syncsort

Trillium Software System™

Repository Administrator’s Guide

Version 15.7

Trillium for Microsoft Dynamics CRM User's Guide
August 2018, 8118

Trillium Software, Inc
1700 District Avenue, Suite 300, Burlington, MA 01803, USA
978-901-0000 (Worldwide Headquarters)
+44 (0) 118-940-7676 (European Headquarters)

Contents

Preface

This Repository Administrator's Guide describes the Trillium Software System™ **Repository Manager** and **Repository Administration Command Prompt** applications. The Repository Manager application is the TSS graphical user interface for managing repositories, TSS users and data connections, and for performing other system tasks. The Repository Administration Command Prompt is a command line utility. Most of the commands available in the Repository Manager can also be performed from the Repository Administration Command Prompt.

Intended Audience

This document is intended for administrators and advanced TSS users who will manage TSS repositories, users and connections to data sources. The document presumes the repository administrator is familiar with database structure and with standard expression language.

The information in this guide describes the procedures for setting up Trillium Software System for general users, data analysts and profilers, corporate managers, and database administrators. Each of these user types depends on the completion of the tasks in this guide before they can begin their work.

Related Documents

Refer to the following documents to learn about TSS:

- Trillium Software System *Help*, which provides detailed reference information for any user activity you may need to perform in the Control Center client application.
- Administration Center *Help*, which describes how to perform administrative tasks for Discovery Center and Control Center users.
- *TSS Beyond the Basics* - provides supplemental information, sample tutorials and in-depth analysis of key TSS concepts and processes

Access the TSS *Help* and *Beyond the Basics* from the **Help** option (?) in the Control Center and Repository Manager. You can also open them from the

Windows Start menu (**Start > Programs > Trillium > TSS 15> Documentation**).

The Administration Center help is available in the Administration Center interface by clicking **Help > Open Online Help**.

You may also want to visit the Trillium website at:

`http://www.trilliumsoftware.com`

The website is a comprehensive resource library of white papers, case studies, product descriptions, solution summaries, webcasts, and related information.

ⓘ *If you find errors in this or any Trillium Software documentation, have suggestions for additional topics, or have any other comments, please let us know. You can contact the Documentation team through Trillium's Customer Support department.*

Customer Support

Trillium Software Customer Support is available Monday through Friday, 24 hours a day, to help you with any questions you may have about the Trillium Software System. This service is available to all customers who have active Trillium Software maintenance plans.

- **E-mail**: **TrilliumSupport@syncsort.com**
- **Tel**: **978-901-0000 (Worldwide Headquarters)**
+44 (0) 118 940 7676 (European Headquarters)

CHAPTER 1

Introduction

This chapter introduces you to the Trillium Software System (TSS) Repository Manager and its interface. It includes the following sections:

Overview

The Repository Manager is a secure application that gives the repository administrator access to all repository servers and allows secure management of repositories, imported data, and data source connections. It has a separate user interface and requires an administrative ID and password—set during installation—to log on.

The administrator also uses the Repository Manager to set up user profiles and to create loader connections for data sources.

ⓘ *Administrators can also add and manage repositories and loader (data) connections in the Administration Center browser-based application.*

Repositories

A repository stores data and metadata (information about your data). Using TSS, you can discover the following types of information about your data:

- Data structures, contents, and relationships
- Data compliance with business rules
- Data statistics, drill-down details, and data patterns
- Data trends and changes over time
- Data quality processing and results

- Documentation of data observations, compliance issues, and more

You can create one or more repositories to store your data. Each repository belongs to a specific repository server.

Repository Server

A repository server is a collection of one or more repositories. At the time of installing the Control Center, you created the primary repository server. You can create an additional server during installation, or you can create multiple repository servers after installation.

Repository Administrator

The repository administrator is the TSS user who is responsible for creating and maintaining repository servers, repositories, users and loader connections. This user is also known as the repository server boot user.

The repository administrator account is established during system installation. The administrator account can be from a Windows administrative group or can be a TSS-only user account that was created during TSS installation.

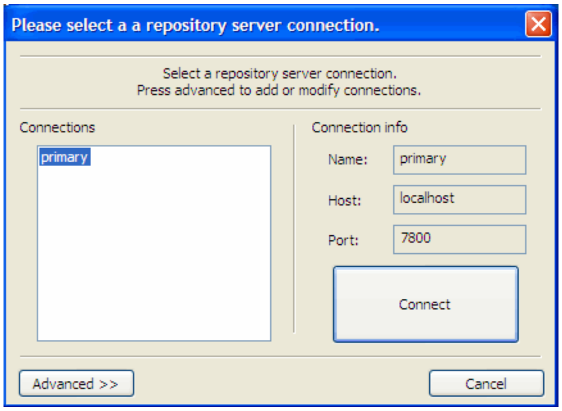
ⓘ *Only a TSS user with administrator permission can log on to the Repository Manager.*

Starting the Repository Manager

You can access the Repository Manager from the Windows **Start** menu.

▶ **To start the Repository Manager**

1. From the Start menu, select **Start > All Programs > Trillium Software> TSS 15 > Repository Manager**. The Connection window opens.



Please select a a repository server connection.

Select a repository server connection.
Press advanced to add or modify connections.

Connections: primary

Connection info

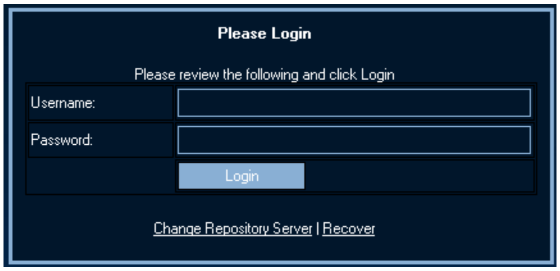
Name: primary

Host: localhost

Port: 7800

Figure 1-1 Connection Window

2. From the Connections list, select a connection name (default is primary) and click **Connect**.



Please Login

Please review the following and click Login

Username:

Password:

Login

Change Repository Server | Recover

Figure 1-2 Repository Manager Login Window

If you are using Windows authentication, your login credentials are verified and you are automatically logged in. If you are using Trillium authentication, the Repository Manager Login Window opens.

ⓘ *You may be prompted to login even when using Windows authentication if there were issues connecting to the repository server. Once you establish the connection however, you will not have to repeat the procedure.*

3. In **Username**, enter the name of the repository administrator.
4. In **Password**, enter the password for the repository administrator.
5. Click **Login**. The Repository Manager opens.

You can return to the Repository Manager login window at any time to change users, repositories, or repository servers.

▶ To return to the login window:

From the Application menu, select **Switch User or Repository** or **Switch Repository Server**.

Repository Manager User Interface

The Repository Manager user interface has the following elements:

- Application button/Application menu
- Quick Access Toolbar
- Command tabs and Command sets
- Getting Started Wizard
- Navigation View
- List View
- Message View
- Status Bar

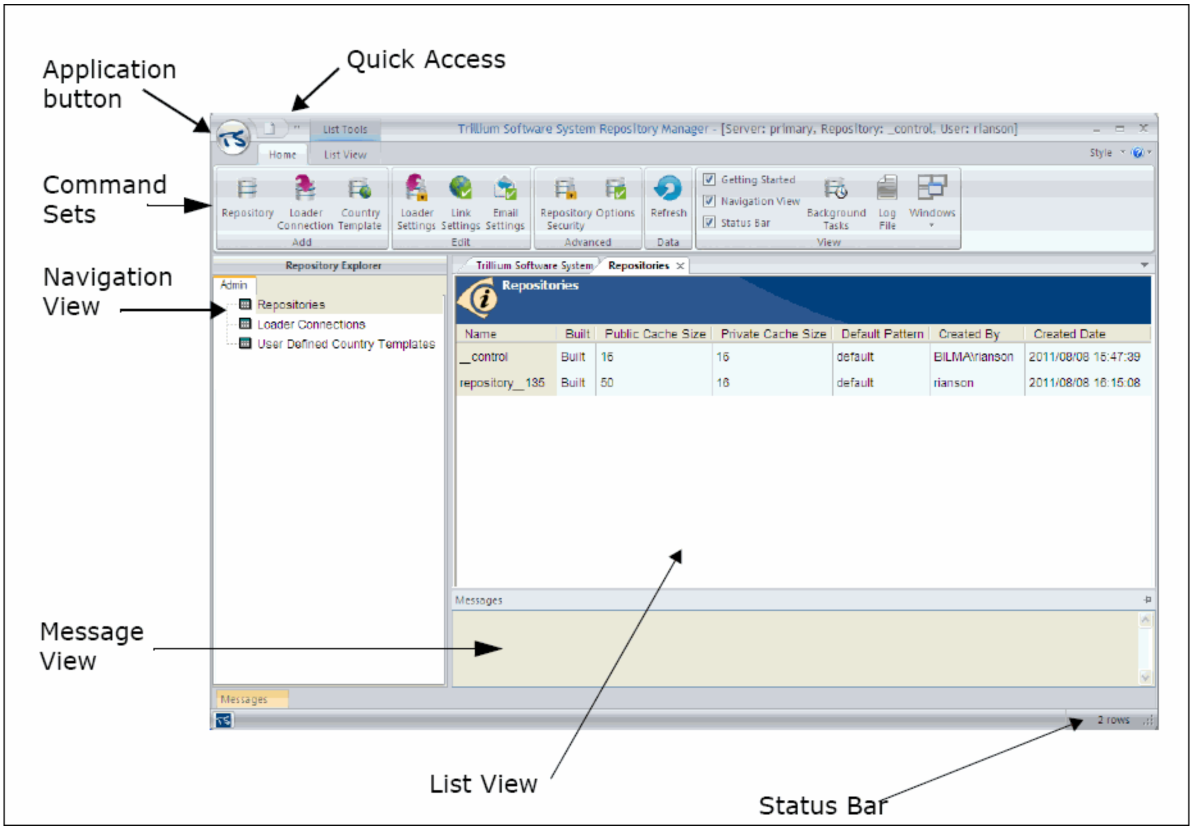
ⓘ *You can use Home tab commands to turn off the display of the Getting Started Wizard, Navigation View, or Status Bar.*

Application Button/Application Menu

The **Application** button in the upper left corner of the user interface opens the Application menu. This menu contains commands that apply to the Repository Manager session, such as environment settings, active repository server, and so on. The right pane of the application menu reflects the command option you selected. For example, if you select the **Open** command, a list of recent sessions appears in the right pane.

Quick Access Toolbar

The Quick Access Toolbar contains frequently used commands that apply to the Repository Manager session. You can customize the tool bar by adding or deleting commands or by repositioning the commands.



Trillium Software System Repository Manager - [Server: primary, Repository: _control, User: rianson]

Name	Built	Public Cache Size	Private Cache Size	Default Pattern	Created By	Created Date
__control	Built	16	16	default	BILMA\rianson	2011/08/08 15:47:39
repository__135	Built	50	16	default	rianson	2011/08/08 16:15:08

Figure 1-3 Repository Manager Main Window

Command Tabs and Command Sets

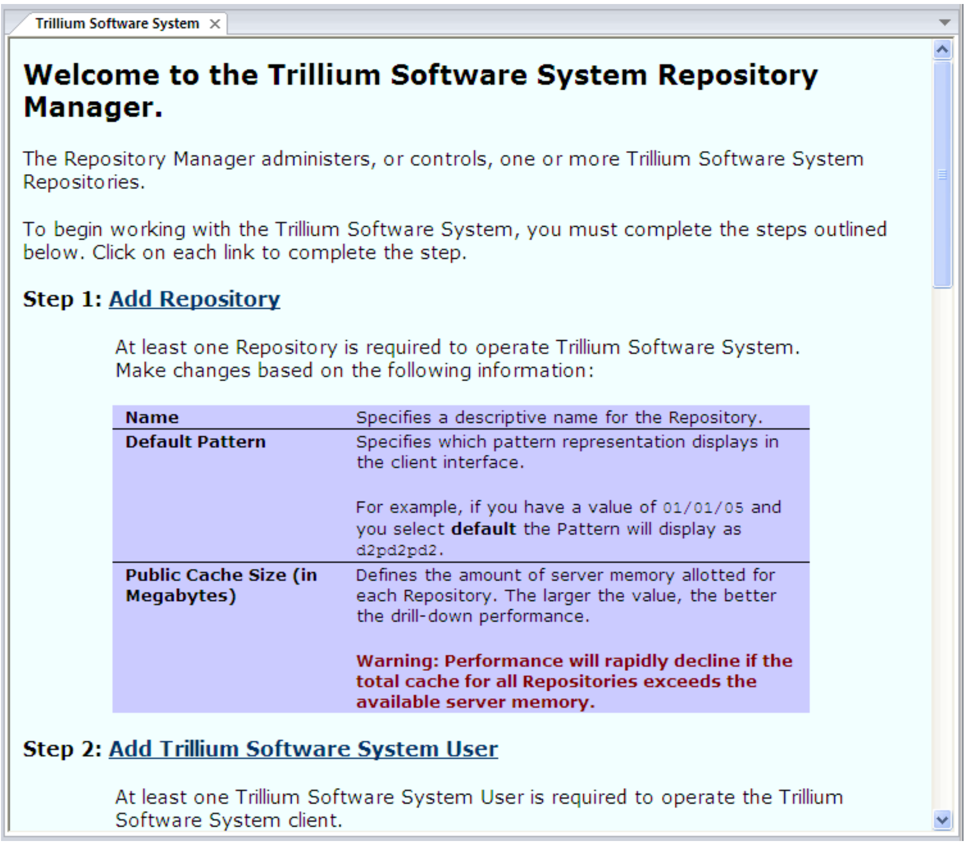
Command tabs relate to the current task in the Repository Manager. The Home tab is always displayed. When you open a List View, the List View tab appears. When you close the List View, the List View tab also closes.

Within each command tab are several command sets, organized by function. You click the icons in the command sets to execute commands, open windows, or display menus.

Getting Started Wizard

To configure TSS, the repository administrator must first add and configure repositories. If you are a first time user, consult the Getting Started Wizard (Figure 1-4) where you can find at a glance, the instructions to configure a repository, add users and loader connections, and modify settings. The wizard also links you to the appropriate window to start each process.

ⓘ *You will see the Getting Started Wizard when you first open the Repository Manager.*



Welcome to the Trillium Software System Repository Manager.

The Repository Manager administers, or controls, one or more Trillium Software System Repositories.

To begin working with the Trillium Software System, you must complete the steps outlined below. Click on each link to complete the step.

Step 1: Add Repository

At least one Repository is required to operate Trillium Software System. Make changes based on the following information:

Name	Specifies a descriptive name for the Repository.
Default Pattern	Specifies which pattern representation displays in the client interface. For example, if you have a value of 01/01/05 and you select **default** the Pattern will display as d2pd2pd2.
Public Cache Size (in Megabytes)	Defines the amount of server memory allotted for each Repository. The larger the value, the better the drill-down performance. **Warning: Performance will rapidly decline if the total cache for all Repositories exceeds the available server memory.**

Step 2: Add Trillium Software System User

At least one Trillium Software System User is required to operate the Trillium Software System client.

Figure 1-4 Repository Manager Getting Started Wizard

▶ To use the Getting Started Wizard

1. Open the Repository Manager. The Welcome page (Getting Started Wizard) opens. Alternately, click the **Getting Started** check box on the Home tab to open the wizard.
2. Locate the task you want to perform. You can, for example, add a repository and or user, assign user privilege and also add loader connections.
3. Click a link to begin a task. This will launch the appropriate window. Complete the task following the instructions in the Wizard.

Navigation View

You use the tree in the Navigation View to quickly drill down to view a list of repositories and loader connections. You can also use the Navigation View to *create* new repositories and loader connections and to *edit* loader connections. The following chapters provide detailed information on performing these tasks.

ⓘ *The objects displayed in the Navigation View may vary, depending on whether your are using Trillium authentication or Windows authentication.*

List View

The List View displays details about the object you selected in the Navigation View. You can customize the way the data is displayed in a number of ways, as explained in the following sections.

Opening Multiple List Views

You can have multiple lists open at the same time. For example, you might want to display a filtered list of users and a complete list of users, or a list of all users and a list of repository users.

▶ To display multiple lists

1. On the Home tab, click **Windows** and then select **New Window**.
2. From the Navigation View, double-click the object you want to display in the window.
3. Repeat steps 1 and 2 for each list you want to display.

▶ To switch between views

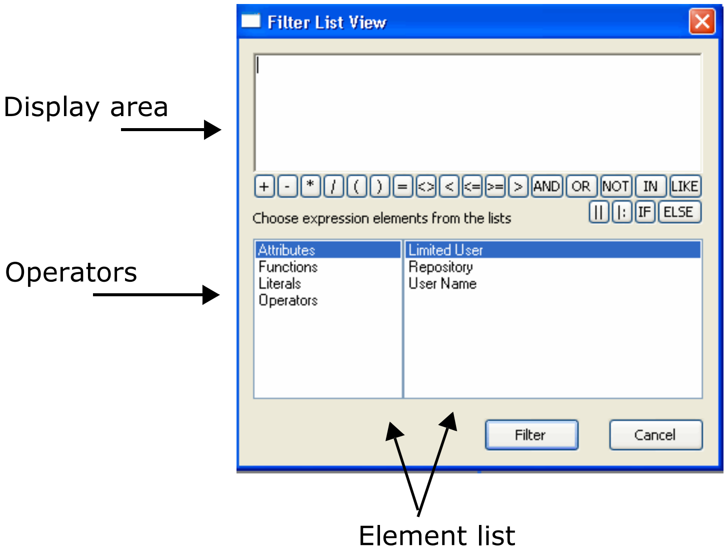
To activate a different list, click on the appropriate tab at the top of the List View pane. Tabs are labeled so you can easily identify the list you want.

Filtering

If the number of entries in the List View is long, you may want to filter the view to display only the entries that are of interest.

▶ To filter the entries in the List View

1. From the Navigation View, double-click the object you want to review.
2. On the List View tab, click **Filter**, or right-click in the List View and select **Filter**. The Filter List View window opens.



Choose expression elements from the lists

Figure 1-5 Filter Window

3. Construct the filter expression. The Repository Manager filters use standard expression language.

   a. In the first element list, select **Attribute**. The second element list refreshes to display the list of all the attributes that are part of the current list. (In this context, the attribute is used as the column name in the List View.)

   b. Double-click the attribute on which you want to base the filter. For example, double-click **User Name**. The attribute name appears in the display area of the window.

   c. In the first element list, click **Functions** or **Operators**. The Filter List View window refreshes to display a third element list that contains categories of functions or operations.

   d. Double-click the operator or function that is appropriate for the expression you are defining. For example, select the equals sign (**Operators > Comparison > =**).

ⓘ *You can also select an operator by clicking its icon in the middle of the window.*

e. Complete the expression by entering a literal value in the display area, or using the expression builder to create the value. Continuing with the same example, type "Joey" in the display area.

   You now have an expression that looks like this:

   ```
   [User Name] = "Joey"
   ```

f. (Optional) Use a logical operator to connect the expression you just created to another one.

ⓘ *When building a complex filter expression, it is a good idea to make a copy of it periodically. Select the expression and press Ctrl+C. If you need to backtrack to the saved version or use it as a basis for another filter, you can paste it into the display area of the Filter List View window.*

4. Click **Filter**. The List View is updated to display only the entries that meet the conditions of the filter expression.

Sorting

You can change the order in which rows are displayed by changing the sort order. The Repository Manager lets you sort by the values in a column or by the data length of a column. The sort can be in ascending or descending order.

ⓘ *After you sort a column, an arrow appears next to the column name. The arrow points up for ascending order and down for descending order.*

▶ To change the sort order of a list by value

1. Right-click the label of the column on which you want to base the sort.
2. Select **Sort > Ascending or Sort > Descending**.

   ⓘ *Click the column label to toggle between ascending and descending order.*

▶ To change the sort order of a list by attribute length

1. Right-click the label of the column on which you want to base the sort.
2. Select Sort by Length > Ascending or Sort by Length > Descending.

▶ To base the sort on multiple columns

1. Right-click any column label.
2. Select **Multi-Column Sort**. The Multi-Column Sort window opens.

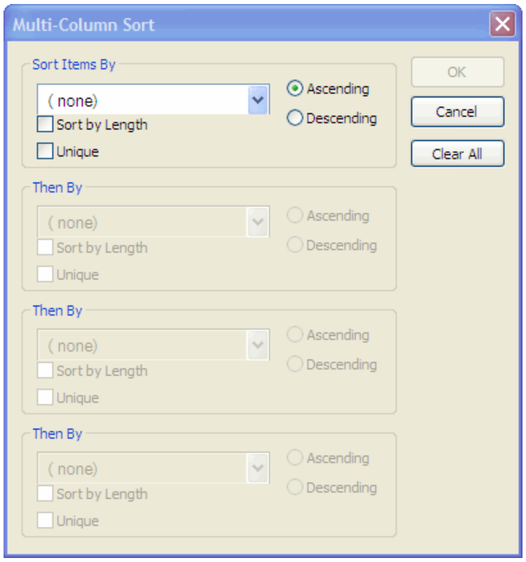


Figure 1-6 Multi-Column Sort Window

3. In the Sort Items By field, specify the following information for the *primary* sort key:

- Column name
- Direction (ascending or descending)
- Type of sort: by default, Repository Manager sorts by value; if you want to sort by data length, click the Sort by Length check box.
- Uniqueness: Check the Unique check box if you want the list to display only rows that contain unique values in this column.

4. Repeat step 3 for each level of sort you want to apply.

5. Click **OK**.

Message View

The Message View displays information about repository activity. By default, the Message View is hidden.

▶ To display the Message View

Hover over the Messages tab at the bottom of the window. The Messages View opens. It closes again as soon as you move the pointer outside the Message View pane.

▶ To keep the Message View open

1. Hover over the Messages tab to open the view.
2. Click the Auto-Hide icon in the upper right corner of the view. Message View remains open until you repeat step 2 to restore the Auto-Hide function.

ⓘ *You can resize the Message View pane by dragging the top border.*

CHAPTER 2

Repositories

This chapter provides instructions for managing Trillium Software System repositories and includes the following topics:

Only the Repository Manager can add and delete users or modify user permissions.

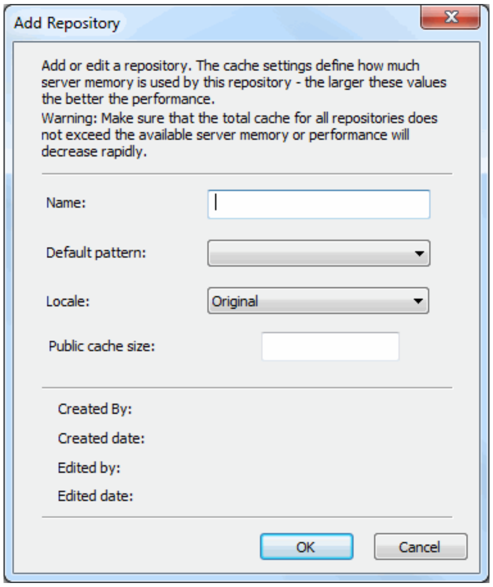
Adding a Repository

You must first create one or more repositories to enable users to process data in TSS. You must create a minimum of one repository.

Each repository server can hold multiple repositories. You can add a new repository at any time.

▶ To add a repository

1. Log in to the Repository Manager.
2. In the Navigation View, right-click **Repositories** and select **Add Repository**. Alternately, on the **Home** tab and click **Repository**. The Add Repository window opens.



Add Repository

Add or edit a repository. The cache settings define how much server memory is used by this repository - the larger these values the better the performance.

Warning: Make sure that the total cache for all repositories does not exceed the available server memory or performance will decrease rapidly.

Name:

Default pattern:

Locale: Original

Public cache size:

Created By:

Created date:

Edited by:

Edited date:

Figure 2-1 Add Repository Window

3. In **Name**, enter a descriptive name for the repository, using a combination of lowercase English alpha characters and numbers so that they are acceptable to all file systems. Symbols and non-English characters are not valid in repository names as these names are also used as directory names. Once the repository creation process is complete, you cannot rename the repository.

4. In **Default pattern**, select the default pattern for data using the information in the following table. The patterns help create a shorthand version of the data and are useful in identifying data deviations.

Table 2-1 Data Patterns

Pattern	Examples
default	Jane Smith would be displayed as a4_a5; 05/31/2008 would be displayed as d2pd2pd4.
rich	Jane Smith would be displayed as ul3_ul4; 05/31/2008 would be displayed as d2Sd2Sd4.
long	Jane Smith would be displayed as AAAA AAAAA; 05/31/2008 would be displayed and NN/NN/NNNN.
Greek Hebrew Turkish	Use default pattern style.

ⓘ *Refer to Pattern Styles on page 143 for more information about pattern styles.*

5. In the **Locale** field, select a desired locale (your unique language or region identifier) for your collation method. You can also keep the default option **Original** if you want to use the non-collated sort method.

Once you select a locale, you cannot change your selection. This prevents compatibility issues, and also allows you to upgrade existing repositories without corrupting them.

6. In **Public cache size**, specify in MB the server memory for the repository. This value is site-specific. In general, the larger the value, the better the drill-down performance, but we do not recommend that you use a value larger than 512 MB.

   ⓘ *Performance will rapidly decline if the total cache size for all repositories exceeds available server memory.*

   The four fields at the bottom of the Add Repository window are automatically filled in and do not require manual configuration.

7. Click **OK** to add the repository.

Note About Using Locale in Control Center

TSS Control Center has the capability to display numeric values conforming to locale settings in the repository server. For example, if you set, the locale option as German, you will see the numeric value of one thousand displayed as 1.000,00.

Additionally, the Control Center coverts localized numbers to computer readable format prior to numeric operations and then converts them back to the localized format for display. As a result, non-computer readable numbers with periods and comma, like (1,000.00) or German (1.000,00) are converted to standard computer readable number string for numeric processing, and then converted back based on the locale in use.

ⓘ *The Control Center uses standard locale library ICU v4.4.*

Editing a Repository

Once you have created a repository, you can change the pattern and cache size associated with it. You cannot rename a repository. Also, you cannot edit the _control repository.

▶ To edit a repository

1. Log in to the Repository Manager.
2. View the log file to ensure that no one is using the repository you want to edit.
3. In the Navigation View, double-click **Repositories**. A list of all repositories defined in this repository server opens in the List View.
4. Right-click the repository name and select **Edit.**
5. Modify the **Default pattern** and **Public cache size** fields as needed.

ⓘ *Changing the cache size should be done with caution. We recommend that you back up the repository before making this change.*

6. Click **OK**. If you change the **Public cache size** setting, you must perform a recovery procedure to activate it. See To activate a new Public cache size setting for more information.

▶ To activate a new Public cache size setting

1. Ensure that all users are logged off the Control Center.
2. From within the Repository Manager, click the **Application** button and select **Switch User or Repository**.
3. In the Login window that opens, enter the username and password.
4. Click **Recover**.

Deleting a Repository

Follow the procedure to delete a repository that is no longer in use. You cannot delete the _control repository.

1. Log in to the Repository Manager.
2. View the log file to ensure that no one is using the repository you want to edit.
3. In the Navigation View, double-click **Repositories**. A list of all repositories in this repository server displays.
4. Right-click a repository name and select **Delete...**.
5. Verify that the repository name is no longer listed.

Creating an Alternate Repository

The data files created by a Quality project can be very large. It is recommended that you create an alternative area for just those files. Everything except the data files remains in the existing repository server directory.

⚠ *You must create the alternative repository* ***before*** *you create any projects for that repository in the Control Center.*

▶ To define an alternative repository directory

1. Shut down TSS, including all Control Centers.
2. If the new repository does not yet exist, create it now using the Repository Manager.

3. Stop the TSS Scheduler service.

4. Configure as shown below:

   - On Windows, click **Start > Programs > Trillium > TSS 15 > Repository Administration Command Prompt** to start the Repository Application Command Prompt.
   - On UNIX, enter `./mtb_admin`.

5. When prompted, log on as the repository administrator to the repository where you want to define the alternative repository.

   ⓘ *The chosen repository must not contain any entities or projects. Also, if you choose the _control repository, it will set the alt_ metabase for all repositories.*

6. Verify that an alternative repository location has not already been defined by entering the following command:

   ```
   define alt_metabase
   ```

   The command should return the following failure message: `"No setting with name 'alt_metabase' found in table 'default_settings'."`

   If a value is returned, remove the current setting using instructions in Removing Alternative Repository Definition.

7. Configure the alternative repository directory.

   a. At the mtb_admin prompt, type:

   ```
   expert
   ```

   b. Define the alternative location by entering the following command `define alt_metabase [file nativename {d:\newmetabasedir}]` where *`d:\newmetabasedir`* is the path of the alternative location.

8. Press **Enter**.

9. Type `exit` to close the command prompt window.

10. Restart the Scheduler and the Repository Manager.

11. Create entities and projects in the new repository and verify that data is being stored in the alternate location.

Removing Alternative Repository Definition

Follow the procedure to remove an alternate repository definition.

▶ To remove the alternative repository definition

1. Close all TSS applications.
2. Stop the TSS Scheduler service.
3. On Windows, click **Start > Programs > Trillium > TSS 15 > Repository Administration Command Prompt** to start the Repository Application Command Prompt.

   Alternately, on UNIX, enter `./mtb_admin`.
4. When prompted, log on as the repository administrator to the _control repository.
5. Enter the following command: `undefine alt_metabase`
6. Press **Enter**.

CHAPTER 3

Users

This chapter provides instructions for managing Trillium Software System users. TSS supports two mutually exclusive user categories:

- Windows-authenticated users
- Trillium-authenticated users

The user category for your environment is defined during the installation of the Repository Server and cannot be changed. This chapter includes the following topics:

- Configuring Windows-authenticated Users on page 26
- Configuring Trillium-authenticated Users on page 31
- Managing User Access from the Control Center on page 35

Configuring Windows-authenticated Users

This section includes:

- Overview
- Before You Begin
- User Permissions
- Adding Windows-Authenticated User
- Modifying Windows-authenticated User
- Deleting Windows-authenticated User

Overview

Windows Authenticated users have valid Active Directory accounts. Windows Authentication gives users the permission to read, create, modify, execute, delete, and export projects, entities, business rules, and Business Rules Library objects in one or more repositories.

There are several advantages to using Windows-authenticated users for TSS:

- **Improved security**. A multi-layered system of permissions is available.
- **Ease of configuration**. No need to create user IDs and passwords. You can directly select TSS users from those listed in Active Directory. Optionally, you can also select a user group.
- **Single sign-on**. Users can access the Control Center without having to enter their user ID and password.

Before You Begin

Before you configure access permissions, develop a plan by answering the following questions:

- **Which users or groups in the Windows Active Directory will use the Control Center?** The Active Directory typically contain hundreds or thousands of users. Make a list of those who will be using the Control Center.
- **What data does each user need to access?** Does the user need to work with all repositories or a single repository? For example, a database analyst might need access to multiple the repositories, while a salesperson may just need to access to say, the Prospects repository.

  ⓘ *The repository administrator has access to all repositories.*
- **What tasks will the user perform?** Will the user create, edit and run projects or does the user needs read-only permission.

User Permissions

By default, all TSS users have read-only permission. You can also grant some permissions on the *repository server level* and others on a *repository level*. For simplicity, we recommend that you add users and establish permissions at the repository level instead of the server level. Assign repository server level permissions to repository administrators and other management staff. Table 3-1 describes the permission options available for each user or user group.

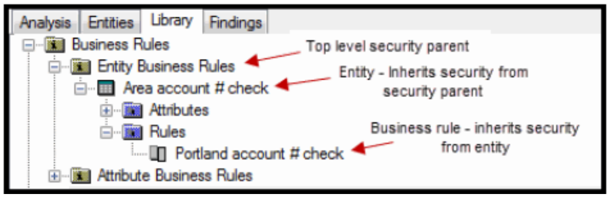
ⓘ *Permissions can be set at the repository level in the Repository Manager and at the object level in the Control Center.*

To avoid overwrite of work resulting from concurrent access to the same repository object by multiple users with the same permission level, it is recommended that you verify that the object is not being accessed by other users.

Table 3-1 Permission Options for Windows-Authenticated Users

Permission	Description
Create	Allows users to create projects, entities, business rules, and Business Rules Library objects in the Control Center. If this is denied on a parent object, such as an entity, then you would not be allowed to create a child object, such as a business rule. ⓘ *Disable the create permission if the user should not convert a view to an entity in the Control Center.*
Read	Allows users to view projects, entities, business rules, and Business Rules Library objects in the Control Center. ⓘ *By default, all TSS users have read-only permission.*
Edit	Allows users to modify projects, entities, business rules, and Business Rules Library objects. Users can also enable and disable business rules from the List View right-click menu.
Execute	Allows users to run any project or business rule analysis.
Delete	Allows users to delete projects, entities, business rules, and Business Rules Library objects.
Export	Allows users to run the Control Center's deployment tools to export a project to batch or real time. Users can also export library entities and library attributes that contain library rules.
Permissions	Allows users to give or deny permission to other users for an individual project, entity, business rule, or Business Rules Library object. The type of access (read, edit, and so on) is established on a per-object basis in the Control Center. If you do not have this privilege, you cannot modify permissions for other users. ⓘ *Permission is automatically granted to users for any project, entity, business rule, or Business Rules Library object they create in the Control Center.*
Special Permissions	Allows users to access advanced security settings to change the rule/object owner, add a user to the rule/object, and modify their permissions. ⓘ *Special permissions are automatically granted to users for any project, entity, business rule, or Business Rules Library object they create in the Control Center.*

Default permissions are based on the hierarchy of objects in the Navigation View. For example, the permission for a library entity business rule is inherited from its parent library entity, which in turn inherits permission from the Entity Business Rules entity folder.



Top level security parent

Entity - Inherits security from security parent

Business rule - inherits security from entity

Changing the permissions of a parent object affects children objects. For example, if permission is modified on an entity, then permissions on all business rules associated with that entity also change.

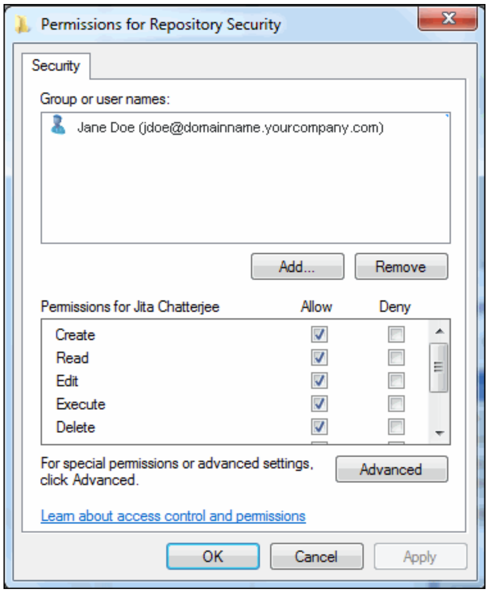
Adding Windows-Authenticated User

Use the following instructions to add Windows-authenticated user or group.

▶ To add a Windows-authenticated user or user group

1. Start the Repository Manager.
2. To **add users for a specific repository**, double-click **Repositories** to show a list of repositories. Right-click the repository to which you want to add users and click **Security**.

   To **add users with access to all repositories**, on to the **Home** tab click **Repository Security**. The Permissions for Repository Security window opens.


Permissions for Repository Security
Security
Add...
Remove
Advanced
OK
Cancel
Apply
Learn about access control and permissions

Group or user names:

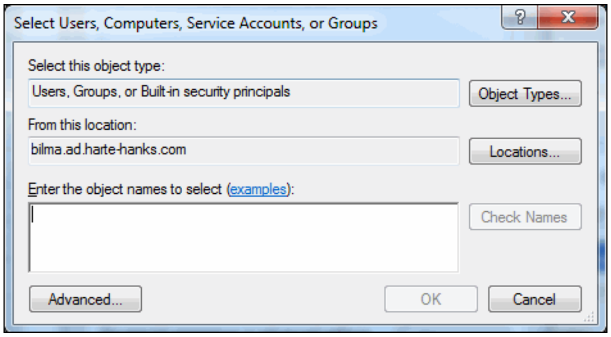
Jane Doe (jdoe@domainname.yourcompany.com)

Permissions for Jita Chatterjee	Allow	Deny
Create	☑	□
Read	☑	□
Edit	☑	□
Execute	☑	□
Delete	☑	□

For special permissions or advanced settings, click Advanced.

Figure 3-1 Windows-Authenticated Users

3. Click **Add**. The Select Users, Computers, Service Accounts or Groups window opens.



Select this object type:

Users, Groups, or Built-in security principals

From this location:

bilma.ad.harte-hanks.com

Enter the object names to select (examples):

Figure 3-2 Select User or Group

4. In the **Enter the object name to select** text box, enter the name of the user or the group you want to add and click **Check Names**. The system verifies the user or the group exists in the Active Directory and returns domain name.

5. Click **OK** to add the user or group.

6. Repeat the steps 4 and 5 to add more user or group if necessary. The user and groups or users you added are listed in the Groups or user name section of the Repository Security window.

7. For each user or group you added, assign permissions. The default permission is Read. Consult Table 3-1 on page 28 for a list of all available permissions.

8. Click **OK** to close the Repository Security window.

Modifying Windows-authenticated User

Follow the instructions to modify Windows-authenticated user permissions.

▶ **To modify user permissions**

1. Refer to the instructions to create a user.

2. Modify as necessary and save your changes.

Deleting Windows-authenticated User

The repository administrator can delete a user who no longer needs access to the Control Center. When deleting users, you must delete users at the parent level. Therefore, if you have added a user at the *repository server* level and at the *repository* level, delete the user from the repository server level.

ⓘ *The deleted user or user group loses the ability to access the Control Center. They are not physically removed from the Active Directory list.*

▶ **To remove a repository-specific user or user group**

1. Log on to the Repository Manager.

2. Display the list of repositories and select the repository in which the user or group was created.

3. Right-click and select **Security** from the pop-up menu. The Repository Security window opens with the names of groups and users that have been added to the repository.

4. Select a group or user name and click **Remove**. The **Remove** button is grayed out if the selected user or group was added at the repository server level. To delete these users, follow the instructions in To remove a user or user group from the repository server on page 31.

5. Repeat Step 4 to remove additional users.

6. Click **OK** to save your changes.

▶ To remove a user or user group from the repository server

1. Log on to the Repository Manager.
2. Select the **Home** tab and click **Repository Security**. The Repository Security window opens with the names of groups and users who have been added to the repository.
3. Users added only at the repository level do not appear in this list. To remove them, follow the instructions in the procedure To remove a repository-specific user or user group.
4. Select a group or user name and click **Remove**.
5. Repeat Step 3 and 4 to remove additional users.
6. Click **OK** to save your changes.

Configuring Trillium-authenticated Users

This section includes:

- Overview
- Adding Trillium-authenticated user
- Modifying Trillium-Authenticated User
- Deleting Trillium-Authenticated User

Overview

When configuring Trillium-authenticated users, consider the following:

- **Security.** You can grant a user full access to a repository or limited access, but there are no other levels of security.
- **Configuration**. You must create a user ID and password for all the users who need access to the Control Center.**Logon.** Users must log on to the Control Center with their Trillium-authenticated user name and password

Adding Trillium-authenticated user

Use the following procedure to add a Trillium-authenticated user.

▶ To add a Trillium-authenticated user

1. Log on to the Repository Manager.

2. In the Navigation View, right-click **Users** and select **Add User**. Alternately, on the Home tab, click **User**. The User Properties window opens.

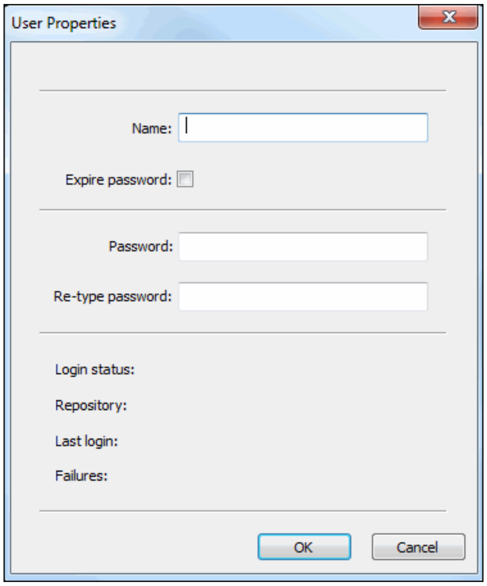


Figure 3-3 User Properties Window

3. In the **Name** field, enter a unique name for the new user. The name can be any combination of alphanumeric characters.

4. Check the **Expire Password** box if you want the user to specify their own password. When you check this option, the password expires the first time the user logs on forcing the user to specify a password. This increases system security since the password is known only to the user.

5. In the **Password** and **Re-type** fields, enter the password you are assigning to the new user.

ⓘ *By default, there are no validation checks, but you can add validation by editing password properties. See Setting Up Password Validation on page 76* for more information.

6. Click **OK**.

7. Repeat the process to add other users.

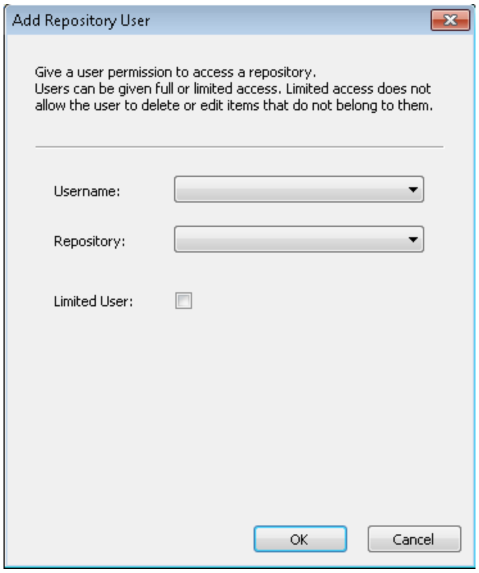
ⓘ *Users at this stage will not have access to the Control Center until you grant them access to at least one repository.*

▶ To grant repository access to users

1. Log on to the Repository Manager.

2. On the Home tab, click **Repository User**. Alternately, in the Navigation View, right-click **Repository Users** and select **Add User to**

Repository from the drop down menu options. The Add Repository User window opens.



Give a user permission to access a repository.
Users can be given full or limited access. Limited access does not allow the user to delete or edit items that do not belong to them.

Figure 3-4 Add Repository User Window

3. From the **Username** drop-down, select a user.

4. From the **Repository** drop-down, select the name of the repository to which user should have access.

   ⓘ *To grant another user repository administrator privileges, select the _control repository.*

5. Optionally, select the **Limited User** box to limit this user's access to entities within the repository. Limited users can only view those entities that they have created or entities that other users give them permission to view. If the limited user is not granted access to an entity, the user cannot see any object involving that entity.

6. Click **OK**.

7. Repeat this procedure to grant the user access to other repositories as needed.

8. Notify the new users of their login credentials, including user name, password, and repositories.

Modifying Trillium-Authenticated User

The following procedures provide instructions on modifying a Trillium - authenticated user's:

- password
- switch repository

- accessing type (full or limited)

Modifying password

▶ To change a Trillium-authenticated user’s password

1. Log on to the Repository Manager.
2. From the Navigation View, double-click **Users** to open a list of users in the List View.
3. Right-click the name of the user and select **Edit** from the drop down menu options to open the User Properties window.
4. Modify the **Password** fields as appropriate and click **OK**.
5. Notify the user of the changes made to his password.

Note about User Properties window

The User Properties window that opens when modifying a user password is similar to the one that opens when you are adding a user, but they have the following differences:

- You cannot edit the **Name** field.
- The Repository Manager has filled in one or more of the following fields if a user has been active:
- Login Status - Indicates if the user is currently logged on to TSS.
- Repository - The name of the repository to which the user is logged on.
- Last Login - Date and time the user last logged in to TSS.
- Failures - The number of failures the user experienced when using TSS.

Removing User Access

▶ To remove a Trillium-authenticated user’s access to a repository

1. From the Navigation View, double-click **Repository Users**.
2. In the List View, select the user-repository combination you want to delete.
3. Right-click and select **Delete** from the drop-down menu options.

Modifying Repository Access Type (Full or Limited)

▶ To change the type of repository access

1. From the Navigation View, double-click **Repository Users**.
2. In the List View, select the user-repository combination you want to change.
3. Right-click and select **Edit** to open the Edit Repository User window.

4. Change the **Limited User** selection as desired.
5. Click **OK** to save your changes.

Deleting Trillium-Authenticated User

You cannot delete the repository administrator. Follow the instructions below to delete all other users.

▶ To delete a user

1. Log on to the Repository Manager.
2. From the Navigation View, double-click **Users** to open a list of all users in List View.
3. Right-click the name of the user you want to delete and select **Delete** from the drop-down menu option.
4. You are prompted to confirm the delete: **Are you sure you want to delete this User?**
5. Click **Yes** to delete the user.

Managing User Access from the Control Center

The repository administrator uses the Repository Manager application to establish and manage users. The repository administrator can also log on to the Control Center application and configure user access to the Control Center.

In addition, users with the appropriate access level can modify other users' ability to access projects, entities, business rules and Business Rules Library objects.

Specifically, from the Control Center a user can:

- Change permissions on an object (this applies to Windows-authenticated users).
- Add or remove a user's access to an entity (this applies to Trillium-authenticated users).

▶ To change security permissions on a repository object

1. In the Control Center Navigation View, right-click an object and select **Security**.

   ⓘ *The Security option is grayed out if you do not have the appropriate permissions. For more information, see Table 3-1 on page 28.*

   A security window opens, listing all users and user groups who have access to the selected repository object.

2. Select a user or group and go to the Permissions section.

3. Select the **Allow** or **Deny** check boxes as appropriate.
4. Click **OK.**
5. Repeat Steps 2 and 3 as necessary to change permissions for other users.

▶ To remove user access to an entity

1. Log on to the Control Center as a user with unlimited access to the repository that contains the entity.
2. From any tab or list displaying entities, right-click an entity name and select **Manage User Access**. The Manage User Access window opens.

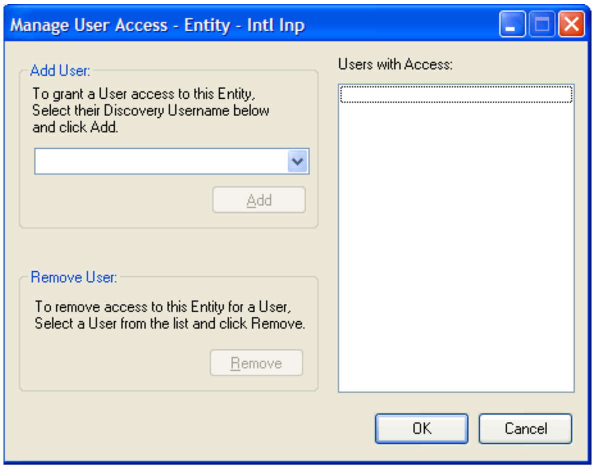

Manage User Access - Entity - Intl Inp
Add User:
To grant a User access to this Entity, Select their Discovery Username below and click Add.
Add
Remove User:
To remove access to this Entity for a User, Select a User from the list and click Remove.
Remove
Users with Access:
OK
Cancel

Figure 3-5 Manage User Access Window

3. Select a name from the **Users with Access** list on the right.
4. Click **Remove**.
5. Repeat steps 2 through 4 to remove other users as needed.
6. Click **OK** to close the Manage User Access window.

▶ To grant a limited user access to an entity

1. Log on to the Control Center as a user with unlimited access to the repository that contains the entity.
2. From any tab or list displaying entities, right-click an entity name and select **Manage User Access**. The Manage User Access window opens.
3. In the **Add User** field, select a user and click **Add**. The user name appears in the **Users with Access** list.
4. Repeat Step 3 for other users if needed.
5. Click **OK.**

CHAPTER 4

Loader Connections

Loader connections enable Trillium Software System to link to an external database and import data. This chapter discusses the supported databases and loader connections, and includes the following topics:

- Overview on page 37
- Default Loader Connections on page 43
- TSS ODBC (Trillium Supplied) Loader Connections on page 43
- Creating a Loader Connection on page 53
- Editing a Loader Connection on page 58

Overview

In order for TSS to connect to an external data source, the repository administrator must set up at least one loader connection for each type of database you will profile. TSS can import or link to both flat file data sources and relational databases.

When you create an entity, you are storing a reference to the associated loader connection data. The reference information is useful when you move the data on disks; you update the loader connection information and the associated entities will continue to work.

ⓘ *You cannot delete a loader connection as that will delete the data connection reference information the entities store. You can disable a loader connection. Once disabled, a loader connection is no longer accessible from the Create Entity Wizard in the Control Center.*

Flat File Data Sources

TSS can import or link to data from any of the following flat file data source:

- Trillium flat files (created by TSS modules when they process data.)

  ⓘ *Duplicate row checksum calculation, sparse row counting, and row length statistics are not supported for Trillium connections.*

- Delimited files
  - With ASCII, extended ASCII, or hexadecimal delimiters
  - With or without ANSI SQL DDL
- COBOL flat fixed length files
  - Data must match copybook layout
  - Various character encoding including ASCII, EBCDIC, and Unicode
  - Big or little Endian byte orders
  - One- or two-byte data alignment
  - Common COBOL copybook features such as:
    - REDEFINES clause
    - OCCURS clause
    - FILLER clause
    - COBOL data types: 9, X, A, B
    - Numeric storage formats: COMP-3, Packed-DECIMAL, COMP, COMPUTATIONAL, or BINARY
    - Comments beginning with *
    - Descriptors such as USAGE IS, DISPLAY, SEPARATE, LEADING, and TRAILING
    - Unnamed fields
    - Unsigned COMP-3 fields

  ⓘ *Copybook features such as Multiple Record Types and OCCURS DEPENDING must be manually removed from the Copybook and the data file prior to loading into the TSS.*

  ⓘ *For more information about supported and unsupported COBOL copybook features, refer to Appendix B.*

Relational Data Sources

If you are adding loader connections to point to relational databases, make sure that your relational databases are properly configured before trying to create entities.

ⓘ *For Oracle connections, 32-bit Oracle libraries are required, 64-bit libraries are not supported.*

TSS can import or link to data from Relational Database Management Systems (RDBMS) with:

- Direct connection to Oracle and DB2
- ODBC-compliant RDBMS connection

  ⓘ *Duplicate row checksum calculation, sparse row counting, and row length statistics for ODBC connections are not supported.*

- RDBMS extraction into a delimited file with a corresponding ANSI DDL.

  ⓘ *NATIONAL data type for RDBMS loads is not supported.*

- TSS-supplied ODBC loader connection (for a list of supported databases and corresponding drivers, see TSS ODBC (Trillium Supplied) Loader Connections on page 43.

Table 4-1 lists supported relational database clients, their version and driver information.

Table 4-1 Supported Relational Databases

Database	Version	Driver
Supported Database Clients for Direct Connections		
DB2	9.1, 8.1 (FixPack 5 or later)	IBM
Oracle[1]	11G, 10G, 9i ⓘ *11G is supported for TSS ODBC Loader Connection only.*	Oracle
Tested ODBC Connections		
DB2	9.1, 8.1 (FixPack 5 or later)	IBM
Informix	10.0 using Informix Connect 3.5	IBM driver
Microsoft Access	Access 2002 with SP3	Microsoft driver
Microsoft SQL Server	SQL Server 2008, SQL Server 2005, and SQL Server 2000 with SP3	Microsoft driver
Oracle	11G, 10G, 9i ⓘ *11G is supported for TSS ODBC Loader Connection only.*	Oracle
Sybase	12.5.x	Sybase driver
Teradata	v13, v12	Teradata driver
ⓘ *ODBC connections to any other database or database version require verification from Trillium Software Customer Support.*		

[1] If you plan to use the Oracle Instant Client with the Oracle (Vendor-Supplied) loader connection, you must choose the Administrator or Runtime installation type when installing the Oracle Client. Other installation types are not supported.

Configure rdbms Key in TSS config.txt File

Configure the rdbms key in the TSS config.txt file to:

- Modify the number of records returned (when fetching data from a RDBMS). See Modifying the Number of Records Returned for details.
- Configure DB2 Adapter on AS/400 Platform. See Configuring DB2 Adapter on AS/400 Platform for details.
- Configure special query acceleration register if you are connecting to DB2 v10 on a Mainframe z/OS and need to load data using SQL queries. See Configuring Current Query Acceleration Special Register for details.
- Enable uncommitted read. See Enabling Uncommitted Read in TSQDA on page 166) for details.

The config.txt file is located in `C:\Program Files x64\Trillium Software\MBSW\15\etc (config.txt)` on Windows and `../Metabase/etc (config.txt)` on UNIX.

Modifying the Number of Records Returned

You can modify the number of records that are loaded when creating an entity pointing to a RDBMS. This is useful when you want to restrict the size of the data that should be fetched.

▶ To modify the number of records that are returned

1. Open ...\`MBSW\15\etc\config.txt` file.
2. Modify the `records_read_odbc` parameter of the key rdbms section. For example:

```
 key rdbms {
    value uncommitted_read "on"
    value records_read 100
    value records_read_odbc 1000
}
```

3. Save your changes. This will return 1000 records.

Configuring DB2 Adapter on AS/400 Platform

When adding DB2 data connections to a DB2 server on AS/400 platform, modify the TSS config.txt file by adding the entry `value iseries 1` to the key rdbms section.

▶ To add DB2 data connection

1. Open `...\MBSW\15\etc\config.txt` file.
2. Add the iseries entry to the key rdbms section.

   For example:

```
key rdbms {
    value uncommitted_read "off"
    value records_read 100
    value records_read_odbc 1
    value iseries 1
}
```

Make this change either before or after you add the loader connection in the Repository Manager, but use the connection only after the change.

Configuring Current Query Acceleration Special Register

If you are connecting to DB2 v10 on a Mainframe z/OS and need to load data using SQL queries, you must add special register CURRENT query acceleration in the TSS config.txt file. The value you specify for this register identifies when DB2 submits dynamic SQL queries to an accelerator server and how DB2 manages accelerator server failures.

In the key rdbms section of the config.txt file, you add the following entry:

```
value db2_current_query_acceleration "value"
```

The following values (as specified by IBM) are supported:

- ALL. Specifies that queries are accelerated if they are eligible for acceleration. DB2 does not use cost information to determine whether to accelerate the queries. Queries that are not eligible for acceleration are not executed by DB2, and an SQL error is returned. If an accelerator failure occurs while a query is running or if the accelerator returns an error, DB2 returns a negative SQLCODE to the application.
- NONE. Specifies that no queries are sent to an accelerator server.
- ENABLE. Specifies that queries are accelerated only if DB2 determines that it is advantageous to do so. If an accelerator failure occurs while a query is running or if the accelerator returns an error, DB2 returns a negative SQLCODE to the application.
- ENABLE_WITH_FAILBACK. Specifies that queries are accelerated only if DB2 determines that it is advantageous to do so. If the accelerator returns an error during the PREPARE or first OPEN for the query, DB2 executes the query without the accelerator. If the accelerator returns an error during a FETCH or a subsequent OPEN, DB2 returns the error and does not execute the query.
- ELIGIBLE. Specifies that queries are accelerated if they are eligible for acceleration. DB2 does not use cost information to determine whether to accelerate the queries. Queries that are not eligible for acceleration are executed by DB2. If an accelerator failure occurs while a query is running or if the accelerator returns an error, DB2 returns a negative SQLCODE to the application.

For more information about query acceleration, see the IBM Knowledge Center.

▶ To configure query acceleration

1. Open `...\MBSW\15\etc\config.txt` file.
2. Add the entry and specify a supported value to the key rdbms section. For example:

   ```
   key rdbms {
       value uncommitted_read "on"
       value records_read 100
       value records_read_odbc 1000
       value db2_current_query_acceleration "ALL"
   }
   ```

3. Save your changes and close the file.

Security Considerations

Before you create a loader connection, verify that security for your databases is adequate and it can prevent unauthorized access. Note the following security guidelines:

- When you create a loader connection, you have the option to select which repositories are allowed to access the connection. This helps restrict user access to certain data sources.
- If you are creating loader connections to SQL Server Open Database Connectivity (ODBC) data types on Windows systems, you can configure these connections for single sign-on that supports a user's Windows system credentials. Also, a user accessing the connection (to create an entity in the Control Center, for example) must be authenticated to access the server machine on which the data source resides.
- To increase security, create separate loader connections to point to different file and database locations.

  For example, assume two data analysis teams work with delimited data sources stored on the same UNIX server. If you create two loader connections, one for each team, to point to the separate file locations, data access by the team members can be streamlined and effectively controlled. Team 1 members will have access to only that data that is relevant for the Team 1 and Team 2 members will have access to only that data that is relevant for the Team 2.

Default Loader Connections

When you install TSS, you are presented with the option of installing the following sample data files:

- Delimited data file
- Trillium data file

To support the sample data files, TSS delivers a default loader connection for each. You can use these connections as is, modify them, or create new versions of them.

TSS ODBC (Trillium Supplied) Loader Connections

The TSS ODBC loader connection allows you to directly connect to an external database without installing any external component. For example, using the TSS ODBC loader connection, you can read from and write to an Oracle database without the Oracle client.

⚠ *The Database Write process will overwrite the database table and will not append or update the data. The attributes in the table that are not selected for mapping will be blanked out. Therefore, the database write operation must always be performed in a staging area and not directly in the original tables.*

For instructions on configuring a TSS ODBC loader connection, see Creating a Loader Connection. TSS ODBC loader connections can be configured on Windows and UNIX systems, unlike the ODBC loader connection which works only on Windows. Also, the TSS ODBC loader connection does not require a port. It uses the same port as the server and is installed as part of the server.

Prerequisites

Configure a Data Source Name (DSN) with information about the specific database you want the loader connection to point to. See Configuring DSN on Windows and Configuring DSN on UNIX for more information.

Supported Databases

Table 4-2 lists the supported databases and their corresponding TSS ODBC drivers.

Table 4-2 Supported Databases for TSS ODBC (Trillium-Supplied) Loader Connections

Database	Version	Platform	TSS ODBC Driver
Btrieve	Pervasive.SQL 7.0, 8.5, 2000 Btrieve 6.15	Windows only	TSS 15 Btrieve Driver
IBM DB2	DB2 V10.1, V10.5	Linux, UNIX, Windows (LUW)	TSS 15 DB/2 Driver
	DB2 V9.1, V9.5, V9.7, V9.8	Linux, UNIX, Windows	
	DB2 V8.x	Linux, UNIX, Windows	
	DB2 10, 11	z/OS	
	DB2 V9.1	z/OS	
	DB2 UDB V8.1	z/OS	
	DB2 i 7.1, 7.2	DRDA	
	DB2 i 6.1	DRDA	
	DB2 for i5/OS V5R3, V5R4	DRDA	
dBase	dBASE IV, V Clipper FoxPro 6.0 with 3.0 functionality only FoxPro 3.0 FoxPro 3.0 database container (DBC) FoxPro 2.5, 2.6		TSS 15 dBase Driver
Greenplum	Pivotal HAWQ™ 1.1, 1.2 Greenplum version 4.0, 4.1, 4.2, 4.3 Greenplum version 3.3		TSS 15 Greenplum Driver
Apache Hive	See Apache Hive Hadoop Distribution Support		TSS 15 Hadoop Hive Driver
IBM Informix	Informix Dynamic Server 12.1 11.0, 11.5, 11.7, 10.0, 9.2, 9.3, 9.4		TSS 15 Informix Driver
Impala	Cloudera Impala 2.0, 2.1, 2.2 Cloudera Impala 1.0, 1.1, 1.2, 1.3, 1.4		TSS 15 Impala Wire Protocol Driver
MySQL	MySQL Enterprise Edition 5.0.x server, 5.1 server, 5.5 server, 5.6		TSS 15 MySQL Driver

Table 4-2 Supported Databases for TSS ODBC (Trillium-Supplied) Loader Connections (Continued)

Database	Version	Platform	TSS ODBC Driver
Oracle	12c R1 (12.1) 11g R1, R2 (11.1, 11.2) 10g R1, R2 (10.1, 10.2)		TSS 15 Oracle Driver
PostgreSQL	8.2, 8.3, 8.4 9.0, 9.1, 9.2, 9.3, 9.4		TSS 15 PostgreSQL Driver
Progress OpenEdge	11.0, 11.1, 11.2, 11.3, 11.4, 11.5, 10.1.x, 10.2.x	Not supported on AIX	TSS 15 Progress OpenEdge Wire Protocol Driver
Salesforce	Salesforce API versions 33, 34		TSS 15 Salesforce Driver
Microsoft SQL Server	On premise: 2000, 2005, 2008 R1 + R2, 2012, 2014 Cloud: Windows Azure SQL Database 11.0 and later		TSS 15 SQL Server Driver
	On premise: 7.0, 2000, 2005, 2008 R1 + R2, 2012, 2014		TSS 15 SQL Server Legacy Driver
Sybase	SAP Adaptive Server Enterprise 16.0 Sybase Adaptive Server Enterprise 15,15.5, and 15.7 Sybase Adaptive Server Enterprise 12.0, 12.5, and 12.5.x Sybase Adaptive Server Enterprise 11.9		TSS 15 Sybase Driver
Sybase IQ	SAP IQ 16.0 Sybase IQ Server 15.0, 15.1, 15.2, 15.3, 15.4		TSS 15 Sybase IQ Wire Protocol Driver
Teradata	12.0, 13.0, 13.1, 14.0, 14.10, 15.0 V2R6.0, V2R6.1, V2R6.2		TSS 15 Teradata Driver
XML (tabular/ hierarchical format)		Windows only	TSS 15 XML Driver

If you are running TSS applications on a 64-bit system and are planning to configure ODBC, you must configure odbc32, not odbc64.

The following databases require the database client to be installed on the same system:

- Btrieve
- Teradata

All supported TSS ODBC drivers are installed with the Repository Manager.

Using the TSS 15 Salesforce Driver

If you plan to use the TSS 15 Salesforce Driver on Windows 10 or Windows Server 2012 R2 platforms, there are additional steps necessary to configure your environment, including installing Java Virtual Machine (JVM) J2SE 5 or higher, setting the TSS 15 Salesforce Driver to operate in Server Mode rather than Direct Mode (the default), creating the Salesforce SQL engine Windows Service, and setting environment variables.

For more information, contact technical support at TrilliumSupport@syncsort.com.

Creating DSN

This section contains instructions on how to create and configure DSN for TSS ODBC loader connections.

- Configuring DSN on Windows
- Configuring DSN on UNIX

Configuring DSN on Windows

Before you create a DSN entry, you must collaborate with your database administrator to gather relevant information like hostname, database name, user login credentials and other relevant information. This is important because you may need to provide all or some of the information when creating the TSS ODBC DSN entry.

▶ To add a DSN entry on Windows

1. Open the command prompt.
2. Type in `C:\Windows\SysWOW64\odbcad32.exe`. The ODBC Data Source Administrator opens.
3. Click the **System DSN** tab.
4. Click **Add** to open the Create New Data Source window.
5. Scroll down and select the appropriate driver, for example, TSS 15 Oracle Driver.
6. Click **Finish**. The ODBC configuration page opens. The parameters in the ODBC configuration page will vary depending on the data source.
7. Specify the required information in the configuration page. See Configuring TSS Oracle Driver on Windows (Example) on page 47.

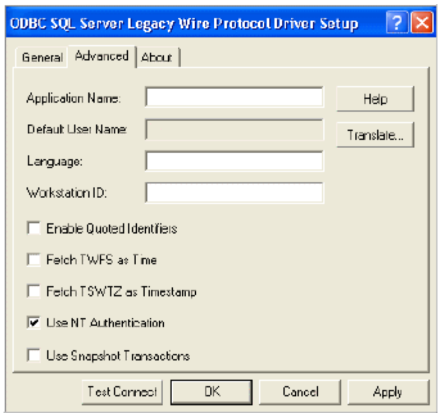
   If you are configuring a TSS 15 SQL Server Legacy Driver, see SSO and the TSS 15 SQL Server Legacy Driver.

   If you are configuring a TSS 15 Hadoop Hive Driver, see Note About TSS 15 Hadoop Hive Driver.

8. Click **Apply** to save your changes.
9. Click **OK** to close the configuration window.
10. Create TSS ODBC loader connection.

SSO and the TSS 15 SQL Server Legacy Driver

For TSS 15 SQL Server Legacy Driver, to use SSO, you must select the option **Use NT Authentication** option on the Advanced tab of the driver configuration page.



Application Name:
Default User Name:
Language:
Workstation ID:

□ Enable Quoted Identifiers
□ Fetch TWFS as Time
□ Fetch TSWTZ as Timestamp
☑ Use NT Authentication
□ Use Snapshot Transactions

Figure 4-1 Configuring TSS 15 SQL Server Legacy Driver

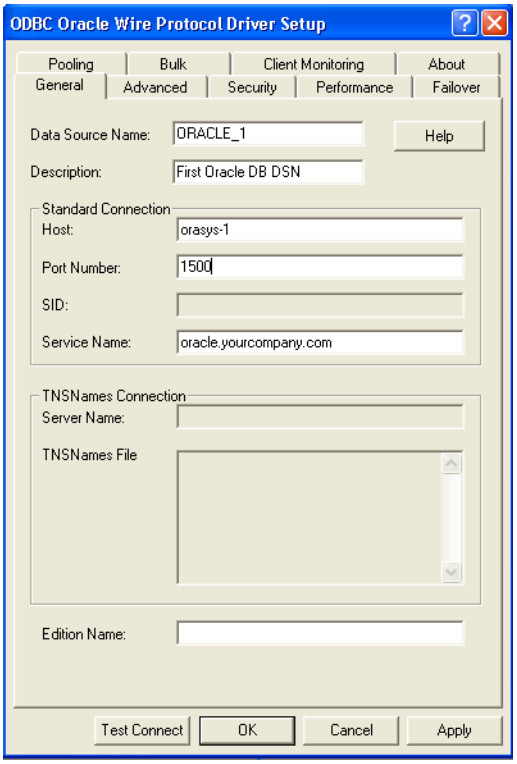
Configuring TSS Oracle Driver on Windows (Example)

Before you create the DSN entry, collaborate with your database administrator and get all information you will be required to provide during the configuration.

▶ To configure TSS Oracle Driver

1. From the Windows Start menu, click **Start > Settings > Control Panel > Administrative Tools > Data Sources (ODBC)**. The ODBC Data Source Administrator opens.
2. Click the **System DSN** tab.
3. Click **Add** to open the Create New Data Source window.
4. Scroll down and select the **TSS 15 Oracle Driver** option.
5. Click **Finish**. The ODBC Oracle Wire Protocol configuration page opens.
6. Enter information in the following fields: Data Source Name, Description, Host, Port Number, and Service Name. In this example, the name of the Data Source you are creating is ORACLE_1. The name of

system hosting the Oracle database is orasys-1, its port number is 1500 and the Service Name is oracle.yourcompany.com.



Data Source Name: ORACLE_1

Description: First Oracle DB DSN

Host: orasys-1

Port Number: 1500

SID:

Service Name: oracle.yourcompany.com

Server Name:

TNSNames File

Edition Name:

Figure 4-2 ODBC Oracle Wire Protocol Driver Setup

ⓘ *The information you provide in the Standard Connection section is defined in the tnsnames.ora file. You can get this information from the Oracle database administrator.*

7. Click **Apply** to save your changes.

8. Click **OK** to close the window.

Configuring DSN on UNIX

For UNIX systems, you must manually add data sources by editing the ODBC configuration file: `.odbc.ini`, a plain text file located in the user's home directory. You must create a data source entry corresponding to each supported database that you intend to use. If you plan to use TSS 15 SQL Server Driver, see Configuring Authentication for TSS 15 SQL Server Driver on UNIX.

odbc.ini details

There are two distinct sections in the `.odbc.ini` file. At the beginning, there is a section named `[ODBC Data Sources]` containing a list of the DSN and the matching installed driver pairs, for example:

```
[ODBC Data Sources]
Oracle=TSS 15 Oracle Driver...
```

where **Oracle** is the data source name and **TSS 15 Oracle Driver** is the associated driver for the Oracle database.

The next section named `[ODBC]` allows ODBC tracing. This capability allows you to trace calls to ODBC drivers and create a log of the traces for troubleshooting purposes.

ⓘ *The following keywords all control tracing: Trace, TraceFile, TraceDLL, ODBCTraceMaxFileSize, and ODBCTraceMaxNumFiles.*

Configuring Authentication for TSS 15 SQL Server Driver on UNIX

In order to configure SQL Server or Windows authentication for a TSS 15 SQL Server Driver on a UNIX system, modify the `odbc.ini` file to use the correct authentication values:

- For SQL Authentication, set `AuthenticationMethod=1`
- For Windows Authentication, set `AuthenticationMethod=9`

Configuring TSS ODBC Connection String on UNIX

In this example, you are configuring the TSS ODBC connection string for UNIX. You will not need to install the Oracle Client.

▶ To add a data source entry to connect to Oracle database

1. Make a backup copy of the original file in case you delete or overwrite the file with a new installation or update.
2. Open the `.odbc.ini` file using a text editor like Notepad plus. Only the TSS administrator has permissions to configure the .odbc.ini file.
3. Locate and copy the example section pertaining to the Oracle driver, for example:

```
[Oracle]
Driver=/home/.../tsq15.0.0/RepoServer/TSS/lib/HTora25.so
Description=TSS 15 Oracle Driver
....
HostName=
PortNumber=
ServiceName=
TNSNamesFile=<tnsnames.ora_filename>
........
```

4. Paste the section at the end of the list of definitions listing.

5. In the copied definition, change the data source name. The data source name is between square brackets at the beginning of the definition, for example, `[Oracle1]`.

   Modify as follows:

   Change data source name to `[Oracle1]`.

   Change `HostName` to reflect the Oracle TNS and installation.

   Change `PortNumber` to reflect the Oracle TNS and installation.

   Change `ServiceName` to reflect the Oracle TNS and installation.

ⓘ *When using* `ServiceName`*, clear the default comments (shown below) for ServerName and SID as defined in the .odbc.ini including the opening and closing brackets "<>". That is:*
ServerName=<server_name in tnsnames.ora>
SID=<Oracle_System_Identifier>

ⓘ *The information you provide here is defined in the tnsnames.ora file. You can get this information from the Oracle database administrator.*

6. Go to the beginning of the file to the [ODBC Data Sources] section. Add a new data source name as shown below:

   ```
   Oracle1=TSS 15 Oracle Driver
   ```

   This is how the changed sections will look like:

   ```
   [ODBC Data Sources]
   ...
   Oracle=TSS 15 Oracle Driver
   Oracle1=TSS 15 Oracle Driver
   ....
   [Oracle1]
   Driver=/home/.../tsq15.0.0/RepoServer/TSS/lib/HTora25.so
   Description=TSS 15 Oracle Driver
   ....
   HostName=orasys-1
   PortNumber=1500
   ServiceName=oracle.yourcompany.com
   TNSNamesFile=<tnsnames.ora_filename>
   ........
   ```

7. Save the changes and close the file.

Apache Hive Data Connection

You can load data into TSS from an Apache Hive environment using the TSS ODBC Hadoop Hive Driver loader connection.

ⓘ *You can only read data from a Apache Hive environment. You cannot write to it.*

Table 4-3 Apache Hive Hadoop Distribution Support

Hadoop Distribution	Distribution Version	Apache Hive Version
Amazon Elastic MapReduce (Amazon EMR)	AMI 2.1.4	Hive 0.8.x
	AMI 2.2.4-3.1.4	Hive 0.11X
	AMI 3.2-3.3.1	Hive 0.13.x
Apache Hive	N/A	Hive 0.8.x Hive 0.9.x Hive 0.10.x Hive 0.11.x Hive 0.12.x Hive 0.13.x Hive 0.14.x Hive 1.0.x Hive 1.1.x Hive 1.2.x
Cloudera's Distribution Including Apache Hadoop (CDH)	CDH 3 update 4 CDH 4.0 CDH 4.1 CDH 4.2 CDH 4.5 CDH 5.0 CDH 5.1 CDH 5.2 CDH 5.3 CDH 5.4	Hive 0.7.1 Hive 0.8.x Hive 0.9.x Hive 0.12.x Hive 0.13.x Hive 1.1.x
Hortonworks Distribution for Apache Hadoop	HDP 1.3	Hive 0.11.x
	HDP 2.0	Hive 0.12.x
	HDP 2.1	Hive 0.13.x
	HDP 2.2	Hive 0.14.x
	HDP 2.3	Hive 1.2.1
	HDP 2.4	Hive 1.2.1
IBM BigInsights	BigInsights 3.0	Hive 0.12.x
	BigInsights 4.0	Hive 1.1.x
	BigInsights 4.1	Hive 1.2.1

Table 4-3 Apache Hive Hadoop Distribution Support

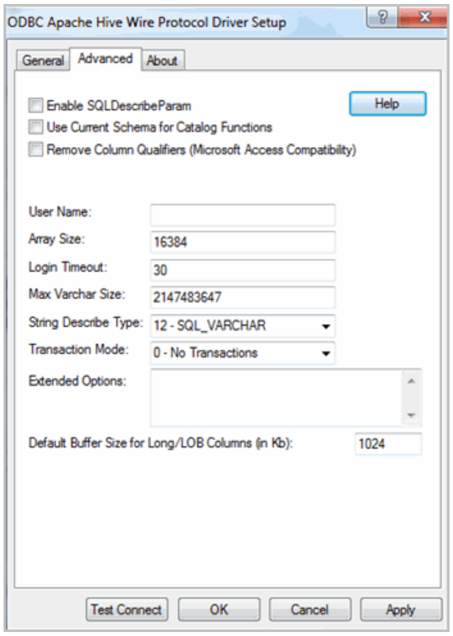
Hadoop Distribution	Distribution Version	Apache Hive Version
MapR Distribution for Apache Hadoop	MapR 1.2 MapR 2.0	Hive 0.7.1, Hive 0.9.x
Pivotal HD Enterprise Distribution for Apache Hadoop	PHD 2.0.1	Hive 0.12.x
	PHD 2.1	Hive 0.12.x
	PHD 3.0	Hive 1.1.x

Note About TSS 15 Hadoop Hive Driver

Create the Apache Hive data source by selecting the TSS 15 Hadoop Hive Driver from the list of supported drivers (Table 4-2) and following the instructions in Creating DSN. Configure the Advanced settings by changing the Max Varchar field from 2GB (default) to 32KB. Otherwise, you may not be able to see the data correctly.

▶ To change the Max Varchar

1. Open the **Advanced** tab of the ODBC Apache Hive Wire Protocol Driver Setup window.



□ Enable SQLDescribeParam
□ Use Current Schema for Catalog Functions
□ Remove Column Qualifiers (Microsoft Access Compatibility)

User Name:
Array Size: 16384
Login Timeout: 30
Max Varchar Size: 2147483647
String Describe Type: 12 - SQL_VARCHAR
Transaction Mode: 0 - No Transactions
Extended Options:
Default Buffer Size for Long/LOB Columns (in Kb): 1024

Figure 4-3 Configuring TSS 15 Hadoop Hive Driver

2. In **Max Varchar Size**, enter 32768.

3. Click **Apply**.

Creating a Loader Connection

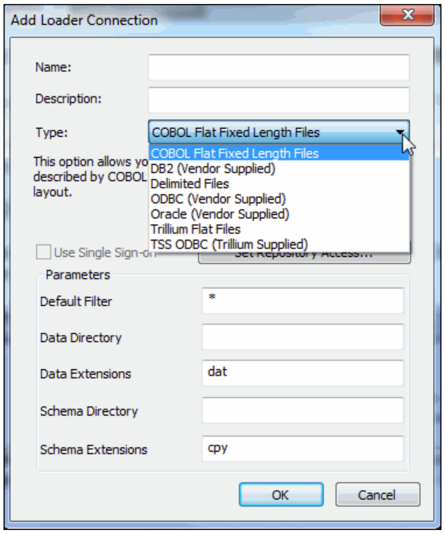
You can create loader connections using the Repository Manager application and the Repository Administration Command Prompt utility. For information on how to create a loader connection using the utility, see Command Line Utility on page 97.

ⓘ *When adding loader connections to relational databases, ensure that the relational data sources are properly configured before you create entities.*

When you add a loader connection, it is enabled by default. You can disable it if needed. For instructions, see Enabling or Disabling a Loader Connection.

▶ To add a loader connection from the Repository Manager

1. In the Repository Manager's Navigation View, right-click **Loader Connections**.
2. Select **Add Loader Connection**. The Add Loader Connection window opens.



Add Loader Connection

Name:

Description:

Type: COBOL Flat Fixed Length Files

This option allows yo
described by COBOL
layout.

□ Use Single Sign-on

Parameters

Default Filter: *

Data Directory:

Data Extensions: dat

Schema Directory:

Schema Extensions: cpy

Figure 4-4 Add Loader Connection Window

3. In the **Name** field, specify a name.
4. In the **Description** field, add a brief description.
5. From the **Type** drop-down list, select a data source type.

6. Configure the loader connection configuration using the information in Table 4-4 on page 54 as a guideline.

Table 4-4 Loader Connection Properties

Loader Connection	Property	Description
COBOL Flat Fixed length Files		This loader connection option allows you to connect to flat, fixed length files described by COBOL Copybooks. The data must match copybook layout.
	Default Filter	Default value is *. Specify a default filter to limit the number of files that will get listed in the Create Entity Wizard of the Control Center. For example, to list only those files/tables that contain cust, enter cust in the Default Filter field.
	Data Directory	Directory path to data files.
	Data Extensions	Specify expected data file extensions. Only files with these extensions display in the Create Entity Wizard for this connection. If none are specified, dat is the default. Separate each extension with a space and omit the leading period. If you want to display files that have no extension, add **{}** to the list. **Example** If you only want to display data files with the extensions **dat**, **txt**, and files without extensions, type the following entry: dat txt {}
	Schema Directory	Directory path to copybook files.
	Schema Extensions	Specify expected schema file extensions. Only files with these extensions display in the Create Entity Wizard for this connection. If none are specified, cpy is the default. Separate each extension with a space and omit the leading period. If you want to display files that have no extension, add **{}** to the list. **Example** If you only want to display schemas with the extensions **cpy** and **cbl**, and schemas without extensions, type the following entry: cpy cbl {}
DB2 (Vendor Supplied)		This loader connection option allows you to connect to DB2 databases.

Table 4-4 Loader Connection Properties (Continued)

Loader Connection	Property	Description
	Default Filter	Default value is *. Specify a default filter to limit the number of files that will get listed in the Create Entity Wizard of the Control Center. For example, to list only those files/tables that contain cust, enter cust in the Default Filter field.
	Alias	DB2 alias.
Delimited Files		This loader connection option allows you to connect to ASCII, extended ASCII, or hexadecimal delimiters.
	Default Filter	Default value is *. Specify a default filter to limit the number of files that will get listed in the Create Entity Wizard of the Control Center. For example, to list only those files/tables that contain cust, enter cust in the Default Filter field.
	Data Extensions	Specify expected data file extensions. Only files with these extensions display in the Create Entity Wizard for this connection. If none are specified, txt, csv and dat are the default file extensions. Separate each extension with a space and omit the leading period. If you want to display files that have no extension, add **{}** to the list. **Example** If you only want to display data files with the extensions **dat** and **txt**, and files without extensions, enter: dat txt {}
	Schema Directory	Directory path to schema files such as DDLs.
	Schema Extensions	Specify expected schema file extensions. Only files with these extensions display in the Create Entity Wizard for this connection. If none are specified, ddl is the default. Separate each extension with a space and omit the leading period. If you want to display files that have no extension, add **{}** to the list. **Example** If you only want to display schemas with the extensions **ddl** and **ddt**, and schemas without extensions, type the following entry: ddl ddt {}

Table 4-4 Loader Connection Properties (Continued)

Loader Connection	Property	Description
ODBC (Vendor Supplied)		This loader connection option allows you to connect to a number of databases using ODBC drivers supplied by database vendors. Choose this to connect to a database that is not available through TSS ODBC. ⓘ *If you are running TSS applications on a 64-bit system and are planning to configure ODBC, you must configure odbc32, not odbc64.*
	Default Filter	Default value is *. Specify a default filter to limit the number of files that will get listed in the Create Entity Wizard of the Control Center. For example, to list only those files/tables that contain cust, enter cust in the Default Filter field.
	Machine Name	Name of the system or the IP address of the system where the ODBC Adapter is installed. (Applicable only if the ODBC Adapter is installed on a remote system.)
	Port Number	Port number of TSS ODBC Service—specified during TSS ODBC Support installation.
	DSN Name	The ODBC system DSN on the server where TSS ODBC Support was installed.
	SSO	Optional. If you are using a Windows operating system, check the Use Single Sign-on option if you want to apply Windows authentication to users of this connection.
Oracle (Vendor Supplied)		This loader connection option allows you to connect to an Oracle database. This connection requires an Oracle client to be installed.
	Default Filter	Default value is *. Specify a default filter to limit the number of files that will get listed in the Create Entity Wizard of the Control Center. For example, to list only those files/tables that contain cust, enter cust in the Default Filter field.
	TNS Name	TNS Name is the name of the entry in tnsnames.ora file which is kept in `$ORACLE_HOME/network/admin`
	SSO	Optional. If you are using a Windows operating system, check the Use Single Sign-on option if you want to apply Windows authentication to users of this connection.

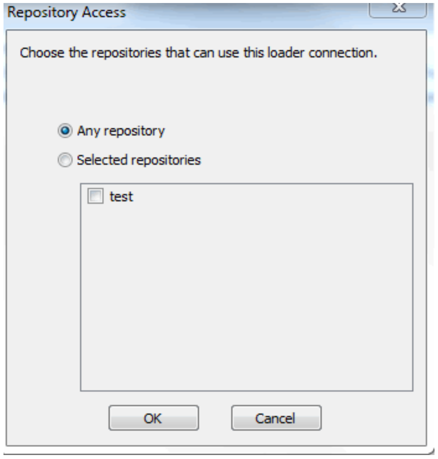
Table 4-4 Loader Connection Properties (Continued)

Loader Connection	Property	Description
Trillium Flat Files		This option allows you to connect to a flat file created by TSS modules when they process data.
	Default Filter	Default value is *. Specify a default filter to limit the number of files that will get listed in the Create Entity Wizard of the Control Center. For example, to list only those files/tables that contain cust, enter cust in the Default Filter field.
	Data Directory	Directory path to data files.
	Data Extensions	Specify expected data file extensions. Only files with these extensions display in the Create Entity Wizard for this connection. If none are specified, no extension is the default. Separate each extension with a space and omit the leading period. If you want to display files that have no extension, add **{}** to the list. **Example** If you only want to display data files with the extensions **dat** and **txt**, and files without extensions, type the following entry: dat txt {}
	Schema Directory	Directory path to schema files such as DDLs.
	Schema Extensions	Specify expected schema file extensions. Only files with these extensions display in the Create Entity Wizard for this connection. If none are specified, ddl and ddx are the defaults. Separate each extension with a space and omit the leading period. If you want to display files that have no extension, add **{}** to the list. **Example** If you only want to display schemas with the extensions **ddl** and **ddt**, and schemas without extensions, type the following entry: ddl ddt {}
TSS ODBC (Trillium Supplied)		This loader connection option allows you to connect to the widest selection of external databases, including Oracle, DB2, SQL and so on without installing any external component.
	Default Filter	Default value is *. Specify a default filter to limit the number of files that will get listed in the Create Entity Wizard of the Control Center. For example, to list only those files/tables that contain cust, enter cust in the Default Filter field.

Table 4-4 Loader Connection Properties (Continued)

Loader Connection	Property	Description
	DSN Name	The DSN name as defined on the server for the TSS ODBC driver.
	SSO	Optional. If you are using a Windows operating system, check the Use Single Sign-on option if you want to apply Windows authentication to users of this connection.

7. To limit repository access, repository, select **Set Repository Access**. The Repository Access window opens.



Repository Access

Choose the repositories that can use this loader connection.

● Any repository

○ Selected repositories

□ test

Figure 4-5 Repository Access Window

8. Click **Selected repositories** and then select the repositories that can be accessed using the loader connection.

9. Click **OK** to save your changes.

ⓘ *By default, all repositories in the repository server are granted permission to use the connection.*

10. (Optional) Specify a default filter to limit the number of files that will get listed in the Create Entity Wizard of the Control Center. For example, to list only those files/tables that contain cust, enter `cust` in the Default Filter field.

ⓘ *The default is *, to list all available files.*

11. Click **OK** to create the loader connection.

12. Repeat the steps to create additional loader connections if needed.

Editing a Loader Connection

This section includes:

- Reviewing a List of Loader Connections
- Enabling or Disabling a Loader Connection
- Enabling a Loader Connection After Logging Out of Repository Manager
- Modifying a Loader Connection Definition
- Defining Performance Settings

Reviewing a List of Loader Connections

You can display a list that includes all the loader connections defined for a repository server (and all its repositories), along with the details of each connection.

▶ To display a list of connections

1. Log in to the Repository Manager.
2. In the Navigation View, double-click **Loader Connections**. The Loader Connections List View opens, showing all loader connections for this repository server. If a connection is restricted to a subset of repositories, the Repository Access column shows which repositories have been granted access to the connection. The Single Sign-on column shows whether an SQL Server ODBC connection is enabled for single sign-on.

   ⓘ *For information on restricting access to repositories or establishing single sign-on, refer to page Add Loader Connection Window.*

Enabling or Disabling a Loader Connection

When you create a loader connection, it is enabled by default. You cannot delete a loader connection as that will delete the loader connection reference information the entities store. You can however, disable a loader connection if you do not need to use it.

ⓘ *Once disabled, a loader connection is no longer accessible from the Create Entity Wizard in the Control Center.*

▶ To enable or disable a loader connection

1. Log on to the Repository Manager.
2. In the Navigation View, double-click **Loader Connections**. The Loader Connections List View opens, showing all loader connections for this repository server.
3. To enable: select a loader connection from the list. Right click and select **Enable** to enable the connection.

   ⓘ *Only those loader connections that are disabled in the current session are listed for enabling.*

4. To disable: select a loader connection from the list. Right-click and select **Disable** to disable the connection.

   ⓘ *Only those loader connections that are enables in the current session are listed for disabling.*

   If you disabled a connection and logged out of Repository Manager and wish to re-enable the connection, follow instructions in Enabling a Loader Connection After Logging Out of Repository Manager).

Enabling a Loader Connection After Logging Out of Repository Manager

Once you disable a loader connection and log out of the Repository Manager, the loader connection is no longer listed in the Loader Connection’s List View page. To enable the connection, you must go to the tree view and search for disabled loader connections.

▶ To search for and enable a loader connection from the tree view:

1. In the Repository Manager’s Navigation View, select **Loader Connections**.
2. Right click and select **Disabled Loader Connections**. The Loader Connections List View opens, showing all disabled loader connections for this repository server.

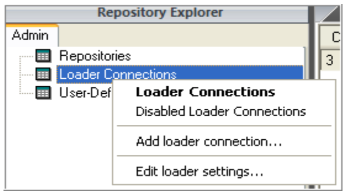


Figure 4-6 Disabled Loader Connection Option

3. Select the connection you want to enable.
4. Right-click and select **Enable** from the drop-down menu option.

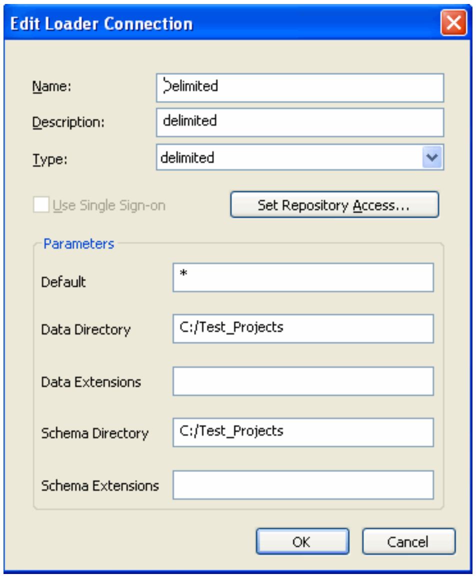
Modifying a Loader Connection Definition

Use the following procedure to modify a loader connection in situations where you had entered an incorrect value or if the requirements have been modified.

▶ To modify a loader connection definition

1. In the Repository Manager’s Navigation View, double-click **Loader Connections**. The Loader Connections List View opens, showing all loader connections for this repository server.

2. Select the connection you want to modify. Right-click and select **Edit** from the drop-down menu option. The **Edit Loader Connection** window opens.



Name: Delimited

Description: delimited

Type: delimited

□ Use Single Sign-on

Default: *

Data Directory: C:/Test_Projects

Data Extensions:

Schema Directory: C:/Test_Projects

Schema Extensions:

Figure 4-7 Edit Loader Connection Window

3. Make the necessary modifications, including resetting repository access if necessary.

ⓘ *If you select a different data source Type, the parameter fields may change to reflect the new type.*

4. Click **OK** to save your changes. The changes take effect immediately.

Defining Performance Settings

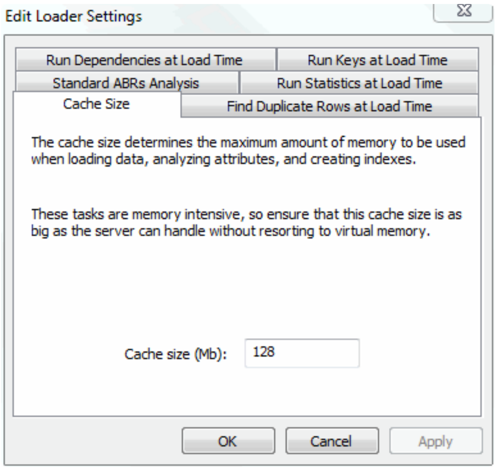
The following settings in the Repository Manager can have an impact on data import performance:

- Cache Size
- Dependency Analysis
- Key Analysis
- Statistics Analysis
- Duplicate Key Discovery
- Copy Standard Attribute Business Rules (ABRs)
- Standard ABRs Analysis

ⓘ *They apply to all loader connections in the repository server.*

▶ To define or modify performance settings

1. In the Repository Manager's Navigation View, right-click **Loader Connections** and select **Edit Loader Settings** from the drop-down list. The Edit Loader Settings window opens.



Edit Loader Settings
Run Dependencies at Load Time
Run Keys at Load Time
Standard ABRs Analysis
Run Statistics at Load Time
Cache Size
Find Duplicate Rows at Load Time
OK
Cancel
Apply

The cache size determines the maximum amount of memory to be used when loading data, analyzing attributes, and creating indexes.

These tasks are memory intensive, so ensure that this cache size is as big as the server can handle without resorting to virtual memory.

Cache size (Mb): 128

Figure 4-8 Edit Loader Settings Window

2. Review the current cache size setting and make the necessary change. You can set the loader connection cache size between 128 and 2000 MB (2 GB).

3. Click the other tabs, and modify as necessary using the instructions in Table 4-5 on page 62.

Table 4-5 Performance Setting Descriptions

Setting	Description
Cache Size	Specifies the maximum memory, in megabytes, to be used when loading data, analyzing attributes, and creating indexes. Define this setting to be as high as your hardware allows (cache size between 128 and 2000 MB) without relying on virtual memory. ⓘ *On multiple CPU systems, the load process can run multiple times in parallel. In these situations, memory is shared across all processes to ensure that it does not exceed the limit you define.*
Run Dependencies at Load Time	By default, TSS automatically performs dependency analysis on a sample of your data (10,000 rows) during data import to find possible dependencies. Disabling the dependency analysis improves the load performance. To disable this option, clear the **Discover Dependencies** at **Load Time** selection. ⓘ *If you choose to not run the dependency analysis during data import, remember to manually run the analysis at a later time from the Control Center.*

Table 4-5 Performance Setting Descriptions (Continued)

Setting	Description
Run Keys at Load Time	By default, TSS automatically performs key analysis on a sample of your data (10,000 rows) during data import to find potential keys. Disabling the key analysis improves the load performance. To disable this option, clear the **Discover Keys at Load Time** selection. ⓘ *If you choose to not run the key analysis during data import, remember to manually run the analysis at a later time from the Control Center.*
Run Statistics at Load Time	By default, TSS automatically performs statistical analysis of numeric attributes during data import to calculate the standard deviation value. Disabling statistical analysis improves load performance. To disable this option, clear the **Run Statistics at Load Time?** selection. ⓘ *If you choose to not run the statistical analysis during data import, remember to manually run the analysis at a later time from the Control Center.* To find other attribute statistics like percentile value, run percentile analysis of the attribute with numeric data in the Control Center.
Find Duplicate Rows at Load Time	By default, TSS identifies duplicate rows discovered during the data import process. To improve system performance, you can disable this feature or you can limit the number of duplicates you want found. To disable this option, clear the **Discover duplicated rows** selection. To set a processing limit, enter a number in the **Potential duplicates** field. The default value is 10,000.
Standard ABRs Analysis	During entity creation process, this option allows the TSS to copy standard ABRs to the new entity. Option is disabled by default to speed up the entity creation process and also to control the size of the business rules table. To enable copy of standard ABRs, select the **Copy standard attribute business rules to attributes** option. Now if you create a new entity in the Control Center, all standard ABRs are copied. To disable the copy function, remove the check from the **Copy standard attribute business rules to attributes** option.

4. Click **OK** to save your changes.

CHAPTER 5

User-Defined Country Templates

Country templates enable users to quickly create a Quality project based on predefined country settings. Trillium Software delivers country templates for many, but not all, countries.

This chapter includes:

- Overview of Country Templates on page 64
- Creating User-Defined Country Template on page 65

Overview of Country Templates

A country template is useful in building a standard project for a specific country. A standard project contains definitions for all the required settings for each process, therefore, a user can create and run projects without spending a lot of time configuring the project.

As you become familiar with using the TSS application, you will typically customize projects settings to exactly match your needs. However, the country templates enable Trillium users to use the product “out of the box.” Trillium Software has developed country-specific support for over 190 countries. The level of support varies by country, ranging from robust to basic. Each support level is defined as follows:

- **Robust**—Country-specific standardization and matching rules. At minimum, street-level address correction, validation, and formatting. Over thirty countries supported, including US, Canada, France, Australia, United Kingdom, and Germany.
- **Standard**—Country-specific standardization and matching rules. City correction, validation, and address formatting. Sixteen countries supported, including Turkey, Venezuela, and Saudi Arabia.
- **Basic**—City-level correction and validation. Includes support for over 180 countries not included in the Robust and Standard categories.

ⓘ *For a complete list of countries and the support level of each, contact your account manager.*

When you install TS Quality, the countries covered by your license agreement are automatically delivered. Templates are provided for countries in the robust and standard support levels, but not for the basic level. You can expand the level of coverage for one or more of the basic countries by defining a template based on the template of a standard country.

ⓘ *The Add Country Template command is available only if your license includes the basic country package offered by Trillium Software.*

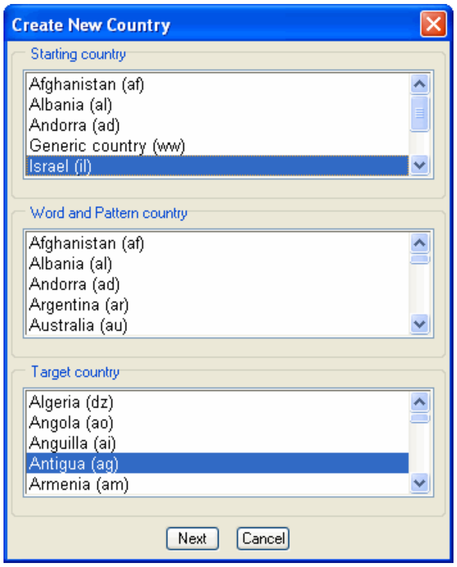
You cannot use a country that has a Postal Matcher as the basis for a new template. All of the settings and rules for those countries are based on the use of a Postal Matcher, but user-defined country templates do not have a Postal Matcher. You can, however, use the word and pattern tables from a country that has a Postal Matcher

Creating User-Defined Country Template

When creating a new country template, it is most effective to base it on an existing template; that is for a country with similar address and language patterns.

▶ To create a user-defined country template

1. In the Navigation View, right-click **User-Defined Country Templates** and select **Add user-defined country template**. Alternately, on the Home tab, click **Country Template**. The Create New Country window opens.



Create New Country

Starting country

Afghanistan (af)
Albania (al)
Andorra (ad)
Generic country (ww)
Israel (il)

Word and Pattern country

Afghanistan (af)
Albania (al)
Andorra (ad)
Argentina (ar)
Australia (au)

Target country

Algeria (dz)
Angola (ao)
Anguilla (ai)
Antigua (ag)
Armenia (am)

Figure 5-1 Create New Country Window

2. In the **Starting country** list, select a country as base the new template. The selected country provides the default values for the

encoding to be used in the template. Alternatively, you can select the **Generic country (ww)** entry.

Countries in the **Starting country** list include the standard countries for which you have a license, plus any user-defined countries that you created previously. It does not include countries that use a Postal Matcher.

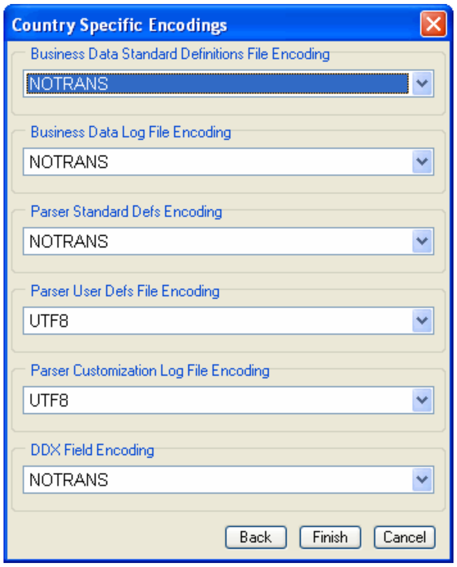
3. In the **Word and pattern country** list, select the country whose language patterns most closely resemble the country for which you are creating this template. For example, if you are creating a template for Bermuda, select United Kingdom.

   Countries in the **Word and pattern country** list include robust and standard countries for which you have a license.

4. In the Target Country list, select the country for which you are creating the template. The list comprises all the countries for which Trillium Software provides only basic support, excluding those for which you have already created a template.

5. Click **Next**. The Country-Specific Encodings window opens.

6. The values that appear in the fields are the default values for the starting country you selected in step 2.



Country Specific Encodings

Business Data Standard Definitions File Encoding: NOTRANS

Business Data Log File Encoding: NOTRANS

Parser Standard Defs Encoding: NOTRANS

Parser User Defs File Encoding: UTF8

Parser Customization Log File Encoding: UTF8

DDX Field Encoding: NOTRANS

Figure 5-2 Country Specific Encodings Window

ⓘ *Encoding (also called code page) is a mapping of binary values to code positions which represent characters of data.*

ⓘ *NOTRANS means No Translation. Quality processes use the default encoding for the host computer.*

7. If necessary, change one or more of the default encodings.

8. Click **Finish**. The necessary files are copied and modified to create the new country template. TSS users will see the new template in the selection list the next time they create a project.

About User Defined Country Projects

When you create a project based on a user-defined template, you cannot export the project to run on a different server unless you recreate the template on that server first. This applies to both export/import and batch exports.

If you modify a project that was created with a user-defined template, the edits only apply to that project. You cannot save them to the master template on the server.

CHAPTER 6

User-Defined Notes Class

Notes provide an easy way to collaborate within and across teams with relevant information about your projects.

This chapter includes:

Overview of Notes Classes and Sub-Classes

You can add notes to a variety of objects including:

- Baseline Analysis, Join and Quality projects
- Attributes
- Entities
- Permanent joins
- TS Quality processes and output entities

For each note, you can define note classes and subclasses. There are different classifications of notes that are available out of the box, see Table 6-1, "Default Note Classes and subclasses". You can also create custom classes and subclasses.

The note class and subclasses are useful in identifying the contents of a note and allows you to group notes together for reporting purposes.

Table 6-1 Default Note Classes and subclasses

Class	Subclass	Use this class to note
TS Quality		
	To Do List	List of items to attend to.

Table 6-1 Default Note Classes and subclasses (Continued)

Class	Subclass	Use this class to note
	Processing Details	Detailed description of a Quality data process steps.
	Processing Results	Results of a data quality process. You can save this information as either as a summary you have written and/or as copied data from List Views.
	Issues to Resolve	Data issues and resolution requirements.
Business		
	Meaning	Detailed description of the issue or data.
Reconciliation		
	Sum	Expected sum for integer/decimal attributes.
	Record Count	Expected record count for an entity.
	Data Volumes	Data volume of an entity in a repository.
Attribute		
	Nulls	Issue related to NULL values (blank fields) for an attribute.
	Ranges	Issue related to the range of values for an attribute.
	Phonetic	Issue with values that sound alike. For example, potential misspellings.
	Abnormal Values	Issue related to value(s) with unexpected high or low frequency.
	Patterns	Issue related to value(s) with unexpected pattern.
	Datatype	Issues with unexpected data types.
	Technical Cleanse Rule	Definition of the technical cleansing rules required or done for system reasons.
	Business Cleanse Rule	Definition of the business cleansing rules required or done for system reasons.
Mapping		
	Mapped From	Object from which an object is mapped. Mappings must be defined bi-directionally.
	Mapped To	Object to which an object is mapped. Mappings must be defined bi-directionally.
	Out of Scope	Object that is not considered in scope for mapping.
	Default Value	Attribute that is not mapped from another attribute; in such cases, note the default value used.
	Error Processing	Error processing rule that defines what to do if an error occurs.

Table 6-1 Default Note Classes and subclasses (Continued)

Class	Subclass	Use this class to note
	Complex Transformation	Definition of complex data transformation rule.
	Medium Transformation	Definition of medium data transformation rule.
	Simple Transformation	Definition of simple data transformation rule.
	Select Criteria Rule	Definition of how rows should be selected for extraction from the source.
Join		
	Outer Join Rule	If an outer join is required for this join.
	Cardinality	Definition of unexpected cardinality issue.
	Optionality	Definition of unexpected optionality issue.
	RI Rule	Definition of unexpected Referential Integrity (RI) issue.
Key		
	Business Meaning	Business significance of a key
	Duplication Issue	Data duplication issue.
	Technical Cleanse Rule	Technical description of how to correct an issue
	Business Cleanse Rule	Business description of how to correct an issue.
Dependency		
	Business Meaning	Business significance of the dependency.
	Conflict Issue	Description of conflicts.
	Technical Cleanse Issue	Technical description of how to correct the issue.
	Business Cleanse Issue	Business description of how to correct the issue.

Creating a User-Defined Notes Class

You can create custom class and subclasses for Notes in the Repository Manager.

▶ To create a user-defined notes class

1. In the Navigation View, right-click **User-Defined Note Class** and select **Add User Defined Notes Class**. Alternately, on the Home tab, click **Notes Class**. The Add Notes Class window opens.

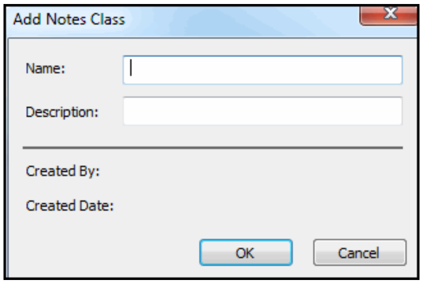


Figure 6-1 Add Note Class Window

2. In the **Name** field, enter a descriptive name like Insurance policy.
3. In the **description** field, enter a brief description about the note class.
4. Click **OK** to save your changes. The new note class is listed on the details pane.

Creating a User-Defined Notes Subclass

You can create custom class and subclasses for Notes in the Repository Manager.

▶ To create a user-defined notes subclass

1. In the List View, select the note class to which you want to add a subclass.
2. Right click and select **Add Subclass...** from the drop down menu options.

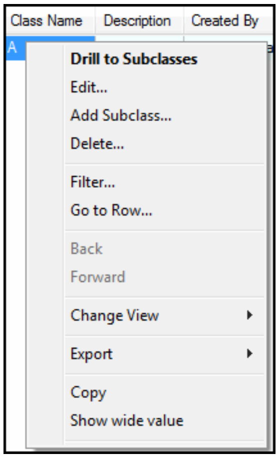


Figure 6-2 Add Note Subclass

The Add Notes Subclass window opens.

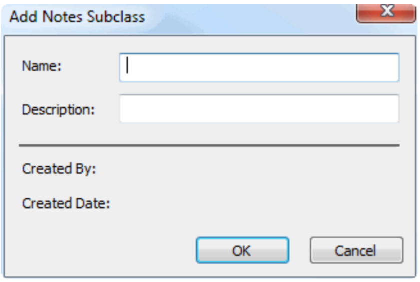


Figure 6-3 Add Notes Subclass Window

3. In the Name field, enter a descriptive name like Flood Insurance.
4. In the description field, enter a brief description about the note subclass.
5. Click **OK** to save your changes.

Editing Note Class and Subclass

You can edit a class or a subclass to a note.

▶ To edit a note class

1. In the User Defined Note Classes list view, select the note class you want to edit.
2. Right-click and select **Edit** from the drop-down menu options. The Edit Note class window opens.
3. Modify the note class description as needed.
4. Click **OK** to save your changes.

▶ To edit a note subclass

1. In the User Defined Note Classes list view, double-click the note class with the note subclass you want to edit. The User Defined Notes Subclasses page opens.
2. Select the subclass you want to edit.
3. Right-click and select **Edit** from the drop-down menu options. The Edit Note subclass window opens.
4. Modify the note class description as needed.
5. Click **OK** to save your changes.

CHAPTER 7

Custom Settings

This chapter explains how to configure Control Center features and how to customize the Repository Manager user interface. It includes the following topics:

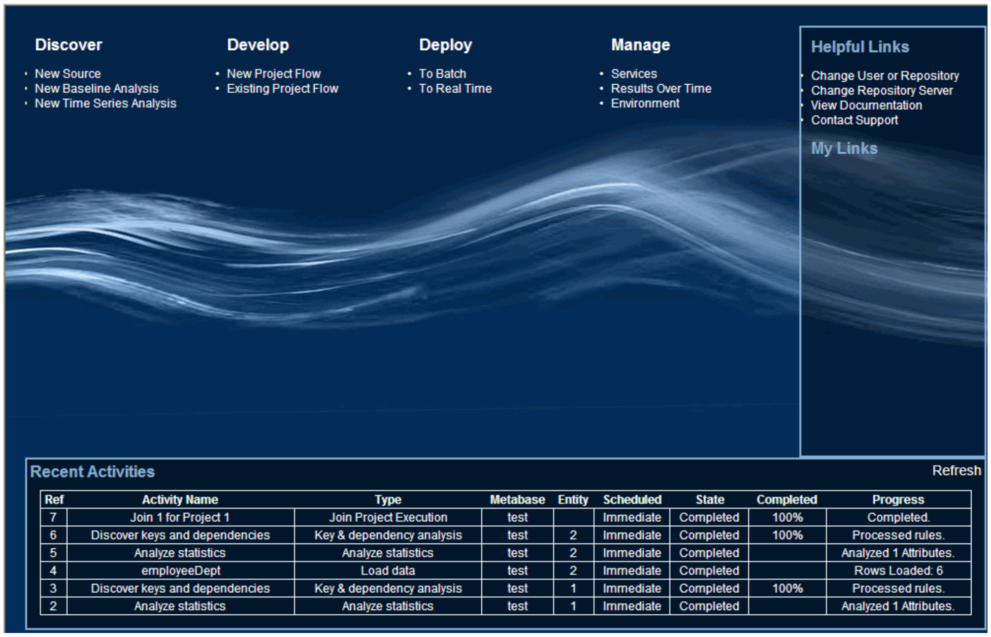
Enabling Control Center Options

The Repository Administrator can configure certain features in the Control Center that apply to all Control Center users.

- Setting Up User-Defined Links
- Setting Up Password Validation
- Enabling E-mail Notification
- Changing HTML Format

Setting Up User-Defined Links

The Getting Started page in the Control Center has a section entitled My Links.



Ref	Activity Name	Type	Metabase	Entity	Scheduled	State	Completed	Progress
7	Join 1 for Project 1	Join Project Execution	test		Immediate	Completed	100%	Completed.
6	Discover keys and dependencies	Key & dependency analysis	test	2	Immediate	Completed	100%	Processed rules.
5	Analyze statistics	Analyze statistics	test	2	Immediate	Completed		Analyzed 1 Attributes.
4	employeeDept	Load data	test	2	Immediate	Completed		Rows Loaded: 6
3	Discover keys and dependencies	Key & dependency analysis	test	1	Immediate	Completed	100%	Processed rules.
2	Analyze statistics	Analyze statistics	test	1	Immediate	Completed		Analyzed 1 Attributes.

Figure 7-1 Control Center Getting Started Page

By default, this area is blank, but you can populate it with links to useful sites on the Internet or your company's Intranet. The links you define here will be displayed on the Getting Started page of all users.

▶ To define a link

1. Log in to the Repository Manager.
2. On the Home tab, click **Link Settings**. The Link Settings window opens. If you have already created links, they appear in this window.

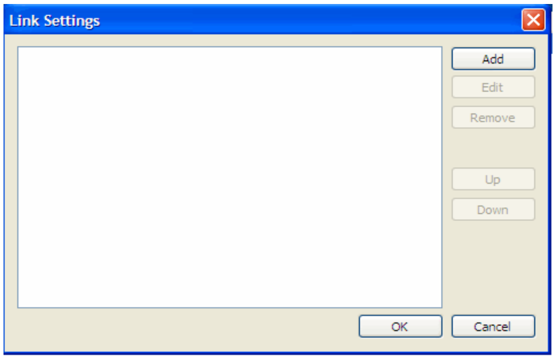


Figure 7-2 Link Settings window

3. Click **Add**. The Add Link window opens.

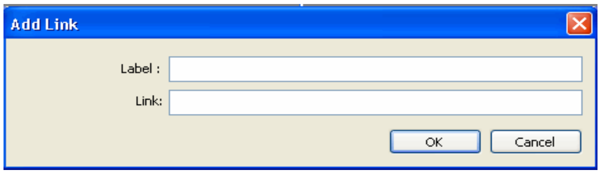


Figure 7-3 Add Link Window

4. In the **Label** field, enter a descriptive name for this link. The label is what the user will see in the Getting Started window.
5. In the **Link** field, enter the URL.
6. Click **OK**. Repeat steps 3 through 6 to define another link.
7. Click **OK** to save your changes. The new links will display the next time a user starts the Control Center.

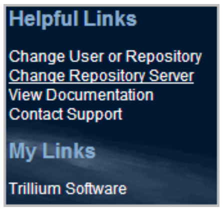


Figure 7-4 Custom links

▶ To delete a link

1. Log in to the Repository Manager.
2. On the Home tab, click **Link Settings**. The Link Settings window opens listing all defined links.
3. Select the link you want to delete and click **Remove**.
4. Click **OK**.

▶ To change the label or URL of an existing link

1. Log in to the Repository Manager.
2. On the Home tab, click **Link Settings**. The Link Settings window opens listing all defined links.
3. Select the link you want to change and click **Edit**. The Edit Link window opens.
4. Make your changes to the label or URL and click **OK**.
5. Click **OK** on the Link Settings window to apply the changes.

Setting Up Password Validation

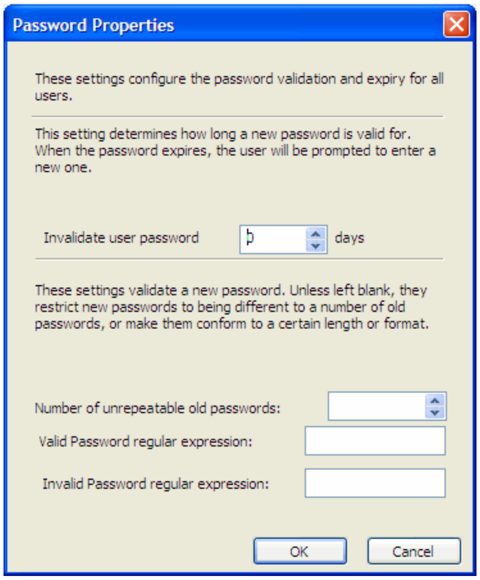
Password rules apply only to Trillium-authenticated users and not to Windows-authenticated users.

By default, there are no password validation rules. Users can create a password of any size and character combination and the password never expires. However, the repository administrator can establish the following rules:

- Length of time the password is valid
- Validity of reusing old passwords
- Valid and invalid password combinations

▶ To edit password properties

1. Log in to the Repository Manager.
2. On the Home tab, click **Password Properties** or from the Navigation View, right-click **Users** and select **Edit Password Properties** from the drop-down menu. The Password Properties window opens.



Password Properties

These settings configure the password validation and expiry for all users.

This setting determines how long a new password is valid for. When the password expires, the user will be prompted to enter a new one.

Invalidate user password: 0 days

These settings validate a new password. Unless left blank, they restrict new passwords to being different to a number of old passwords, or make them conform to a certain length or format.

Number of unrepeatable old passwords:

Valid Password regular expression:

Invalid Password regular expression:

Figure 7-5 Password Properties Window

3. In **Invalidate user password**, select the number of days after which the password expires. The default is 0, which means the password never expires.
4. In **Number of unrepeatable old passwords**, enter the number of recent passwords that cannot be repeated. Setting a value of 2, for example, forces the user to create a new password at least three times before he is able to recycle. By default, the user can reuse the same password every time.

5. In **Valid Password regular expression** and in **Invalid Password regular expression**, enter the expression defining the restrictions on the password length and composition. Leave the fields blank if you do not want to impose any constraints on the passwords.

   Expressions are enclosed in curly braces and use the following metacharacters:

^	Start of string
$	End of string
\w	Represents alpha-numerics (a-z, A-Z, 0-9, _)

   Table 7-1 provides several examples of password expressions.

Table 7-1 Sample Password Expressions

Expression Type	Example	Description
Valid	.{6}	Passwords must be at least 6 characters long.
	{^\w{5,16}$}	Passwords must be between 5 and 16 characters long and can contain only alphanumeric characters and an underscore. Explanation of expression: ■ Match whole string using the **^** (start) and **$** (ending) metacharacters ■ Match alphanumerics, which are represented by the **\w** metacharacter ■ Match a length of 5 characters minimum and 16 characters maximum: **{5,16}**
	{^.*(?=.{10,})(?=.*\d)(?=.*[a-z])(?=.*[A-Z])(?=.*[@#$%^&+=]).*$}	Passwords must be at least 10 characters long and contain at least one lowercase letter, one uppercase letter, one digit, and one special character. Explanation of expression: ■ Match whole string using the **^** (start) and **$** (ending) metacharacters ■ Match a length of at least 10 characters: **(?=.{10,})** ■ Match at least one digit: **(?=.*\d)** ■ Match as least one lowercase letter: **(?=.*[a-z])** ■ Match as least one uppercase letter: **(?=.*[A-Z])** ■ Match as least one of the following special characters: **(?=.*[@#$%^&+=])**
Invalid	.{9}	Passwords cannot exceed eight characters in length.
	{^[a-zA-Z]*$}+	Passwords that contain *only* alphabetic characters are invalid. In other words, passwords must contain at least one number or special character.

6. Click **OK** to save your changes.

Enabling E-mail Notification

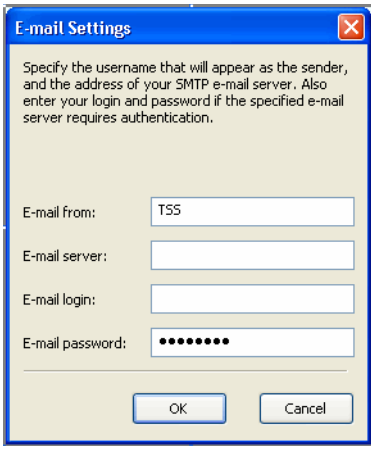
You can set up an authenticated SMTP server to enable Trillium Software System software to send e-mail messages to TSS users if one of the following events occur:

- Business rules fail
- Business rules succeed
- Exported rows complete
- Exported rows fail
- Load job completes
- Load job fails

Using the **Email Settings** window in the Repository Manager, you can set up an authenticated SMTP server. An authenticated SMTP server requires a username and password to send mail or to change the port on which the SMTP server listens.

▶ To enable e-mail notification with authentication

1. Log in to the Repository Manager.
2. On the Home tab, click **E-mail Settings**.The E-mail Settings window opens.



E-mail Settings

Specify the username that will appear as the sender, and the address of your SMTP e-mail server. Also enter your login and password if the specified e-mail server requires authentication.

E-mail from: TSS

E-mail server:

E-mail login:

E-mail password: ●●●●●●●●●

Figure 7-6 Email Setting Window

3. In **E-mail From**, enter the name you want to appear in the From field of the recipient’s e-mail messages.
4. In **E-mail Server**, enter the host name of the SMTP server that will send the e-mail.
5. In **E-mail Login**, enter the username that is required to log on to the server.
6. In **E-mail Password**, enter the password that is required to log on to the server.

7. Click **OK** to save your changes.

   ⓘ *After you have enabled e-mail notification, each user can configure his personal preferences for how the e-mail messages are constructed. That configuration is done in the Control Center.*

Changing HTML Format

Each TSS client references an extensible style sheet to generate HTML reports. This style sheet can be customized to meet your site requirements. The most likely modification that you will make is to edit the headers and footers to reflect your company name and legal statements.

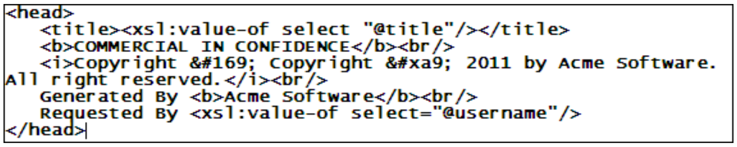
▶ To customize HTML headers and footers

1. On the server machine, go to the directory:

   `<server path>/TSSUI/bin`

   where *`server path`* is the installation location of the TSS repository server.

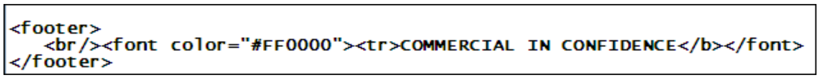
2. Locate the file named **Default.xslt**.
3. Make a backup of this file.
4. Open the file for editing.
5. Edit the header section to reflect your site requirements.



```
<head>
   <title><xsl:value-of select "@title"/></title>
   <b>COMMERCIAL IN CONFIDENCE</b><br/>
   <i>Copyright &#169; Copyright &#xa9; 2011 by Acme Software.
All right reserved.</i><br/>
   Generated By <b>Acme Software</b><br/>
   Requested By <xsl:value-of select="@username"/>
</head>
```

Figure 7-7 Header section

6. Edit the footer section to reflect your site requirements.



```
<footer>
   <br/><font color="#FF0000"><tr>COMMERCIAL IN CONFIDENCE</b></font>
</footer>
```

Figure 7-8 Footer section

7. Save the file and close it.

Repository Manager Options

The Repository Manager enables you to specify your preferences for several environment settings, including two startup options and several List View options. It also enables you to control the style of the user interface by specifying the font and color palette you want to apply. These options do not affect the Control Center.

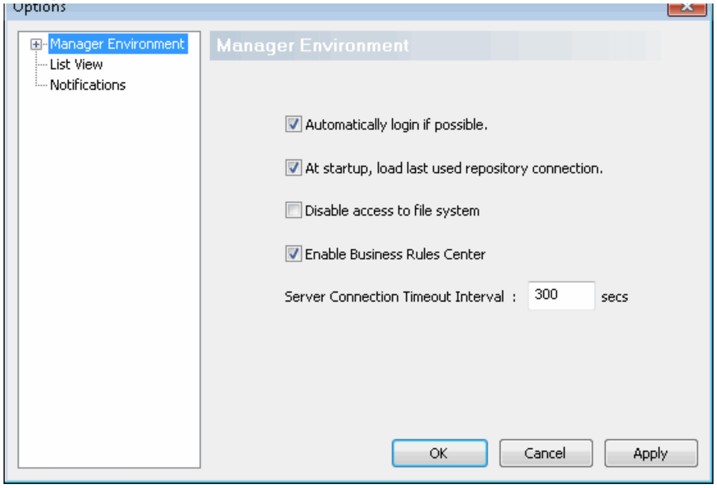
Startup Options

When you start up the Repository Manager, TSS verifies your credentials. Once verification is complete, TSS automatically connects you to the repository connection you last used. You can modify the way the startup process works:

- If you are using Windows-authenticated users, TSS bypasses the login process by default. If you prefer, you can require the repository administrator to log into the Repository Manager.
- By default, TSS automatically connects you to the most recently used repository connection. If you use several repository connections, you can opt to display the Connection window at startup so you can select the one you want to use.

▶ To define your startup options

1. On the Home tab, click **Options**. The Manager Environment Options window opens.



☑ Automatically login if possible.

☑ At startup, load last used repository connection.

□ Disable access to file system

☑ Enable Business Rules Center

Server Connection Timeout Interval : 300 secs

Figure 7-9 Manager Environment Window

2. Select from the following options:

   - **Automatically login if possible**. This option is applicable for Windows-authenticated users only. Automatic login is selected by default. Clear the option if you want the user to authenticate themselves during login.
   - **At startup, load last repository connection**. When you select this option, the Connection window is by passed during the login. Disable this option if you regularly connect to different repository servers.
   - **Disable access to file system**. When you select this option, access to local file system is disabled. As a result, in both the Repository Manager and Control Center applications:
     - New, Open, Save Session and Save as session options from the Application menu are disabled.

- User is not prompted to save changes to the session when switching the user or repository server or when closing an application.
- User cannot access Trillium Software documentation by clicking the **Help** option (?) and then clicking on **Manuals**.
- Print and Quick Print options are disabled.
- The Metadata Summaries print icon (🖶) is disabled.

In the Control Center:

- User does not see the Contact Support and View Documentation in the Control Center Start page.
- Create entity from client setting is disabled.
- In the List View:
  - Export to local system menu options (for entity, projects etc) and corresponding ribbon bar button are disabled.
  - Copy menu item also disabled.
  - Save Layout, Load Layout and Export options (from both menu drop-downs and the Ribbon bar buttons) for E-R diagram are disabled.
- In the Icon View the Save button, and Export and Copy options are disabled.
- In the Library tab, the export to local file system or import from local file system is disabled for Business Rules, Word definitions, Ignore words tables in the Library tab of the Control Center.
- Export and Generate Schema option for entity is disabled.
- Import project for TS Quality projects and Generate DDL for TS Quality entities are disabled.
- In Quick Access Toolbar, any buttons related to above operations are disabled.
- On the Manage bar, when you right-click an icon, the View Source option is disabled.
- When creating and editing notes, the Print and Export buttons are disabled in the New Note or Note for *note_name* windows.
- On the Advanced tab, selecting the Edit Schema icon is disabled.
- The message `This feature is not supported` is displayed:
  - When user clicks Help > Manuals
  - When user clicks View Documentation in the Getting started page

3. **Enable Discovery Center**. When you select this option, users can access the Discovery Center application from the Control Center. The Discovery Center is a web-based application that allows Trillium Software business users to manage their business rules and entities (data sources), add and run rules to view analysis results, create new data sources, modify multiple rules at a time, and create and manage

rules sets (groups of library rules). See also, Logging in to the Discovery Center.

4. Click **OK** to save your changes.

List View Options

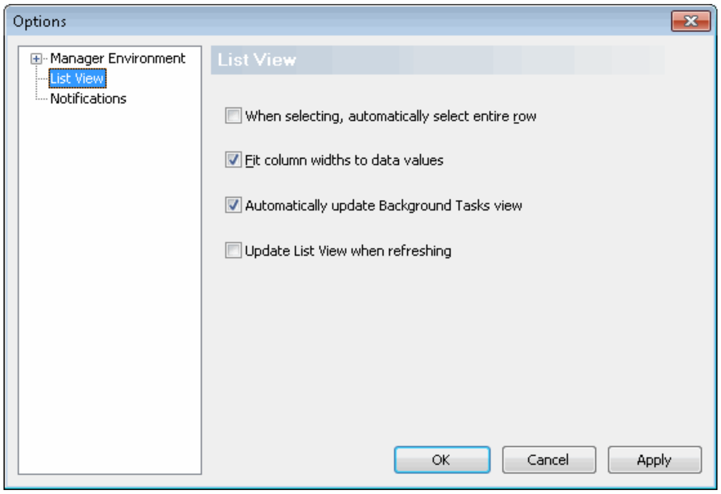
You use List Views to display information about loader connections, repositories, and other objects. By default, List Views have the following characteristics:

- Column widths are determined by the width of the data they contain.
- Background Tasks View is automatically updated as the status of tasks changes.
- List View is not automatically updated when the Refresh option is selected.

▶ To change the List View options

1. On the Home tab, click **Options**. The Manager Environment Options window opens.

2. Click **List View** in the Navigation View. The List View Options window opens.



□ When selecting, automatically select entire row

☑ Fit column widths to data values

☑ Automatically update Background Tasks view

□ Update List View when refreshing

Figure 7-10 List View Options Window

3. Enable and disable the features as necessary. The options are:

   - **When selecting, automatically select entire row**.This option is not enabled by default.
     When this option is selected, if you click a cell in List View, the entire row is selected. Un-check this option so that only the selected cell and not the entire row is selected.
   - **Fit column widths to data values**. This option is enabled by default.

Column widths are sized to fit the widest data value in the column. Un-check this option so that column widths are based on the size of the column label.

- **Automatically update Background Task View**. This option is enabled by default.
  The Background Tasks View is automatically updated as the status of tasks changes. Un-check this option if you require a static display of the Background Tasks view.
- **Update List View when refreshing**. This option is not enabled by default.
  Enable this option if you prefer that the List View data is automatically updated when you click the Refresh button.

4. Click **Apply** and then click **OK** to save your changes.

Selecting a Display Style

You can customize the color palette and font of the Repository Manager display. When you change the palette, the ribbon and border colors are affected. When you change the font, you change the text that appears in the Navigation and List Views.

▶ To change the color palette

1. Click **Style** on the right side of the ribbon. A list of palette options opens.

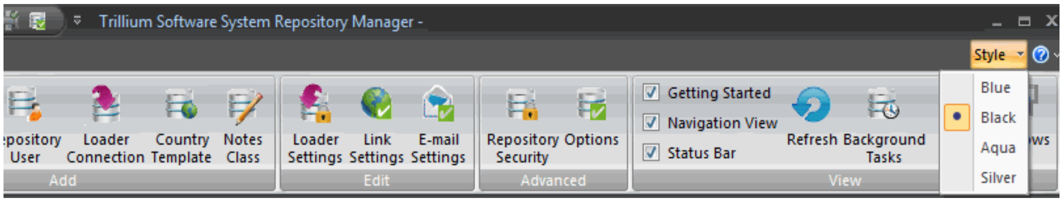


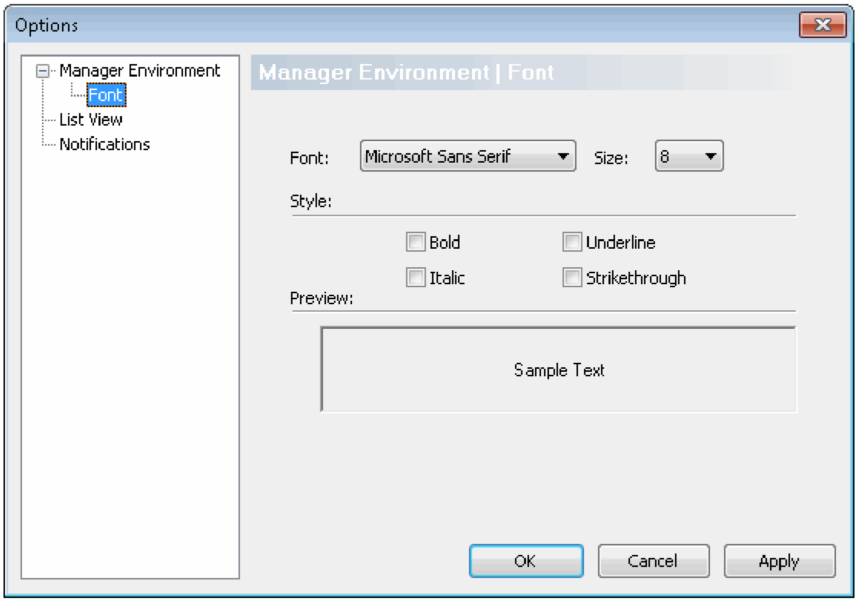
Figure 7-11 Style Options

2. Click the name of the color palette you want to apply. (The default palette is silver.)

▶ To change the font

1. On the Home tab, click **Options**. The Manager Environment Options window opens.
2. Expand **Manager Environment** and click **Font**.

   The Manager Environment/Font window opens.



Font: Microsoft Sans Serif Size: 8

Style:

□ Bold □ Underline

□ Italic □ Strikethrough

Preview:

Sample Text

Figure 7-12 Font Window

3. From the Font drop-down, select a font.
4. From the Size drop-down, select a font size.
5. Select from Bold, Underline, Italic and Strike-through formats as necessary.

ⓘ *The default font is Microsoft Sans Serif and the default size is 8.*

6. Click **Apply** to see how the changes look in the interface.
7. Click **OK** to save your changes.

Logging in to the Discovery Center

When the **Enable Discovery Center** option is selected (default), in the Options window (see Startup Options), users can access the Discovery Center from the Trillium Control Center.

Users can also access the Discovery Center from a web browser. You must provide the users the URL they should use to access the Discovery Center.

The URL consists of the web server hostname, port number, and the term discovery; for example: `https://localhost:3000/discovery`

where:

- `localhost` is the Web server machine name
- `3000` is the port number the TS Web Server will listen on

The web server is required for communication between the repository server and the Discovery Center and Administration Center. On Sun SPARC and AIX the web server is not installed by default and you need to install it to a Windows system. For more information on the installing the standalone TS Web Server, see the *Trillium Software System Installation Guide*.

ⓘ *If the Discovery Center and Administration Center do not open, verify that the TSS - RestAPIs service is started on the TSS server system.*

Logging in to the Administration Center

Discovery Center administrators access the Administration Center from a web browser in much the same way that Discover Center is accessed.

The URL consists of the web server hostname, port number, and the term admin; for example: `https://localhost:3000/admin/`

Configuring Locale for Language Packs

When you install a language pack, you must set the REST API locale parameter in the TSS server `config.txt` file. Specifying the REST API locale ensures that the proper country locale is configured for applications that use the Trillium REST API, including Discovery Center, Quality Monitor, and Administration Center.

Run this procedure after you install a TSS language pack in your environment.

▶ **To configure Trillium REST API locale**

1. Stop the scheduler and Rest APIs services.
2. Open the `config.txt` file on your server system. The file is in the `etc` directory in your TSS server installation location. For example, on Windows this default location is: `C:\Program Files x86\Trillium Software\MBSW\15\etc`
3. Locate and edit the `rest_api` parameter for your language. The following languages and values are supported:
    - English = en (default)
    - Brazilian Portuguese = pt_BR
    - Chinese, Mainland = zh_CN
    - Chinese, Taiwan = zh_TW
    - French = fr
    - German = de
    - Italian = it
    - Japanese = ja

- Korean = ko
- Spanish = es

ⓘ *You may need to add the key/value pair in the file if not already there.*

For example, to set the locale to display strings in Taiwan Chinese, add or edit the settings as follows:

```
key rest_api {
    value locale "zh_TW"
}
```

If the locale setting has not been set or has not been added to the `config.txt` file, the locale defaults to English.

4. Save and close the file.
5. Restart the scheduler and Rest APIs services.

ⓘ *You must also configure the locale environment variable (TSSLOCALE) to use the Control Center and Repository Manager in your language. See the TSS Help for more information.*

CHAPTER 8

Repository Maintenance

This chapter discusses the procedures related to maintaining the TSS repository. It includes the following topics:

Recovering the Control Center

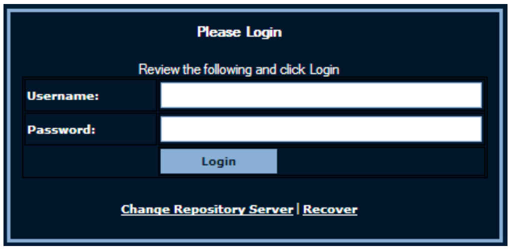
The Recover function enables the repository administrator (also known as the *boot user*) to unlock the Control Center if it is malfunctioning. For example, you will recover the Control Center if a TSS user is unable to log on.

The Recover function deletes temporary files and stops any processes that are running. It also stops the TSS Scheduler, deletes the Schedule.db file, and restarts the Scheduler.

ⓘ *Use the Recover function only when instructed to by Trillium Software Customer Support.*

▶ To use the Recover function

1. In the Repository Manager, click the Application button menu (top left) and select the **Switch User or Repository** option. The Switch User command automatically logs you out and opens the Login prompt.



Please Login

Review the following and click Login

Username:

Password:

Figure 8-1 Repository Manager Login Window

2. In the **Username** text box, enter the name of the repository administrator.

3. In the **Password** text box, enter the password associated with the user.

4. Click **Recover**.

5. Follow any additional instructions provided by Customer Support.

Adding a Repository Server

The primary repository server is created during the installation of Trillium Control Center. You can create additional repository servers during initial installation or you can add them later following the instructions below.

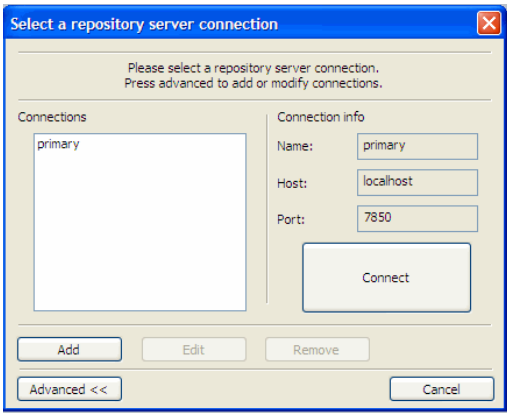
ⓘ *Before you begin, make note of the host names and ports of the repository servers you want to add.*

▶ To add a repository server

1. On a client machine, start the Control Center or Repository Manager and log in.

   ⓘ *Login as the Repository Manager or as a member of the Windows Administrators group to make the new repository server available to all users on the system. Otherwise, the new repository server will be available only to the user who creates it.*

2. Click the **Application** button and select **Switch Repository Server**. The current session closes and the Select Repository Server Connection window opens (Figure 8-2).



Select a repository server connection

Please select a repository server connection.
Press advanced to add or modify connections.

Connections: primary

Connection info

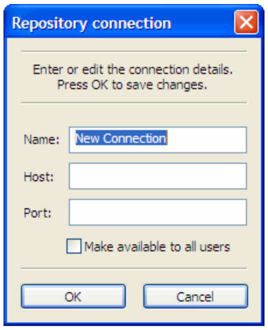
Name: primary

Host: localhost

Port: 7850

Figure 8-2 Select Repository Server Connection

3. Click **Advanced**, then click **Add**. The Repository Connection window opens.



Repository connection

Enter or edit the connection details.
Press OK to save changes.

Name: New Connection

Host:

Port:

□ Make available to all users

Figure 8-3 Add Repository Server Connection

4. In the **Name** field, enter a descriptive name.

5. In the **Host** field, enter the name of the host server.

6. In the **Port** field, enter the port number of the host server.

7. To make the repository available to all users, select the **Make available to all users** option.

8. Click **OK** to save your changes.

Backing Up a Repository

Trillium Software recommends that you back up the repositories as a routine maintenance task. You do not need to back up the `tmp` directory since the files within this folder are transient in nature and are not needed for a successful repository restoration.

ⓘ *Before you perform a repository backup or restore, verify that all users are logged off Trillium Software System and no background tasks are running.*

Windows Systems

▶ To back up a repository on Windows

1. Log on to Trillium Software System as a Windows administrator.
2. Verify that no TSS activities are running on the server. (See Monitoring Repository Activities on page 93 for information.).

   ⚠ *This is a **very** important verification. Stopping the TSS Scheduler while a TSS activity (data imports, join processing, etc.) is running can leave a repository in an unstable condition.*

3. Open the Windows **Start > Control Panel > System and Security > Administrative Tools > Services**.
4. Stop the TSS - inetd and TSS Scheduler services:
   a. Right-click on **TSS 15 - Scheduler** and select **Stop**.
   b. Right-click on **TSS - inetd** and select **Stop**.
5. Leave the **Services** window open.
6. Locate the repository path.

   ⓘ *When you installed TSS, you were asked to define a path to the repository database (for example, C:/Documents and Settings/ All Users/Application Data/Trillium Software//MBSW/Data).*

7. Make a copy of the entire data directory structure, *excluding* the **tmp** folder.
8. Save the backup copy on a different server or on external media.
9. After the backup is complete, return to the Services window and restart the TSS - inetd and TSS Scheduler services:
   a. Right-click on **TSS - inetd** and select **Start**.
   b. Right-click on **TSS 15 - Scheduler** and select **Start**.

UNIX Systems

▶ To back up a repository on UNIX

1. Log on to the TSS Server as the UNIX root user.
2. Verify that no TSS activities are running on the server. (See Monitoring Repository Activities on page 93 for information.).

   ⚠ *This is a **very** important verification. Stopping the TSS Scheduler while a TSS activity (data imports, join processing, etc.) is running can leave a repository in an unstable condition.*

3. Change directories to **`/<Server Path>/metabase/bin`**.

   ⓘ *When you installed Trillium Software System, you were asked **Where would you like to install the Repository Server?** This is the directory represented by **<Server Path>**.*

4. Stop the TSS Scheduler:

   ```
   ./scheduler -stop
   ```

5. Locate the repository directory on the server (`/<Repository Path>/version/Data`).
6. Make a backup copy of the entire directory structure, *excluding* the **`tmp`** directory.
7. After the backup is complete, change directories to `/<Server Path>/metabase/bin`
8. Start the Scheduler by typing the command:

   ```
   ./scheduler -start
   ```

Restoring a Repository

This section includes:

- Restoring a Repository on Windows Systems
- Viewing Repository Background Tasks

Restoring a Repository on Windows Systems

Use these instructions to restore a repository on a Windows system.

▶ To restore a repository on Windows

1. Log on to TSS as the Windows administrator.

2. Verify that no TSS activities are running on the server. (See Monitoring Repository Activities on page 93 for information.).

   > ⚠ *This is a **very** important verification. Stopping the TSS Scheduler while a TSS activity (data imports, join processing, etc.) is running can leave a repository in an unstable condition.*

3. Open the Windows **Start > Control Panel > System and Security > Administrative Tools > Services**.
4. Stop the TSS - inetd and TSS Scheduler services:

   a. Right-click on **TSS 15 - Scheduler** and select **Stop**.

   b. Right-click on **TSS - inetd** and select **Stop**.
5. Leave the **Services** window open.
6. Locate the repository path. This is the path you defined when you installed TSS (for example, C:/Documents and Settings/All Users/Application Data/Trillium Software//MBSW/Data).
7. Make a backup of the **logs** directory and its contents.
8. Restore the repository backup to the repository directory to the exact location from where the backup was made.
9. Copy the backup of the **logs** directory (that you created in step 7) on top of the old **logs** directory that was just restored.
10. Change to the **tmp** directory and delete its contents.
11. Return to the **Services** window and restart the TSS - inetd and TSS - Scheduler services:

    a. Right-click on **TSS - inetd** and select **Start**.

    b. Right-click on **TSS 15 - Scheduler** and select **Start**.

Restoring a Repository on UNIX Systems

Use these instructions to restore a repository on a UNIX operating system.

▶ To restore a repository on UNIX

1. Log on to the repository server machine as a UNIX root user.

2. Verify that no TSS activities are running on the server. (See Monitoring Repository Activities on page 93 for information.).

   ⚠ *This is a **very** important verification. Stopping the TSS Scheduler while a TSS activity (data imports, join processing, etc.) is running can leave a repository in an unstable condition.*

3. Change directories to **`/<Server Path>/metabase/bin`**.

   ⓘ *When you installed Trillium Software System, you were asked **Where would you like to install the Repository Server?** This is the directory represented by **<Server Path>**.*

4. Stop the TSS Scheduler:

   ```
   ./scheduler -stop
   ```

5. Change directories to the repository path.

   ⓘ *When you installed Trillium Software System, you were asked **Where would you like to install your Repository Environment?** Go to this directory.*

6. Make a backup of the **logs** directory and all of its contents.
7. Restore the repository backup to the repository directory.

   ⓘ *The repository must be restored to the exact location from which the backup was made.*

8. Copy the backup of the **logs** directory (that you created in step 6) on top of the logs directory that was just restored.
9. Change to the **tmp** directory and delete its contents.
10. Change directories to **`/<Server Path>/metabase/bin`**.
11. Start the Scheduler by typing the command

    ```
    ./scheduler -start
    ```

Monitoring Repository Activities

The following tasks help you monitor activities from the Repository Manager:

- Viewing Repository Background Tasks on page 94
- Viewing Log File on page 94
- Managing Log Files on page 95
- Setting Log File Logging Level on page 96
- Viewing Messages on page 96

ⓘ *The Control Center provides similar tools Control Center users to monitor activities that affect the entities to which they have access.*

Viewing Repository Background Tasks

You can display a list of background tasks to monitor activities related to all repositories in the repository server. The list includes detailed information about each task, including:

- The type of activity
- The names of the repository and entity related to the activity
- The current status of the activity

▶ To display a list of background tasks

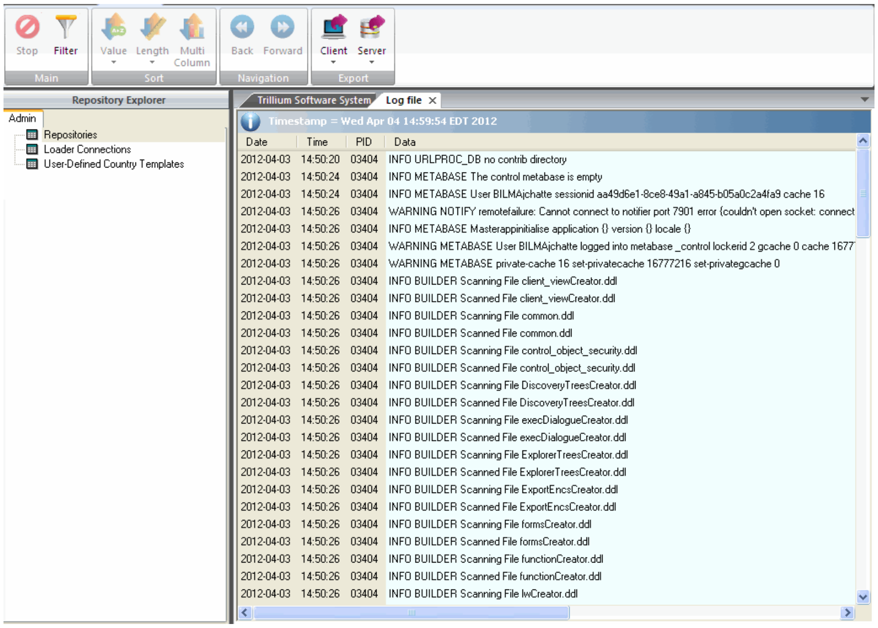
On the Home tab, click **Background Tasks**.

Viewing Log File

The Repository Manager makes a log entry each time an activity that affects the repository takes place in the log file. The log file, `mtb_server.log`, is located in `C:\ProgramData\MBSW\Data\logs` directory. The entry records the date, time, identification number, and affected data for each activity. The log files provide valuable information in tracing TSS user activity and help in troubleshooting data profiling and cleansing efforts as well as any unplanned events.

▶ To display the log file

On the Home tab, click **Log File**. The log opens in the List View pane.



```
Timestamp = Wed Apr 04 14:59:54 EDT 2012
Date        Time      PID    Data
2012-04-03  14:50:20  03404  INFO URLPROC_DB no contrib directory
2012-04-03  14:50:24  03404  INFO METABASE The control metabase is empty
2012-04-03  14:50:24  03404  INFO METABASE User BILMAjchatte sessionid aa49d6e1-8ce8-49a1-a845-b05a0c2a4fa9 cache 16
2012-04-03  14:50:26  03404  WARNING NOTIFY remotefailure: Cannot connect to notifier port 7901 error {couldn't open socket: connect
2012-04-03  14:50:26  03404  INFO METABASE Masterappinitialise application {} version {} locale {}
2012-04-03  14:50:26  03404  WARNING METABASE User BILMAjchatte logged into metabase _control lockerid 2 gcache 0 cache 1677
2012-04-03  14:50:26  03404  WARNING METABASE private-cache 16 set-privatecache 16777216 set-privategcache 0
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File client_viewCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanned File client_viewCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File common.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanned File common.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File control_object_security.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanned File control_object_security.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File DiscoveryTreesCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanned File DiscoveryTreesCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File execDialogueCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanned File execDialogueCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File ExplorerTreesCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanned File ExplorerTreesCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File ExportEncsCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanned File ExportEncsCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File formsCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanned File formsCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File functionCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanned File functionCreator.ddl
2012-04-03  14:50:26  03404  INFO BUILDER Scanning File lwCreator.ddl
```

Figure 8-4 Repository Manager Log File

Managing Log Files

By default, when the log reaches 1 GB, the mtb_server.log file is renamed with a timestamp and a new log is started. The log files can use up a lot of disk space and this could lead to poor performance. To avoid this problem, you can control the file size before renaming occurs and specify how many backups to keep in your disk.

▶ To manage the server log files

1. Close the Repository Manager.
2. Go to the server installation directory and look for the etc folder. For example, if you selected the default directory during the server installation, the etc folder is in C:/Program Files (x86)/Trillium Software/MBSW/15. Within that folder is the config.txt file.
3. Open the config.txt file in a text editor.
4. Locate the following parameters under the key server section:

```
value logfile_size_max 1024
value logfile_backups 16
```

   - **logfile_size_max** - The maximum size of the log file in megabytes. For example, logfile_size_max 500 means a maximum log file size of 500MB. If this parameter is omitted or set to '0' a default log file size of 1024MB is applied.
   - **logfile_backups** - The number of backup versions of the log file. For example, logfile_backups 4 means keep a maximum of 4 backup files. If this parameter is omitted or set to '0' the maximum number of log files is set to 255.

   When the current mtb_server.log file exceeds its configured maximum size a new backup file is created, and if the total number of backup files now exceeds the user-configured maximum, the files with the oldest timestamp are deleted until only the maximum number of backup files remain.

   Backup files are identified as the filename pattern of mtb_serverYYYYMMDD.log where YYYYMMDD is a datestamp in the inclusive range 10000101 through 29991231.

   ⓘ *Usually only a single backup file will be deleted, but if you have recently reconfigured the value of logfile_backups then more than one backup file may be deleted.*

5. Change the default values of the parameters.

6. Save and close the file.
7. Restart the Repository Manager.

Setting Log File Logging Level

The `mtb_server.log` log file is used to help track repository server activities and troubleshoot issues. You can set the level of logging information returned to the file by adjusting the loglevel entry in the TSS `config.txt` file on your server system.

There are three valid levels of logging available, including:

- **INFO** (default). Returns all available details for repository server activities.
- **WARNING**. Returns warning information about repository server activities.
- **FATAL**. Returns information about fatal repository server problems.

▶ To set the log file logging level

1. Close the Repository Manager.
2. Go to the server installation directory and look for the `etc` folder. For example, if you selected the default directory during the server installation, the `etc` folder is in `C:/Program Files (x86)/Trillium Software/MBSW/15`. Within that folder is the `config.txt` file.
3. Open the `config.txt` file in a text editor.
4. Locate the `value loglevel` parameter under the key server section.
5. By default, the value is set to INFO. Change this value to WARNING or FATAL as needed.
6. Save and close the file.
7. Restart the Repository Manager.

Viewing Messages

You can view messages about changes to the repository as they occur by enabling the **Messages** option.

▶ To display messages

At the bottom of the Repository Manager window, click **Messages**.

The message pane opens at the bottom of the window. To enlarge the pane to view more messages without scrolling, drag the top border up.

CHAPTER 9

Command Line Utility

This chapter describes how to use the Repository Server command line utility, mtb_admin to perform many of the same tasks. It includes:

Command Line Options

You can run the Repository Server command line utility, mtb_admin, from:

- The operating system command prompt
- The Repository Administration command prompt

ⓘ *You must have repository administrator privilege to use the mtb_ admin utility.*

Running mtb_admin utility from System Command Prompt

This section describes how to run the mtb_admin utility directly from a UNIX or Windows command prompt. You can enter commands directly or include them in a script.

▶ **To run mtb_admin from an operating system command prompt**

1. Open the command prompt, change to the directory that contains the `mtb_admin.exe`. (It is stored in the `C:/Program Files (x86)/ Trillium Software/MBSW/bin`.)
2. Enter the following command on a single line:

```
mtb_admin -user userid -password password -metabase
repository command
```

For example:

```
mtb_admin -user admin@xxx.abc-corp.com -password
```

```
admin -metabase _control jobs
```

3. Press **Enter**. This command starts the utility, logs the repository administrator onto the _control repository, and runs the command jobs as described in Command—jobs.

   ⓘ *If you are using Windows-authentication with TSS, you must log in with your fully qualified user name.*

▶ To run a script from the operating system command prompt

1. Create the script as a text file.
2. At the command prompt, change to the directory that contains the `mtb_admin.exe`. The utility is stored in the `C:/Program Files (x86)/Trillium Software/MBSW/bin`.
3. Enter the following command on a single line:

```
mtb_admin -user userid -password password -metabase
repository -readfile path\script.txt
```

   For example:

```
mtb_admin -user admin@xxx.abc-corp.com -password
admin -metabase _control -readfile c:\myscript.txt
```

4. Press Enter

Repository Administration Command Prompt

The administrator uses the Repository Administration Command Prompt to perform tasks from a command line interface for the mtb_admin utility. The `mtb_admin` utility is located in the bin subdirectory of the server application install location (for example, C:/Program Files (x86)/Trillium Software/MBSW).

▶ To access the administration command prompt from Windows

1. Log on to the server machine where Trillium Software System is installed.
2. From the Start menu, select **Start > All Programs > Trillium Software > TSS 15 > Repository Administration Command Prompt**. The command window opens.
3. At the login prompt, enter the repository administrator name and press **Enter**.

   ⓘ *If you have set up TSS to use Windows-authenticated users, you must use your fully qualified user name to log in (for example, admin@networkdomain.com).*

4. At the password: prompt, enter the password for the repository administrator and press **Enter**.

   ⓘ *If only the _control repository is defined, you are automatically* *logged into that repository. If multiple repositories exist, proceed to step 5.*

5. At the repository: prompt, do one of the following:

   a. Enter the name of the repository in which you want to work and press Enter.

   b. Press Enter to log on to the _control repository.

▶ To access the command prompt from UNIX

1. Log on to the server where Trillium Software System is installed.

2. Change to the directory <*serverpath*>/metabase/bin, where <*serverpath*> is the location of the Trillium Software System server application (for example, /workarea/TrilliumSoftware/MBSW).

3. Type the following command at the command prompt:
   ```
   ./mtb_admin
   ```

4. At the login: prompt, enter the repository administrator name and press Enter.

   ⓘ *The repository administrator or boot user is defined during installation. Refer to Trillium Software System Installation Guide for more information.*

5. At the password: prompt, enter the password for the repository administrator and press Enter.

   ⓘ *If only the _control repository is defined, you are automatically logged into that repository. If multiple repositories exist, proceed to step 6.*

6. At the repository: prompt, do one of the following:

   a. Enter the name of the repository in which you want to work and press Enter.

   b. Press Enter to log on to the _control repository.

Commands

Table 9-1 is an alphabetical list of the commands you can issue from the command prompt. Each command is described in greater details in the subsequent sections.

Table 9-1 Commands and their description

Command	Description
add	Adds a repository object or script.
copyentity	Copies the specified entity.

createentity	Creates a dynamic entity and returns the entity's identification number, referred to as the *entity_id*.
define alt_metabase	Creates an alternative directory for TS Quality data files. This command is described in *Chapter 2: Repositories on page 20*.
defineudpvdir	Defines virtual directories for executables used in the User-Defined Procedure module of TS Quality.
definesrtvdir	Defines virtual directories for the temporary work files used by the Sort process.
delete entity	Deletes a specified entity.
delete userscripts	Deletes a user-defined script registered with the Control Center.
deletetxnlogfiles	Deletes the internal transaction log files associated with the current repository.
deletealltxnlogfiles	Deletes the internal transaction log files associated with all repositories in the repository server.
edit userscripts	Modifies a user-defined script or the parameters associated with the script.
expert	Invokes "super-user" mode which expands the list of commands that are available.
export	Exports the List View contents to a file that you specify.
fixRules	Updates every rule in the (currently logged in) repository. Displays the rule ids whose expressions have been changed.
job	Shows the status of a specific running or completed job. Requires a job identification number.
jobs	Shows the status of all running or completed jobs.
loaddata	Creates an entity by loading data from a data source into a repository.
loadentity	Loads entities.
newuser	Creates a new user.
purge	Deletes temporary indexes.
print	Prints a response to the mtb_admin prompt after a job completes.
runrules	Runs Entity and Attribute Business Rules.
set	Assigns a variable.
show userscripts	Displays all user-defined scripts registered with the Control Center.
tscreate	Creates a Time Series project.
tsgeneration	Generate the next entity in a Time Series project.
wait	Blocks use of the command line until a job completes.

Command–`add`

You use the `add` command to add a repository object, such as a loader connection or user.

You can also use the `add` command to add the name of a user-defined script to an internal repository table that is displayed in the Control Center. Users can then run the script and pass arguments to it by selecting its name from the **Execute Server Action** list in the Control Center.

ⓘ *The command must be typed on a single line. The parameters vary, depending on the type of object you are adding.*

The add userscripts command must be run from the _control repository.

Syntax

```
add <repository_object> <param1 value> <param2 value>...
```

Syntax (add loader_connection)

```
add loader_connection NAME "<connection name to display>"
TYPE <Loader Connection Type>
DESCRIPTION {"<description to display>"} PARAMETERS {DATA_
DIRECTORY "<data file path using forward slashes>"SCHEMA_
DIRECTORY "<data file path using forward slashes>"DATA_
EXTENSIONS {<ext1> <ext2> <ext3>} SCHEMA_EXTENSIONS {<ext1>
<ext2> <ext3>}}
```

See Table 4-4 on page 54 for a description of the variables in this command.

Example

The following example creates a loader connection to a COBOL data source. In practice, the command must be typed on *one line* but is shown here on multiple lines for readability.

```
add loader_connection
     NAME “Team A Data”
     TYPE cobol
     DESCRIPTION “data from the marketing team”
          PARAMETERS {
          DATA_DIRECTORY “marketing/data/”
          SCHEMA_DIRECTORY “marketing/schemas/”
          DATA_EXTENSIONS {dat cbl}
          SCHEMA_EXTENSIONS {cpy cbl}
          }
```

Syntax (add userscripts)

```
add userscripts NAME "<display name>" DESCRIPTION
"<description of script>" LOCATION "<full path>"
```

where

NAME "<display name>"	Name of the script as you want it to appear in the Execute Server Action submenu in the Control Center.

DESCRIPTION "<description>"	Description of the user-defined script.
LOCATION "<full path>"	Location of the script, including filename. Use forward slashes to define the path.

Optional Parameters

Parameter	Description
SECURITY <1|0>	Type 1 to hide the script from limited users; type 0 to display the script to all users. (The default is 0.)
FIXEDARGS <arg1 arg2>	Arguments that are always passed to the script. You can use a literal value as an argument or one of the following: $METABASE passes the name of the active repository $USERNAME passes the name of the logged on user $DATADIR passes the name of the repository data directory $BINDIR Passes the name of the binaries directory
USERARGS <n>	The number of arguments that need to be passed to the script by the user. The Control Center prompts only once for these arguments, and the user must respond with all the values, separated by a space. An individual argument cannot contain a space.

ⓘ *The add userscripts command must be run from the _control repository. If the add userscripts command contains both FIXEDARGS and USERARGS, the FIXEDARGS are passed to the script first.*

The add userscripts command returns an ID number for the script. Make a record of that ID for future reference; you will need it if you want to delete the script from the table at some point.

Example

The following example adds a user-defined script named **Trim Log** to the internal repository table that is displayed in the **Tools > Execute Server Action** submenu. The command returns an ID number for the script, which you will need if you want to delete the script at a later time.

ⓘ *In practice, the command must be typed on one line but is shown here on multiple lines for readability.*

```
add userscripts
    NAME "Trim Log"
```

```
DESCRIPTION "user-defined script to trim log files"
PARAMETERS {
     LOCATION "C:/temp/trim.bat/"
     SECURITY 1
     FIXEDARGS 100 log
     USERARGS 2
     }
```

Example of a Script

The following is a Windows batch file, shown here as an example of a user script.

```
echo off
set myparam1=%1
set myparam2=%2
set myparam3=%3
set myparam4=%4
echo %myparam1% > c:\temp\log1.log
echo %myparam2% > c:\temp\log2.log
echo %myparam3% > c:\temp\log3.log
echo %myparam4% > c:\temp\log4.log
```

Command—copyentity

The copyentity command lets you copy any entity stored in a repository. This includes entities that are created in the Control Center, as well as entities created using the command line interface (mtb_admin). A new entity is created to hold the copy.

The command requires that you know the identifier of the entity you want to copy. TSS gives each entity a unique identification number when it is created. This identifier is a number and displays in parenthesis in the Navigation View entity lists.

Syntax

```
copyentity {<entity_id1> <entity_id2 . . .<entity_idn>}
```

where *<entity_id1> <entity_id2. . .<entity_idn>* are the numbers that identify the entities you want to copy.

Optional Parameters

Parameter	Description
-workgroup <1|0>	Type 1 to create the copied entity in the same project as the source entity. Type 0 to create the copied entity outside of the project. If this parameter is not specified, the default is 1.

-joins <open|closed|empty>	Type open to copy all permanent joins; type closed to copy only joins in the listed entities; type empty if you do not want to copy any joins.

Example

This command copies the entity identified as "23" and ensures that the entity copy is *not* placed in the project that contains the source entity. It also ensures that joins are not copied.

```
copyentity 23 -workgroup 0 -joins empty
```

Command—createentity

The createentity command creates a **dynamic entity**. A dynamic entity is an entity that has data stored externally instead of in a repository. When you create a dynamic entity, the TSS performs only a limited analysis of the data.

ⓘ *To create a real entity, that is an entity that loads the data into TSS, you must use the loaddata command (see Command—loadentity on page 130).*

The createentity command returns the identifier of the entity that is created.

The required syntax for the createentity command depends on the data source type you choose for the entity. The command syntax is different if you are connecting to data in a Trillium file as compared to connecting to data that uses a COBOL copybook file.

The command supports both flat file sources and relational database sources. Refer to the Overview section of Chapter 4, for more information about supported data source types.

Command—createentity for COBOL Files

You can create a dynamic entity using a COBOL flat file that has a companion COBOL copybook (schema).

Before you issue the createentity command, create a loader connection for the COBOL data source. The loader connection specifies where the data source files are located and allows the createentity command to connect to the data source and initiate the entity creation process. (For information about creating a loader connection, see Chapter 4: Loader Connections on page 37).

Copybook and data file directories do not need to be specified when they are referenced within the loader connection set up by the repository administrator.

Syntax

```
createentity <loader_connection> datafile <filename>
schemafile <copybook_filename>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection.
<filename>	Name of the COBOL flat file that contains the data.
<copybook_filename>	Name of the schema file that corresponds to the COBOL flat file you specified as the data file.

Optional Parameters

Parameter	Description
username <user_name>	User ID required to validate the connection to the data source. Use this parameter only if a login name and password are required. Do not use the `username` and `password` parameters if the mtb_admin user is the data file owner.
password <password>	Password required to validate the connection to the data source. Use this parameter only if a login name and password are required.
jobname <job_name>	Job ID or name of the data load job.
charset <value>	Indicates the character encoding used by the COBOL data and copybook files; the options are `EBCDIC` or `ASCII`.
endian <value>	Indicates the byte order. The options are `big` (for Big Endian) and `little` (for Little Endian).
upacked <value>	Indicates the structure of the data. Set to `1` if you want to treat unsigned comp-3 fields as comp-6.
align <value>	Indicates one- or two-byte data alignment. The options are `one` and `two`.
redefines <value>	Indicates how to manage REDEFINES clauses in a copybook. The options are `all` and `first`. Type `all` to account for all REDEFINES clauses in a copybook. If this option is selected and the system encounters a REDEFINES clause, it removes the REDEFINES clause and keeps both representations of the data in the copybook. The data file will be populated to match the copybook. Type `first` to ignore all REDEFINES clauses in the copybook.

terminator <value>	Indicates how records in the data file are terminated. The options are lf (linefeed), cr (carriage return), crlf (linefeed and carriage return), and none. Typically, COBOL data files are fixed length and have no record delimiter. However, if COBOL data is exported from the original application and transferred into other file systems (especially UNIX), they could contain record delimiters added by the export or transfer process. If the file originated from: Windows, type crlf UNIX, type lf IBM mainframe, type none.
encoding <value>	Indicates the character encoding used by the data file. The options are EBCDIC and ASCII. EBCDIC data is translated into a correct ASCII representation on load. Generally, UNIX COBOL files will be ASCII and IBM mainframe data will be EBCDIC.
national <value>	National character encoding used by the data file. If your COBOL copybooks define national data items holding Unicode strings (such as PIC clause containing N and USAGE NATIONAL) or your compiler options from the data source are set to NSYMBOL (NATIONAL) or CODEPAGE, you should specify the National Character encoding standard of the data source.
columns <names>	Indicates the names of the columns from which to import data.
skip <number>	The number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file to load (for example, the first 1000 records).
random <percentage>	Indicates the degree to which you want to randomly sample a percentage of records from the file.

Example

This command uses a loader connection named cobolconn, provides a username and password to access the data directories, and then loads data from three columns in the testcp932.dat data file.

```
createentity cobolconn username radmin password serverheaven
encoding cp932 datafile q34fin95.dat schemafile q34fin95.cpy
columns {{Ref Id} Source Amount}
```

When multiple columns are represented within a space-delimited line of column names, be sure to enclose them in braces ({}). If a column name contains whitespace, enclose the column name in braces ({}) also.

Command—createentity for Delimited Files

You can create a dynamic entity using a delimited file that may or may not have a companion schema Data Dictionary Language (DDL) file.

Before you issue the createentity command, create a loader connection for the delimited data source. The loader connection specifies where the data source files are located and allows the createentity command to connect to the data source and initiate the entity creation process. (For information about creating a loader connection, see Chapter 4: Loader Connections on page 37.)

Syntax

```
createentity <loader_connection> datafile <filename>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection.
<filename>	Name of the delimited file that contains the data.

Optional Parameters

Parameter	Description
username <user_name>	User ID required to validate the connection to the data source. Use this parameter only if a login name and password are required. Do not use the username and password parameters if the mtb_admin user is the data file owner.
password <password>	Password required to validate the connection to the data source. Use this parameter only if a login name and password are required.
schemafile <filename>	Name of the schema file that corresponds to the delimited file you specified as the data file.
jobname <job_name>	Job ID or name of the data load job.
attr <header>	Indicates the header line. The options are: names—names on first line none—no attribute names specified ddl—names in schema file
delimiter <character>	Indicates the character that is used as the data delimiter in the file. A field can be delimited by whitespace, tabs, commas (CSV), periods (.), or other characters.

terminator <value>	Indicates how records in the data file are terminated. The options are lf (linefeed), cr (carriage return), and crlf (linefeed and carriage return). Typically, if the file resides on a Windows system, type crlf. If on a UNIX system, type lf.
encoding <name>	Character encoding used by the data file. The options are EBCDIC and ASCII. This parameter controls the character set for the file. EBCDIC data is translated into a correct ASCII representation on load. Generally, UNIX COBOL files will be ASCII and IBM mainframe data will be EBCDIC.
columns <names>	Indicates the names of the columns from which to import data.
skip <number>	Number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file (for example, the first 1000 records) to load.
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.

Example

This command creates a variable named entid that references the createentity command shown in brackets ([]).

```
set entid [createentity delimconn datafile nurserystock.txt
attr names delimiter . quote \"]
```

Command—createentity for ODBC-Compliant Databases

You can create an entity by importing data from an ODBC-compliant relational database.

Before you issue the createentity command, create a loader connection for the ODBC-compliant data source. The loader connection specifies where the data source files are located and allows the createentity command to connect to the data source and initiate the entity creation process. (For information about creating a loader connection, see Chapter 4: Loader Connections on page 37.)

Syntax

```
createentity <loader_connection> username <username>
password <password> table <table_name>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection.
<username>	User ID required to validate the connection to the data source. You must always provide the username and password parameters when importing data from a relational database that uses an ODBC connection.
<password>	Password required to validate the connection to the data source.
<table_name>	Name of the relational database table to be imported.

Optional Parameters

Parameter	Description
jobname <job_name>	Job ID or name of the data load job.
columns <names>	Indicates the names of the columns from which to import data.
skip <number>	Number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file to load (for example, the first 1000 records).
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.

Example

This command connects to an ODBC-compliant database and creates a dynamic entity for three columns in the Pilgrim Fund portfolio table.

```
createentity odbcconn username odbcuser password odbcpwd
table pilgrimfdfolio columns {stocks bonds {mutual funds}}
```

When multiple columns are represented within a space-delimited line of column names, be sure to enclose them in braces ({}). If a column name contains white space, enclose the column name in braces ({}) also.

Command—createentity for RDBMS Connections

You can create a dynamic entity that has data stored externally in an IBM DB2 or Oracle relational database management system (RDBMS). (For a list of database versions and platforms supported by Trillium Software System, see Relational Data Sources on page 38.)

Before you issue the `createentity` command, create a loader connection for the DB2 or Oracle data source. The loader connection specifies where the data source files are located and allows the `createentity` command to connect to the data source and initiate the entity creation process. (For information about creating a loader connection, see Managing Loader Connections on page 40)

Syntax

```
createentity <loader_connection> username <username>
password <password> table <table_name>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection.
<username>	User ID required to validate the connection to the data source. You must always provide the username and password parameters when connecting directly to a relational database management system (RDBMS).
<password>	Password required to validate the connection to the data source. Use this parameter if a login name and password are required.
<table_name>	Name of the relational database table to be imported.

Optional Parameters

Parameter	Description
jobname <job_name>	Job ID or name of the data load job.
columns {names}	Indicates the names of the columns from which to import data.
skip <number>	Number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file to load (for example, the first 1000 records).
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.

Example

This command connects to a loader connection named `db2conn` for an IBM DB2 database. There are three table columns that are used to create the dynamic entity: div A, player, and score.

```
createentity db2conn username db2user password db2pwd
jobname db2_dyn1 table littleleague columns {{div a} player
score}
```

When multiple columns are represented within a space-delimited line of column names, be sure to enclose them in braces ({}). If a column name contains white space, enclose the column name in braces ({}) also.

Command—`createentity` for Trillium Files

You can create a dynamic entity using Trillium files that are created as output by the TS Quality application.

Before you issue the `createentity` command, create a loader connection for the DB2 or Oracle data source. The loader connection specifies where the data source files are located and allows the `createentity` command to connect to the data source and initiate the entity creation process. (For information about creating a loader connection, see Chapter 4: Loader Connections on page 37).

Required Syntax

```
createentity <loader_connection> datafile <filename>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection.
<filename>	Name of the Trillium file that contains the data.

Optional Parameters

Parameter	Description
username <user_name>	User ID required to validate the connection to the data source. Use this parameter if a login name and password are required. Do not use the username and password parameters if the mtb_admin user is the data file owner.
password <password>	Password required to validate the connection to the data source. Use this parameter if a login name and password are required.
schemafile <filename>	Name of the schema file that corresponds to the Trillium file you specified as the data file.
jobname <job_name>	Job ID or name of the data load job.
skip <number>	Number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file (for example, the first 1000 records) to load.
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.

Example

This command uses a loader connection called trilconn and creates an entity by loading data from the testtril file.

```
createentity trilconn datafile testtril schemafile
testrill.ddt jobname trilliudyn1
```

Command—defineudpvdir

The defineudpvdir command defines the directories where executables used by a TS Quality User-Defined Process (UDP) can be stored.

As a security measure, Trillium Software System only allows users to run third-party executables or scripts that have been authorized by the repository administrator. This authorization is granted when you designate one or more server directories as virtual user-defined process directories. Only executables and scripts that are located in the designated directories may be run from within a User-Defined Process.

Syntax

```
defineudpvdir <name> <path>
```

where <name> is the name you want to assign to the virtual UDP directory and <path> is the directory that contains the executables and scripts you are authorizing. If the path name includes spaces, enclose it in braces.

Examples

```
defineudpvdir UDP C:/Executables
defineudpvdir app {C:/Program Files}
```

Variations

When issued without arguments, the defineudpvdir command returns a list of existing UDP virtual directories.

When issued with only the *name* argument, the defineudpvdir command returns the path of the named virtual directory.

You can edit the path used by a existing virtual directory by issuing the command with a new path name. For example:

```
defineudpvdir UDP
```

returns C:/Executables, the current path for the UDP directory.

```
defineudpvdir UDP {C:/Programs/UDP}
```

replaces the original path (C:/Executables) with the new path {C:/Programs/UDP}.

To delete an existing udp virtual directory, issue the defineudpvdir command with an empty path.

```
defineudpvdir UDP {}
```

Command—definesrtvdir

The definesrtvdir command defines the directories to be used for the temporary work files that are created while a Sort process runs. These files can be very large and so there is often a requirement for them to use a location other than the default. The work files are deleted when the Sort process completes.

ⓘ *If you do not define a virtual sort directory, the default is TS_WORK. If you do define a virtual sort directory, it becomes the default.*

Syntax

```
definesrtvdir <name> <path>
```

where <name> is the name you want to assign to the virtual sort directory and <path> is the directory where the temporary work files will be stored. If the path name includes spaces, enclose it in braces.

Example

```
definesrtvdir sortwrk C:/Temp
definesrtvdir SortTmp {C:/Work Files/Sort}
```

Variations

When issued without arguments, the definesrtvdir command returns a list of existing sort virtual directories.

When issued with only the *name* argument, the definesrtvdir command returns the path of the named virtual directory.

You can edit the path used by a existing virtual directory by issuing the command with a new path name. For example:

```
definesrtvdir sortwrk
```

returns C:/Temp, the current path for the Sort work files directory.

```
definesrtvdir sortwrk {C:/Work Files/sort}
```

replaces the original path (C:/Temp) with the new path {C:/Work Files/sort}.

To delete an existing virtual directory, issue the definesrtvdir command with an empty path.

```
definesrtvdir sortwrk {}
```

Command—delete entity ENTITY_ID

The delete entity ENTITY_ID command deletes the specified entity.

Syntax

```
delete entity ENTITY_ID <entity_id>
```

where *<entity_id>* is the number that identifies the entity you want to delete.

ⓘ ***ENTITY_ID*** *must appear in uppercase, as shown.*

Example

This command deletes the entity identified as "12."

```
delete entity ENTITY_ID 12
```

An error message results if the entity you want to delete does not exist.

Command–delete userscripts

You use the `delete userscripts` command to delete a user-defined script from the repository table that is displayed in the Control Center. This is also a useful procedure when you are creating and debugging scripts.

Once the script is deleted, it is no longer displayed in the **Execute Server Action** list in the Control Center.

To delete a userscript, you must know its ID number, which was returned when you ran the `add userscripts` command. You can also use the `show userscripts` command to get the ID of the script you want to delete.

ⓘ *The command must be typed on a single line. The delete userscripts command must be run from the _control repository.*

Syntax

```
delete userscripts SEQNO <sequence number>
```

where

SEQNO <sequence number>	Sequence number or ID of the user-defined script that is to be deleted. This is the ID that was originally returned when the add userscripts command was executed.

Example

The following example deletes a user-defined script with ID 1.

```
delete userscripts SEQNO 1
```

Command—deletetxnlogfiles

This command deletes *internal* transaction log files created for the TSS database software. The transaction log files hold details of a transaction while it is in progress; once the transaction is complete, the data in the log is simply historical. The most current log file is essential to the operation of TSS. The historical log files are not needed and should be deleted periodically to free up disk space.

ⓘ *The internal transaction log files are stored in the <repository_path>/MBSW/<version>/Data/metabase/<repository_name>/_txnlogs directory.*

Syntax

```
deletetxnlogfiles
```

This command deletes all the historical log files from the *specified* repository.

Command—`deletealltxnlogfiles`

This command also deletes *internal* transaction log files created for the TSS database software. It operates in the same way as the `deletetxnlogfiles` command, except it deletes historical transaction files from all repositories in the repository server.

Syntax

```
deletealltxnlogfiles
```

This command deletes all the *historical* log files from all repositories in the current repository server.

Command–`edit userscripts`

This command is used to modify a user-defined script that is registered with the Control Center. In order to use this command, you must know the ID of the script that you want to modify.

If you type `edit userscripts` without any parameter in mtb_admin you will initially be prompted for the sequence number (SEQNO). This is the number you get back when you firstly create a userscript (i.e. add userscripts). You can also see this number using `show userscripts` command.

The SEQNO parameter is followed by all the other parameters. To change a parameter (e.g. LOCATION), type the new value when prompted and hit Enter. To keep a parameter unchanged, just press Enter at the prompt.

When you are done, you can review the modifications using `show userscripts`.

ⓘ *The command must be typed on a single line. The parameters vary, depending on the type of object you are adding.The edit userscripts command must be run from the _control repository.*

Syntax

```
edit userscripts SEQNO <seqno> NAME "<display name>"
DESCRIPTION "<description of script>" LOCATION "<full path>"
```

where

SEQNO <seqno>	The ID of the script that needs to be modified. This is the ID that was returned when `add userscripts` was executed.

NAME "<display name>"	Name of the script as you want it to appear in the Execute Server Action submenu in the Control Center.
DESCRIPTION "<description>"	Description of the user-defined script.
LOCATION "<full path>"	Location of the script, including filename. Use forward slashes to define the path.

Optional Parameters

Parameter	Description
SECURITY <1|0>	Type 1 to hide the script from limited users; type 0 to display the script to all users. (The default is 0.)
FIXEDARGS <arg1 arg2>	Arguments that are always passed to the script. You can use a literal value as an argument or one of the following: $METABASE passes the name of the active repository $USERNAME passes the name of the logged on user $DATADIR passes the name of the repository data directory $BINDIR Passes the name of the binaries directory
USERARGS <n>	The number of arguments that need to be passed to the script by the user. The Control Center prompts only once for these arguments, and the user must respond with all the values, separated by a space. An individual argument cannot contain a space.

ⓘ *The edit userscripts command must be run from the _control repository. If the edit userscripts command contains both FIXEDARGS and USERARGS, the FIXEDARGS are passed to the script first.*

Command—`expert`

The `expert` extends the authority of the repository administrator when issuing commands from the command line. It is required to execute some commands, such as the define commands, and is recommended for use with all commands.

To use this command, enter it immediately after you log in to the Repository Administration Command Prompt. The command remains in effect until you exit the session.

Syntax

```
expert
```

Command—`export`

The `export` command exports information from the repository to a virtual directory named export on the TSS server.

ⓘ *The Control Center supports a local export (to the client machine), and a server export. The command line only supports a server export to a virtual directory named export.*

Syntax

```
export <object_type> entity <entity_id> filename
</export/<filename> filetype <filetype>
```

where

<object_type>	Indicates the type of List View information you want to export. The options are: `rows`—exports rows in a List View `rule`—exports all business rules associated with a specific entity `rules`—exports all business rules in the repository `passing_rows`—exports rows that pass the conditions or restrictions of the specified business rule `failing_rows`—exports rows that fail the conditions or restrictions of the specified business rule `list_entities`—exports details for all entities `list_attributes`—exports details for all attributes in a specified entity `list_dependencies_per_attribute`—exports details about dependencies for a specified attribute and entity `list_contradictions`—exports dependency conflict details for a specified entity The syntax for the export command is determined by which of these options you select.
<entity_id>	Number that identifies the entity you want to export.
</export/filename>	Name of the virtual directory and file to which the data will be exported. (File extensions are implied by the file type and do not need to be specified.) The virtual directory must be named export and it must be explicitly defined in the command.

<filetype>	Indicates the type of file to which the data will be exported. The options are: xml csv txt ddl ddx In txt and csv output files, the quote (&quot;) are used around data values that contain the delimiter. Delimiter for output files: txt - tab csv - comma *The Control Center supports HTML file as an output file type when you use the local export function. That function is not available from the command line.*

Optional Parameters

Parameter	Description
attribute <id>	The name and ID number of the attribute you want to export. Use when exporting failing or passing rows for an attribute business rule.
br_name <busrule_name>	Name of the business rule you want to use to filter passing or failing rows before export. Use this parameter when the export object type is either `passing_rows` or `failing_rows`.
canon	Specifies the rows to be exported.
derived_from <parent>	If you are exporting rows from an inherited/ derived rule, you must specify the name of the parent rule within the library. Any changes to the inherited/derived (child) rule as a result of running the export command will be propagated to the parent rule. For example, see Export All Rows that Pass a Library Business Rule.
encoding <encoding_name>	Character encoding used by the data file.
keypattern	Specifies the entity number.
output <colname colname>	List of attribute columns, separated by whitespace, to export. If the `output` parameter is not specified, all columns in the List View are exported. To add additional columns, such as a business rule XML expression, specify all required fields with this parameter.

parent_eid <library entity ID>	Number that identifies the ID of the parent of the entity you want to export.

Examples

These command line examples show the different types of information you can export by specifying the repository object you want to export. You can export rows or attributes, business rules, and passing or failing rows associated with a business rule. See specific command line examples below.

Export All Entity Rows to an XML file

```
export rows entity 11 filename /export/usdatarows filetype
xml encoding utf-8
```

Export All Entity Rows to a CSV file

```
export rows entity 11 filename /export/usdatarows filetype
csv encoding utf-8
```

Export Surname and First Name Attributes in an Entity

```
export rows entity 33 filename /export/two filetype xml
encoding utf-8 output {surname {first name}}
```

ⓘ *When multiple column names are represented within a space-delimited line, be sure to enclose them in braces ({}). If a column name contains white space, enclose the column name in braces ({}) also.*

Export All Business Rules for a Repository

```
export rules filename /export/allrules filetype xml encoding
utf-8
```

ⓘ *The repository from which the business rules are exported is the repository you logged on to when you responded to the repository: prompt.*

Export All Business Rules for an Entity

```
export rule entity 6 filename /export/ent6rules filetype xml
encoding utf-8
```

Export All Rows that Fail an Entity Business Rule

```
export failing_rows entity 22 br_name my_entity_businessrule
filename /export/ent22failrows filetype xml
```

Export All Rows that Pass an Attribute Business Rule

```
export passing_rows entity 51 attribute 12 filename /export/
attribute12passrows filetype xml br_name ABR_boston_sales
```

Export All Rows that Pass a Library Business Rule

```
export passing_rows entity 22 br_name my_businessrule
filename /export/ent22passrows filetype xml derived_from
parent_eid 10
```

Export Details for All Entities in the Repository

```
export list_entities filename /export/ents filetype xml
```

Export Details for All Attributes in an Entity

```
export list_attributes params {-keypattern 21} filename /
export/exportedfile filetype xml
```

ⓘ *In this example, the keypattern parameter specifies the entity number*

Export the Attribute Metadata For a Given Attribute (8) and Entity (11)

```
export list_attributes_pivoted params {-keypattern {11 8}}
filename /export/attribute filetype xml
```

Export Dependency Data

```
export list_dependencies_per_attribute params [list -entity
$eID -attribute $attrID -in {STATUS {Discovered}}] filename
/export/Dependencies filetype csv
```

where, $eID and $attrID are entity and attribute IDs respectively.

Export Dependency Conflict Data

For a single left-hand side attribute:

```
export list_contradictions params [list -entity $EID -
keypattern [list $rhsAID $lhsAID] -canon [list RH_ATTR
$rhsAID LH_ATTRS $lhsAID]] filename /export/Dependencies_
Conflicts filetype csv
```

For multiple left-hand side attributes:

```
export list_contradictions params [list -entity $EID -
keypattern [list $rhsAID [list $lhsAID1 $lhsAID2]] -canon
[list RH_ATTR $rhsAID LH_ATTRS [list $lhsAID1 $lhsAID2]]]
filename /export/Dependencies_Conflicts filetype csv
```

where, $EID is the entity ID, $rhsAID is the right-hand side attribute ID, and $lhsAID is left-hand side attribute ID.

Export Failing Rows Using ID of Parent Entity

```
export failing_rows entity 1 br_name test filename /export/
testfile filetype txt parent_eid 2
```

where parent entity id is 2

Command—fixRules

The fixRules command updates every rule in the current repository and displays the Rule Ids whose expressions have been changed. As a result of running this command, every rule in the repository becomes out-of-date and needs to be re-analyzed.

Syntax

```
fixRules [-verbose|-quiet]
```

where

-verbose	With this option, the command displays on the console every rule that it checks, allowing you to monitor progress.
-quiet	With this option, nothing is displayed on the console.

By default, only the modified Rule Ids are displayed.

Command—job

The job command displays the status of a specific running or completed job. This information is similar to the details shown in the Background Tasks list in the Control Center.

The command requires that you know the job identifier for the job you want to examine. A job identifier gets created when you issue a command that requires the server to run a job in order to complete the request.

Syntax

```
job <job_id>
```

where *<job_id>* is the job identifier.

Command—jobs

The jobs command displays the status of all running or completed jobs. This information is similar to the details shown in the Background Tasks list in the Control Center.

Syntax

```
jobs
```

Command—renamejob

The renamejob command allows you to edit the default job name and assign a custom name before scheduling the job. Renaming is useful when you are running multiple jobs concurrently and need a way to track the different jobs.

Syntax

```
renamejob <jobid> <new job name>
```

where

<jobid>	Is the job identifier.
<new job name>	Custom name to be assigned to the job.

Command—loaddata

The loaddata command creates an entity by importing data from a data source and storing it in a repository.

ⓘ *If you want to create a dynamic entity, you must use the createentity command.*

The required syntax for the loaddata command depends on the data source type you choose for the entity. The command syntax is different if you are importing data from a Trillium file as compared to importing data that uses a COBOL copybook file. The command returns the job identifier for the loader job.

The command supports five data source types, which are listed in the following table.

Data Source Type	Description
Delimited data file	Data stored in a delimited flat file database that separates records by rows, and fields by fixed length with padding or delimited by whitespace, tabs, commas (CSV) or other characters; for example, a basic name and address list in spreadsheet format. Supports ASCII, extended ASCII or hexadecimal delimiters. Delimited files with multiple global character encodings are not supported.
COBOL copybook	Data stored in flat, fixed length files that have a companion COBOL copybook (schema) file. Supports multiple character encoding, including EBCDIC, ASCII, and Unicode; Big Endian and Little Endian byte orders, and one or two byte data alignment.
ODBC-compliant relational database source	Databases supported using an ODBC-compliant RDBMS connection. See Relational Data Sources on page 38 for a list of supported sources. Data imports do not support analysis of duplicate row checksum calculation, sparse row counting, or row length statistics. NATIONAL data type is not supported.
Direct connection to Relational Database Management System (RDBMS)	Direct connections to IBM DB2 and ORACLE relational database systems. See Relational Data Sources on page 38 for a list of supported sources.

Trillium file	Output files created by the TS Quality application. Data imports do not support analysis of duplicate row checksum calculation, sparse row counting, or row length statistics.

ⓘ *You can extract data from a Relational Database Management System (RDBMS) to a delimited file with a corresponding ANSI DDL schema file. If you choose this method, you perform the extraction first and use the `loaddata` command syntax for delimited data sources.*

Command—`loaddata` for COBOL Files

You can create an entity using a COBOL flat file that has a companion COBOL copybook (schema).

Before you issue the `loaddata` command, create a loader connection for your COBOL data source. The loader connection specifies where the data source files are located and allows the `loaddata` command to connect to the data source and initiate the data import process.

Copybook and data file directories do not need to be specified when they are referenced within the loader connection you set up.

Required Syntax

```
loaddata <loader_connection> datafile <filename> schemafile
<copybook_filename>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection created for the COBOL data source.
<filename>	Name of the COBOL flat file that contains the data.
<copybook_filename>	Name of the schema file that corresponds to the COBOL flat file you specified as the data file.

Optional Parameters

Parameter	Description
username <user_name>	User ID required to validate the connection to the data source. Use this parameter only if a login name and password are required. Do not use the username and password parameters if the mtb_admin user is the data file owner.
password <password>	Password required to validate the connection to the data source. Use this parameter only if a login name and password are required.
jobname <job_name>	Job ID or name of the data load job.

charset <charset>	Character encoding used by the COBOL data and copybook files. The options are: EBCDIC or ASCII.
endian <value>	Indicates the byte order. The options are big (for Big Endian) and little (for Little Endian).
upacked <value>	Indicates the structure of the data. Set to 1 if you want to treat unsigned comp-3 fields as comp-6.
align <value>	Indicates one- or two-byte data alignment. The options are one and two.
redefines <value>	Indicates how to manage REDEFINES clauses in a copybook. The options are all and first. Type all to account for all REDEFINES clauses in a copybook. If this option is selected and the system encounters a REDEFINES clause, it removes the REDEFINES clause and keeps both representations of the data in the copybook. The data file will be populated to match the copybook. Type first to ignore all REDEFINES clauses in the copybook.
terminator <value>	Indicates how records in the data file are terminated. The options are lf (linefeed), cr (carriage return), crlf (linefeed and carriage return) and none. Typically, COBOL data files are fixed length and have no record delimiter. However, if COBOL data is exported from the original application and transferred into other file systems (especially UNIX), they could contain record delimiters added by the export or transfer process. If the file originated from: Windows, type crlf UNIX, type lf IBM mainframe, type none.
encoding <value>	Indicates the character encoding used by the data file. The options are EBCDIC and ASCII. EBCDIC data is translated into a correct ASCII representation on load. Generally, UNIX COBOL files will be ASCII and IBM mainframe data will be EBCDIC.
national <value>	National Character encoding used by the data file. If your COBOL copybooks define national data items holding Unicode strings (such as PIC clause containing N and USAGE NATIONAL) or your compiler options from the data source are set to NSYMBOL (NATIONAL) or CODEPAGE, you should specify the National Character encoding standard of the data source.
columns <names>	Indicates the names of the columns from which to import data.

skip <number>	The number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file (to load (for example, the first 1000 records).
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.

Example

The command uses a loader connection named `cobolconn`, provides a username and password to access the data directories, and then loads data from three columns in the `testcp932.dat` data file.

```
loaddata cobolconn username myuser password mypsd encoding
cp932 datafile testcp932.dat schemafile testcp932.cpy
columns {{Ref Id} Source Amount}
```

When multiple columns are represented within a space-delimited line of column names, be sure to enclose them in braces ({}). If a column name contains whitespace, enclose the column name in braces ({}) also.

Command—`loaddata` for Delimited Files

You can create an entity using a delimited file that may or may not have a companion schema Data Dictionary Language (DDL) file.

Before you issue the `loaddata` command, create a loader connection for your "Delimited" data source. The loader connection specifies where the data source files are located and allows the `loaddata` command to connect to the data source and initiate the data import process.

Required Syntax

```
loaddata <loader_connection> datafile <filename>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection.
<filename>	Name of the delimited file that contains the data.

Optional Parameters

Parameter	Description
username <user_name>	User ID required to validate the connection to the data source. Use this parameter only if a login name and password are required. Do not use the username and password parameters if the mtb_admin user is the data file owner.
password <password>	Password required to validate the connection to the data source. Use this parameter only if a login name and password are required.
schemafile <filename>	Name of the schema file that corresponds to the delimited file you specified as the data file.
jobname <job_name>	Job ID or name of the data load job.
attr <header>	Indicates the header line. The options are: `names`—names on first line `none`—no attribute names specified `ddl`—names in schema file
delimiter <character>	Indicates the character that is used as the data delimiter in the file. A field can be delimited by whitespace, tabs, commas (CSV), periods (.) or other characters.
terminator <value>	Indicates how records in the data file are terminated. The options are `lf` (linefeed), `cr` (carriage return), and `crlf` (linefeed and carriage return). Typically, if the file resides on a Windows system, type `crlf`. If on a UNIX system, type `lf.`
encoding <name>	Character encoding used by the data file. The options are `EDCDIC` and `ASCII`. This parameter controls the character set for the file. EBCDIC data is translated into a correct ASCII representation on load. Generally, UNIX COBOL files will be ASCII and IBM mainframe data will be EBCDIC.
columns <names>	Indicates the names of the columns from which to import data.
skip <number>	Number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file (for example, the first 1000 records) to load.
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.

Example

This command loads data from three columns in the testdel.txt file and uses the delimconn Loader Connector to connect to the file.

```
loaddata delimconn datafile testdel.txt attr names delimiter
. quote \" terminator crlf columns {{Ref Id} Source Amount}
```

When multiple columns are represented within a space-delimited line of column names, be sure to enclose them in braces ({}). If a column name contains whitespace, enclose the column name in braces ({}) also.

Command—loaddata for ODBC-Compliant Databases

You can create an entity by importing data from an ODBC-compliant relational database.

Before you issue the loaddata command, create a loader connection for your data source. The loader connection specifies where the data source files are located and allows the loaddata command to connect to the data source and initiate the data import process.

Required Syntax

```
loaddata <loader_connection> username <username> password
<password> table <table_name>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection.
<username>	User ID required to validate the connection to the data source. You must always provide the username and password parameters when importing data from a relational database that uses an ODBC connection.
<password>	Password required to validate the connection to the data source.
<table_name>	Name of the relational database table to be imported.

Optional Parameters

Parameter	Description
jobname <job_name>	Job ID or name of the data load job.
columns <names>	Indicates the names of the columns from which to import data.
skip <number>	Number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file to load (for example, the first 1000 records).
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.

Example

This command connects to an ODBC-compliant database and creates an entity for three columns in the `footballteams` table.

```
loaddata odbcconn username odbcuser password odbcpwd table
footballteams columns {team player score}
```

Command—`loaddata` for RDBMS Connections

You can connect directly to an IBM DB2 or ORACLE relational database management system (RDBMS) and create an entity by importing data from database tables stored in the system.

Before you issue the `loaddata` command, create a loader connection for your DB2 or ORACLE data source. The loader connection specifies where the data source files are located and allows the `loaddata` command to connect to the data source and initiate the data import process.

Syntax

```
loaddata <loader_connection> username <username> password
<password> table <table_name>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection.
<username>	User ID required to validate the connection to the data source. You must always provide the username and password parameters when connecting directly to a relational database management system (RDBMS).

<password>	Password required to validate the connection to the data source. Use this parameter if a login name and password are required.
<table_name>	Name of the relational database table to be imported.

Optional Parameters

Parameter	Description
jobname <job_name>	Job ID or name of the data load job.
columns {names}	Indicates the names of the columns from which to import data.
skip <number>	Number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file to load (for example, the first 1000 records).
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.

Example

This command loads data over a special loader connection named `db2conn` which has been set up for an IBM DB2 database. There are three table columns that will be loaded: team blue, player, and score.

```
loaddata db2conn username db2user password db2pwd jobname
db2load1 table championship columns {{team blue} player
score}
```

When multiple columns are represented within a space-delimited line of column names, be sure to enclose them in braces ({}). If a column name contains whitespace, enclose the column name in braces ({}) also.

Command—`loaddata` for Trillium Files

You can create an entity using Trillium files that are created as output by the TS Quality application.

Before you issue the `loaddata` command, create a loader connection for your "Trillium" data source. The loader connection specifies where the data source files are located and allows the `loaddata` command to connect to the data source and initiate the data import process.

Syntax

```
loaddata <loader_connection> datafile <filename>
```

where

<loader_connection>	Name assigned by the repository administrator to the loader connection.
<filename>	Name of the Trillium file that contains the data.

Optional Parameters

Parameter	Description
username <user_name>	User ID required to validate the connection to the data source. Use this parameter if a login name and password are required. Do not use the username and password parameters if the mtb_admin user is the data file owner.
password <password>	Password required to validate the connection to the data source. Use this parameter if a login name and password are required.
schemafile <filename>	Name of the schema file that corresponds to the delimited file you specified as the data file.
jobname <job_name>	Job ID or name of the data load job.
skip <number>	Number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file (for example, the first 1000 records) to load.
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.

Example

This command uses a loader connection called `qualconn` and creates an entity by loading data from the `testqual` file.

```
loaddata qualconn datafile testtril schemafile testqual.ddt
jobname qualityload1
```

Command—loadentity

The `loadentity` command creates a real entity by loading data to a copied entity. The entity is first created using the `copyentity` command. The examples below show how to use the `copyentity` and `loadentity` commands together.

The command requires that you know the entity identifier for the entity into which you want to load the data. TSS assigns entity ids when it creates the entity.

When the command completes, it returns a job identifier for the load job.

Syntax

```
loadentity <entity_id>
```

where *<entity_id>* is the number that identifies the entity you want to copy.

Optional Parameters

Parameter	Description
datasource <name>	Name of the data source file or table from which the data will be loaded. Use quotes around the file or table name if the name includes spaces.
skip <number>	Number of rows to skip before starting to import data rows. All rows after the skipped rows will be loaded to the repository. For example, if your file has 300 rows and you select to skip the first 99, the system will load 200 rows, starting with the 100th row.
first <number>	Number of records from the beginning of the file (for example, the first 1000 records) to load.
random <percentage>	The degree to which you want to randomly sample a percentage of records from the file.
passrule <business_rule_ name>	Name of the business rule which defines which passing rows will be used as the row filter during the load process.
failrule <business_rule_ name>	Name of the business rule that defines the failing rows that will be used to filter rows during the load process.

Example

This command sequence creates a variable for the `copyentity` operation that copies entity 15 to a new entity. The `loadentity` command loads the first 10 rows of data in the “sequence two.txt” file and creates it as an entity. It also returns the job identifier for the load job.

ⓘ *It is important that the data source you specify matches the schema of the original source entity specified in the `copyentity` command.*

```
set entid [copyentity 15]
45
loadentity $entid datasource "sequence two.txt" first 10
Successfully started job 394
```

This sequence creates a variable for `copyentity` that copies entity 15 to a new entity. The `loadentity` loads only rows that pass the “highseqrule” business rule into the entity. It returns the job ID for the load job.

```
set entid [copyentity 15]
46
loadentity $entid passrule highseqrule
Successfully started job 395
```

Command—newuser

The newuser command creates a new Trillium-authenticated user and gives the user access to the repositories you specify. Make a note of the user names you are creating, along with the passwords and repository to which each user has access.

Syntax

```
newuser name <username> password <password> metabases
{<repository1 repository2>}
```

where

name <username>	Logon ID for the new user.
password <password>	Password you are assigning to the new user.
metabases {<repository1 repository2>}	Repositories to which this user will have access. Separate multiple repositories with a space and enclose them in curly braces, as shown. If you want to give the new user access to all repositories, use an * as the argument.

Examples

```
newuser name joey password abc123 metabases {demo accounts}
newuser name joey password abc123 metabases *
```

Command—purge

You use the purge command to remove the unwanted indexes. TSS uses indexes for many purposes. For example:

- List views create a temporary index when displayed.
- The join engine uses indexes extensively.
- av2t indexes are used to speed-up drill downs to rows.

Most of these indexes result in files with an extension of .ndx and .nta. The indexes are automatically deleted when the task with which they are associated is complete and:

- When a user logs out, all of the temporary indexes are deleted.
- Join indexes are deleted when the joins job is complete.
- The av2t indexes generally remain for as long as the entity remains.

However, there are times when an index is left behind (for example, if you kill a process before it completes).

Command—purgeentities

The purgeentities command deletes the av2t indexes for a specific entity or list of entities. It updates the catalog and removes the index files.

Syntax

```
purgeentities {<entity_id> <entity_id> . . .}
```

where *<entity_id>* is the number that identifies the entity you want to copy.

Examples

The following example removes the av2t indexes for entity 13 from the catalog and deletes the related index files.

```
purgeentities 13
```

The following example removes the av2t indexes for entities 4 and 5 from the catalog and deletes the related index files.

```
purgeentities {4 5}
```

Command—purgeall

The purgeall command deletes the av2t indexes for all entities in the current repository. It updates the catalog and removes the index files.

Syntax

```
purgeall
```

Command—purgetemporaries

The purgetemporaries command removes all the unused *temporary* indexes from the catalog and deletes the related files. It only removes the temporary indexes that should have been automatically deleted but were left behind for some reason (such as a system crash).

Syntax

```
purgetemporaries
```

Command—print

The print command displays a message at the mtb_admin command line prompt. You can use the command to print a response when a job completes

Syntax

```
print <message>
```

where *<message>* is one or more words that you want to display at the command prompt.

Example

This example shows how to print the words "Job Completed" at the command prompt after a `loaddata` job completes.

```
wait [loaddata delim_conn datafile usdata.txt attr names
delimiter . quote \" terminator crlf]
print "Job Completed"
```

Command—runrules

The `runrules` command enables you to run Attribute Business Rules or Entity Business Rules outside the Control Center. Command options are designed to give you maximum flexibility when you run a rules job. For example, you can run all rules from a specified entity, selected rules in an entity, or only rules that need to be analyzed.

The following terms are used in the `runrules` command syntax. Understanding their meaning will help you structure the command to run the appropriate rules.

Term	Definition
dirty	A *dirty* entity or attribute is one contains rules that are out-of-date; that is, rules that need to be analyzed or re-analyzed.
model	A *model* entity or attribute is one that is in the *library*. Non-library objects are called *real* objects.

ⓘ *Attribute Business Rules (ABRs) and Entity Business Rules (EBRs) have an ID number. By default, this number is hidden in the Control Center, but you can see it by displaying a List View of the rules and adding the column Sequence Number to the list.*

Syntax

EBRs: `runrules <eid> <qualifier>`

ABRs: `runrules <eid> -attribute <aid> <qualifier>`

where

<eid>	An entity ID or 0. When you specify 0, the command runs the rules for all the entities in the repository.
<aid>	An attribute ID.
<qualifier>	One of the following values:

-all	All rules for the specified entity or attribute.
-allentity	All rules associated with the specified real entity or attribute.
-allmodel	All rules associated with the specified library entity or attribute.
-dirty	Only rules for the specified entity or attribute that need to be analyzed.
-dirtyentity	Only rules in the specified entity that need to be analyzed.
-dirtymodel	Only rules that need to be analyzed and are associated with the specified library entity or attribute.
-selected {#, # }	Only the specified rules, identified by rule number.

Examples

The following command line examples demonstrate the different types of rules jobs you can run.

Run Entity Business Rules in a Real Entity

Run all the EBRs in entity 62.

```
runrules 62 -all
```

Run all the EBRs in entity 62 that need to be analyzed.

```
runrules 62 -dirty
```

Run the specified EBRs in entity 62.

```
runrules 62 -selected {16,18,24}
```

Run Attribute Business Rules in a Real Entity

Run all the ABRs for attribute 24 in entity 31.

```
runrules 31 -attribute 24 -all
```

Run all the ABRs that need to be analyzed for attribute 24 in entity 31.

```
runrules 31 -attribute 24 -dirty
```

Run the specified ABRs for attribute 24 in entity 31.

```
runrules 31 -attribute 24 -selected {1,2,4}
```

Run All Business Rules in a Real Entity

Run all the EBRs and ABRs for entity 31.

```
runrules 31 -allentity
```

Run all the EBRs and ABRs in entity 31 that need to be analyzed.

```
runrules 31 -dirtyentity
```

Run All Business Rules in the Repository

Run all the EBRs and ABRs for all entities in the repository.

```
runrules 0 -allentity
```

Run all the EBRs and ABRs for all entities in the repository that need to be analyzed.

```
runrules 0 -dirtyentity
```

Run Entity Business Rules Associated with a Library Entity

Run all the EBRs associated with library entity 43.

```
runrules 43 -allmodel
```

Run only the EBRs associated with library entity 43 that need to be analyzed.

```
runrules 43 -dirtymodel
```

Run Attribute Business Rules Associated with a Library Attribute

Run all the ABRs in the repository that are associated with library attribute 54.

```
runrules 0 -attribute 54 -allmodel
```

Run only the ABRs in the repository that are associated with library attribute 54 and need to be analyzed.

```
runrules 0 -attribute 54 -dirtymodel
```

Command—set

The set command assigns a variable. When you use a variable created by the set command, the variable must be preceded by a $(dollar sign) notation.

Syntax

```
set <variable> [<mtb_admin_command>]
```

where <variable> is the variable name and <mtb_admin_command**>** indicates the mtb_admin operation that the variable will reference.

Examples

This command assigns a variable to a loaddata operation.

```
set thisjob [loaddata delim_conn datafile usdata.txt attr
names delimiter . quote \" terminator crlf]
```

The variable can then be used to obtain details about the operation using this command:

```
job $thisjob
```

This example assigns a variable to a createentity operation.

```
set myentity [createentity delim_conn datafile usdata.txt
attr reps delimiter . quote \"]
```

Command–`show userscripts`

Example (showing a userscript)

You can show all registered userscripts. This is useful when you want to check what scripts are defined.

You use the `show userscripts` command to display all user-defined scripts that are registered in the Control Center. This is very useful find the SEQNO value for a script that you want to edit or delete.

ⓘ *The command must be typed on a single line. The show userscripts command must be run from the _control repository.*

Syntax

```
show userscripts
```

Example

The following example shows all user-defined scripts currently registered with the Control Center.

```
show userscripts
SEQNO NAME      DESCRIPTION LOCATION          SECURITY
FIXEDARGS USERARGS
-------------------------------------------------------------
-------------
1     myscript1 A script    c:/temp/test.bat 1
$METABASE 2
```

Command—`tscreate`

The `tscreate` command creates a new Time Series project. It returns the ID of the new project.

ⓘ *This command does not schedule automatic creation of new iterations. The command assumes that the automatic interval will be defined in the Control Center or that a new iteration will be initiated manually, either from the Control Center or from the `tsgeneration` command.*

Syntax

```
tscreate <entity_id> name <name>
```

where

<entity_id>	Number that identifies the entity you want to use to create the project.
name <name>	Name that identifies the new project.

Optional Parameters

Parameter	Description
description <desc>	Description of the new project.
datasource <name>	Name of the data source file or table from which the data will be loaded. Use quotes around the file or table name if the name includes spaces.

Example

This example creates a new Time Series project named Tahoe from entity 28.

```
tscreate 28 name Tahoe description “tourist registration”
```

Command—tsgeneration

The tsgeneration command generates a new entity for the specified Time Series project.

Syntax

```
tsgeneration project_num datasource <name>
```

where

project_num	Description of the new project.
datasource <name>	Name of the data source file or table from which the data will be loaded for this entity. Use quotes around the file or table name if the name includes spaces.

Example

In the following example, the command generates a new entity for project ID 3, using the data in the source file 2009 Sales.

```
tsgeneration 3 datasource "2009_Sales.txt"
```

Command—wait

The wait command causes all command activity to wait until a particular job completes. It effectively blocks the command line until the job is finished. For jobs that require a loaddata operation, the command line waits only until the "Load Data" activity is finished. Subsequent analysis jobs are ignored.

Syntax

```
wait <job>
```

where *<job>* can be either a command that specifies a job or a variable that references the command.

Example

This example shows how to specify a loaddata command as part of the wait command.

```
wait [loaddata delim_conn datafile usdata.txt attr names
delimiter . quote \" terminator crlf]
```

You can create a variable for the loaddata job and then use the variable in the wait command. This blocks the command line until the loaddata job completes.

```
set thisjob [loaddata delim_conn datafile usdata.txt attr
names delimiter . quote \" terminator crlf]
wait $thisjob
```

Command—waitall

The waitall command causes all command activity to wait until a particular job completes. It effectively blocks the command line until the job and its spawns are finished.

For jobs that require a loaddata operation, the command line waits until the "Load Data" activity and the subsequent analysis jobs are completed.

Syntax

```
waitall <job>
```

where <job> can be either a command that specifies a job or a variable that references the command.

The examples shown for the wait command are also valid for the waitall command.

CHAPTER 10

Troubleshooting

This chapter discusses potential problems you may encounter and the steps to resolve them. It includes:

- User Permissions on page 140
- Control Center Sessions File Error on page 141
- Delay Reading Data -TSS SQL Server Driver on page 141

User Permissions

Problem:

A TSS user is having trouble performing a task or accessing a repository object.

Solution:

Verify security settings by logging in to the Control Center and checking permission settings. If necessary, switch to the repository in which the user is working.

▶ To check on a user's security settings at an object level

1. Locate the object (repository, project, entity, business rule, or objects in the Business Rules Library) that is in question.
2. Right-click and select **Security**.
3. Review the security settings for the selected object and look for conflicting security settings. For example, you may have given the user permission to edit a project at the server level but denied that permission at the repository level. In this case, TSS applies the most restrictive permission.
4. In the Repository Manager application, make any required adjustments.

Control Center Sessions File Error

Problem:

A TSS user opens the Control Center after saving a repository connection and a message opens alerting the user that the saved session file cannot be found.

Solution:

The message displays when you attempt to open the Control Center after you save a session file then remove or rename the file. Use this procedure to bypass the error, open the Control Center, and save another session.

▶ To bypass the Cannot find session file message

1. Click **OK** to close the Cannot find this file message.
2. Open the environment options window.
3. Clear the option **At startup, load last used repository server connection**.
4. Click **Apply** and **OK**.
5. Close and reopen the Control Center and use your credentials to log in to the repository.
6. To save another session file:
   a. From the Control Center, click the TS Application button in the upper left of the interface and select **Save Session as**. The Save As window opens.

   b. Browse to the directory in which you want to save the session and enter the name you want to assign. (Sessions use the file extension .tss.)

   c. Click **Save**.

   d. Select the environment option **At startup, load last used repository server connection**. The next time you start the Control Center, you do not have to select a connection.

Delay Reading Data -TSS SQL Server Driver

Problem:

Using the TSS 15 SQL Server Driver, you attempt to read the SQL Server database from the Control Center by clicking the preview option in the Create Entity wizard. If you click preview multiple times, the Control Center hangs. In addition, activities that connect to a database (load data, preview) may take a long time to disconnect and complete even though all data appears to have been processed.

Solution:

Configure the TSS 15 SQL Server Driver and set the parameter EnableScrollableCursors to 3.

▶ To configure EnableScrollableCursors value

1. Open the ODBC Data Source Administrator window.
2. Open the DSN for the TSS 15 SQL Server Driver, then open the **Advanced** tab of the ODBC SQL Server Wire Protocol Driver Setup window.
3. In the Extended Options section, add `EnableScrollableCursors=3`.
4. Click **Apply** ad then **OK** to save your changes.

APPENDIX A

Pattern Styles

Trillium Software System uses patterns to standardize the format of data in the repository data. Patterns allow you to quickly identify deviations from the norm when you are analyzing data. This chapter includes:

Overview

When you create a new repository, one of the parameters you define is the data pattern. The options are:

- Default
- Rich
- Long
- Greek
- Hebrew
- Turkish

Default Pattern

The default pattern represents data in shorthand notation by counting the number of characters represented by a code type and displaying that count next to the code. Default Pattern Codes describes the codes used in the default pattern and Default Pattern Examples provides examples.

The codes and examples also apply to the Greek, Hebrew, and Turkish patterns.

Table A-1 Default Pattern Codes

Code	Represents
a	alpha character
d	digit
p	punctuation
_	space
z	null
.	unprintable
!	non-ASCII
c.	carriage return.

Table A-2 Default Pattern Examples

Value	Represents
Jane Smith	a4_a5
5.00E+02	dpd2apd2
$400.00	pd3pd2
05/31/2005	d2pd2pd4
jane_smith@abc.com	a4pa5pa3pa3

Rich Pattern

The rich pattern represents data in shorthand notation by counting the number of characters represented by the code and displaying that count next to the code. It is similar to the default pattern, but uses different codes. Rich Pattern Codes describes the codes used in the rich pattern and Rich Pattern Examples provides examples.

Table A-3 Rich Pattern Codes

Code	Represents
l	lowercase alpha
u	uppercase alpha
d	digit
p	punctuation
q	Apostrophe (’) Double quotes (“) Single quote (‘)
S	symbol

_	space
z	null
.	unprintable
m	currency
+	plus sign
-	minus sign or dash

Table A-4 Rich Pattern Examples

Value	Representation
Jane Smith	ul3_ul4
5.00E+02	dSd2u+d2
$400.00	md3Sd2
05/31/2005	d2Sd2Sd4
jane_smith@abc.com	l4Sl5pl3Sl3

Long Pattern

The long pattern represents in long-hand notation by explicitly displaying each character representation. Long Pattern Codes describes the codes used in the long pattern and Long Pattern Examples provides examples.

ⓘ *This is the default mask pattern used by the TS Quality component.*

Table A-5 Long Pattern Codes

Code	Represents
A	alpha
N	digit
explicit	any special characters (not alpha or numeric) are displayed as they appear in the value.

Table A-6 Long Pattern Examples

Value	Representation
Jane Smith	AAAA AAAAA
5.00E+02	N.NNA+NN
$400.00	$NNN.NN
05/31/2005	NN/NN/NNNN
jane_smith@abc.com	AAAA_AAAAA@AAA.AAA

Greek, Hebrew, and Turkish Patterns

These patterns use the same codes as the default pattern. See Default Pattern on page 143.

APPENDIX B

COBOL Copybooks

Trillium Software System can import or link to data from a variety of data sources, including RDBMS, flat files, delimited files, and Trillium files. This appendix focuses on flat files described by COBOL copybooks. It lists the COBOL copybook features that are supported and those that are not, and explains how TSS handles some of the COBOL clauses.

ⓘ *For information on other data source types and on creating and validating loader connections, refer to Chapter 4: Loader Connections on page 37.*

This appendix chapter includes:

- Supported Copybook Features on page 147
- Unsupported Copybook Features on page 153

Supported Copybook Features

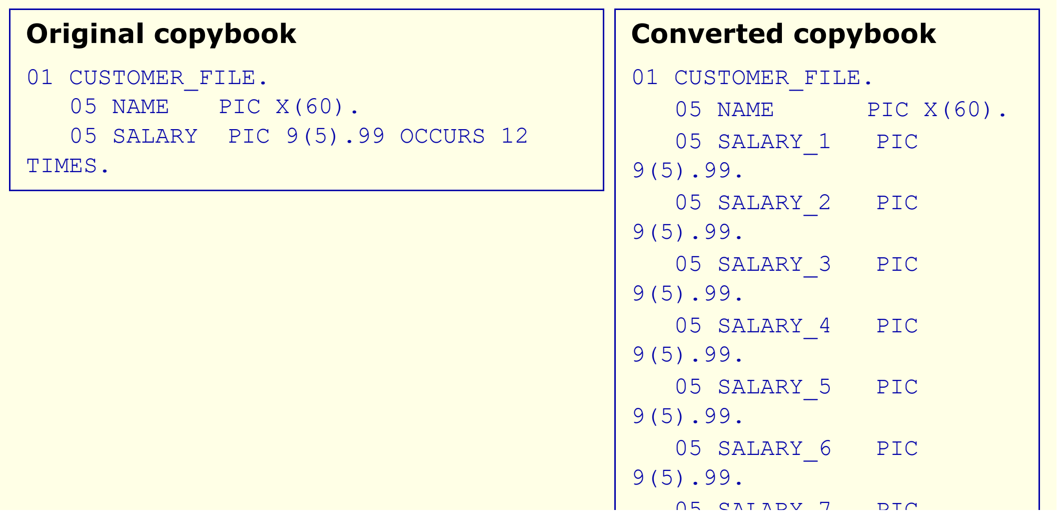
Trillium Software System supports the most common COBOL copybook features:

- REDEFINES clause
- OCCURS clause
- FILLER clause
- COBOL data types: 9, X, A, B
- Numeric storage formats: COMP-3, Packed-DECIMAL, COMP, COMPUTATIONAL, or BINARY
- Comments beginning with “*”
- Descriptors such as USAGE IS, DISPLAY, SEPARATE, LEADING, and TRAILING
- Unnamed fields
- Unsigned COMP-3 fields

How Are OCCURS Clauses Handled?

TSS automatically replaces OCCURS clauses in a copybook with one field per instance of the array, as shown in the following example.

Example - TSS converting an OCCURS clause



Original copybook

```
01 CUSTOMER_FILE.
   05 NAME    PIC X(60).
   05 SALARY  PIC 9(5).99 OCCURS 12
TIMES.
```

Converted copybook

```
01 CUSTOMER_FILE.
   05 NAME       PIC X(60).
   05 SALARY_1   PIC
9(5).99.
   05 SALARY_2   PIC
9(5).99.
   05 SALARY_3   PIC
9(5).99.
   05 SALARY_4   PIC
9(5).99.
   05 SALARY_5   PIC
9(5).99.
   05 SALARY_6   PIC
9(5).99.
   05 SALARY_7   PIC
```

How Are REDEFINES Clauses Handled?

When creating entities described by COBOL copybooks that contain REDEFINES clauses, you have two options:

- Ignore all REDEFINES clauses in your COBOL copybooks, or
- Represent all data redefinitions (including nested REDEFINES).

ⓘ *You do not have the option to "pick and choose" REDEFINES clauses during an import. If you have a copybook where only a few of the REDEFINES clauses are required, manually modify the copybook to remove the unnecessary REDEFINES clauses prior to importing the data into TSS. Make sure that you maintain the original record length if some REDEFINES blocks of the same fields are of different sizes.*

▶ To ignore all REDEFINES clauses in a copybook

In the Create Entity Wizard, select the option **Redefines: First**.

When TSS encounters a REDEFINES clause, it ignores the REDEFINES clause and retains only the **first** representation of the data in the copybook.

Example—Results of the Option Redefines First

Original copybook with a REDEFINES clause (BEFORE)

```
01   CUSTOMER_FILE.
     05 CUSTOMER_FULL_CODE  PIC X(7)
     05 SPLIT_CODE REDEFINES CUSTOMER_FULL_
CODE.
        10 LOCATION          PIC X(2).
        10 ID                PIC 9(5).
```

Converted copybook with REDEFINES clause ignored (AFTER)

```
01   CUSTOMER_FILE.
     05 CUSTOMER_FULL_CODE   PIC X(7).
```

▶ To represent all REDEFINES clauses in a copybook

In the Create Entity Wizard, select the option **Redefines: All**.

When TSS encounters a REDEFINES clause, it removes the REDEFINES clause and keeps both representations of the data in the copybook. The data file is then populated to match the copybook.

Example—Results of the Option Redefines All

Original copybook with a REDEFINES clause (BEFORE)

```
01    CUSTOMER_FILE.
      05 CUSTOMER_FULL_CODE        PIC X(7)
      05 SPLIT_CODE REDEFINES CUSTOMER_FULL_CODE.
         10 LOCATION               PIC X(2).
         10 ID                     PIC 9(5).
```

Converted copybook with REDEFINES clause removed (AFTER)

```
01    CUSTOMER_FILE.
      05 CUSTOMER_FULL_CODE PIC X(7).
      05 LOCATION           PIC X(2).
      05 ID                 PIC 9(5).
```

The corresponding data file is modified by TSS to match the copybook. In this example, TSS will add seven bytes of data to the file to account for the redefinition.

How Are Multiple Record Types Handled?

There are two possible methods for handling multiple record types. Either manually separate multiple record types into separate files, or filter for each record type during entity creation. Both methods assume that the data file has fixed record lengths and the COBOL copybook matches the data file.

Method 1—Manually Separate Multiple Record Types

▶ To manually separate multiple record types

1. Manually extract each record type into its own file.
2. Create a copybook to match each new data file.
3. Create Entities based on new data files and copybooks.

Example 1 - Remove Header and Trailer Information

Assume that you have one data file and one copybook that contains header, trailer, and account information.

Original Data File and copybook with Multiple Record Types

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D	J	O	H	N				S	M	I	T	H			
H	B	R	A	N	C	H	-	A	B	C					
D	J	A	N	E				N	O	R	M	A	N		
F	0	0	2												
D	S	I	M	O	N			P	A	U	L				

```
01 ACCOUNTS.

  03 REC-TYPE          PIC X.
  03 ACCT-DETAIL       PIC X(15).
  03 HEADER REDEFINES ACCT-DETAIL.
    05 SOURCE-INFO     PIC X(15).
  03 ACCT-RECORD REDEFINES ACCT-
  DETAIL.
    05 FIRST-NAME      PIC X(7).
    05 LAST-NAME       PIC X(8).
  03 TRAILER REDEFINES ACCT-DETAIL.
    05 COUNT           PIC 999.
    05 FILLER          PIC X(12).
```

If the header and trailer are not necessary for your data profiling efforts, remove the header and trailer from the data file and the copybook. Create 1 entity based on the new data file and copybook.

ACCOUNTS-DETAIL.CPY

```
01 ACCOUNTS-DETAIL.
   03 REC-TYPE   PIC X.
   03 FIRST-NAME PIC X(7).
```

ACCOUNTS-DETAIL.DAT

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
D	J	O	H	N				S	M	I	T	H		

Example 2 - Maintain Header and Trailer Information

Assume that you have one data file and one copybook that contains header, trailer, and account information.

Original Data File and copybook with Multiple Record Types

1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1
									0	1	2	3	4	5	6
D	J	O	H	N				S	M	I	T	H			
H	B	R	A	N	C	H	-	A	B	C					
D	J	A	N	E				N	O	R	M	A	N		
F	0	0	2												
D	S	I	M	O	N			P	A	U	L				

```
01 ACCOUNTS.

   03 REC-TYPE            PIC X.
   03 ACCT-DETAIL         PIC X(15).
   03 HEADER REDEFINES ACCT-DETAIL.
      05 SOURCE-INFO     PIC X(15).
   03 ACCT-RECORD REDEFINES ACCT-
   DETAIL.
      05 FIRST-NAME      PIC X(7).
      05 LAST-NAME       PIC X(8).
   03 TRAILER REDEFINES ACCT-DETAIL.
      05 COUNT           PIC 999.
      05 FILLER          PIC X(12).
```

In order for TSS to represent this data correctly, create 3 copybooks to represent each record type and 3 data files to match each copybook. You will then create 3 Entities based on these sets.

ACCOUNTS-HEADER.CPY

```
01 ACCOUNTS-HEADER.
   03 REC-TYPE     PIC X.
    03 SOURCE-INFO PIC
X(15).
```

ACCOUNTS-DETAIL.CPY

```
01 ACCOUNTS-DETAIL.
   03 REC-TYPE    PIC
X.
   03 FIRST-NAME PIC
X(7).
   03 LAST-NAME  PIC
X(8).
```

ACCOUNTS-TRAILER.CPY

```
01 ACCOUNTS-TRAILER.
   03 REC-TYPE PIC X.
   03 COUNT    PIC 999.
   03 FILLER   PIC
X(12).
```

ACCOUNTS-HEADER.DAT

```
1 2 3 4 5 6 7 8 9 1 1 1 1 1 1 1
                  0 1 2 3 4 5 6
```

ACCOUNTS-DETAIL.DAT

```
1 2 3 4 5 6 7 8 9 1 1 1 1 1 1
                  0 1 2 3 4 5
```

ACCOUNTS-TRAILER.DAT

```
1 2 3 4 5 6 7 8 9 1 1 1 1 1 1
                  0 1 2 3 4 5
```

Method 2—Filter Each Record Type During Entity Creation

ⓘ *Only apply this method if the following are true: Data file is delimited with line feeds, and Record types all have the same length.*

▶ To manage multiple record types with a filter

1. Make sure that there is one COBOL copybook for each record type that exists in the data file.
2. Once you have split your copybook to represent each of the record types, follow the steps for creating an entity. (This procedure is described in TSS online help.)

3. When asked to select a data file, choose the one that contains the multiple record types.

4. When asked to select a corresponding schema, choose one of the copybooks (that you created) to represent a record type.

5. When you get to the schema settings window, apply the schema settings and click **Preview**. The data displays in the pane below the wizard. From this pane, you can organize how the data should be represented in the new entity.

6. From the preview pane below the wizard, right-click and select **Filter**.

7. Build a filter expression to select the record type that corresponds to the copybook and click **Apply**.

   ⓘ *Refer to the TSS online help for information on filtering.*

   Only records with a single record type should show in the preview pane.

   Example

   Assume the following:

   - Record type field is REC-TYPE
   - Valid values for REC-TYPE are H, D, or F
   - Selected copybook describes records with a REC-TYPE of D

   You would build a filter expression that looks like:
   REC-TYPE = D

8. Continue with the Create Entity Wizard.

9. Once the entity is created, repeat steps 3-8 to create an entity for each record type.

Example

Assume that you have one data file and one copybook that contains header, trailer, and account information.

Original Data File and copybook with Multiple Record Types

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
D	J	O	H	N				S	M	I	T	H			
H	B	R	A	N	C	H	-	A	B	C					
D	J	A	N	E				N	O	R	M	A	N		
F	0	0	2												
D	S	I	M	O	N			P	A	U	L				

```
01 ACCOUNTS.

  03 REC-TYPE          PIC X.
  03 ACCT-DETAIL       PIC X(15).
  03 HEADER REDEFINES ACCT-DETAIL.
    05 SOURCE-INFO     PIC X(15).
  03 ACCT-RECORD REDEFINES ACCT-
  DETAIL.
    05 FIRST-NAME      PIC X(7).
    05 LAST-NAME       PIC X(8).
  03 TRAILER REDEFINES ACCT-DETAIL.
    05 COUNT           PIC 999.
    05 FILLER          PIC X(12).
```

In order for TSS to represent this data correctly, you must create 3 copybooks to represent each record type:

```
01 ACCOUNTS-HEADER.
   03 REC-TYPE    PIC X.
   03 SOURCE-INFO PIC
X(15).
```

```
01 ACCOUNTS-DETAIL.
   03 REC-TYPE   PIC X.
   03 FIRST-NAME PIC
X(7).
   03 LAST-NAME  PIC
X(8).
```

```
01 ACCOUNTS-TRAILER.
   03 REC-TYPE PIC X.
   03 COUNT    PIC 999.
   03 FILLER   PIC
X(12).
```

Step through the Create Entity Wizard 3 times. Each time applying a different copybook representing each record type and filtering out the record types that do not match the copybook. In this example, the result should be 3 entities—Accounts Header, Accounts Detail, and Accounts Trailer.

Unsupported Copybook Features

TSS ignores the following COBOL copybook features:

- Any printer formatting information such as SKIP or PAGE
- Currency symbols, padding characters, and thousands separators (treated as strings on import)
- VALUE clause representing default values
- INDEX clause
- Descriptors such as RENAMES, JUSTIFIED, and SYNCHRONIZED
- LEVEL 66, 77, and 88
- POINTER
- PIC 1

TSS will not represent data correctly if there are occurrences of:

- OCCURS DEPENDING clause
- COMP-2, COMP-4, and COMP-5. (IBM-formatted COMP-2 data *is* supported by TSS.)

ⓘ *Although the data will not be displayed correctly, you could replace the COMP-2, 4, 5 copybook information with pictures such as* `PIC X(n)` *fields of the correct length. The benefit of making this change is that you would not have to remove the data from the data file. The fields can then be ignored at the loader preview stage—as long as the COMP-2,4,5 pictures have been replaced with* `PIC X(n)` *fields of the correct length.*

Ensure that any instances of these unsupported copybook features are removed from the copybook and the data file matches any changes that are made to the copybook prior to creating Entities.

Ensure First Normal Form

Before creating entities from flat files described by COBOL copybooks, you must manually convert the data and the copybook (schema) into First Normal Form (1NF) if any of the following conditions exist:

- Copybook contains OCCURS DEPENDING clauses; or
- Processing entire array as if it were one field.

APPENDIX C

Single Sign-On Support for TSS ODBC Loader Connections to SQL Servers

This appendix describes how to enable single sign-on (SSO) privileges to a SQL Server on a Windows server machine. This allows TSS users to load data using a secure Trillium-supplied ODBC data connection from a SQL Server into a TSS repository server on another Windows server machine.

Loader connection security is configured using the Kerberos authentication method.

This appendix includes the following topics:

Note the following guidelines:

- This configuration applies to Windows Server 2008, 2012, and 2012 R2 and requires that you have implemented Windows-authentication for TSS users. See Users on page 26 for information about TSS users.
- This procedure is valid for Trillium-supplied ODBC loader connections and TSS 15 SQL Server Drivers only.
- Users must have permissions to access the database on the SQL Server system.
- Run this procedure for each SQL Server database to which your TSS users plan on connecting.

Overview

The SSO configuration supports multiple TSS users accessing SQL Server connections using their trusted Windows authentication credentials.

The following steps outline this configuration process:

1. Install TSS Version 15, including the TSS repository server, on a Windows 2008, 2012, or 2012 R2 server system. The server installation program installs the ODBC Trillium-supplied adapter. For installation details, see the *Trillium Software System Installation Guide*.
2. Create a service account on the Windows domain to provide a context for services running on the Windows server. The service account allows multiple TSS client users access to secure SQL Server data sources in the Control Center without having to supply a password.
3. Configure the Windows server system on which Microsoft SQL Server is installed by adding the service account to each database the TSS user will access.
4. Configure the Windows server system on which TSS is installed by granting the service account full rights to the TSS directories.
5. Configure the TSS inetd and TSS scheduler services to log on as the service account.
6. Define a DSN using the TSS 15 SQL Server Driver.
7. In the Repository Manager, create a loader connection for SQL Server using the TSS ODBC (Trillium-supplied) driver. Select the **Use Single Sign-on** check box to enable SSO for that connection. For more information, see Chapter 4: Loader Connections on page 37.

Before You Begin

Before you begin the configuration process, note the following guidelines and prerequisites:

- These procedures must be run by a user who has domain administrator privileges to Active Directory.
- This functionality is supported on Windows Server versions 2008 and later. Verify your system have the most recent Windows updates installed.
- Before users can log on to the Control Center:
  - Ensure users are authenticated to access the SQL Server machine(s) on which the data source resides.
  - Grant Windows-authenticated TSS users access to one or more repositories in the Repository Manager. These users can then use their full domain Windows authentication to log on to the Control Center.

ⓘ *Only TSS users configured to access at least one repository using their Windows authentication can use SSO for loader connections.*

Configuration Options

You can install the ODBC adapter and the Repository Server on the same server system or separate systems.

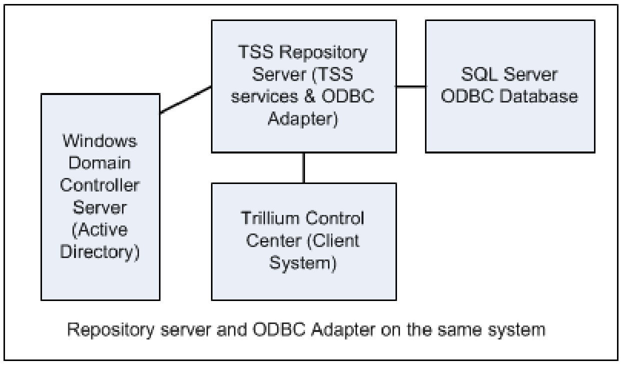

TSS Repository Server (TSS services & ODBC Adapter)
SQL Server ODBC Database
Windows Domain Controller Server (Active Directory)
Trillium Control Center (Client System)
Repository server and ODBC Adapter on the same system

Figure C-1 Repository Server and ODBC Adapter On One System

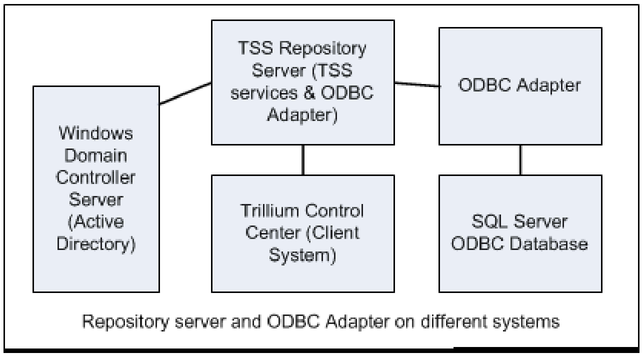

TSS Repository Server (TSS services & ODBC Adapter)
ODBC Adapter
Windows Domain Controller Server (Active Directory)
Trillium Control Center (Client System)
SQL Server ODBC Database
Repository server and ODBC Adapter on different systems

Figure C-2 Repository Server and ODBC Adapter Across Two Systems

Configuring Single-Sign On

Complete the following procedures in the order shown.

- Creating Service Account on Windows Domain on page 158
- Adding Service Account to SQL Server Databases on page 158
- Granting Service Account Rights to TSS Directories on page 158
- Configuring TSS Services to Log On as Service Account on page 159
- Adding New System DSN on page 160
- Creating SQL Server Loader Connection on page 160

Creating Service Account on Windows Domain

On the Windows domain, create a service account (for example, TrilliumSQL@domain.com), and configure it so that the password will not expire.

▶ **To create a Windows domain service account**

1. On the domain controller server system (or a system with domain administration tools installed), from the Control Panel, open **Administrative Tools**.
2. Click **Active Directory Users and Computers**.
3. Navigate to the organizational unit in which you want to create the account, right-click **Users**, and select **New > User**. The New Object - User window opens.
4. Enter the **First name**, **Last name**, and **Full name** for the service account logon.
5. For **User logon name**, enter the name with which the service account will log on. Select the UPN suffix that will be appended to the name, after the @ symbol. Click **Next**.
6. Enter and confirm a new password, and then select **User cannot change password** and **Password never expires**.

   ⓘ *Because of the authority of this user, ensure the password is secure.*

7. Click **Next**.
8. Review the new user account information, make any necessary changes, and click **Finish**.
9. (Optional) Add the new service account to groups as needed.

Adding Service Account to SQL Server Databases

Configure the Windows server system on which Microsoft SQL Server is installed by adding the service account to each database TSS users will access.

Set minimum permissions to **DB Read**. If TSS users will be writing back to the database, then set permissions to **DB Write**.

Granting Service Account Rights to TSS Directories

To ensure the service account has access to the TSS repository, grant the service account full access to the Trillium Software directories on the TSS server system.

Perform this procedure on the Windows server system where the TSS repository is installed.

▶ To grant service account rights to TSS system

1. To grant full rights to the `C:\Program Files (x86)\Trillium Software` directory:

   a. In Windows Explorer, navigate to `C:\Program Files (x86)`.

   b. Right-click the **`Trillium Software`** directory and select **Properties**. The Trillium Software Properties window opens.

   c. Click the **Security** tab, and click **Edit**. The Permissions for Trillium Software window opens.

   d. Click **Add**, select the service account name, and click **OK**.

   e. In the Group or user names list, ensure the service account is selected.

   f. In the Permissions list, for Full control click **Allow**, then click **Apply**. The permissions are applied.

   g. Click **Advanced**. The Advanced Security Settings for Trillium Software window opens.

   h. Click **Change Permissions**, select the service account, and check **Replace all child object permission entries with inheritable permission entries from this object**.

   i. Click **Apply**. Click **OK** to close the window.

2. Repeat Step 1 to grant full rights to the `C:\ProgramData\Trillium Software` directory. If the `ProgramData` directory is hidden, in Windows Explorer, open **Folder options**, click **View**, and select **Show hidden files, folders, and drives**.

Configuring TSS Services to Log On as Service Account

The TSS inetd and Scheduler services must be configured to log on as the new service account.

Perform this procedure on the Windows server system where the TSS repository is installed.

▶ To configure TSS services to log on as the service account

1. From the Control Panel, open **Administrative Tools**. Double-click **Services** to open the Services window.

2. Stop the **TSS - inetd** and **TSS 15 - Scheduler** services.

3. Right-click **TSS - inetd** and select **Properties**. The TSS - inetd Properties window opens.

4. Click the **Log On** tab, select **This account**, and enter the user name and password for the service account. If necessary, click **Browse** to select the user name.

5. Click **Apply**. Click **OK** to close the window.
6. Right-click **TSS 15 - Scheduler** and repeat Steps 3 through 5.
7. Start the TSS - inetd and TSS version - Scheduler services.

Adding New System DSN

You must create and configure a Data Source Name (DSN) for the TSS ODBC loader connection with information about the specific database to which you want the connection to point.

Perform this procedure on the Windows server system where the TSS repository is installed.

▶ To add a new system DSN

1. Click **Start > All Programs > Trillium Software > ODBC Data Source Administrator (32-bit)**. The ODBC Data Source Administrator window opens.
2. Click the **System DSN** tab.
3. Click **Add** to open the Create New Data Source window.
4. Scroll through the list of drivers and select **TSS 15 SQL Server Driver**.
5. Click **Finish**. The ODBC SQL Server Wire Protocol Driver Setup window opens.
6. Configure the data source with the Host Name, Port Number, and Database of the SQL Server system.
7. Click the **Security** tab.
8. For Authentication Method, select **Kerberos**.
9. Click **Apply**. Click **Close** to close the setup window.

Creating SQL Server Loader Connection

In the Repository Manager, create a loader connection for SQL Server using the TSS ODBC (Trillium-supplied) driver. Select the **Use Single Sign-on** check box to enable SSO for that connection. For more information, see Chapter 4: Loader Connections on page 37.

Reset Privileges and Remove SSO (Optional)

At any time after configuring SSO for Trillium-supplied ODBC loader connections, you can change your security model back to a conventional, non-SSO environment.

Complete this optional procedure to remove the TSS service account privileges and add privileges to the Windows SYSTEM account, allowing that account to start the TSS services.

▶ To set privileges back to the Windows SYSTEM account

1. On the Repository Server system running the current TSS services, stop the **TSS - inetd** and **TSS - Scheduler** services and edit them to log on as the original Windows SYSTEM account.
2. Restart the **TSS-inetd** and **TSS-Scheduler** services.
3. In the Repository Manager, open the SQL Server ODBC connection for editing. Deselect the **Use Single Sign-on** check box to disable SSO for that connection, and save the changes.

APPENDIX D

Quality Command Line Options

Chapter 9 describes repository-level command line options. Those commands can be run either from the operating system command line prompt or the Repository Administration command prompt. This appendix also describes command line options, but these commands are specific to Quality projects and can only be run from the operating system command line prompt.

This appendix includes the following topics:

Compressing/Expanding Quality Output Files

When you run a Quality project, each process creates an output file. Depending on the size of your original source file and the complexity of your project, the resulting output files can take up a large amount of disk space. To conserve disk space, Trillium Software System provides two distinct options:

- **Piping data from process to process**. When you pipe data, only the final output file is created. This is a good option if you have already validated each step, but not if you are still testing the project.
- **Compressing output data**. When you compress data, TSS creates an output file for each process but, because the data is compressed, the output files are significantly smaller. (The final output file is not compressed.) You can review individual output files in the same way that you would review non-compressed files.

ⓘ *In some instances, compressing Quality output files adversely affects performance.*

Both piping and compression are enabled through the Control Center. This section describes the command line option that enables you to compress or expand an individual file.

The syntax of the compress command is:

```
tranfrmr -compress filein reclen fileout
```

where *filein* is the name of the Trillium file that you want to compress, *reclen* is the length of the records in the input file, and *fileout* is the name you want to give to the compressed file. For example:

```
tranfrmr -compress tranfrmr04.dat 129 tranfrmr04.compress
```

The syntax of the expand command is:

```
tranfrmr -uncompress filein fileout
```

where *filein* is the name of the file you want to expand and *fileout* is the name of the expanded file. For example:

```
tranfrmr -uncompress tranfrmrZ_04.tsz transfrmrZ_04.expand
```

ⓘ *If you omit the fileout argument from either command, the output is written to the screen.*

Running the Batch Deployment Tool

The Batch Deployment Tool (BDT) creates batch scripts from exported project definitions.

ⓘ *The BDT is an optional installation component valid in 32-bit environments only. If the BDT is not installed you cannot create batch scripts.*

You can create a batch script directly from the Control Center, or you can export a project definition to a machine on which the BDT has been installed and run the BDT from the command line to create the batch script. This allows you to easily move the project to a different system before generating the script files, which ensures that the paths used in the project files are correct for the target system.

ⓘ *If you create a batch script and subsequently move the project files to a different machine, you will have to modify paths in the script files. It is advisable to move the project definition to the destination system before running the BDT.*

The BDT executable, bdt.exe, is located in your installation path in the `\BDT\...\bin` directory and takes arguments described here.

Table D-1 BDT Command Arguments

Argument	Description
-project	Name of the project definition file. The default is project.prj. The path will default to the current working directory.
-proj_compress *value*	Source directory for creating the project definition zip file.
-proj_expand *value*	Project definition zip file to be unzipped.
-use_pipes	If this argument is specified, the generated script will pipe the data through the project’s processes.

Table D-1 BDT Command Arguments (Continued)

-endian_extn	The extension (.len or .ben) to be used for any platform-specific files. The default for Windows and Linux platforms is .len; the default for UNIX is .ben.
-input_delimiter	If the job will use delimited input files, use this argument to specify the delimiter that will be used. Supported delimiters are Comma, Tab, Space, Semicolon, and Pipe (for example, -input_delimiter Tab). Be sure to type the delimiter with an initial cap.
-input_delimiter_ encoding	If the job will use delimited input files, use this argument to specify the encoding of the delimiter that will be used. The default is NOTRANS.
-use_quotes	Include this argument if the delimited input files will have values surrounded by quotes.
-input_headers	Include this argument if the delimited input files will have header records.
-output_delimiter	If you require delimited output files, use this argument to specify the delimiter to be used. Supported delimiters are Comma, Tab, Space, Semicolon, and Pipe. ⓘ *Be sure to type the delimiter type with an initial cap.*
-output_delimiter_ encoding	If you require delimited output files, use this argument to specify the encoding of the delimiter to be used. The default is NOTRANS.
-use_output_quotes	Include this argument if the delimited output files will have values surrounded by quotes.
-use_headers	Include this argument if you want your delimited output files to have header records. Values: 0 = none (default), 1 = header record if file contains data, 2 = header
-with_data	If the project definition includes start data, include this argument if you want the start data files to be included in the batch export. Do not use this argument if you want to work with delimited input files.
-target_platform	The target platform (windows or unix) where the job will be run. The default is the same as the host platform. The value must be entirely in lower case.
-target_dir	Name of the directory where the batch export files are to be created. The default is ./batch, relative to the directory containing the project.prj. Enclose the path in quotes if it includes spaces.
-no_proj_dir	Do not create project directory.

Table D-1 BDT Command Arguments (Continued)

Argument	Description
-tsq_language	Language in which messages are to be output. Values are: ■ es (Spanish) ■ ja (Japanese) ■ pt_BR (Portuguese, Brazil) ■ de (German) ■ fr (French) ■ zh_TW (Chinese, Taiwan) ■ zh_CN (Chinese, Mainland) ■ it (Italian) ■ ko (Korean) ■ en (English, the default)
-tsq_report_encoding	Encoding to be used for the output reports. The value must be specified in uppercase. The default is UTF8.
-ts_quality	Name of the path where TS Quality has been installed. The default is as specified by the TS_QUALITY environment variable.
-help	Display help for BDT command.
-?	Display help for BDT command.
-realtime	Create a project structure that can be used to create a real-time project.
-no_data_compress	Generate uncompressed data in individual output files.

Example

Assuming the project definition files are in `C:\My Projects\project1`, the following example generates a batch script in `C:\Batch` that will run on the host platform using the data that was exported with the project definition.

```
“<BDT_install_path>\bin\bdt>” -project "C:\My
Projects\project1\project.prj" -target_dir C:\Batch -with_
data
```

where <BDT_install_path> is your installation path.

Enabling Uncommitted Read

You can enable uncommitted read (dirty read) if needed. By default it is disabled. Enabling uncommitted read allows you to read the values from the first transaction in a multi-transaction process before the changed values are committed.

Before you enable uncommitted read:

- Understand the implications of uncommitted read (or dirty read) - for example, it is only for transactions that are slow or if the possibility of errors are low.
- Verify that your database supports uncommitted read. Consult with their DBA or Database vendor for additional information.

Enabling Uncommitted Read in TSQDA

You can enable uncommitted read in tsqda, ODBC, DB2, Oracle etc.

▶ To enable uncommitted read in tsqda:

1. Open the configuration file.

   - For Data Adaptor, open `Data Adapter\etc\config.txt`
   - For ODBC, open `ODBC Adapter\etc\config.txt`
   - For other loader connections, for example, DB2 or Oracle, open `MBSW\15\etc\config.txt`

2. Modify the uncommitted_read parameter of the key rdbms section:

   ```
    key rdbms {
       value uncommitted_read "on"
       value records_read 100
       value records_read_odbc 1
   }
   ```

3. Save your changes.

APPENDIX E

Configuring Repositories to Use Profiling Process with Pipes

Before users can use pipes with profiling processes (including Analysis, Business Rules, Dependencies, Keys, and Load) in the Control Center, the repository administrator must configure the repository by adding administrator or generic username password account information to a user-created file called `admin_conf.txt`. In Windows-authenticated environments the SYSTEM security permission to each repository must also be set.

Without setting these two configurations, users in the Control Center will be unable to run Quality profiling processes using pipes.

Note the following guidelines:

- Only one instance of the file `admin_conf.txt` can exist for each repository.
- Username and password account information need only be configured one time per repository.

Complete the procedure for each repository where piping will be used forprofiling processes.

▶ To add account information

1. On the system where your repository is installed, create a text file called admin_conf.txt in the repository directory.

   For example, `C:\ProgramData\Trillium Software\MBSW\15.0\Data\metabase\`*`repository_name`*.

2. In the `admin_conf.txt` file, on separate lines, add one entry for the username and one entry for the password.

   Use the following syntax:

   user=*yourusername*

   password=*yourpassword*

3. Save and close the file.

ⓘ *The SYSTEM permission does not have to be set in Trillium-authenticated environments.*

▶ To add SYSTEM security permission to support Windows authentication

1. In the Repository Manager, on the Home tab, click Repository Security. The Permissions for Repository Security window opens.
2. Click **Add**. The Select Users, Computers, Service Accounts, or Groups window opens.
3. In the Enter object names to select field, enter *system* and click **OK**.
4. In the Group or user names section, select **SYSTEM**.
5. In the Permissions for SYSTEM section, select the **Allow** check box for each permission.
6. Click **OK**.

APPENDIX F

Configuring Business Group Limit Setting

In the Trillium Control Center, scheduled jobs can run consecutively, or concurrently, depending on whether you choose to run the jobs now or at later times. How jobs run, also depend on the `maximumWeight` parameter set in the server configuration (`config.txt`) file.

- If `maximumWeight` is `1`, scheduled jobs run consecutively in sequential order.
- If `maximumWeight` is greater than `1`, scheduled jobs run concurrently up to the set value. For example, if `maximumWeight = 10`, up to ten scheduled jobs can run concurrently.

When you analyze business rules using the Group By option and the `maximumWeight` parameter is greater than `1`, business rule analysis runs concurrently on multiple rules at a time. This can lead to performance issues including a slow down (or hanging) of your system.

You can use the optional `business_group_limit` setting to restrict the concurrent analysis of business rules using the Group By option.

ⓘ *Other types of scheduled jobs, for example, creating entities and running TS Quality processes, are not affected and run in parallel regardless of the business_group_limit value.*

Guidelines:

By default, the `business_group_limit` is disabled, that is set to `0`, and rule analysis jobs can run without any constraint. When `business_group_limit` is set to greater than 1, it allows rule analysis jobs to run concurrently. If the number of business rules using the Group By option is greater than the value set in business_group_limit, the rule analysis jobs run in consecutive order. This is desirable to avoid performance issues.

Example:

Suppose you have five business rules using the Group By option. You modify the server configuration file such that, `maximunWeight = 2`, and `business_group_limit = 4`.

Now when you schedule rule analysis of two business rules, the jobs run in sequential order. As a result, the second rule analysis job is able to obtain the resources it needs and avoid performance issues.

▶ To set business_group_limit

1. On the server system, stop the TSS scheduler service.
2. Open the `config.txt` file for editing:
   - On Windows, file is located in : `C:\Program Files x64\Trillium Software\MBSW\15\etc`
   - On UNIX, file is located in: `UNIX: …/metabase/etc`
3. Go to the key public section.
4. Modify the value for the business_group_limit. For example,

```
key public {
    value max_string_size 32767
    value utf8_length 3
    value str_base 1
    value business_group_limit 5
   }
```

ⓘ *If you upgraded from an earlier version of TSS v15, you must manually enter business_group_limit parameter into the config.txt file.*

5. Go to the key scheduler section.
6. Modify the value of the `maximumWeight` parameter. For example,

```
key scheduler {
    value port [get {public ports scheduler}]
    value timeout [expr {5*60*1000}]
    value maximumWeight 2
```

ⓘ *Ensure the value is greater than 1 (default), otherwise the business_group_limit will be treated as if set to 0 (or turned off).*

7. Save and close the file.
8. Start the TSS scheduler service.

APPENDIX G

How to Set the Dynamic Library Path when Exporting Quality Profiling Projects to Batch (UNIX)

Before you run a batch script for an exported project that contains Quality profiling processes, you must set the dynamic library path to point to your system's libcore library directory.

This procedure applies to exported Quality projects that contain one or more of the following profiling processes:

- Business Rules
- Dependencies
- Keys
- Load

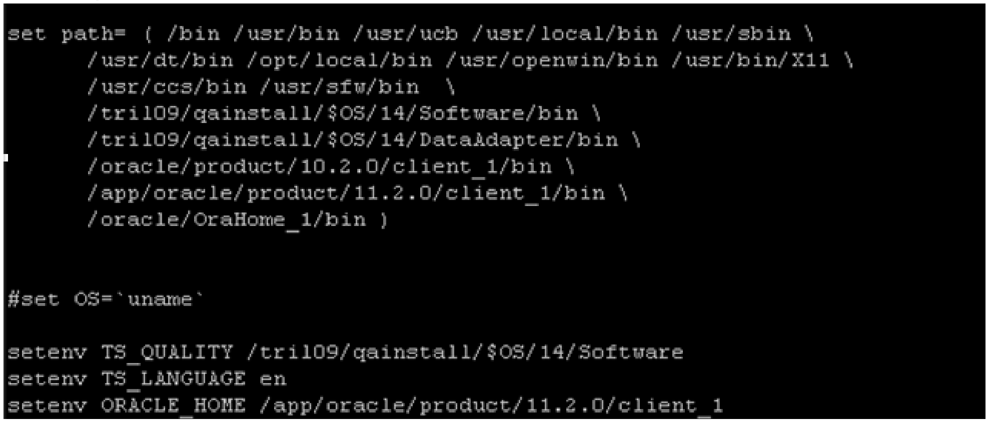
ⓘ *For more information about exporting projects that contain profiling processes, see the Profiling Processes topic in the TSS Help.*

Note the following:

- This information applies to TSS Versions 15.x on UNIX systems only.
- You must be a TSS repository or systems administrator to complete this procedure.
- Set the environment variable at any time before you export the project and run it in batch.
- This procedure permanently sets the dynamic library path in your environment so that you do not have to set it each time you start a session.

▶ To set the dynamic library path on UNIX

1. Open the settings file, located in your home directory, as shown in the example below.



```
set path= ( /bin /usr/bin /usr/ucb /usr/local/bin /usr/sbin \
      /usr/dt/bin /opt/local/bin /usr/openwin/bin /usr/bin/X11 \
      /usr/ccs/bin /usr/sfw/bin  \
      /tril09/qainstall/$OS/14/Software/bin \
      /tril09/qainstall/$OS/14/DataAdapter/bin \
      /oracle/product/10.2.0/client_1/bin \
      /app/oracle/product/11.2.0/client_1/bin \
      /oracle/OraHome_1/bin )


#set OS=`uname`

setenv TS_QUALITY /tril09/qainstall/$OS/14/Software
setenv TS_LANGUAGE en
setenv ORACLE_HOME /app/oracle/product/11.2.0/client_1
```

Figure E-1 .cshrc File Example

2. Add the dynamic library path to the variable list.

3. Set it to point to `Software/bin` directory of your TSS installation. This is used as the libcore library directory. For example:

```
setenv LD_LIBRARY_PATH /.../Linux/15/Software/bin
```

4. Save the file.

5. Source the file, as shown in the command example here:

```
source .cshrc
```

Index