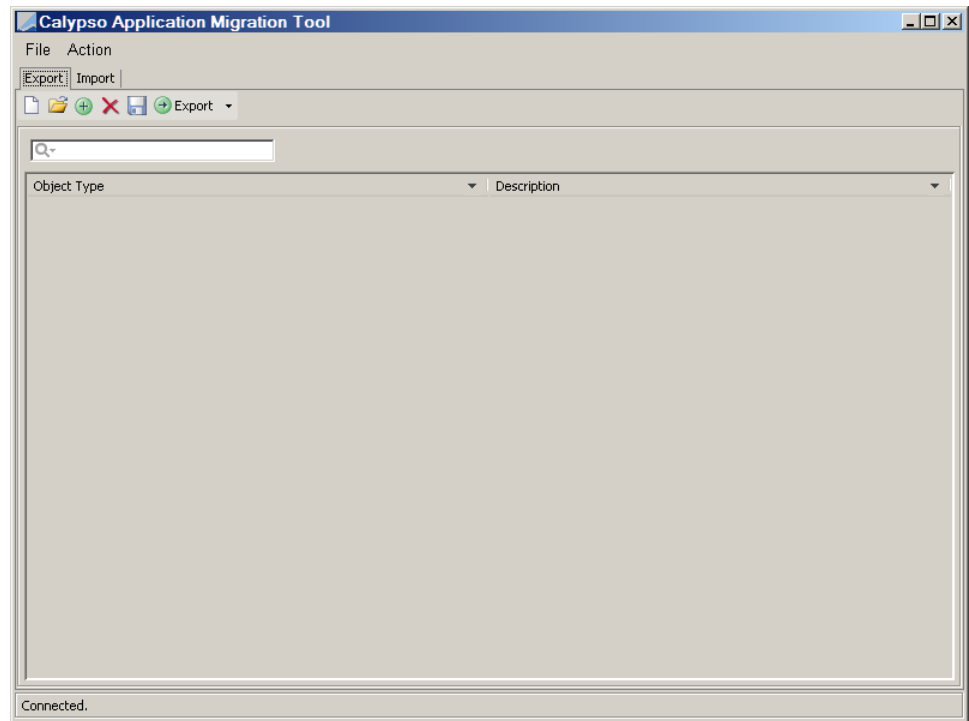
Toolbar buttons, which duplicate all menu functionality, control the operation of the **Export** tab. Refer to the table below for descriptions of each button.

Table 2-1: CAM Tool Toolbar Controls (Export Tab)

| Control | Icon | Description |
|---------|------|-------------|
| Initialize the Selection | | Clears all objects from the **Selected Objects** pane. |
| Load Selection File | | Opens the File Browser to choose a **Selection File**. Once loaded, you can review which objects are included and modify the file as desired. |
| Add Objects | | Add objects to the **Selected Objects** pane. Launches the *Add an Object* window (Figure 2-3). |
| Delete Selected Rows | | Deletes the selected row. |
| Save Selection File | | Saves the selected objects as a **Selection File**. |
| Export Data | Export ▾ | Export the data in the **Selection File** to the specified location. Choose from **Export** or **Export without dependents**. Select **Export** to export the object and it's dependent objects. Select **Export without dependents** to export only objects in the seclection. |

### Add An Object Window

This window is divided into two sections. The top portion contains selectors for **Category**, **Type**, and **SubType**. The lower portion will either contain a set of data selectors based on the top section or a pane of data elements contained in the **SubType** selector. The display will vary according to your selections.

Figure 2-3: Add an Object Window
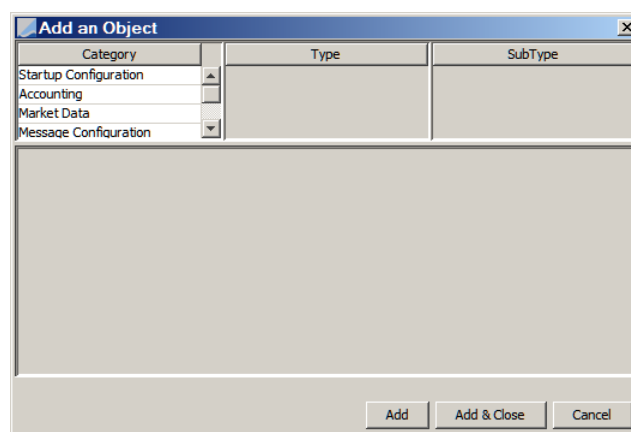
Table 2-2: Add an Object Window Controls

| Control | Description |
|---------|-------------|
| Add | Click to **Add** the object to your selection. |
| Add & Close | Click to **Add** the object to your selection and return to the main CAM Tool window. |
| Cancel | Close the Add an Object window and return to the main CAM Tool window. |

## 2.1.3    Import Tab

Import tab operations include loading a Data File and importing data from the Data File.

Figure 2-4: CAM Tool — Import Tab
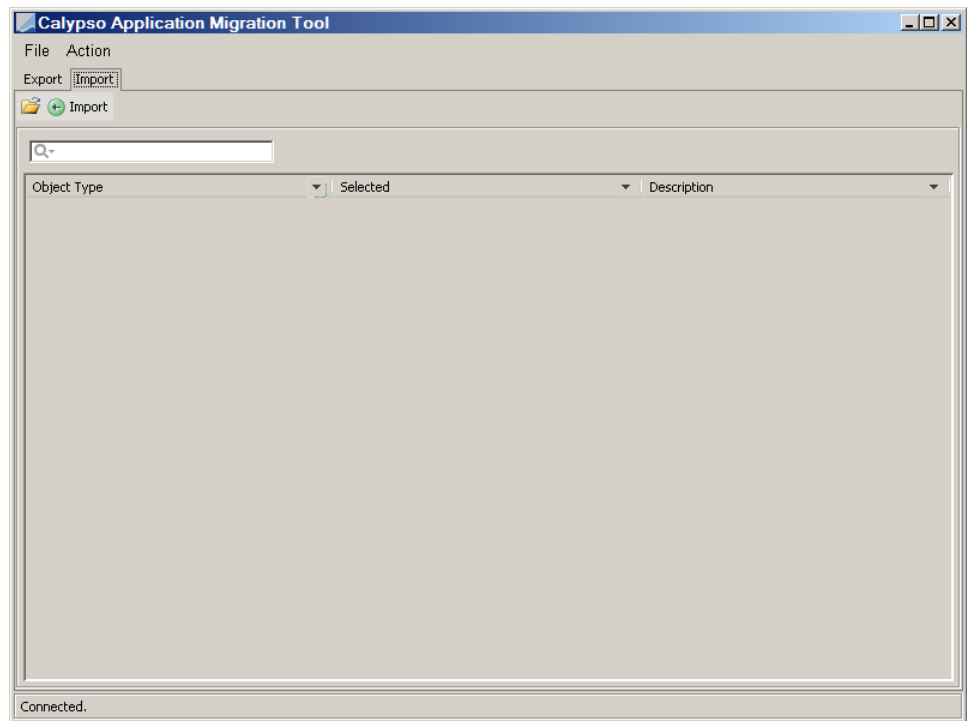
Toolbar buttons, which duplicate all menu functionality, control the operation of the **Import** tab. Refer to the table below for descriptions of each button.

Table 2-3: CAM Tool Toolbar Controls (Import Tab)

| Control | Icon | Description |
|---|---|---|
| Load Data File | | Opens the File Browser to choose a Data File. |
| Import Data | | Loads data from the selected Data File |
| Filter Data | | Enter filtering data. You can filter a specific column or a combination of columns. |

# 2.2    Exporting Data

### 2.2.1    Building a Selection File

A **Selection File** is required to perform export operations. Refer to Figure 2-1, "CAM Tool Main Window (Export tab)" on page 12 while following the instructions, below:

1.  Click to initialize the selection panel.

2.  Click . The *Add an Object* window appears.

3.  In the *Add an Object* window (Figure 2-3), choose the desired **Category**, **Type**, and **Subtype**.

4.  In the lower pane, choose the data you wish to export. Use the column headers to filter and select data from the list.

5.  Click **Add**. Your selection will appear in the **Selected Objects** pane on the **Selection** tab. Repeat steps 3 through 5, as desired.

6.  When you have selected all of the desired objects, click **Cancel** to close the *Add an Object* window and return to the main window.

7.  On the main window, review your selection. If there are any selections in error, you can remove them by selecting each and then clicking **Delete Selection** on the Toolbar.

8.  When you are satisfied with your selections, select Action > Selection Save from the Main Window menu or click to save your **Selection File**.

---

**Note:** Users can manually add additional objects to GUI-created **Selection Files**. Refer to Table 4-1, "CalypsoML to Calypso Objects Reference," on page 23 for a complete list of objects. Additional objects will become available for GUI selection in later releases.

**The CAM Tool exports and import alls CalypsoML Objects**.

---

### 2.2.2    Exporting

To export a selection of objects:

1.  On the Export tab, click 📁 to load an existing **Selection File**, or refer to the previous section to create a new Selection File.

2.  Click ⊕Export ▾ , choose **Export** or **Export without Dependents**, then browse to the directory where you wish to save your **Output File**, enter the filename, then click **Open**.

## 2.3   Importing

### 2.3.1    Basic Importing

1.  On the **Import** tab, click 📁 to use the file browser to locate your data file, then click **Open**.

2.  The application will import all selected items, i.e., those with check-marks. If desired, use the Filter Data query to locate specific objects to import or ignore (see Section 2.3.2, "Filtering Imported Data," on page 17), or click ⊕ to import all data in the data file.

### 2.3.2    Filtering Imported Data
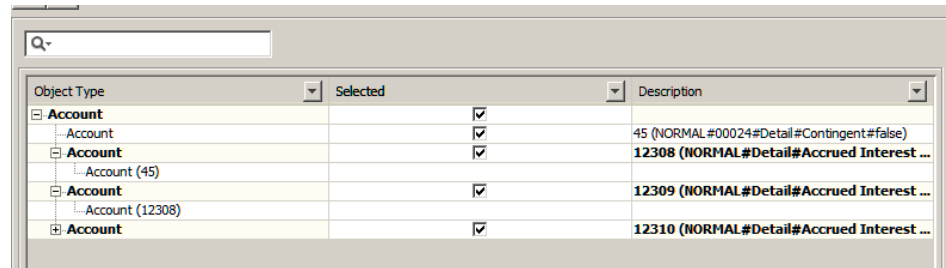
Using the Filter Data Query, you can find and select specific items based on the entered criterion. Only selected crieteria, i.e., those with checkmarks, are used with your entered text. The query permits you a range of filtering attributes:

Figure 2-5: Filtering Attributes



---

For example, the following image shows an unfiltered view:
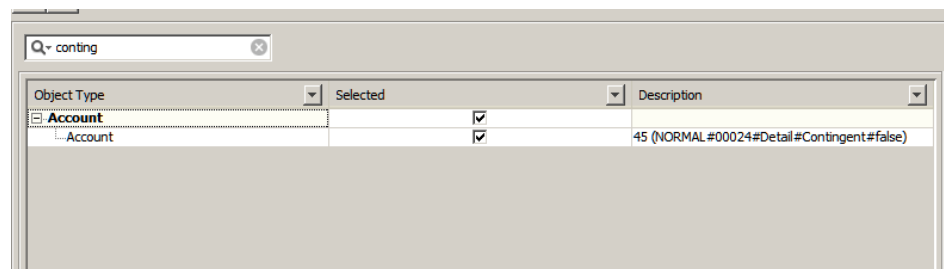
Figure 2-6: Data File — Unfiltered



Typing in a string begins filtering the data according to the selected Filtering Attributes:

Figure 2-7: Data File — Using the Data Filter



When the Import operation is started, only the items visibleon the Selection Panel are imported.

### 2.3.3    Suggested Object Import Order when Building New Databases

Calyspo recommends the following order when importing data into a newly created database:

- Domain Values
- Countries
- Holiday Code
- Holiday Rules
- User Access Permissions
- Currency Defaults
- Currency Pairs
- FX Rate Definition
- Rate Index Defaults
- Rate Index Definition

Followed by instrument data, market data, and trade data. Add additional objects as necessary.

# 2.4   CAM Tool Command-Line Usage

The CAM Tool can facilitate repetitive implementation of a data environment when it may be necessary to restore a database system to a set of default values during testing. Data can be exported from a source system into a file or imported from a file. The user must create a Selection/Query File to select which data to export.

```
java com.calypso.apps.datamigration.Migrator –s env_name [-i | –h | -e]
      [-u user_name] -q query_file [-h] -t env_name [-e] [-p password]
      -f file | URL
```

:

Table 2-4: CAM Tool Command Line Options

| Option & Argument | Description |
|---|---|
| –e | Export mode. The application will export data named in –q *query_file* to the file named by the –f *file* option. |
| –f *file* \| *URL* | The file to import or the file to export to. URL may be used to pro-vide a file location. For example: file://E:/users/file.xml. |
| –h | Displays the internal help. |
| –i | Import mode. The application will read the data file named by –f *file* and import it into the database named in –s *env_name*. |
| –p *password* | The Calypso user's password. Required. |
| –q *query_file* | The file containing the selection of objects to export. **Required for Export mode**. |
| –s *env_name* | Defines the name of the source Calypso environment. **Required (Export mode)**. |
| –t *env_name* | Defines the name of the target Calypso environment. **Required (Import mode)**. |
| –u *user_name* | The name of the Calypso user to use to load the objects. **Required**. |

### Usage Examples:

```
#Import Example
java com.calypso.apps.datamigration.Migrator -i -f input.xml -t calypso_env -u
username -p password

#Export Example
java com.calypso.apps.datamigration.Migrator -e -q query.xml -f output.xml -s
calypso_env
                                  -u username -p password
```

### Example Selection/Query File:

```
<?xml version="1.0" encoding="UTF-8"?>
<selection xmlns="http://www.calypso.com/xml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <items>
```

```
    <item>
        <identifiers>
            <type>com.calypso.tk.core.Book</type>
            <identifier code="TRADINGA" codifier="convention"/>
        </identifiers>
    </item>
    <item>
        <identifiers>
            <type>com.calypso.tk.bo.document.SenderConfig</type>
            <identifier code="3829" codifier="YourDataServerName"/>
        </identifiers>
    </item>
</items>
</selection>
```

# 3       Administrator Information

This section provides detailed instructions for installing and configuring your Calypso implementation to support the CAM Tool.

## 3.1  Required Files

The following third-party jars must be available in your classpath:
- commons-cli-1.0.jar (http://commons.apache.org/cli/)
- commons-lang-1.0.jar (http://archive.apache.org/dist/commons/lang/old/v1.0/)
- j2ee-1.4.jar (http://www.java.com)
- jconn3.jar — or — ojdbc5.jar (Provided by your RDBMS vendor)
- stax-1.2.0.jar (http://dist.codehaus.org/stax/jars/)
- stax-api-1.0.1.jar (http://dist.codehaus.org/stax/jars/)

## 3.2  Installation

### 3.2.1    Third-Party Tools

1. Obtain stax-1.2.0.jar and stax-api.1.0.1.jar from:
   http://dist.codehaus.org/stax/jars/

2. Obtain commons-lang-1.0.jar from:
   http://archive.apache.org/dist/commons/lang/binaries/

3. Obtain commons-cli-1.0.jar from:
   http://archive.apache.org/dist/commons/cli/binaries/

4. Obtain the j2ee-1.4.jar as part of the J2EE 1.4 SDK package from Sun:
   http://java.sun.com/j2ee/1.4/download.html#sdk

### 3.2.2    Calypso Software

1. Obtain the CAM Tool installation file from Calypso.

2. Place the file in `$CALYPSO_HOME` directory.

3. Uncompress the file:

```
jar xvf migrator-2.2.0-install.zip
```

The following files are added to `$CALYPSO_HOME`:

```
+---bin
¦     launch-dataserver.bat
¦     launch-migrator.bat
¦     launch-migrator-mt.bat
¦
+---config
¦     default-migrator-gui-config.xml
¦
+---docs
¦        camtool_release_notes_2.1.1.pdf
¦        camtool_user_guide_2.1.1.pdf
¦
+---jars
         migrator-2.1.1.jar
```

- `bin` — Contains two batch files that can launch the DataServer and the CAM Tool. Refer to these batch files as you create your own shell scripts or batch files.
- `config` — The `default-migrator-gui-config.xml` configuration file.
- `docs` — This documentation and release notes.
- `jars` — Jar files for this release of the CAM Tool.

# 3.3   Configuration

1. Ensure that $CALYPSO_HOME is set to the location of the Calypso release.

2. Ensure that your DataServer's CLASSPATH includes:

   ```
   $CALYPSO_HOME\jars\persistence-service-1.0.0.jar
   ```

   The Persistence Service is required for CAM Tool GUI operation. You must restart the Data Server if you needed to modify the CLASSPATH.

3. Ensure that the CLASSPATH in the script that you provide to end users to launch the CAM Tool includes all of the jars in .

Once the DataServer (the Persistence Service launches automatically with the DataServer) is operating, the end user can use the CAM Tool's GUI to create **Selection Files** and to import/export data. Users may also edit GUI-

based **Selection Files** to manually add CalypsoML objects not available from the GUI.

You can base your own scripts on the `launch-migrator.bat` file available in `$CALYPSO_HOME\bin`.

# 4      CalypsoML to Calypso Objects Reference

The following table provides a list of Calypso Objects with their code and codifier as used in CalypsoML. Use these identifiers to manually construct a Selection/Query File. Note that certain objects are not yet supported. Refer to the table footnotes.

**Note:**     Type, identifier code, and codifier are case sensitive.

**Note:**     You can also use GUI-created **Selection Files** as the basis for your Selection/Query file. After creating a file, simply add objects from the table below before exporting.

When the codifier is "dataServerName," use the actual name of your running DataServer.

Table 4-1: CalypsoML to Calypso Objects Reference

| Calypso Object | Class Name (Type) | [Codifier \| Code] |
|---|---|---|
| Account | **com.calypso.tk.refdata.Account typeOverride=com.calypso.tk.service.AccountingRuleWrapper** | [dataServerName \| id] |
| **AccountInterestConfig** | com.calypso.tk.refdata.AccountInterestConfig | [dataServerName \| id] ["AccountInterestConfigKey" \| name] |
| **AccountSweepingConfig** | com.calypso.tk.refdata.AccountSweepingConfig | [dataServerName \| id] |
| **AccountingBook** | com.calypso.tk.core.AccountingBook | ["convention" \| name] |
| **AccountingEventConfig** | com.calypso.tk.refdata.AccountingEventConfig | [dataServerName \| id] |
| AccountInterestConfig | **com.calypso.tk.refdata.AccountInterestConfig** | \|[dataServerName \| id] ["AccountInterestConfigKey" \| name ] |
| AccountingRule | **com.calypso.tk.refdata.AccountingRule typeOverride="com.calypso.tk.service. AccountingRuleWrapper** | [dataServerName \| id] |
| AdviceConfig | com.calypso.tk.refdata.AdviceConfig | \|[dataServerName \| id] ["AdviceKey" \| processingOrg.code#productType# messageType#receiverRole#senderContactType# receiverContactType#eventType#receiver.code# AddressMethod#filterSet#language#inactive(toString)] |
| **BookHierarchy** | com.calypso.tk.refdata.BookHierarchy | ["convention" \| name] |
| **Book** | com.calypso.tk.core.Book | ["convention" \| name] |
| CalypsoRiskAggregationNodeDb | **com.calypso.tk.refdata.CalypsoRiskAggregationNodeDb** | [dataServerName \| treeNodeId] |

Table 4-1: CalypsoML to Calypso Objects Reference  (Continued)

| Calypso Object | Class Name (Type) | [Codifier | Code] |
|---|---|---|
| CalypsoTreeDb | **com.calypso.tk.refdata.CalypsoTreeDb** | [dataServerName | treeId] |
| CalypsoTreeViewNodeDb | **com.calypso.tk.refdata.CalypsoTreeViewNodeDb** | [dataServerName | treeId] |
| CalypsoVisokioNodeDb | **com.calypso.tk.refdata.CalypsoVisokioNodeDb** | [dataServerName | treeNodeId] |
| CalypsoWorkspaceNodeDb | **com.calypso.tk.refdata.CalypsoWorkspaceNodeDb** | [dataServerName | treeNodeId] |
| **CDSSettlementMatrix** | com.calypso.tk.product.CDSSettlementMatrix | [dataServerName | id] |
| **CDSSettlementMatrixConfig** | com.calypso.tk.product.CDSSettlementMatrixConfig | ["convention" | username] |
| **CFDContractDefinition** | com.calypso.tk.refdata.CFDContractDefinition | [dataServerName | id] |
| **CFDCountryGrid** | com.calypso.tk.refdata.CFDCountryGrid | [dataServerName | id] |
| CommodityFwdPointGenerator | **com.calypso.tk.product.CommodityFwdPointGenerator** | ["convention" | name] |
| CommodityOptionVolType | **com.calypso.tk.marketdata.CommodityOptionVolType** | ["convention" | optionType | delta] |
| CommodityReset | **com.calypso.tk.refdata.CommodityReset** | ["convention" | name] |
| CompositeAnalysisParam | **com.calypso.tk.risk.CompositeAnalysisParam** | ["convention" | name] |
| CorrelationFormula | **com.calypso.tk.marketdata.CorrelationFormula** | ["convention" | type | formula | name] |
| CorrelationMatrix | com.calypso.tk.marketdata.CorrelationMatrix | |["convention" | type#currency#name]+ [datetime] [dataServerName | id] (com.calypso.jaxb.xml.MarketDataIdentifiers includes date for identification) |
| CorrelationSurface | com.calypso.tk.marketdata.CorrelationSurface | ["convention" | type#currency#name]+ [datetime] [dataServerName | id] (com.calypso.jaxb.xml.MarketDataIdentifiers includes date for identification) [a] |
| CorrelationSurfaceUnderlying | com.calypso.tk.marketdata.CorrSurfUnderlying | ["CODIFIER-"+dataServerName | id] |
| CorrSurfUnderlyingCDSIndex-Tranche | **com.calypso.tk.marketdata. CorrSurfUnderlyingCDSIndexTranche** | [dataServerName | id] |
| **Country** | com.calypso.tk.refdata.Country | ["convention" | name] |
| CrossAssetPLParam | **com.calypso.tk.risk.CrossAssetPLParam** | ["convention" | name] |
| CurrencyDefault | **com.calypso.tk.refdata.CurrencyDefault** | [dataServerName | id] |
| **CurrencyPair** | com.calypso.tk.refdata.CurrencyPair | ["convention" | primaryCode#quotingCode] |

Table 4-1: CalypsoML to Calypso Objects Reference  (Continued)

| Calypso Object | Class Name (Type) | [Codifier | Code] |
|---|---|---|
| Curve | com.calypso.tk.marketdata.Curve | ["convention" | type#currency#name]+ [datetime] [dataServerName | id] (com.calypso.jaxb.xml.MarketDataIdentifiers includes date for identification) |
| CurveUnderlying | com.calypso.tk.marketdata.CurveUnderlying | [dataServerName | id] |
| CurveUnderlyingBasisSwap | **com.calypso.tk.marketdata.CurveUnderlyingBasisSwap** | [dataServerName | id] |
| CurveUnderlyingBond | **com.calypso.tk.marketdata.CurveUnderlyingBond** | [dataServerName | id] |
| CurveUnderlyingBondFuture | **com.calypso.tk.marketdata.CurveUnderlyingBondFuture** | [dataServerName | id] |
| CurveUnderlyingCDS | **com.calypso.tk.marketdata.CurveUnderlyingCDS** | [dataServerName | id] |
| CurveUnderlyingCDSIndex | **com.calypso.tk.marketdata.CurveUnderlyingCDSIndex** | [dataServerName | id] |
| CurveUnderlyingETO | **com.calypso.tk.marketdata.CurveUnderlyingETO** | [dataServerName | id] |
| CurveUnderlyingETOEquity | **com.calypso.tk.marketdata.CurveUnderlyingETOEquity** | [dataServerName | id] |
| CurveUnderlyingETOEquityIndex | **com.calypso.tk.marketdata.CurveUnderlyingETOEquityIndex** | [dataServerName | id] |
| CurveUnderlyingFRA | **com.calypso.tk.marketdata.CurveUnderlyingFRA** | [dataServerName | id] |
| CurveUnderlyingFuture | com.calypso.tk.marketdata.CurveUnderlyingFuture | [dataServerName | id] |
| CurveUnderlyingFXForward | **com.calypso.tk.marketdata.CurveUnderlyingFXForward** | [dataServerName | id] |
| CurveUnderlyingMoneyMarket | **com.calypso.tk.marketdata.CurveUnderlyingMoneyMarket** | [dataServerName | id] |
| CurveUnderlyingSpecificMMkt | **com.calypso.tk.marketdata.CurveUnderlyingSpecificMMkt** | [dataServerName | id] |
| CurveUnderlyingSpread | **com.calypso.tk.marketdata.CurveUnderlyingSpread** | [dataServerName | id] |
| CurveUnderlyingSwap | **com.calypso.tk.marketdata.CurveUnderlyingSwap** | [dataServerName | id] |
| CustomObservedData | com.calypso.tk.product.CustomObservedData | Not supported |
| **DateRule** | com.calypso.tk.core.DateRule | ["convention" | Name] |
| **DomainAdapter** | com.calypso.tk.refdata.DomainAdapter | ["convention" | Name] |
| DomainValueAdapter | **com.calypso.tk.refdata.DomainValueAdapter** | ["convention" | domainName#value] |
| DomainValuesAdapter | **com.calypso.tk.refdata.DomainValuesAdapter** | ["convention" | Version] |
| DomainValues$DomainValues-Row | **com.calypso.tk.refdata.DomainValues$DomainValuesRow typeOverride="com.calypso.tk.refdata.DomainValueAdapter** | ["convention" | key#value] |
| **EconomicPLParam** | com.calypso.tk.risk.EconomicPLParam | ["convention" | name] |
| EngineParamAdapter | **com.calypso.tk.refdata.EngineParamAdapter** | ["convention" | engineName] |

Table 4-1: CalypsoML to Calypso Objects Reference  (Continued)

| Calypso Object | Class Name (Type) | [Codifier \| Code] |
|---|---|---|
| **ETOContract** | com.calypso.tk.product.ETOContract | ["convention" \| currency#exchangeCode#name] |
| **EventConfig** | com.calypso.tk.event.EventConfig | ["convention" \| name] |
| ExtendedCalypsoMLGDIdentifiers | ExtendedCalypsoMLGDIdentifiers | not a product |
| **FeeBillingRule** | com.calypso.tk.refdata.FeeBillingRule | [dataServerName \| id] |
| **FeeDefinition** | com.calypso.tk.bo.FeeDefinition | ["convention" \| type] |
| FeeGrid | **com.calypso.tk.refdata.FeeGrid** | [dataServerName \| id] |
| **FeedAddress** | com.calypso.tk.marketdata.FeedAddress | ["convention" \| quoteName#feedName# quoteType#feedAddress] |
| **FeedConfig** | com.calypso.tk.marketdata.FeedConfig | ["convention" \| feedName] |
| **FilterSet** | com.calypso.tk.marketdata.FilterSet | ["convention" \| name] |
| FixingDatePolicy | **com.calypso.tk.product.commodities.schedulegeneration. fixing.FixingDatePolicy** | [dataServerName \| id] |
| FolderDb | **com.calypso.tk.refdata.FolderDb** | [dataServerName \| treeNodeId] |
| ForwardLadderParam | **com.calypso.tk.risk.ForwardLadderParam** | ["convention" \| name] |
| **Fund** | com.calypso.tk.fund.Fund | [dataServerName \| id] |
| **FutureContract** | com.calypso.tk.product.FutureContract | ["convention" \| currency#exchange#name] |
| **FutureOptionContract** | com.calypso.tk.product.FutureOptionContract | ["convention" \| currency#exchange#name] |
| FX Product | com.calypso.tk.product.FX | ["fx_ccy_pair" \| currencyPair(toString)] |
| **FXQVar** | com.calypso.tk.refdata.eXSP.variableI.quotableI.FXQVar | ["convention" \| name] |
| **FXReset** | com.calypso.tk.refdata.FXReset | ["convention" \| primary#quoting#name] |
| **Group** | com.calypso.tk.refdata.Group | ["convention" \| name] |
| com.calypso.tk.core.Holiday | **com.calypso.tk.core.Holiday typeOverride="com.calypso.tk.core.HolidayAdapter** | ["convention" \| code] |
| HolidayAdapter | **com.calypso.tk.core.HolidayAdapter** | ["convention" \| cityCode] |
| **HolidayCode** | com.calypso.tk.refdata.HolidayCode | ["convention" \| code] |
| **HolidayRule** | com.calypso.tk.core.HolidayRule | ["convention" \| name] |
| **InterestRateQVar** | com.calypso.tk.refdata.eXSP.variableI.quotableI.InterestRateQVar | ["convention" \| name] |
| **LEContact** | com.calypso.tk.refdata.LEContact | [dataServerName \| id] |

Table 4-1: CalypsoML to Calypso Objects Reference  (Continued)

| Calypso Object | Class Name (Type) | [Codifier \| Code] |
|---|---|---|
| LegalAgreement | com.calypso.tk.refdata.LegalAgreement | ["convention" \| processingOrgId(converted to code/short name)#legalEntityId(coverted to code/short name)#productFamily#productType] |
| LERegistration | **com.calypso.tk.refdata.LERegistration** | [dataServerName \| id] |
| **LegalEntity** | com.calypso.tk.core.LegalEntity | ["convention" \| Code] |
| LegalEntityRelation | com.calypso.tk.refdata.LegalEntityRelation | ["convention" \| legalEntityId(converted to code)#legalEntityRole#poId] |
| LiquidationInfo | **com.calypso.tk.mo.LiquidationInfo typeOverride="com.calypso.tk.mo.LiquidationInfoAdapter** | ["convention" \| bookId#productType] |
| LiquidationInfoAdapter | **com.calypso.tk.mo.LiquidationInfoAdapter** | ["convention" \| bookId#productType] |
| ManualSDI | **com.calypso.tk.refdata.ManualSDI** | [dataServerName \| id] |
| **MarginCallConfig** | com.calypso.tk.refdata.MarginCallConfig | [dataServerName \| id] |
| **Mutation** | com.calypso.tk.product.Mutation | [dataServerName \| id] |
| NettingConfigAdapter | **com.calypso.tk.bo.NettingConfigAdapter** | ["convention" \| nettingType] |
| NettingConfig$NettingConfigRow | **com.calypso.tk.bo.NettingConfig$NettingConfigRow typeOverride="com.calypso.tk.bo.NettingConfigAdapter** | ["convention" \| type] |
| **NettingMethod** | com.calypso.tk.refdata.NettingMethod | [dataServerName \| id] |
| **ObservedData** | com.calypso.tk.product.ObservedData | [dataServerName \| id] |
| PositionInfo | **com.calypso.tk.mo.PositionInfo typeOverride="com.calypso.tk.mo.PositionInfoAdapter** | ["convention" \| bookId#productType] |
| PositionInfoAdapter | **com.calypso.tk.mo.PositionInfoAdapter** | ["convention" \| bookId#productType] |
| PositionAggregationConfig | **com.calypso.tk.mo.PositionAggregationConfig** | ["convention" \| name] |
| **PricerConfig** | com.calypso.tk.marketdata.PricerConfig | ["convention" \| name] |
| **PricingEnv** | com.calypso.tk.marketdata.PricingEnv | ["convention" \| name] |
| PricingParam | **com.calypso.tk.risk.PricingParam** | ["convention" \| name] |
| **PricingParameters** | com.calypso.tk.marketdata.PricingParameters | ["convention" \| name] |
| Product | com.calypso.tk.core.Product | ["CODIFIER-"+dataServerName \| id] |
| ProductCode | **com.calypso.tk.refdata.ProductCode** | ["convention" \| code] |
| **ProductQVarI** | com.calypso.tk.refdata.eXSP.variableI.quotableI.ProductQVarI | ["convention" \| name] |
| **QuoteSet** | com.calypso.tk.marketdata.QuoteSet | ["convention" \| name] |

Table 4-1: CalypsoML to Calypso Objects Reference  (Continued)

| Calypso Object | Class Name (Type) | [Codifier | Code] |
|---|---|---|
| **RateIndexDefaults** | com.calypso.tk.refdata.RateIndexDefaults | ["convention" | currency#name] |
| **RateIndex** | com.calypso.tk.refdata.RateIndex | ["convention" | currency#name#tenor(toString)#source] |
| RatingValuesAdapter | **com.calypso.tk.refdata.RatingValuesAdapter** | ["convention" | agency | type] |
| ReferenceEntity | com.calypso.tk.product.ReferenceEntity | [dataServerName | id] |
| ReferenceEntityBasket | **com.calypso.tk.product.ReferenceEntityBasket** | [dataServerName | id] |
| ReferenceEntityNthDefault | **com.calypso.tk.product.ReferenceEntityNthDefault** | [dataServerName | id] |
| ReferenceEntitySingle | **com.calypso.tk.product.ReferenceEntitySingle** | [dataServerName | id] |
| ReferenceEntityTranche | **com.calypso.tk.product.ReferenceEntityTranche** | [dataServerName | id] |
| **ReportTemplate** | com.calypso.tk.report.ReportTemplate | [dataServerName | id] |
| RiskConfig | com.calypso.tk.risk.RiskConfig | ["convention" | name] |
| Scenario | com.calypso.tk.risk.Scenario | ["convention" | scenarioName] |
| RiskPresenterConfig | **com.calypso.tk.presentation.risk.RiskPresenterConfig** | ["convention" | name] |
| **ScenarioMarketData** | com.calypso.tk.risk.ScenarioMarketData | ["convention" | scenarioName] |
| **ScenarioParam** | com.calypso.tk.risk.ScenarioParam | ["convention" | scenarioParamName] |
| **ScenarioRule** | com.calypso.tk.risk.ScenarioRule | ["convention" | name] |
| **ScheduledTask** | com.calypso.tk.util.ScheduledTask | [dataServerName | id] [a] |
| **SenderConfig** | com.calypso.tk.bo.document.SenderConfig | [dataServerName | id] |
| SensitivityParam | **com.calypso.tk.risk.SensitivityParam** | ["convention" | name] |
| **SettleDeliveryInstruction** | com.calypso.tk.refdata.SettleDeliveryInstruction | [dataServerName | id] |
| SimulationParam | **com.calypso.tk.risk.SimulationParam** | ["convention" | name] |
| **StaticDataFilter** | com.calypso.tk.refdata.StaticDataFilter | ["convention" | Name] |
| **TaskWorkflowConfig** | com.calypso.tk.bo.TaskWorkflowConfig | [dataServerName | id] |
| TemplateInfo | com.calypso.tk.core.TemplateInfo | ["convention"| user#productType(getType)#name] |
| **Ticker** | com.calypso.tk.product.Ticker | ["convention"| Name] |
| **TradeBundle** | com.calypso.tk.core.TradeBundle | ["convention"| type#name] |
| **TradeFilter** | com.calypso.tk.mo.TradeFilter | ["convention"| Name] |
| Trade | com.calypso.tk.core.Trade | ["CODIFIER-"+dataServerName| id] |

Table 4-1: CalypsoML to Calypso Objects Reference  (Continued)

| Calypso Object | Class Name (Type) | [Codifier | Code] |
|---|---|---|
| **TradeWinClassConfig** | com.calypso.tk.refdata.TradeWinClassConfig | ["convention"| configName] |
| **User** | com.calypso.tk.refdata.User | ["convention"| Name] |
| UserAccessPermission | **com.calypso.tk.refdata.UserAccessPermission** | ["convention"| Version] |
| UserConfiguration | **com.calypso.tk.refdata.UserConfiguration** | ["convention"| userName] |
| UserWorkflowConfig | **com.calypso.tk.bo.UserWorkflowConfig** | ["convention"| user#name] |
| VolatilitySurface3D | com.calypso.tk.marketdata.VolatilitySurface3D | ["convention" | type#currency#name]+ [datetime] [dataServerName | id] (com.calypso.jaxb.xml.MarketDataIdentifiers includes date for identification) |
| **VolSurfaceUnderlying** | com.calypso.tk.marketdata.VolSurfaceUnderlying | [dataServerName | id] |
| CommodityVolPointGenerator | **com.calypso.tk.product.CommodityVolPointGenerator** | ["convention"| Name] |
| VolSurfaceUnderlying | **com.calypso.tk.marketdata.VolSurfaceUnderlying** | [dataServerName | id] |
| VolSurfaceUnderlyingCap | **com.calypso.tk.marketdata.VolSurfaceUnderlyingCap** | [dataServerName | id] |
| VolSurfaceUnderlyingSwaption | **com.calypso.tk.marketdata.VolSurfaceUnderlyingSwaption** | [dataServerName | id] |

a. Support for this object is forthcoming.

# Contacting Calypso

## Americas

### SAN FRANCISCO

595 Market Street
Suite 1800
San Francisco, CA 94105
USA

T +1 415 817 2400
F +1 415 284 1222
info@calypso.com

### NEW YORK

99 Park Avenue,
Suite 930
New York, NY 10016
USA

T +1 212 905 0700
F +1 212 905 0724
sales_nyc@calypso.com

## Europe, Middle East, Africa

### LONDON

17 Dominion Street
London EC2M 2EF
United Kingdom

T +44 20 7826 2500
F +44 20 7826 2501
sales_london@calypso.com

### PARIS

106/108 rue de La Boëtie
75008 Paris
France

T +33 1 44 50 13 99
F +33 1 44 50 12 84
sales_paris@calypso.com

### FRANKFURT

Junghofstrasse 24
2nd floor
60313 Frankfurt Am Main
Germany
T +49 69 920389 0
F +49 69 920389 11
sales_copenhagen@calypso.com

### JOHANNESBURG

Nelson Mandela Square
2nd Floor, West Tower
Maude Street
Sandton, 2196
South Africa
T +27 11 881 5708
F +27 11 881 5611
sales_johannesburg@calypso.com

### Copenhagen

Calypso Technology Denmark
Regus City Center, Office 308
Larsbjornstraede 3,1454 Copenhagen
T +45 33 37 71 79
F +45 33 32 43 70
sales_copenhagen@calypso.com

# Asia Pacific

## TOKYO

Urban Building 10F
2-3 Kojimachi
Chiyoda-ku
Tokyo 102-0083
Japan
T +81 (0)3 5214 1800
F +81 (0)3 5214 1801
sales_tokyo@calypso.com

## SINGAPORE

1 Phillip Street, #12-02
Singapore 048692
Singapore
T +65 6372 1121
F +65 6372 1161
sales_singapore@calypso.com

## SYDNEY

Level 4, 95 Pitt Street
Sydney, NSW 2000
Australia
T +61 2 8249 8115
F +61 2 8249 8116
sales_sydney@calypso.com

## MUMBAI

Unit 401, Akruti Center Point
MIDC Central Road
Near Marol Tel. Exchange
MIDC, Andheri (E)
Mumbai 400 093
India
T +91 22 6681 5600
F +91 22 6681 5611
sales_india@calypso.com