TIBCO BusinessConnect™

Scripting Deployment User's Guide

Version 7.3.0
April 2022

Contents

Chapter 1 Scripting Deployment

This chapter introduces the TIBCO BusinessConnect Scripting Deployment tool, which is used to configure and deploy TIBCO BusinessConnect applications in the administration domain using the command line interface.

Topics

Overview

TIBCO BusinessConnect Scripting Deployment enables the configuration and deployment of a TIBCO BusinessConnect application in a non-GUI mode. This includes deployment related tasks, such as creating and deploying TIBCO BusinessConnect in TIBCO Administrator, as well as configuring the TIBCO BusinessConnect configuration store (creating TIBCO BusinessConnect installation, activating protocols, and importing CSX files).

Scripted deployment itself does not support certain deployment related aspects of the TIBCO BusinessConnect application, such as public and private process transports, file poller configurations, and intercomponent settings. Therefore, these configuration settings need to be extracted from an existing TIBCO BusinessConnect deployment, where they have been already configured using the TIBCO BusinessConnect GUI.

This allows the entire TIBCO BusinessConnect repository and deployment configurations on a TIBCO Administrator domain to be archived and reconstructed on the same or another TIBCO Administrator domain. This archive can be source-controlled for later deployment or redeployment; it can have different service bindings, on different machines, with different clustering, and different load balancing layout.

TIBCO BusinessConnect Scripting Deployment also facilitates the following:

- Automation for moving the TIBCO BusinessConnect application from one environment to another, such as development to QA, or user acceptance to production.
- Deploying of new participants or new operations between environments with the ability to incrementally export the TIBCO BusinessConnect repository, such as participants and operations, and without using the TIBCO BusinessConnect Administrator console.

TIBCO BusinessConnect Scripting Deployment uses various commands to perform configuration and deployment tasks.

Most of the configuration and deployment tasks use the configuration file `commonConfig.properties`. This file is located in the directory *BC_HOME*`/tools/bcappmanage`. For more details, see Chapter 3, Parameters for commonConfig.properties.

TIBCO BusinessConnect Scripting Deployment Architecture

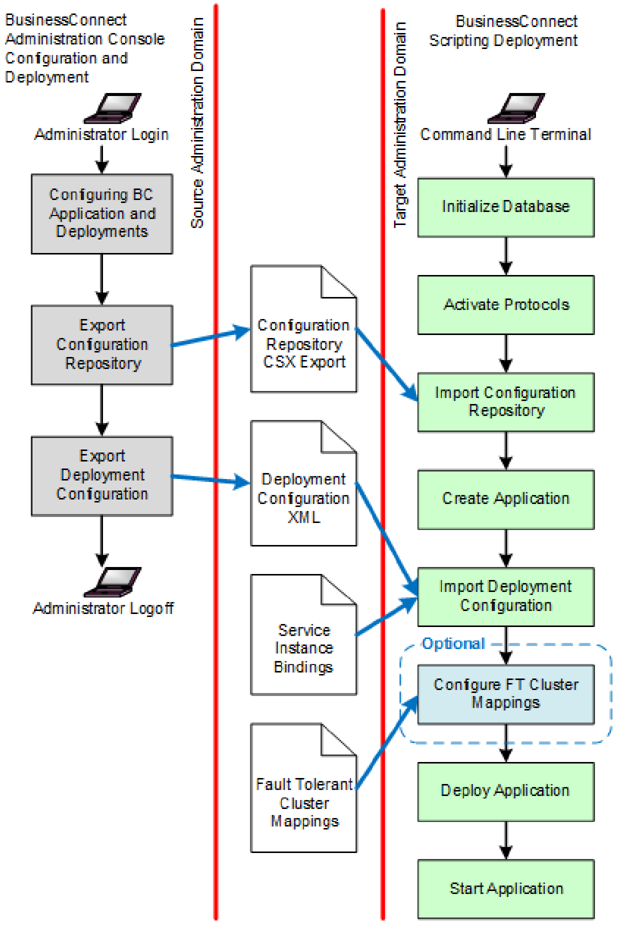
TIBCO BusinessConnect Scripting Deployment (`bcappmanage`) combines the underlying TIBCO BusinessConnect configuration repository management API in the TIBCO Administrator console and the TIBCO Runtime Agent application management utility into a single suite of command line utilities. These utilities create a complete scripting deployment environment for managing of the TIBCO BusinessConnect application.

TIBCO BusinessConnect Scripting Deployment uses four data files to persist all configurations as archives for later deployment:

- **Configuration Repository CSX Export** It contains all configurations in the TIBCO BusinessConnect configuration repository:
  - System settings
  - Trading partner profiles and business agreement settings
  - Operation definitions
  - Public transport and private process Smart Routing rules
  - Private key and certificate credentials
- **Deployment Configuration XML** It contains all configurations in the TIBCO BusinessConnect application archive for a TIBCO Administrator domain:
  - Process definitions that drives the TIBCO BusinessConnect runtime engine
  - Public transport settings
  - Private process and inter-component communication settings
  - Other miscellaneous settings that control the TIBCO BusinessConnect runtime engine
- **Service Instances Binding** It contains all configurations for dynamic binding of machines as the application service containers that execute the TIBCO BusinessConnect runtime engine:
  - Service instances to machines managed by the destination TIBCO Administrator domain
  - Other miscellaneous settings that control each binding service instances
- **FT Cluster Mappings** It contains all configurations for dynamic assignments of service instance bindings to fault-tolerant groups and assignments of fault-tolerant groups to public Smart Routing clusters.

Figure 1 illustrates a typical usage scenario for the TIBCO BusinessConnect Scripting Deployment’s two major tasks: preparation for deployment, and deployment and management of the TIBCO BusinessConnect application. See TIBCO BusinessConnect Scripting Deployment’s Tasks for more information.

Figure 1 TIBCO BusinessConnect Scripting Deployment Architecture


BusinessConnect Administration Console Configuration and Deployment
Administrator Login
Configuring BC Application and Deployments
Export Configuration Repository
Export Deployment Configuration
Administrator Logoff
Source Administration Domain
Configuration Repository CSX Export
Deployment Configuration XML
Service Instance Bindings
Fault Tolerant Cluster Mappings
Target Administration Domain
BusinessConnect Scripting Deployment
Command Line Terminal
Initialize Database
Activate Protocols
Import Configuration Repository
Create Application
Import Deployment Configuration
Optional
Configure FT Cluster Mappings
Deploy Application
Start Application

TIBCO BusinessConnect Scripting Deployment's Tasks

The two major tasks performed by the TIBCO BusinessConnect Scripting Deployment consist of the following:

- Preparation for Deployment
- Deploying and Managing TIBCO BusinessConnect Application

Preparation for Deployment

For more details, see Task A: Preparing for Deployment on page 11.

This task exports the TIBCO BusinessConnect configuration repository CSX export and the deployment configuration XML after completing all the configurations necessary for deployment. It serves as the basis for deploying the TIBCO BusinessConnect application on an TIBCO Administrator domain without the application service containers. Repeat this task when there are any changes in the configurations.

For all TIBCO BusinessConnect configuration related tasks, the JDBC connection parameters for the TIBCO BusinessConnect configuration store need to be specified. These connection parameters are picked up by default from the file `commonConfig.properties`, a regular Java properties file located in the current working directory *BC_HOME*`/tools/bcappmanage`. The location of this file can also be specified with the argument `-configRepoURL`.

Manage Configuration Repository Commands

The following commands are used to prepare for deployment:

- **-createInstallation** Create configuration repository for object models, base metadata, and schemas.
- **-createTables** Create the audit, non-repudiation, and runtime database tables.
- **-activateProtocol** Activate the installed protocol plugins, by performing protocol metadata activation and by creating schemas and supporting runtime tables.
- **-exportConfigRepo** Export the TIBCO BusinessConnect configuration repository into a CSX file with options to allow full installation export or by certificates, operations or participants only.

- **-importConfigRepo** Import the previously exported configuration repository saved as a CSX file.

Argument

The connection parameters are picked up by default from the file `comonConfig.properties`, a regular Java properties file located in the current working directory *BC_HOME*`/tools/bcappmanage`. The location of this file can also be specified with the argument `-configRepoURL`.

- **-configRepoURL** The URL of the configuration repository. This argument follows the syntax `file://`*`file path`*.

Deploying and Managing TIBCO BusinessConnect Application

For more details, see Task B: Deploying and Managing TIBCO BusinessConnect Applications.

This task uses the service instance binding map file to bind service instances dynamically as application service containers for machines managed by a TIBCO Administrator domain at deployment time. Optionally, dynamic assignments of the fault tolerant groups and the public Smart Routing clusters are allowed using the fault tolerant cluster map file.

Deployment tasks require that the administrative domain and authentication credentials are specified. In addition to the listed commands, the command line arguments are also needed. See Deployment Arguments.

Manage Deployment Configuration Commands

- **-exportDeployConfig** Export deployment configuration in XML format.
- **-importDeployConfig** Import deployment configuration in XML format with machine bindings. To learn more, see Binding Maps.
- **-configFTClusterMap** Fault-tolerant and load-balanced mapping for Public Smart Routing.

Manage Application Service Deployment Commands

- **-createApp** Create a TIBCO BusinessConnect application service.
- **-deleteApp** Delete a TIBCO BusinessConnect application service.
- **-deployApp** Deploy a TIBCO BusinessConnect application service.
- **-undeployApp** Undeploy a TIBCO BusinessConnect application service.

Manage Application Service Instances Commands

- **-startApp** Start TIBCO BusinessConnect service instances.
- **-stopApp** Stop TIBCO BusinessConnect service instances.
- **-killApp** Kill TIBCO BusinessConnect service instances.

Deployment Arguments

The mandatory arguments (parameters) for all deployment tasks are as follows:

- **-user** Administrative domain user name
- **-pw** Administrative domain password for the machine
- **-cred** Credentials file. See Credentials File.
- **-domain** Administrative domain

These arguments are not contained in the file `commonConfig.properties.`

Online Help

To invoke the online help for TIBCO BusinessConnect, go to the directory *BC_HOME*`/tools/bcappmanage` and type

`bcappmanage`

The following list of online help commands appears:

```
BCAppManage Usage:

        Manage Configuration Repository:

        -createInstallation     : create Configuration Repository
        -createTables           : create Audit / NR / Runtime database tables
        -activateProtocol       : activate installed Protocol Plugin
        -exportConfigRepo       : export Configuration Repository (csx)
        -importConfigRepo       : import Configuration Repository (csx)
        -createGatewayToken     : create gateway tokens
        -listGatewayToken       : list a gateway token's properties

        Manage Deployment Configuration:

        -exportDeployConfig     : export Deployment Configuration template
        -importDeployConfig     : import Deployment Configuration with machine
bindings
        -configFTClusterMap     : create Fault Tolerance / Load Balance Clustering
for Public Smart Routing

        Manage Application Service Deployment:

        -createApp              : create Application Service
        -deleteApp              : delete Application Service
        -deployApp              : deploy Application Service
        -undeployApp            : undeploy Application Service

        Manage Application Service Instances:

        -startApp       : start Service Instances
        -stopApp        : stop Service Instances
        -killApp        : kill Service Instances

        Manage Gateway Server Instances:

        -assignService          : assign service to gateway server
        -startTransports        : start gateway server transports
        -stopTransports         : stop gateway server transports
```

Chapter 2 Getting Started

This chapter shows the sequence for configuring and deploying TIBCO BusinessConnect application using TIBCO BusinessConnect.

Topics

Starting bcappmanage

To start the `bcappmanage` utility, change to the directory *BC_HOME*`/tools/bcappmanage`, and type the following

```
bcappmanage
```

See the section Online Help for information about using online help.

Once the TIBCO BusinessConnect application has been configured and deployed, you can quickly export the existing configuration and deployment repositories. After these preparation tasks are done, continue by re-installing the application, importing the previously exported configuration, creating tables (if needed), and deploying and managing the application using the command line interface.

To configure and deploy the TIBCO BusinessConnect application with TIBCO BusinessConnect Scripting Deployment, the sequence explained in Managing Applications Using bcappmanage must be followed.

Users who are planning to use the Oracle OCI drivers must specify the Oracle Client Library path in the environment variable `tibco.env.`*ORACLE_CLIENT_PATH* of the file `bcappmanage.tra` file located in the *BC_HOME*`/tools/bcappmanage` directory.

The value must point to the Oracle native client libraries directory.

- On Windows, it is typically *ORACLE_CLIENT_HOME* directory; for example, `C:\oracle\product\11.2.0`.
- On UNIX systems, it is typically the *ORACLE_CLIENT_HOME* directory.

Managing Applications Using bcappmanage

TIBCO BusinessConnect Scripting Deployment cannot be used for the initial TIBCO BusinessConnect application installation and configuration, which must be performed in the TIBCO BusinessConnect GUI using TIBCO Administrator. After you have used the GUI interface to install and configure the application, you can use either the same GUI or the command line interface to perform management and deployment tasks. Before performing any of the listed tasks, make sure to first provide information for the file `commonConfig.properties`. More details about this configuration file are available in Chapter 3, Parameters for commonConfig.properties.

All parameters in the file `commonConfig.properties` will be validated by TIBCO Runtime Agent. If any of the required parameters are missing or have been entered in a format that is not acceptable, TIBCO Runtime Agent will throw an error or insert default values, where appropriate.

To see the completed file commonConfig.properties with added parameters, see Appendix B, Configuration Properties File.

Task A: Preparing for Deployment

To prepare for configuration of a TIBCO BusinessConnect application using TIBCO BusinessConnect Scripting Deployment, the deployment configuration and configuration repository need to be exported. Both tasks are required and can be done in any order.

1. **Export Deployment Configuration**

   ```
   bcappmanage -exportDeployConfig
   ```

   The generated XML file will be used during the task invoked by the command `-importDeployConfig`. Repeat the command `-exportDeployConfig` if any of the configured deployment parameters have changed.

   The complete command reference is available in `-exportDeployConfig`.

2. **Export Configuration Repository**

   ```
   bcappmanage -exportConfigRepo
   ```

   This task is used for generating CSX configuration export files from the TIBCO BusinessConnect configuration repository. It supports four main categories: participants, operations, certificates, and entire installation.

   The complete command reference is available in `-exportConfigRepo`.

Task B: Deploying and Managing TIBCO BusinessConnect Applications

Once the configuration repository and the deployment configuration are exported for the selected application categories, you can install, deploy, and manage the TIBCO BusinessConnect application.

1. **Initialize Database**

   ```
   bcappmanage -createInstallation
   ```

   Same as the Initialize Database task in the TIBCO BusinessConnect GUI, this task creates the TIBCO BusinessConnect installation. The complete command reference is available in `-createInstallation`.

2. **Activate Protocol**

   ```
   bcappmanage -activateProtocol
   ```

   This task allows activation in one pass for all available protocol plugins installed on the machine where `bcappmanage` is running. The complete command reference is available in `-activateProtocol`.

3. **Create Tables (optional)**

   ```
   bcappmanage -createTables
   ```

   This task is optional and is not needed when the audit, non-repudiation, and runtime data will be stored in the database account that is the same as the account of the configuration store. The complete command reference is available in -createTables.

4. **Import Configuration Repository**

   ```
   bcappmanage -importConfigRepo
   ```

   Imports the previously exported configuration repository saved as a CSX file. The complete command reference is available in `-importConfigRepo`.

5. **Create Application**

   ```
   bcappmanage -createApp
   ```

   Creates deployment configuration for a TIBCO BusinessConnect application service. The complete command reference is available in `-createApp`.

6. **Extract Configuration Data**

   ```
   bcappmanage -extractConfigData
   ```

   Extracts configuration data of the participants, agreements, transports, operations and the associated bindings from TIBCO BusinessConnect configuration store. The extracted data is stored as key-value pairs in Configuration Store Reporting Schema and can be queried for generating JasperReports. The complete command reference is available in `-extractConfigData`.

7. **Import Deployment Configuration**

   `bcappmanage -importDeployConfig`

   Imports the previously exported deployment configuration in XML format and also does machine assignments. Names of the Interior hosts in the `commonConfig.properties` file must be specified exactly as they are shown on the screen **Resource Management > Machines** in TIBCO Administrator. Same needs to be ensured for all property settings that use the machine names in their key. To learn more about machine bindings, see Binding Maps. The complete command reference is available in `-importDeployConfig`.

8. **Configure FT Cluster Map (optional)**

   `bcappmanage -configFTClusterMap`

   This command is optional. It configures mappings of machines to fault-tolerant groups and mappings of fault-tolerant groups to Public Smart Routing clusters.

   Configuration of fault-tolerant clusters for Public Smart Routing with TIBCO BusinessConnect Scripting Deployment requires the machine bindings of service instances be configured during the `-importDeployConfig` task. This tool does not recognize machine bindings that are configured using the TIBCO BusinessConnect administration console GUI. The complete command reference is available in `-configFTClusterMap`.

9. **Deploy Application**

   `bcappmanage -deployApp`

   This command corresponds to the Deploy task in the TIBCO BusinessConnect GUI and deploys the TIBCO BusinessConnect application service with an option to suspend starting of all service instances on deploy. The complete command reference is available in `-deployApp`.

10. **Start Application**

    `bcappmanage -startApp`

    This task can be used for starting the TIBCO BusinessConnect engines deployed in scripted mode. The options define whether an individual engine or all engines will be started, and also whether these engines are on the Interior server. The complete command reference is available in `-startApp`.

Task C: Preparing and Managing Gateway Server Tokens

Once the Interior Server is deployed and running, you can create the Gateway Server token and use it to start and deploy the Gateway Server. Two commands for managing these tokens have been supplied:

- `-createGatewayToken`.
- `-listGatewayToken`.

Other Management Tasks

These tasks are dedicated to managing configured and deployed applications:

- Undeploy Application, which is presented in `-undeployApp`.
- Delete Application, which is presented in `-deleteApp`.
- Stop Application, which is presented in `-stopApp`.
- Kill Application, which is presented in `-killApp`.

Chapter 3 Parameters for commonConfig.properties

This chapter explains the parameters that are contained in the file `commonConfig.properties`.

Topics

Overview

When you install TIBCO BusinessConnect, the installation will create the `file commonConfig.properties` in the directory *BC_HOME*/tools/bcappmanage.

This file has not been configured for your site and the configuration values such as host names and passwords are missing. If you try to deploy a TIBCO BusinessConnect application with the initially generated configuration properties file, you will get error prompts asking you to enter the missing values.

To enter the values (parameters or keys) in the file `commonConfig.properties`, either use the values that were available for the TIBCO BusinessConnect application configured and deployed using the TIBCO BusinessConnect GUI, or supply new values.

The values that you have to enter belong to three categories and have to be mapped to the installed parameters as explained in the following sections:

- JDBC Maps
- Binding Maps
- Cluster/FT Map

A sample file `commonConfig.properties` that contains the values needed for deployment is shown in Appendix B, Configuration Properties File.

JDBC Maps

The section JDBC Maps of the file `commonConfig.properties` contains the JDBC connection parameters needed for configuration of repositories and of the audit, non-repudiation, and runtime tables.

Configuration Repository Connection Parameters

Instead of posting your JDBC user ID and password in different files, it is preferred to have them specified as the command line arguments with commands `-configRepoUser` and `-configRepoPw`, respectively.

- **`bc.repo.db.driver =`** *value*
- **`bc.repo.db.url =`** *value*
- **`bc.repo.db.userID =`** *value*

  This parameter is optional if the command `-configRepoUser` is used.
- **`bc.repo.db.password =`** *value*

  This parameter is optional if the command `-configRepoPw` is used .
- **`bc.repo.db.connectionPool.size`** = *value*

Audit, Non-Repudiation, and Runtime Connection Parameters

The audit, non-repudiation, and runtime connection parameters are used by the task `-createTables`.

- **`bc.[audit | nr | runtime].db.driver =`** *value*
- **`bc.[audit | nr | runtime].db.url =`** *value*
- **`bc.[audit | nr | runtime].db.userID =`** *value*

  This parameter is optional if overwritten by the line parameter for the command `-jdbcUser` (see -createTables).
- **`bc.[audit | nr | runtime].db.password =`** *value*

  This parameter is optional if overwritten by the line parameter for the command `-jdbcPw` (see -createTables).
- **`bc.[audit | nr | runtime].db.connectionPool.size`** = *value*

Binding Maps

The section Bindings Maps of the file `commonConfig.properties` contains the values needed to configure Interior server bindings, fault-tolerant parameters, heap sizes, thread stack sizes, and NT service startup parameters.

The binding maps properties are used by the task `-importDeployConfig`.

Machine Mapping to Interior Server Bindings

`bc.appmanage.[interior].hosts = [<host>, ]`

`[interior]`

`The parameter interior` is used for mapping engines that need to be deployed on the Interior Server.

`[<host>, ]`

List of hosts where the Interior servers need to be deployed. Repeat the same host for multiple service instances per host.

Examples:

```
bc.appmanage.interior.hosts = myHostA, myHostA, myHostA
```

Optional General and Java Server Settings Parameters

For more details about server settings parameters, see *TIBCO Administrator User's Guide,* Setting Deployment Options, section Server Settings Tab.

Subsequent configuration properties use the following convention to refer to a unique service instance for each host.

`bc.appmanage.[interior].<host>.<index>.[ startonboot | enableverbose |  maxlogfilesize | maxlogfilecount | threadcount | prependclasspath | appendclasspath | minHeap | maxHeap | threadStack ] = <value>`

`[interior]`

`interior` is used for mapping engines that need to be deployed on the Interior Server.**`<host>`**

List of host names where the Interior server needs to be deployed.

`<index>`

The parameter `<host>` is the name of the machine binding and `<index>` refers to the instance on the same host. If there is only one instance on a machine, an `<index>` of `1` is required.

`myHostA.1, myHostA.2, myHostA.3`

`startonboot`

Boolean value which denotes Start when the machine on which the service instance is installed restarts.

`enableverbose`

Boolean value which denotes the enable verbose tracing

`maxlogfilesize`

Maximum Log File Size in KB

`maxlogfilecount`

Maximum number of log files to use

`threadcount`

Number of threads to use to execute the process instances

`prependclasspath`

Prepend to the `CLASSPATH` environment variable

`appendclasspath`:

Append to the `CLASSPATH` environment variable

`minHeap`

Initial Heap Size (MB)

`maxHeap`

Maximum Heap Size (MB)

`threadStack`:

Java Thread Stack Size (KB)

Examples:

```
bc.appmanage.interior.myHostA.1.minHeap=512 (Mbytes)
bc.appmanage.interior.myHostA.1.maxHeap=1024 (Mbytes)
bc.appmanage.interior.myHostA.2.minHeap=512 (Mbytes)
bc.appmanage.interior.myHostA.2.maxHeap=1024 (Mbytes)
bc.appmanage.interior.myHostA.2.startonboot=true
```

Optional Service Instance NT Service Startup Parameters

`bc.appmanage.[interior].<host>.<index>.ntservice = [automatic | manual].<user>.<pw>`

`[interior]`

`The parameter interior` is used for mapping engines that need to be deployed on the Interior Server.

`<host>`

List of host names where the Interior server needs to be deployed.

`<index>`

A numerical value that denotes the Interior server instances occurring within a host.

`[automatic | manual]`

NT Service start up type

`<user>`

Login user for the service

`<pw>`

Password for the service

Example:

```
bc.appmanage.interior.myHostX.1.ntservice =
automatic.myDomain\\myUserID.myPassword
```

Optional Interior Server Fault Tolerance Parameters

For more details about server fault tolerance parameters, see *TIBCO BusinessConnect Interior Server Administration,* Fault Tolerance Tab.

`bc.appmanage.ftGroupSettings.[service | network | daemonHost | daemonPort | heartbeatInterval | activationInterval | activationDelay]=[value]`

`service`

Service used by the fault tolerance daemon

`network`

Network used by the fault tolerance daemon

`daemonHost`

Daemon host

`daemonPort`

Daemon port

`hearbeatInterval`

Heart beat interval in seconds

`activationInterval`

Activation interval in seconds

`activationDelay`

Activation delay in seconds

Optional Oracle Client Path Parameter

`bc.appmanage.interior.<host>.<index>.oracleclientpath=[value]`

This property has can be updated using the TIBCO BusinessConnect Scripting Deployment.

`<host>`

List of host names where the Interior server needs to be deployed.

`<index>`

A numerical value that denotes the Interior server instances occurring within a host.

Examples:

The Oracle client path for configuring interior engines with an OCI driver can be set as shown below:

```
bc.appmanage.interior.host1.1.oracleclientpath=C:/Oracle1
```

```
bc.appmanage.interior.host2.1.oracleclientpath=/local/Oracle
```

Cluster/FT Map

The section Cluster/FT Map of the file `commonConfig.properties` contains the parameters needed to map fault-tolerant groups to service instances and to map Public Smart Routing clusters to fault-tolerant groups. Both parameter groups are optional. The following properties are used with the task `-configFtClusterMap` and are only applicable for Dual Firewall deployment.

Mapping Fault-Tolerant Groups to Service Instances

`bc.appmanage.ft.group.<groupname> = [<host>.<index>,]`

`<groupname>`

Fault-tolerant group name

`<host>`

Host machine running the service instance occurrence.

`<index>`

Required to denote the service instance occurrence on a host machine. Repeat `<host>.<index>` to assign multiple service instance to a fault-tolerant group.

Examples:

```
bc.appmanage.ft.group.myFTGroupA = myHostX.1, myHostY.2
bc.appmanage.ft.group.myFTGroupB = myHostY.1, myHostX.2
bc.appmanage.ft.group.myDefaultFT = myHostZ.1
```

Mapping Public Smart Routing Clusters to Fault-Tolerant Groups

`bc.appmanage.cluster.queue.[<queuename> | default]= [<groupname>,]`

`<queuename>`

Public Smart Route rule name. Repeat `<groupname>` to assign multiple fault-tolerant groups to a public Smart Routing cluster.

`<default>`

Denotes the `NO MATCHING RULES` rule name. If a service instance binding exists but is not present in these properties, it will automatically be assigned to a default fault-tolerant group named `<domain>.BCFTGROUP.DEFAULT`. All unassigned fault-tolerant groups with be automatically added to the default cluster queue when the default cluster queue property is not present.

Examples:

```
bc.appmanage.cluster.queue.myLargeEmailRule = myFTGroupA
bc.appmanage.cluster.queue.myLargeHTTPRequestRule = myFTGroupB
bc.appmanage.cluster.queue.default = myDefaultFT
```

Appendix A Command Reference

This appendix lists all configuration and deployment commands that you can invoke using TIBCO BusinessConnect.

Topics

- -undeployApp, page 58

-activateProtocol

Synopsis

```
bcappmanage -activateProtocol
```

Description

This command activates the specified TIBCO BusinessConnect protocol, which involves the creation of database tables and other protocol specific initialization steps. If ALL is specified, all installed protocols will be activated. In case of the base EDI protocol, it checks whether it is enabled before the HIPAA protocol can be activated. This task is equivalent to the activate protocol task under **BusinessConnect > Manage** Installation in the TIBCO BusinessConnect GUI.

Options

```
-protocol [ tibEDI | tibHIPAA | B2BPlugin | RosettaNet | SOAP |
ebXML | cXML | TCM | CMI | EZComm | FTPS | SFTP | PX | XCFI | ALL ]
```

Protocol plugin to activate

```
{ -tmpdir  <temp directory> }
```

Temporary directory used to un-war the protocol plugin.

(default= Java system property - `java.io.tmpdir`)

```
{ -configRepoURL [<props_file_url> | <jdbc_url>] }
```

Optional URL to property file specifying the configuration repository connection parameters or a standard JDBC URL.

If specifying a property file URL, the syntax is `file://<file location>, ex file://c:/temp/config.properties.`

(default=`file://./commonConfig.properties`)

```
{ -configRepoUser <user> -configRepoPw <password> }
```

Optional configuration repository user and password. It must be specified if the `configRepoURL` parameter is a JDBC URL, or if they are not specified in the connection properties file.

Example:

- Activates all installed protocols.

```
bcappmanage -activateProtocol -protocol ALL -configRepoURL
file://c:/temp/config.properties
```

- Activates the EDI protocol.

```
bcappmanage -activateProtocol -protocol tibEDI -configRepoURL
file://c:/temp/config.properties
```

Dependencies

```
-createInstallation
```

-createApp

Synopsis

```
bcappmanage -createApp
```

Description

This task is equivalent to the step Create Deployment Configuration invoked from the TIBCO BusinessConnect GUI, which creates the TIBCO BusinessConnect application that can be managed from TIBCO Administrator.

Options

`-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]`

administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

`{ -configRepoURL [<props_file_url> | <jdbc_url>] }`

Optional URL to property file specifying configuration repository connection parameters or a standard JDBC URL

If specifying a property file URL, the syntax is file://<file location>, ex file://c:/temp/config.properties.

(default = `file://./commonConfig.properties`)

`{ -configRepoUser <user> -configRepoPw <password> }`

Optional configuration repository user and password. It must be specified if the `configRepoURL` parameter is a JDBC URL, or if they are not specified in the connection properties file

Example

- Create the Single Server deployment configuration for the administrative domain `default-lt` using the configuration connection parameters from a Java properties file other than default.

```
bcappmanage -createApp -domain default-lt -cred
c:/temp/cred.properties -configRepoURL
file://c:/temp/config.properties
```

Dependencies

`-importConfigRepo`

-extractConfigData

Synopsis

```
bcappmanage -extractConfigData
```

Description

This command is used for extracting configuration data of participants, agreements, transports, operations and the associated bindings from TIBCO BusinessConnect configuration store. The extracted data is stored in the new tables and can be queried for reporting when required. For more details, see *TIBCO BusinessConnect Interior Server Administration,* Configuration Store Reporting Schema Details topic.

Only the TIBCO BusinessConnect Super User can perform this task. To learn more about TIBCO BusinessConnect Super Users and how they are created, see *TIBCO BusinessConnect Concepts*, TIBCO BusinessConnect Super User, and *TIBCO BusinessConnect Trading Partner Administration Guide*, Managing Groups with TIBCO BusinessConnect User Management.

> Only the CSX files belonging to TIBCO BusinessConnect 5.3.3 and later are supported.

It is required that TIBCO Administrator application is running while performing this task.

Options

```
-domain <domain name> [-user <domain user> -pw <domain password> |
-cred <domain cred_file>]
```

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

```
{ -configRepoURL [<props_file_url> | <jdbc_url>] }
```

Optional URL to property file specifying configuration repository connection parameters or a standard JDBC URL.

If specifying a property file URL, the syntax is file://*file location*, such as `file://c:/temp/config.properties`.

(default=`file://./commonConfig.properties`)

```
{ -configRepoUser <user> -configRepoPw <password> }
```

Optional configuration repository user and password. It must be specified if the `configRepo`URL parameter is a JDBC URL, or if they are not specified in the connection properties file.

Example

- Extract structured configuration data from unstructured configuration data for Jasper Reporting in TIBCO BusinessConnect.

```
bcappmanage -extractConfigData -domain default-lt -user joe -pw
changeme -configRepoURL file://c:/temp/config.properties
```

Dependencies

--createInstallation

-exportConfigRepo

-importConfigRepo

-activateProtocol (if any)

-createTables (optional)

-createGatewayToken

Synopsis

```
bcappmanage -createGatewayToken
```

Description

This command is used to create a Gateway Server token. The Gateway Server container reads a Gateway Server token to start establishing a secure trusted connection with the Interior Server.

The Gateway Server token allows for a secure configuration of a management port and the DDTP port. It defines the communication of the Gateway Server: the default settings given in General Settings are overwritten by the data in the Gateway Server token. It also defines policy control and validates policies upon startup.

Options

- configRepoURL

The TIBCO BusinessConnect repository URL, mandatory

- configRepoUser

The TIBCO BusinessConnect repository user, mandatory

- **configRepoPw**

The TIBCO BusinessConnect repository password, mandatory

- name

Token name, optional

- desc

Optional description of the token

- managementPort

Optional management port for monitoring, defaults to 11000.

- dataPort

Optional data port used for streaming data transfer, defaults to 12000.

- publicBinding

Optional public binding address set in the token for transport listeners.

- systemBinding

Optional system binding address set in the token for internal management and streaming data transfer.

- jmsJndiUrl

Optional URL set in the token for communication with the EMS Server from the Gateway Server.

-exportPath

Optional export file path. The default is `./securetoken.dat`.

- overwrite

Optional flag indicating to overwrite an existing token file.

Example:

Create a Gateway Token for the TIBCO BusinessConnect configuration repository URL jdbc:oracle:thin:@host:1521:orcl and the user Joe. The password for this URL and user is `changeme`, the token name is `token1`, and the rest of the data is also entered for the management port (13000), data port (14000) and for the export path of the token (c:\temp\token1.dat).

```
bcappmanage -createGatewayToken -configRepoURL
jdbc:oracle:thin:@10.100.100.10:1521:orcl -configRepoUser joe
-configRepoPw changeme -name token1 -managementPort 13000
-dataPort 14000 -exportPath c:/temp/token1.dat
```

-assignService

Synopsis

```
bcappmanage -assignService
```

Description

This command is used to delete the predefined service or to assign a service to a Gateway Instance. This task is equivalent to perform under **BusinessConnect > Gateway > Gateway Instances** in TIBCO Administrator.

Options

`-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]`

Administration domain name and credentials. Either user/password or credential file location must be specified.

`{ -configRepoURL [<props_file_url> | <jdbc_url>] }`

Optional URL to property file specifying configuration repository connection parameters or a standard JDBC URL.

If specifying a property file URL, the syntax is file://<file location>; for example, `file://c:/temp/config.properties`.

The default value is `file://./commonConfig.properties`.

`{ -configRepoUser <user> -configRepoPw <password> }`

Optional configuration repository user and password. It must be specified if the `configRepoURL` parameter is a JDBC URL, or if they are not specified in the connection properties file.

`{ -prefix <installation prefix> }`

TIBCO BusinessConnect installation prefix. The default value is `AX.BC`.

`{ -instance <gateway server instance> }`

Optional BC Gateway Server Instance name.

`{ -transport <gateway server transport> }`

Optional BC Gateway Server transport name. If you want to assign a service to a Gateway Server Instance, you have to set the service transport name option.

`{ -service <gateway server service> }`

Optional BC Gateway Server service name. If you want to assign a service to a Gateway Server Instance, you have to set this option.

Example:

The following examples are listed when using the default settings.

- Specify credential file for administrative domain authorization and a file URL for the configuration repository connection properties.

  ```
  bcappmanage -assignService -domain default-lt -cred
  c:/temp/cred.properties -configRepoURL
  file://c:/temp/config.properties
  ```

- Specify user ID/password for administrative domain authorization, and a JDBC URL and `userid/password` for the configuration repository connection properties.

  ```
  bcappmanage -assignService -domain default-lt -user joe -pw
  changeme
  ```

  ```
  -configRepoURL jdbc:oracle:thin://host:1521;SID=bcdb
  -configRepoUser joe -configRepoPw changeme
  ```

-startTransports

Synopsis

```
bcappmanage -startTransports
```

Description

This command is used to start all available transports for a Gateway Server. This task is equivalent to perform under **BusinessConnect > Gateway > Gateway Instances** in TIBCO Administrator.

Options

`-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]`

administration domain name and credentials. Either user/password or credential file location should be specified.

`{ -configRepoURL [<props_file_url> | <jdbc_url>] }`

Optional URL to property file specifying configuration repository connection parameters or a standard JDBC URL.

If specifying a property file URL, the syntax is file://<file location>, ex `file://c:/temp/config.properties`.

(default = `file://./commonConfig.properties`)

`{ -configRepoUser <user> -configRepoPw <password> }`

Optional configuration repository user and password. It must be specified if the `configRepoURL` parameter is a JDBC URL, or if they are not specified in the connection properties file.

`{ -prefix <installation prefix> }`

TIBCO BusinessConnect installation prefix. The default value is `AX.BC`.

`{ -instance <gateway server instance> }`

Optional BC Gateway Server Instance name.

Example:

- Specify credential file for administrative domain authorization and a file URL for the configuration repository connection properties.

  bcappmanage -startTransports -domain default-lt -cred **`c:/temp/cred.properties -configRepoURL file://c:/temp/config.properties`**

- Specify user ID/password for administrative domain authorization, and a JDBC URL and `userid/password` for the configuration repository connection properties.

  bcappmanage -startTransports -domain default-lt -user joe -pw **`changeme`**

```
-configRepoURL jdbc:oracle:thin://host:1521;SID=bcdb
-configRepoUser joe -configRepoPw changeme
```

-stopTransports

Synopsis

```
bcappmanage -stopTransports
```

Description

This command is used to stop all available transports for a Gateway Server. This task is equivalent to perform under **BusinessConnect > Gateway > Gateway Instances** in TIBCO Administrator.

Options

-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]

administration domain name and credentials. Either user/password or credential file location should be specified.

{ -configRepoURL [<props_file_url> | <jdbc_url>] }

Optional URL to property file specifying configuration repository connection parameters or a standard JDBC URL.

If specifying a property file URL, the syntax is file://<file location>, ex `file://c:/temp/config.properties`.

(default = `file://./commonConfig.properties`)

{ -configRepoUser <user> -configRepoPw <password> }

Optional configuration repository user and password. It must be specified if the `configRepoURL` parameter is a JDBC URL, or if they are not specified in the connection properties file.

{ -prefix <installation prefix> }

TIBCO BusinessConnect installation prefix. The default value is `AX.BC`.

{ -instance <gateway server instance> }

Optional BC Gateway Server Instance name.

Example:

- Specify credential file for administrative domain authorization and a file URL for the configuration repository connection properties.

  ```
  bcappmanage -stopTransports -domain default-lt -cred
  c:/temp/cred.properties -configRepoURL
  file://c:/temp/config.properties
  ```

- Specify user ID/password for administrative domain authorization, and a JDBC URL and `userid/password` for the configuration repository connection properties.

```
bcappmanage -stopTransports -domain default-lt -user joe -pw
changeme

-configRepoURL jdbc:oracle:thin://host:1521;SID=bcdb
-configRepoUser joe -configRepoPw changeme
```

-createInstallation

Synopsis `bcappmanage -createInstallation`

Description The command -createInstallation creates a new TIBCO BusinessConnect installation with option to disallow overwriting. This step initializes the TIBCO BusinessConnect configuration store and is the functional equivalent of the step Initialize Database performed by TIBCO Administrator GUI.

It creates the TIBCO BusinessConnect configuration schema in the database account specified by the argument `-configRepoURL`, where the configuration related tables are initialized with bootstrap data. This task also creates the audit, runtime and non-repudiation schemas in the same database account, and initializes the configuration store JDBC parameters used by TIBCO Administrator in the **BusinessConnect > Manage**.

Only the TIBCO Administrator Super User and TIBCO BusinessConnect Super User can perform the task `-createInstallation`.

It is required that TIBCO Administrator application is running while performing this task.

Options **`-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]`**

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

`{ -configRepoURL [<props_file_url> | <jdbc_url>] }`

Optional URL to property file specifying configuration repository connection parameters or a standard JDBC URL.

If specifying a property file URL, the syntax is file://<file location>; for example, `file://c:/temp/config.properties`. (default=`file://./commonConfig.properties`)

`{ -configRepoUser <user> -configRepoPw <password> }`

Optional configuration repository user and password. Must be specified if the `configRepoURL` parameter is a JDBC URL, or if they are not specified in the connection properties file

`{ -prefix <installation prefix> }`

The TIBCO BusinessConnect installation prefix (default=`AX.BC`)

`{ -overwrite }`

Optional flag to enable overwriting if the database account has an existing TIBCO BusinessConnect installation. By default existing TIBCO BusinessConnect installation is not overwritten.

Examples

- Specifying credential file for admin domain authorization and a file URL for the configuration repository connection properties.

```
bcappmanage -createInstallation -domain default-lt -cred
c:/temp/cred.properties -configRepoURL
file://c:/temp/config.properties
```

- Specifying user ID/password for administrative domain authorization and a JDBC URL and `userid/password` for the configuration repository connection properties.

```
bcappmanage -createInstallation -domain default-lt -user joe -pw
changeme -configRepoURL jdbc:oracle:thin:@10.100.100.10:1521:orcl
-configRepoUser joe -configRepoPw changeme
```

-createTables

Synopsis

```
bcappmanage -createTables
```

Description

This command is used to create audit, runtime, and non-repudiation schemas in database accounts that are different from the ones in the TIBCO BusinessConnect configuration store. This is accomplished by specifying the category (audit/non-repudiation/runtime) as well as the JDBC connection parameters for TIBCO BusinessConnect configuration store.

This task is optional and is not needed when the audit, non-repudiation, and runtime data will be stored in the database account that is the same as the account of the configuration store.

Options

`-category [ audit | nr | runtime ]`

Table category to drop and recreate.

`{ -jdbcURL [<props_file_url> | <jdbc_url>] }`

Optional URL to a property file specifying database tables connection parameters or a standard JDBC URL.

If specifying a property file url the syntax is `file://<file location>`; for example, `file://c:/temp/audit.properties.`

(default=`file://./commonConfig.properties)`

`{ -jdbcUser <user> -jdbcPw <password> }`

Optional database tables connection user and password. It must be specified if the JDBC URL parameter is a JDBC URL, or if they are not specified in the connection properties file.

`{ -configRepoURL [<props_file_url> | <jdbc_url>] }`

Optional URL to property file specifying configuration repository connection parameters or a standard JDBC URL

If specifying a property file URL, the syntax is file://<file location>; for example, `file://c:/temp/config.properties.`

(default=`file://./commonConfig.properties`)

`{ -configRepoUser <user> -configRepoPw <password> }`

Optional configuration repository user and password. It must be specified if the `configRepoURL` parameter is a JDBC URL, or if they are not specified in the connection properties file

`{ -overwrite }`

Optional flag to enable overwriting if the database account has existing audit, non-repudiation, or runtime data. By default existing data is not overwritten.

Examples

- Recreates and drops audit tables by specifying the audit connect entries via a properties file other than default.

```
bcappmanage -createTables -category audit -configRepoURL
file://c:/temp/config.properties -jdbcURL
file://c:/temp/audit.properties
```

Dependencies `-createInstallation`

-configFTClusterMap

Synopsis

```
bcappmanage -configFTClusterMap
```

Description

This task is optional and is used for changing the default fault-tolerant assignments for interior engines, as well as for mapping the Public Smart Routing cluster queues to fault-tolerant groups.

Clusters for fault-tolerant groups and for Smart Routing are mapped as follows:

- Fault-tolerant groups to machine bindings
- Public Smart Routing cluster queues to fault-tolerant groups

The mappings are specified via a Java properties file or a commonConfig.properties file located under *BC_HOME*`/tools/bcappmanage`.

> The command `configFTClusterMap` can work only with the service instances bindings that were previously configured using the command `-importDeployConfig`. See step 7.

For more in formation, see Cluster/FT Map.

Options

`-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]`

administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

`{-map <ft_cluster_map> }`

location to properties file hosting fault tolerance and smart routing cluster map configuration (default=./commonConfig.properties)

Example

- Perform cluster mapping using a Java properties file `C:/temp/cluster.properties` other than default.

```
bcappmanage -configFTClusterMap -domain default-lt -cred
c:/temp/cred.properties -map C:/temp/cluster.properties
```

Dependencies

`-importDeployConfig`

(service instances bindings must be previously configured using the command `-importDeployConfig`)

-deleteApp

Synopsis

```
bcappmanage -deleteApp
```

Description

Deletes deployment configuration for a TIBCO BusinessConnect application service. This task is equivalent to deleting the TIBCO BusinessConnect application from TIBCO Administrator after undeploying the application.

⚠ Undeploy the application service before invoking this task.

Options

```
-domain <domain name> [-user <domain user> -pw <domain password> |
-cred <domain cred_file>]
```

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

Example

- Deletes the TIBCO BusinessConnect application using the credential file `C:/temp/cred.properties`.

```
bcappmanage -deleteApp -domain default-lt -cred
c:/temp/cred.properties
```

-deployApp

Synopsis `bcappmanage -deployApp`

Description Deploys the configured TIBCO BusinessConnect application with an option to suspend starting of all service instances.

Options **`-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]`**

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

`{ -configRepoURL [<props_file_url> | <jdbc_url>] }`

Optional URL to property file specifying configuration repository connection parameters or a standard JDBC URL.

If specifying a property file URL, the syntax is `file://<file location>`; for example, `file://c:/temp/config.properties`.

(default=`file://./commonConfig.properties`)

`{ -configRepoUser <user> -configRepoPw <password> }`

Optional configuration repository user and password. It must be specified if the `configRepoURL` parameter is a JDBC URL, or if they are not specified in the connection properties file.

`{ -nostart }`

Optional argument. If it is specified, do not start application service instances on deploy.

Examples

- Deploys TIBCO BusinessConnect application service without starting the service instances.

```
bcappmanage -deployApp -domain default-lt -cred
c:/temp/cred.properties -nostart
```

Dependencies **`-importDeployConfig`**

`-configureFTClusterMap` (optional)

-exportConfigRepo

Synopsis **`bcappmanage -exportConfigRepo`**

Description The command `-exportConfigRepo` is used for exporting CSX component files from the TIBCO BusinessConnect configuration store. It supports four main categories: participants, operations, certificates, and entire installation. It typically exports all operations or participants, but it can be used also to export only selected ones. For example, if you have to export a participant Company2 into production environment, this command coupled with `-importConfigRepo` will allow you to add the participant without making any changes to the GUI.

When exporting the configuration repository, target files are saved in the CSX format, and configuration repositories are created for the following:

- Participants and agreements
- Operations and bindings
- Key and certificate credentials
- System and server settings

Exporting is performed as follows:

- For participants, operations, and certificates: a differential export is possible.
- For participants and certificates: a comma-separated list of entities to be exported can be specified.
- For operations: the XPath of the operation tree to be exported can be specified in addition to the protocol standard, which is mandatory for this category.
- In all these cases the exported CSX file can be optionally protected with a password.
- For certificate export: each exported certificate is written out as a file `<cert-alias>.cert` in the folder specified by the `-p7b` argument.
- The default export directory is the `p7b_certs` folder in the current working directory.

Options **`-csx <config_export>`**

Configuration repository export (csx) file to import/export . Not applicable for certificates export.

`{ -pw <password> }`

Optional password to encrypt/decrypt configuration repository import/export (csx). Not applicable for certificates export.

{-overwrite}

Overwrites existing configuration repository export (csx) file if specified. Not applicable for certificates export.

{ -category [installation | participants | operations | certificates] }

Export category type (default = `installation`)

{ -protocol [EZComm | X12 | EDIFACT | TEXT | Gateway | Service | RosettaNet | SOAP | ebXML | cXML] }

Protocol standard for operation export applicable to category = `operations` only

{ -op <delta_pattern> }

The operation sub-branch to be exported for category = `operation`, if not specified the entire operation tree is exported; for example, `-op` /00401 for protocol = X12

{ -alias <delta list> }

A comma separated list of participants for category = `participants` or aliases for category = `certificates`, if not specified the entire set of participants or certificates is exported

{ -tp <trading partner> }

The trading partner name for category = `certificates`

{ -p7b <p7b cert dir> }

The export directory for category = `certificates`

{ -configRepoURL [<props_file_url> | <jdbc_url>] }

Optional URL to property file specifying configuration repository connection parameters or a standard JDBC URL. If specifying a property file URL, the syntax is `file://<file location>`; for example, `file://c:/temp/config.properties.` `(default=file://./commonConfig.properties)`

{ -configRepoUser <user> -configRepoPw <password> }

Optional configuration repository user and password. It must be specified if the `configRepoURL` parameter is a JDBC URL or if they are not specified in the connection properties file.

Examples

- Exporting two participants

```
bcappmanage -exportConfigRepo -category participants -alias
tantony-lt,tantony-dt -configRepoURL
file://c:/temp/config.properties -csx c:/temp/participants-1.csx
```

- Exporting all participants

```
bcappmanage -exportConfigRepo -category participants
-configRepoURL file://c:/temp/config.properties -csx
c:/temp/participants-1.csx
```

- Exporting certificate with alias cert1 for partner `tantony-dt` into folder `c:/temp` (as file `cert1.cert`)

```
bcappmanage -exportConfigRepo -category certificates -alias cert1
-tp tantony-dt -configRepoURL file://c:/temp/config.properties
-p7b c:/temp
```

- Exporting the operation branch /00401 for protocol standard X12

```
bcappmanage -exportConfigRepo -category operations -protocol X12
-op /00401 -configRepoURL file://c:/temp/config.properties -csx
c:/temp/operations-401.csx
```

- Exporting the entire operation tree for protocol standard X12

```
bcappmanage -exportConfigRepo -category operations -protocol X12
-configRepoURL file://c:/temp/config.properties -csx
c:/temp/operations-401.csx
```

-exportDeployConfig

Synopsis

```
bcappmanage -exportDeployConfig
```

Description

Exports the deployment configuration into an XML file for future scripting deployment. Any future import of this configuration will contain all public and private process communication parameters specific to TIBCO BusinessConnect application. By default, the output deployment configuration XML file is set as `BC-<domain>-deploy-config.xml` and resides in the directory *BC_HOME*`/tools/bcappmanage/domain/<domain>`, if not specified otherwise.

Options

```
-domain <domain name> [-user <domain user> -pw <domain password> |
-cred <domain cred_file>]
```

Administration domain name and credentials. Either `user/password` or credential file location should be specified. See Credentials File.

```
{-out <deploy_config_xml> }
```

File path to export the TIBCO BusinessConnect application configuration into (`default = BC-<domain>-deploy-config.xml`)

```
{-overwrite}
```

Overwrites existing deployment configuration XML file if specified

In addition to an user ID and password, TIBCO BusinessConnect also supports the TIBCO Runtime Agent credentials file, which can hold an obfuscated (encrypted) password created by the `obfuscate` utility. If you do not want to use the credentials file to store login and password, you can supply them directly on the command line.

Credentials File

To read more about the credentials file, see *TIBCO Runtime Agent Scripting Deployment User's Guide*, section Working with Passwords.

Example

- Exports the deployment configuration for the administration domain `default-dt` and the credential file located in `C:\temp\cred.properties`, and saves it to the file `C:\temp\config.xml`.

```
bcappmanage -exportDeployConfig -domain default-lt -cred
c:/temp/cred.properties -out c:/temp/config.xml
```

-importConfigRepo

Synopsis

```
bcappmanage -importConfigRepo
```

Description

This command is used for importing CSX component files into the TIBCO BusinessConnect configuration store by specifying the location of the CSX file and the password, in case the file was password protected. It imports the file that was previously exported using step 2.

Only the TIBCO BusinessConnect Super User can perform this task. To learn more about TIBCO BusinessConnect Super Users and how they are created, see *TIBCO BusinessConnect Concepts*, TIBCO BusinessConnect Super User, and *TIBCO BusinessConnect Trading Partner Administration Guide*, Managing Groups with TIBCO BusinessConnect User Management.

Only the CSX files belonging to TIBCO BusinessConnect 5.3.3 and later are supported.

When `importConfigRepo` is executed, the installation name created during the task `createInstallation` will be replaced with the installation name from the imported CSX file.

It is required that TIBCO Administrator application is running while performing this task.

Options

`-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]`

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

`-csx <config_export>`

Configuration repository export (csx) file to import/export

`{ -csxpw <password> }`

Optional password to encrypt/decrypt the configuration repository import or export (csx).

`{ -configRepoURL [<props_file_url> | <jdbc_url>] }`

Optional URL to property file specifying the configuration repository connection parameters or a standard JDBC URL. If specifying a property file URL, the syntax is `file://`*file location*, such as `file://c:/te/config.properties`. (default=`file://./commonConfig.properties`)

`{ -configRepoUser <user> -configRepoPw <password> }`

Optional configuration repository user and password. It must be specified if the `configRepoURL` parameter is a JDBC URL, or if they are not specified in the connection properties file.

Example

- Import the configuration repository that was previously exported into the file `C:\tempmisc\operations-x12-test.csx.`

```
bcappmanage -importConfigRepo -domain default-lt -cred
c:/temp/cred.properties-configRepoURL
file://c:/temp/config.properties -csx
c:/temp/misc/operations-x12-test.csx
```

Dependencies

`-createInstallation`

`-exportConfigRepo`

`-activateProtocol` (if any)

`-createTables` (optional)

-importDeployConfig

The task `-importDeployConfig` allows only import of the same type of XML deployment configuration; for example, the XML configuration exported from a Single Server deployment mode can be only imported on a Single Server deployment configuration.

Synopsis

`bcappmanage -importDeployConfig`

Description

This command is used for configuring the Interior server engine instances and for specifying the attributes of the deployed engines, such as Java heap, stack size settings, and NT service settings.

Input the bindings using the following:

- The Java properties file `commonConfig.properties`, whose location is a command line argument for this task or *BC_HOME*`/tools/bcappmanage`
- Configuration file that was exported using the command `-exportDeployConfig`. By default, the input XML file for the task `-importDeployConfig` is set as `BC-<domain>-deploy-config.xml`, which resides under *BC_HOME*`/tools/bcappmanage/domain/<domain>`

While exporting the deployment configuration, you do not need to specify any machines names. Therefore, the step invoked by the command `-importDeployConfig` is used to define to the exact machines that will be used for the deployment and to configure their bindings. To learn more, see Binding Maps.

Names of the Interior hosts must be specified exactly as they are shown on the screen **Resource Management > Machines** in TIBCO Administrator. Same needs to be ensured for all property settings that use the machine names in their key. See Binding Maps for more information.

Options

`-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]`

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

`{ -config <deploy_config_xml> }`

File path to the exported TIBCO BusinessConnect application configuration (default=`BC-<domain>-deploy-config.xml`)

`{ -bindings <machine_binding_map> }`

Location to properties file hosting the Interior server host definitions (default=`./commonConfig.properties`)

Example

- Import the deployment configuration for the administrative domain `default-lt` from the file `C:\temp\config.xml`.

```
bcappmanage -importDeployConfig -domain default-lt -cred
c:/temp/cred.properties -config c:/temp/config.xml -bindings
C:/temp/bindings.properties
```

Dependencies

`-exportDeployConfig`

`-createApp`

-killApp

Synopsis

```
bcappmanage -killApp
```

Description

Kills (forcefully stops) TIBCO BusinessConnect application service instance.

This task can be used for killing the TIBCO BusinessConnect engines deployed in scripted mode.

The command `killApp` can work only with the service instances bindings that were previously configured using the command `-importDeployConfig`. See step 7.

Options

`-domain <domain name> [-user <domain user> -pw <domain password> | -cred <domain cred_file>]`

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

`{ -engine [interior.<host>.<index>] }`

Application service instance identifier; for example, `interior.host1.1`.

Example

- Kills all engines.

```
bcappmanage -killApp -domain default-lt -cred
c:/temp/cred.properties
```

- Kills interior `instance 1` on the host `Machine_X`.

```
bcappmanage -killApp -domain default-lt -cred
c:/temp/cred.properties -engine interior.Machine_X.1
```

-listGatewayToken

Synopsis

```
bcappmanage -listGatewayToken
```

Description

Lists properties of a Gateway token.

Options

`-filename`

Name of the token file (mandatory)

Example

- List properties of the Gateway Token file `token1.dat`.

```
bcappmanage -listGatewayToken -fileName c:/temp/token1.dat
```

-startApp

Synopsis

```
bcappmanage -startApp
```

Description

Starts a TIBCO BusinessConnect application service instance by service instance identifier on the Interior Server service.

The command `startApp` can work only with the service instances bindings that were previously configured using the command `-importDeployConfig`. See step 7.

Options

```
-domain <domain name> [-user <domain user> -pw <domain password> |
-cred <domain cred_file>]
```

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File

```
{ -engine [interior.<host>.<index>] }
```

Application service instance identifier: `interior.host1.1`

Example:

- Start all engines

```
bcappmanage -startApp -domain default-lt -cred
c:/temp/cred.properties
```

- Start interior instance 1 on host Machine_X

```
bcappmanage -startApp -domain default-lt -cred
c:/temp/cred.properties -engine interior.Machine_X.1
```

Dependencies

`-deployApp` (with the **`-nostart`** option)

-stopApp

Synopsis

```
bcappmanage -stopApp
```

Description

Stops a single TIBCO BusinessConnect application service instance by service instance identifier on the Interior Server.

This task can be used for stopping the TIBCO BusinessConnect engines deployed in scripted mode.

The command `stoppApp` can work only with the service instances bindings that were previously configured using the command `-importDeployConfig`. See step 7.

Options

```
-domain <domain name> [-user <domain user> -pw <domain password> |
-cred <domain cred_file>]
```

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

```
{ -engine [interior.<host>.<index>] }
```

Application service instance identifier; for example, `interior.host1.1`.

Example

- Stop all engines.

```
bcappmanage -stopApp -domain default-lt -cred
c:/temp/cred.properties
```

- Stop interior `instance 1` on host `Machine_X`.

```
bcappmanage -stopApp -domain default-lt -cred
c:/temp/cred.properties -engine interior.Machine_X.1
```

-undeployApp

Synopsis

```
bcappmanage -undeployApp
```

Description

Undeploys the configured TIBCO BusinessConnect application.

Options

```
-domain <domain name> [-user <domain user> -pw <domain password> |
-cred <domain cred_file>]
```

Administration domain name and credentials. Either user/password or credential file location should be specified. See Credentials File.

Example

- Undeploys the TIBCO BusinessConnect application using the credential file `C:/temp/cred.properties`.

```
bcappmanage -undeployApp -domain default-lt -cred
c:/temp/cred.properties
```

Appendix B Configuration Properties File

This appendix contains a sample file `commonConfig.properties`, which is installed by the application TIBCO BusinessConnect in the directory *BC_HOME*`/tools/bcappmanage`.

Topics

commonConfig.properties

The file `commonConfig.properties` is initially installed by the application TIBCO BusinessConnect in the directory *BC_HOME*`\tools\bcappmanage`. In this example, user information is entered in bold text.

More details about this file are available in Chapter 3, Parameters for commonConfig.properties.

Example 1 commonConfig.Properties

```
#
# Copyright (c) 2003-2015 TIBCO Software Inc.
# All Rights Reserved.
#
# This software is the confidential and proprietary information of
# TIBCO Software Inc.
#
#==========================================================
# *** Common variables ***
#==========================================================

#================================
# *** JDBC Maps ***
#
# Samples JDBC driver and URLs where <type> = <repo|audit|nr|runtime>:
#
# bc.<type>.db.driver = com.mysql.jdbc.Driver
# bc.<type>.db.url = jdbc:mysql://[host]:[port]/[database]
#
# bc.<type>.db.driver = oracle.jdbc.driver.OracleDriver [ deprecated by Oracle ]
# bc.<type>.db.driver = oracle.jdbc.OracleDriver [ new Driver class introduced by 
Oracle from ojdbc5.jar ]
# bc.<type>.db.url = jdbc:oracle:thin:@[host]:[port]:[sid]
#
#
#
# Native Vendor Driver support
# ============================
#
# Oracle OCI
# ----------
# bc.<type>.db.driver = oracle.jdbc.OracleDriver
# bc.<type>.db.url = jdbc:oracle:oci:@<Service Name>
#
# Microsoft SQL Server driver
# ---------------------------
# bc.<type>.db.driver =  com.microsoft.sqlserver.jdbc.SQLServerDriver
# bc.<type>.db.url    = jdbc:sqlserver://<host>:<port#>;databaseName=<databaseName>
#
# DB2 Universal Driver
# ---------------------
# bc.<type>.db.driver = com.ibm.db2.jcc.DB2Driver
# bc.<type>.db.url    = jdbc:db2://<host>:<port#>/<databaseName>
#================================

# Repository connection prameters
#
bc.repo.db.driver=com.mysql.jdbc.Driver
bc.repo.db.url=jdbc:mysql://axw3k1:3306/john
bc.repo.db.userID=john1
bc.repo.db.password=john1
bc.repo.db.connectionPool.size=32

# Audit connection parameters
#
bc.audit.db.driver=com.mysql.jdbc.Driver
bc.audit.db.url=jdbc:mysql://axw3k1:3306/john
bc.audit.db.userID=john1
bc.audit.db.password=john1
bc.audit.db.connectionPool.size=

# NR connection parameters
#
bc.nr.db.driver=com.mysql.jdbc.Driver
bc.nr.db.url=jdbc:mysql://axw3k1:3306/john
bc.nr.db.userID=john1
bc.nr.db.password=john1
bc.nr.db.connectionPool.size=
#
# Runtime connection parameters
#
bc.runtime.db.driver=com.mysql.jdbc.Driver
bc.runtime.db.url=jdbc:mysql://axw3k1:3306/john
bc.runtime.db.userID=john1
bc.runtime.db.password=john1
bc.runtime.db.connectionPool.size=

#================================
# Bindings Map
#================================

#
# List of hosts where interior servers need to be deployed. Repeat the host name for
multiple interior server
# instances on the same host. Ex,
# interior.hosts=host1,host2,host3,host1
#
bc.appmanage.interior.hosts=

# (Optional)
# The min, max heap sizes, thread stack size and classpath extensions for the
interior engines can be specified(in MBs) as shown below.
# The first token in the key is interior, the next token refers to the host
# name, the next token specifies the instance occurence within that host (for
interiors)and the last token specifies
# whether the value is min / max heap size, thread stack size or append/prepend
classpath.
```

```
# Since this is a java properties file, prepend/append classpaths should always use
'/' as the path separator.
# Ex,
#
# bc.appmanage.interior.host1.1.minHeap=500
# bc.appmanage.interior.host1.1.maxHeap=1000
# bc.appmanage.interior.host1.1.threadStack=512
# bc.appmanage.interior.host1.1.appendclasspath=c:/temp
# bc.appmanage.interior.host1.1.prependclasspath=c:/temp2

# bc.appmanage.interior.host1.2.minHeap=800
# bc.appmanage.interior.host1.2.maxHeap=1200
# bc.appmanage.interior.host1.2.threadStack=1024

# (Optional)
# The general settings for deployed engines can be set as shown below.
# The first token in the key is interior, the next token refers to the host
# name, the next token specifies the instance occurence within that host (for
interiors)and the last token specifies
# the specific attribute being set which is one among - startonboot, enableverbose,
maxlogfilecount & maxlogfilesize.
# Ex,
#

# bc.appmanage.interior.host1.1.startonboot=true
# bc.appmanage.interior.host1.1.enableverbose=true
# bc.appmanage.interior.host1.1.threadcount=14

# (Optional)
# The NT service settings for deployed engines can be set as shown below.
# The first token in the key is interior, the next token refers to the host
# name, the next token specifies the instance occurence within that host (for
interiors) and the last token identifes
# this key as defining the NT service setting. The value has three tokens of which
the first one indicates the
# service startup type which can be either 'manual' or 'automatic'. The second and
third tokens specify the user id
# and password for the service's logon account. Note that the userid should
mandatorily specify the domain. The double
# backslashes are required, slash being part of the properties file syntax. To run
the service under the default local
# system account, leave the userid and password empty.
#
# bc.appmanage.interior.host1.1.ntservice=automatic.domain\\joe.changeme
# bc.appmanage.interior.host1.2.ntservice=manual.domain\\joe.changeme

# (Optional)
# Interior Server Fault Tolerance Group Settings can be set as shown below.
#
# Note: The properties for heartbeatIntervals, activationInterval and
activationDelay are specified in seconds.
#
# bc.appmanage.ftGroupSettings.service=7505
# bc.appmanage.ftGroupSettings.network=
# bc.appmanage.ftGroupSettings.daemonHost=
# bc.appmanage.ftGroupSettings.daemonPort=7505
# bc.appmanage.ftGroupSettings.heartbeatInterval=5
# bc.appmanage.ftGroupSettings.activationInterval=15
# bc.appmanage.ftGroupSettings.activationDelay=10

# (Optional)
# The Oracle client path for configuring interior engines with OCI driver can be set
as shown below.
# bc.appmanage.interior.host1.1.oracleclientpath=<Oracle Home Path1>
# bc.appmanage.interior.host2.1.oracleclientpath=<Oracle Home Path2>

#================================
# Cluster/FT Map
#================================

#
# Mapping of FT groups to interior engine instances. Key is the group name and value
is a comma separated list of
# interior engine instances under that group with the syntax <host.index>. Repeat
the entry for multiple
# FT groups. Ex
# bc.appmanage.ft.group.FTG1=host1.1,host2.1
# bc.appmanage.ft.group.FTG2=host3.1, host1.2
#
bc.appmanage.ft.group.<groupname>=

# (Optional)
# Mapping of public smart routing cluster queues to FT groups. Key is the cluster
queue name name and value is a
# comma separated list of FT groups asigned to that cluser. Repeat the entry for
multiple cluster queues. Ex,
# bc.appmanage.cluster.queue.Q1=FTG1
# bc.appmanage.cluster.queue.Q2=FTG2
#
#bc.appmanage.cluster.queue.<queuename>=

#
# A comma separated list of FT groups assigned to the default (NO MATCHING RULES)
cluster queue.
#
bc.appmanage.cluster.queue.default=
```

Index

TIBCO Product Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join the TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website, mainly in HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

The following documentation for this product is available on the TIBCO BusinessConnect page.

- *TIBCO BusinessConnect Release Notes*
- *TIBCO BusinessConnect Installation and Configuration*
- *TIBCO BusinessConnect Concepts*
- *TIBCO BusinessConnect Scripting Deployment User's Guide*
- *TIBCO BusinessConnect Interior Server Administration*
- *TIBCO BusinessConnect Trading Partner Administration*

Other TIBCO Product Documentation

When working with TIBCO BusinessConnect, you may find it useful to read the documentation of the following TIBCO products:

- TIBCO Administrator™: This software allows you to manage users, machines and applications defined in a TIBCO Administration Domain. The TIBCO Administrator graphical user interface enables users to deploy, monitor, and start and stop TIBCO applications.
- TIBCO ActiveMatrix BusinessWorks™: This software is a scalable, extensible, and easy to use integration platform that allows you to develop integration projects. TIBCO ActiveMatrix BusinessWorks includes a graphical user interface (GUI) for defining business processes and an engine that executes the process.
- TIBCO Designer™: This graphical user interface is used for designing and creating integration project configurations and building an Enterprise Archive

(EAR) for the project. The EAR can then be used by TIBCO Administrator for deploying and running the application.

- TIBCO Runtime Agent™: This software suite is a prerequisite for other TIBCO software products. In addition to TIBCO Runtime Agent components, the software suite includes the third-party libraries used by other TIBCO products such as TIBCO Designer, Java Runtime Environment (JRE), TIBCO Hawk®, and TIBCO Rendezvous®.
- TIBCO Rendezvous®: This software enables programs running on many different kinds of computers on a network to communicate seamlessly. It includes two main components: the Rendezvous programming language interface (API) in several languages, and the Rendezvous daemon.
- TIBCO Enterprise Message Service™: This software provides a message service that enables integration of applications within an enterprise based on the Java Message Service (JMS) specification. This software is a prerequisite for other TIBCO software products.
- TIBCO BusinessEvents®: This software helps companies identify and quantify the impact of events; it notifies people and systems about meaningful events so processes can be adapted on-the-fly. TIBCO BusinessEvents uses a unique model-driven approach to collect, filter, and correlate events and deliver real-time operational insight.
- TIBCO Hawk®: This software is a tool for monitoring and managing distributed applications and operating systems. The software is designed specifically for monitoring distributed systems, so there is no centralized console or frequent polling across the network.
- tibbr®, tibbr Service, tibbr Community, and tibbr Community Service: This software is the first workplace communication tool with which you can follow subjects that relate to your work and interests besides following people as you do in typical social networking applications. That way, you have much more flexibility in obtaining the right information at the right time in the right context. In fact, the information will find you.
- TIBCO BusinessConnect™ Palette: This software is about the resources available in the TIBCO BusinessConnect Palette for TIBCO ActiveMatrix BusinessWorks.

How to Contact TIBCO Support

Get an overview of TIBCO Support. You can contact TIBCO Support in the following ways:

- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support website.

- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to TIBCO Support website. If you do not have a user name, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the TIBCO Ideas Portal. For a free registration, visit TIBCO Community.

Legal and Third-Party Notices

This software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the readme file for the availability of this software version on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENT AT ANY TIME.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

This and other products of TIBCO Software Inc. may be covered by registered patents. Please refer to TIBCO's Virtual Patent Marking document (https://www.tibco.com/patents) for details.