Siebel Application Deployment Manager Guide

Version 8.0, Rev. A

April 2007

ORACLE

PRODUCT MODULES AND OPTIONS. This guide contains descriptions of modules that are optional and for which you may not have purchased a license. Siebel's Sample Database also includes data related to these optional modules. As a result, your software implementation may differ from descriptions in this guide. To find out more about the modules your organization has purchased, see your corporate purchasing agent or your Siebel sales representative.

Contents

Chapter 4: Configuring and Administering the ADM Framework

Chapter 5: Creating an ADM Package

Chapter 6: Deploying an ADM Package

Chapter 7: Guidelines for Deploying ADM

Chapter 8: ADM Logging Capabilities and Troubleshooting

Appendix A: ADM Supported Data Types

Appendix B: ADM Support for Additional Data Types

Index

1 What's New in This Release

What's New in Siebel Application Deployment Manager Guide, Version 8.0, Rev. A

Table 2 lists changes described in this version of the documentation to support release 8.0 of the software.

Table 1. What's New in Siebel Application Deployment Manager Guide, Version 8.0, Rev. A

Topic	Description
"About the Customizations Migration Process Using ADM" on page 27	Added language limitation for customizations deployment.
"About the ADM Command Flow" on page 31	New topic and diagram. Describes the path of ADM commands through the Siebel Enterprise Server.
"About the ADM Framework Installation" on page 35	Added more information on Application Deployment Manager (ADM) framework installation, including Java and Perl requirements.
"Configuring the Siebel Server for ADM" on page 36	Corrected the directory path in step 4.
"Configuring the Siebel Management Server and Siebel Management Agent after Installation" on page 39	Added the getserver.pl and makeagentconfig.pl Perl scripts as requirements.
"Configuring the ADM Enterprise Profile after Installation" on page 40	Added multiserver setup instructions.
"Deploying a Sample ADM Package" on page 44	Added a sample output of a successful sample package deployment.
"About ADM Configurations for New Data Types" on page 51	New topic. Describes ADM configurations for new data types.
"Creating Integration Objects for ADM" on page 52	This topic has been moved to "About ADM Configurations for New Data Types" on page 51.
"Creating Content Objects for ADM" on page 53	This topic has been moved to "About ADM Configurations for New Data Types" on page 51.
"Configuring ADM Parent Child Relationships on Integration Objects" on page 54	This topic has been moved to "About ADM Configurations for New Data Types" on page 51.

Table 1. What's New in Siebel Application Deployment Manager Guide, Version 8.0, Rev. A

Topic	Description
"Configuring the ADM Registry" on page 51	New topic. Describes how to configure the ADM registry.
"Restoring the ADM Registry" on page 52	New topic. Describes how to restore the ADM registry from backup.
"Server Components That Must Be Enabled And Online" on page 56	New topic. Describes server component requirements to run an ADM deployment.
"About the ADM Package Utility Modes and Switches" on page 60	Added a table for the ADM Package Utility return codes.
"Process of Creating an ADM Package" on page 63	Added new step 1.
"Reviewing the ADM Data Type Requirements" on page 63	New topic. Describes the importance of reviewing the data type requirements prior to creating ADM packages.
"Deploying Customizations Using the ADM Framework" on page 28, "Creating an ADM Package" on page 59, and "Deploying an ADM Package" on page 83	Added process flow diagrams to illustrate the high-level process of migrating customizations using ADM.
"Reviewing ADM Data Types" on page 86	New topic. Describes reviewing the ADM data types before deploying the ADM packages.
"Choosing to Use a Staged or Full Deployment" on page 87	New topic. Describes the differences between using a staged deployment and a full deployment.
"Validating an ADM Deployment Environment" on page 91	Added note about ADMProc server component.
"Creating a Destination Map for File Directories" on page 99	New topic. Describes how to create a report listing the files that are deployed to the target environment.
"Scenario for a Minor Functional Update Using ADM" on page 106	Revised packaging guidelines.
"Synchronizing Repository Updates" on page 111	New topic. Describes how to make sure the repository data in the source and target environments are synchronized.
"Integrating ADM with Change Management Systems" on page 111	New topic. Describes the integration of ADM and change management systems.
"About ADM Framework Logging Options" on page 113	Added information on the contents of each log file.
"About SRProc and SRBroker Log Files" on page 117	New topic. Describes the SRProc and SRBroker log files.

Table 1. What's New in Siebel Application Deployment Manager Guide, Version 8.0, Rev. A

Topic	Description
"Configuring ADM Log Files on Siebel Management Server and Siebel Management Agent" on page 118	Combined the two topics, Configuring ADM Log Files on Siebel Management Server and Configuring ADM Log Files on Siebel Management Agent.
"Reviewing ADM Log Files" on page 119	New topic. Describes the log files to review after installing the ADM framework and after executing an ADM deployment.
"Copy Command Error: Multiple Matches for Integration Component User Key" on page 121	New troubleshooting tip.
"ADM Data Type Details" on page 135	Added new steps 8 and 9 for the Webmaster data type.
Appendix B, "ADM Support for Additional Data Types"	New appendix. Describes additional ADM data type information to support adding new data types.

What's New in Siebel Application Deployment Manager Guide, Version 8.0

Siebel Application Deployment Manager Guide is a new title on the *Siebel Bookshelf* and replaces the former Application Deployment Manager (ADM) documentation contained in *Going Live with Siebel Business Applications*.

Table 2 lists chapter descriptions of the *Siebel Application Deployment Manager Guide* documentation to support release 8.0 of the software and highlights the new functionality included in Oracle's 8.0 Siebel Application Deployment Manager feature.

Table 2. What's New in Siebel Application Deployment Manager Guide, Version 8.0

Topic	Description
Chapter 2, "Overview of the ADM Architecture"	Describes the new ADM architecture.
Chapter 3, "Overview of the ADM Deployment Process"	Describes the new ADM deployment process and other deployment options.
Chapter 4, "Configuring and Administering the ADM Framework"	Describes the tasks necessary to configure and administer the ADM framework.
Chapter 5, "Creating an ADM Package"	Describes the creation of the ADM package using new utilities.

Table 2. What’s New in Siebel Application Deployment Manager Guide, Version 8.0

Topic	Description
Chapter 6, “Deploying an ADM Package”	Describes the ADM package deployment process that includes: ■ Support for the deployment of customized objects in the Siebel Repository ■ Support for the deployment of customized files to the Siebel Server (including the custom files destined for the Siebel Web server) ■ Expanded support for the deployment of run-time database customizations ■ Support for the backup, online activation, and restoring from a backup within the context of an ADM deployment package ■ New utilities to deploy the application customizations in concert with existing ADM functionality
Chapter 7, “Guidelines for Deploying ADM”	Describes some best practices when using the ADM feature and use scenarios.
Chapter 8, “ADM Logging Capabilities and Troubleshooting”	Describes the ADM logging and status-reporting capabilities, as well as options for troubleshooting ADM deployments.
Appendix A, “ADM Supported Data Types”	Lists and describes a subset of ADM supported data types.

2 Overview of the ADM Architecture

This chapter contains an overview of the Application Deployment Manager (ADM) architecture and information on various ADM entities categorized based on the environment that hosts them. It includes the following topics:

- "ADM Framework High-Level Overview" on page 13
- "About ADM, Siebel Management Server, and Diagnostic Tool" on page 15
- "About the ADM Source Environment" on page 15
- "About the ADM Client Environments" on page 16
- "About the ADM Orchestration Environment" on page 18
- "About the ADM Target Environment" on page 24

The new ADM framework works with the new Siebel infrastructure components, Siebel Management Server, and Siebel Management Agents. For more information on the Siebel Management Server and Siebel Management Agents and other documentation related to ADM, see:

- ***Siebel Installation Guide* for the operating system you are using.** This guide describes the installation of the Siebel Management Server and Siebel Management Agents, which are required for a functioning ADM framework.
- ***Siebel System Administration Guide*.** This guide describes the administrative tasks for the Siebel Management Server and Management Agents.
- ***Siebel System Monitoring and Diagnostics Guide*.** This guide describes the new Diagnostic Tool, which also uses Siebel Management Server and Management Agents. This tool is not required to use ADM.

ADM Framework High-Level Overview

The *Application Deployment Manager (ADM) framework* is the recommended support structure for migrating application customizations from a source environment to one or more target environments. To manage and execute the application customizations migration, the ADM framework includes an orchestration environment hosted within the Siebel Management Server. Each environment in the migration process contains ADM-specific entities involved in the migration. For a high-level overview of the ADM entities of the ADM framework, see the following environment-specific topics:

- **ADM Source Environment.** This topic describes the origin of the application customizations and the initial preparation of these customizations prior to deployment to a new environment. For more information, see "About the ADM Source Environment" on page 15.
- **ADM Client Environment.** This topic describes the various client interfaces used to control the ADM deployment process. For more information, see "About the ADM Client Environments" on page 16.

- **ADM Orchestration Environment.** This topic describes the ADM framework components that manage and execute the ADM deployment. For more information, see "About the ADM Orchestration Environment" on page 18.
- **ADM Target Environment.** This topic describes the target environment that receives the ADM-migrated customizations and the necessary ADM framework components that facilitate the migration on the target environment side. For more information, see "About the ADM Target Environment" on page 24.

See Figure 1 for a view of the ADM environments.

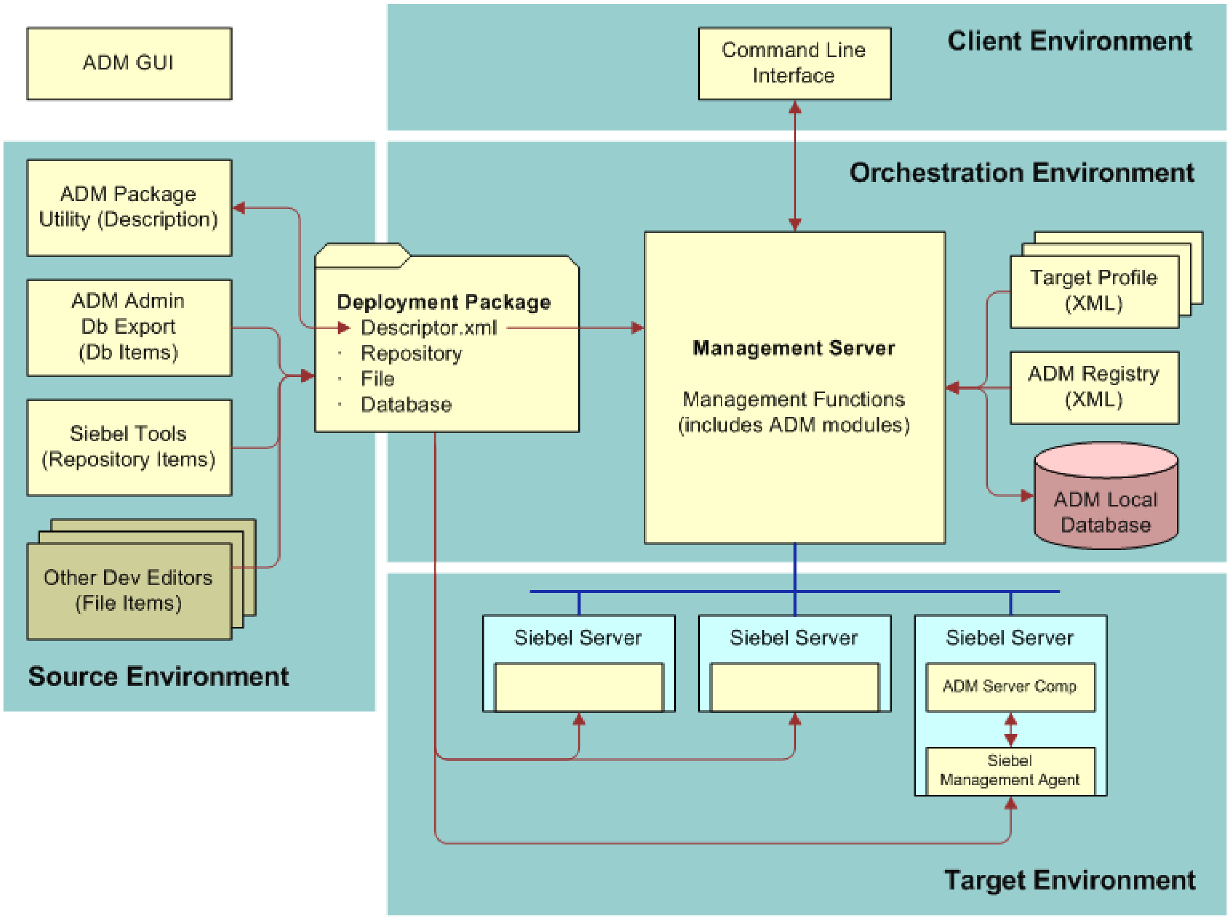


Figure 1. High-Level ADM Environment Overview

About ADM, Siebel Management Server, and Diagnostic Tool

The Siebel Management Server and Siebel Management Agents are Siebel Business Application infrastructure components used to host both the Application Deployment Manager (ADM) feature and the Siebel Diagnostic Tool. They provide a framework for the functionality of ADM and the Siebel Diagnostic Tool. Siebel Management Server and Siebel Management Agents are not unique or exclusive to either feature.

You can use ADM to target multiple Siebel Enterprise Servers either with the Siebel Diagnostic Tool installed or without. ADM does not require Diagnostic Tool to operate. The Siebel Management Server and Siebel Management Agent components that you install and use are based on your business requirements, for example, whether the environment is used for development, testing, or production.

At the installation of the Siebel Management Server, specific configurations are required at that time if you are using the ADM functionality. For more information on the installation of the Siebel Management Server and Siebel Management Agents, see "About the ADM Framework Installation" on page 35. For information about postinstallation configurations necessary for ADM, see "Process of Configuring the ADM Framework after Installing Siebel Management Server and Siebel Management Agent" on page 36. For more information on the Siebel Diagnostic Tool and its use of the Siebel Management Server, see *Siebel System Monitoring and Diagnostics Guide*.

About the ADM Source Environment

The Application Deployment Manager (ADM) source environment refers to the Siebel Business Application environment where the application customizations originate. This environment can be, for example, a development or test environment. The ADM framework automates the collection, packaging, and migration of application customizations from the source environment to the target environment. For example, one source environment can be used to create an ADM package for migration into multiple target environments.

ADM data types represent the different customizable entities in the source environment. For more information on ADM data types, see "About ADM Data Types" on page 15.

You begin the ADM deployment process by packaging ADM data types in the source environment. For information on this process, see "About the ADM Packaging Process" on page 30.

You can also use the source environment to deploy certain customizations outside the ADM framework. For more information on this deployment option, see "Other Options for Deploying Customizations Using ADM" on page 32.

About ADM Data Types

Application Deployment Manager (ADM) data types represent the customized entities that require migration to new environments. For example, an assignment rule, a List of Values, application responsibilities, or Web templates represent the type of configurations you make to one environment that must be migrated to another environment.

There are three general categories of ADM data types:

- **ADM Database Types.** Siebel run-time customizations residing in the Siebel database, for example, a Product Line or Assignment Rule.
- **ADM Repository Types.** Siebel objects residing in the Siebel Repository, for example, a Business Component.
- **ADM File Types.** Siebel Business Application files residing on the Siebel Server, Web server, or a supported third-party server, for example, a Web Template.

Some data types are hierarchical and contain more than one object; this occurs when the parent object is dependent on one or more child objects. For example, the Workflow Policy data type also migrates associated child Workflow Actions.

One or more ADM data types are consolidated into a deployment unit, which makes up an ADM package. You deploy the ADM package to migrate the application customizations.

The packaging and deployment of ADM data types are dependent on their category. Individual data types may also have other specific configurations, limitations, or dependencies on other ADM data types. For full details and a listing of all supported ADM data types, see Appendix A, “ADM Supported Data Types”. For more information on packaging and deploying different ADM data types, see Chapter 3, “Overview of the ADM Deployment Process.”

About the ADM Client Environments

The Application Deployment Manager (ADM) client environments include command-line interface programs and GUI applications that allow you to create and manage a deployment of application customizations between environments. Other traditional Siebel application interfaces are also available to assist with ADM deployments. The ADM client environments are:

- **ADM Package Utility.** This command-line utility creates and validates ADM packages containing application customizations. For more information on this utility, see “About the ADM Package Utility” on page 17.
- **ADM Command-Line Interface.** This interface controls the ADM deployment through the Siebel Management Server. For more information on this interface, see “About the ADM Command-Line Interface” on page 17.
- **Siebel Web Client ADM Screen.** The Siebel Web Client ADM screen assists in the creation of ADM deployment units for any database data types that make up an ADM package. It can also serve as a means of deploying ADM database types outside the new ADM framework (as used in ADM before Siebel 8.0). For more information on this interface, see “About the ADM Screen” on page 17.
- **Siebel Server Manager Command-Line Interface.** This command-line interface administers the Siebel Server. It can also create ADM deployment units for database data types used in ADM packages and can also deploy ADM database data types outside the ADM framework. For more information on:
  - Creating ADM deployment units using the Siebel Server Manager command-line interface, see “Creating ADM Deployment Units from Database Types Directly from Command-Line Interface” on page 74.

  - Deploying ADM database types outside the ADM framework, see "Deploying ADM Sessions Using Command-Line Interface" on page 102.
  - Using the Siebel Server command-line interface, see *Siebel System Administration Guide*.
- **Siebel Tools.** The Siebel Tools GUI creates deployment units of the repository ADM data types. For information on this task, see "Creating ADM Deployment Units from Repository Types" on page 75. For information on Siebel Tools, see *Using Siebel Tools*.

About the ADM Package Utility

The Application Deployment Manager (ADM) Package Utility creates and validates the ADM deployment package, which can then be used by the ADM framework to deploy Siebel application customizations to a target environment. It also creates the ADM package descriptor file that is used during the deployment process.

The ADM Package Utility is a stand-alone, Java-based program that is invoked through a batch file (admpkgr.bat) to execute various package commands at the command-line. By default, it is installed with the Siebel Management Server and Siebel Tools, and you can copy it to other Windows or UNIX machines. For more information on copying this utility to a UNIX environment, see "Copying the ADM Package Utility to a UNIX Environment" on page 62. For more information on the ADM Package Utility modes and a list of available switches, see "About the ADM Package Utility Modes and Switches" on page 60. For more information on the commands associated with the ADM Package Utility, see Chapter 5, "Creating an ADM Package."

About the ADM Command-Line Interface

The Application Deployment Manager (ADM) command line functions as the interface with the Siebel Management Server to execute the actual ADM deployment of application customizations and also manages any associated deployment tasks.

The ADM command-line interface is a Java-based tool executed by running a batch file with the appropriate commands and parameters. A command-line tool for the target Siebel Enterprise Server is configured during the installation to use a separate batch file for each Siebel Enterprise Server named: deploy_*enterprise_name*.bat. This batch file is located in the Siebel Management Server installation directory: `MgmtSrvrInstallDir\ADM`. For information on the ADM command-line syntax, see "About the ADM Command-Line Interface Syntax" on page 84. For more information on the commands associated with the ADM command-line interface, see Chapter 6, "Deploying an ADM Package."

About the ADM Screen

The Application Deployment Manager (ADM) GUI consists of a screen and views from the Siebel application site map. The screen name is Application Deployment Manager, and from this screen you create deployment units consisting only of ADM database data types.

Deployment units form the ADM package used to deploy application customizations. For more information on creating ADM database type deployment units using the ADM GUI, see Chapter 5, “Creating an ADM Package.”

The ADM GUI also facilitates the deployment of ADM database types without using the ADM framework, for backwards compatibility with ADM before Siebel 8.0. For more information on this deployment option, see “Other Options for Deploying Customizations Using ADM” on page 32.

The Siebel Tools GUI is also used in the ADM packaging process to create deployment units based on ADM repository data types. For information on this process, see “Creating ADM Deployment Units from Repository Types” on page 75.

About the ADM Orchestration Environment

The ADM orchestration environment is the environment where the actual ADM deployment is coordinated and executed on the target environment. For a high-level overview of this environment and its components, see Figure 2. The prime infrastructure entity that manages the ADM deployment is the Siebel Management Server. It interacts with various orchestration entities to deploy application customizations through Siebel Management Agents on the target environment’s Siebel Servers. For more information on the orchestration environment architecture, see:

- “About the Siebel Management Server and ADM Deployment Engine” on page 19
- “About the ADM Local Database” on page 21
- “About the ADM Registry” on page 22
- “About the ADM Enterprise Profile” on page 23

- "About the ADM Enterprise Profile" on page 23

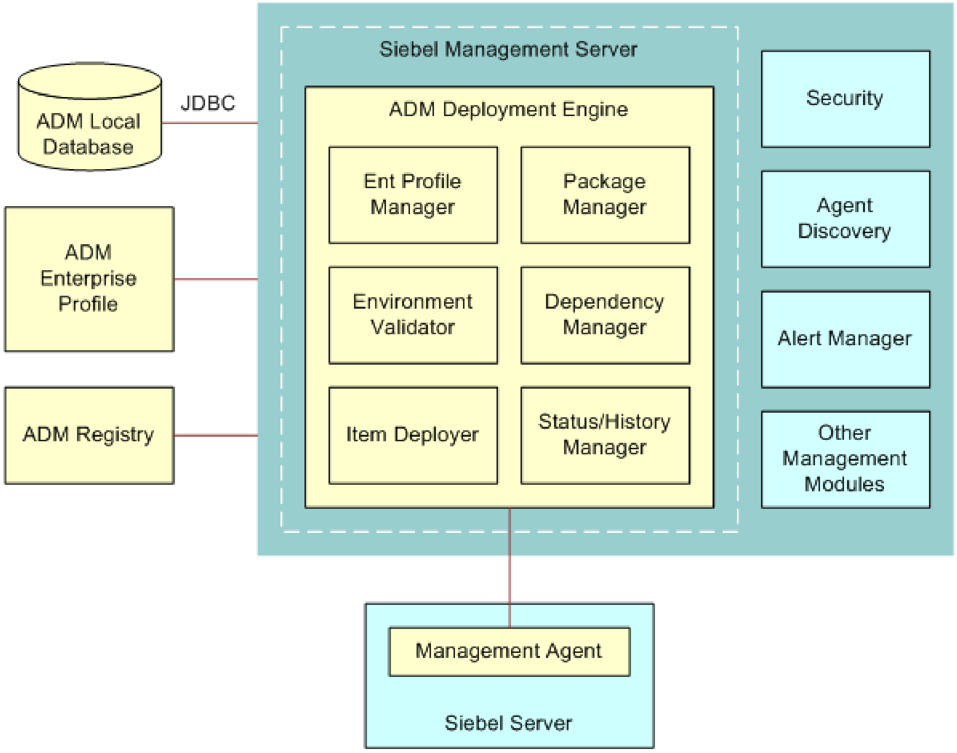


Figure 2. High-Level Overview of ADM Orchestration Environment

About the Siebel Management Server and ADM Deployment Engine

The Siebel Management Server is the prime infrastructure component that orchestrates an ADM deployment; that is, from initial loading to deployment. It also activates or restores the ADM deployment packages, if applicable, and tracks the status of the ADM deployment during this process. The Siebel Management Server maintains connections to Siebel Management Agents to initiate the ADM deployment process on the target environment. It also maintains connections to the ADM local database to store necessary details of the ADM deployment.

The ADM deployment engine resides in the Siebel Management Server along with the other system management and monitoring components. The ADM deployment engine is the portion of the Siebel Management Server that manages the deployment of an ADM package. It includes the following functionality:

- **Enterprise Profile Manager.** This component of the ADM deployment engine manages the ADM enterprise profile, which stores the following information:
  - Siebel Server names and installation directories
  - Relative deployment path for certain groups of file items
  - Siebel Servers that can perform database or repository item deployment

  One Siebel Enterprise Server can have multiple profiles for you to select when deploying a package.
- **ADM Package Manager.** This component of the ADM deployment engine opens an ADM package, loads the descriptor contents and dependencies into memory, validates the package, and manages the section of an ADM enterprise profile for the opened package. This module performs validation on the integrity of the individual files and the existence of all the files included in the ADM package.
- **Environment Validator.** This component of the ADM deployment engine verifies that all file deployment target Siebel Servers are able to handle file deployment, the Siebel Server root path and deployment path are accessible, and database or repository deployment target Siebel Servers are able to run ADM server components.
- **Dependency Manager.** ADM deployment dependencies between ADM deployment units can be organized in two places:
  - In the ADM package descriptor file. The deployment of units is performed in the sequence listed in the ADM package descriptor file. By modifying the package descriptor file, you can control the deployment sequence so that the prerequisite items are deployed before the dependent items.
  - In the Category Dependency section of the enterprise profile. By modifying the Category Dependency parameters, you can define the dependency of one category on another category's status for an action to start. For example, the execution of an action on the database category can depend on the status of the repository category to be MIGRATION_COMPLETE. This ensures that database units are not deployed unless all the repository units have the status of MIGRATION_COMPLETE.
- **Item Deployer.** This component of the ADM deployment engine sends deployment requests to the Siebel Management Agent on Siebel Server machines through the JMX interface. It passes the necessary information to the Siebel Management Agent and the target MBean (either Data MBean or File MBean).
- **Status and History Manager.** This component of the ADM deployment engine monitors the status of each deployment subsession and records the deployment history. It receives notifications from Siebel Management Agent about the deployment status and stores it in the ADM local database.

Siebel Management Server Installation and Configuration

For installation and configuration information on Siebel Management Server, see *Siebel Installation Guide* for the operating system you are using. For information on administering Siebel Management Server, see *Siebel System Administration Guide*.

For ADM-specific configuration information on Siebel Management Server, see "Configuring the Siebel Management Server and Siebel Management Agent after Installation" on page 39.

About the ADM Local Database

The Application Deployment Manager (ADM) local database is a small SQLAnywhere database that functions with the Siebel Management Server during the deployment of application customizations. The ADM local database serves as persistent storage for deployment data, environment information, and deployment status details for each ADM deployment. Specifically, the ADM local database contains:

- **ADM Package Information.** During the ADM deployment process, you load an ADM package's description into the local database. The Siebel Management Server parses the ADM descriptor file of an ADM deployment package and stores this package-specific information. This information includes general package information, deployment groups, deployment dependencies, and deployment unit information for each deployment group.

  NOTE: The physical ADM package files are not stored in the ADM local database.

- **Siebel Server Information.** The ADM local database contains information on the Siebel Enterprise Server, Siebel Servers, Web servers, and other entities of the application framework. This information is cached in the ADM local database.
- **ADM Deployment History Information.** The ADM local database stores records on each ADM deployment session, including details on general session information, when it deploys, and its status throughout the deployment.
- **ADM Enterprise Profile.** When you create an ADM deployment session, the ADM local database stores a copy of the ADM enterprise profile referenced in the create session command. An ADM enterprise profile stores the deployment configuration information and defines a set of default deployment parameters. For more information on this file, see "About the ADM Enterprise Profile" on page 23.

The ADM local database is stored in the following location on the Siebel Management Server: `*MgmtSrvrInstallDir*\ADM\admdb.dbf`, and accesses the Siebel Management Server through Java Database Connectivity (JDBC).

NOTE: The configuration of the ADM local database generally occurs during the installation. However, further updates to the ADM local database's properties can be made at other times by updating the file, configuration.adm.xml. It is not required to directly alter or manipulate any data in the ADM local database. All necessary changes are made automatically.

About the ADM Registry

The Application Deployment Manager (ADM) registry is an XML file located in the *MgmtSrvrInstallDir*\ADM directory of the Siebel Management Server. The default filename for the preconfigured ADM registry is adm_registry.xml. This file defines the ADM deployment behavior by declaring deployment attributes and characteristics for each supported ADM deployment type, including:

- Listing all supported ADM categories; that is, file, repository, database types
- Listing all supported ADM data types within the categories
- Defining common deployment parameters
- Defining ADM data type-specific parameters
- Defining the business services and methods necessary to deploy the ADM data types

ADM Registry File Structure. The ADM registry file is a hierarchical XML file with the top level defined as the deployment category (i.e. data category); that is, file, repository, and database categories. All common deployment parameters in a category are defined at this level. The categories are then further divided into the sublevels representing the ADM deployment types (i.e. data types). The deployment behavior specific to each ADM data type is defined at this level.

At the deployment category level, the ADM registry records the following properties:

- MBean name (the MBean that handles the deployment of this category)
- MBean version
- Validation rules for the category
- Default operations for the category

At the deployment type level, the ADM registry records the following properties:

- Deployment parameters
- Activation parameters
- Restore parameters (if any)
- ADM data type-specific parameters

The parameters defined at the ADM category level are inherited by each individual ADM data type. However, an individual ADM data type can override these parameters with other values or provide its own properties if necessary.

NOTE: The “deployment type” and “deployment category” labels used in ADM registry XML file respectively equate to the “data type” and “data category” labels referenced elsewhere throughout this document.

ADM Registry File Configuration. You can configure the ADM registry file to modify the existing properties or support additional ADM data types. For information on configuring the ADM registry file, see “Modifying ADM Parameters on Siebel Management Server” on page 48 and “Configuring the ADM Registry” on page 51.

For information on restoring the ADM registry, see “Restoring the ADM Registry” on page 52.

About the ADM Enterprise Profile

The application deployment manager (ADM) enterprise profile is an XML file located in the *MgmtSrvrInstallDir*\ADM directory. The default filename for the preconfigured enterprise profile is entprofile.xml. A separate file, entprofile_*enterprise_name*.xml, is created for each Siebel Enterprise Server registered with ADM. Each file contains a collection of default deployment parameters that define the deployment behavior for a specific Siebel Enterprise Server. Each deployment of an ADM package references an ADM enterprise profile to facilitate the deployment of application customizations to a specific Siebel Enterprise Server. You can define multiple enterprise profiles and assign an individual enterprise profile to an ADM package at the time deployment.

The enterprise profile is organized into three levels:

- **Profile.** Parameters defined at this level serve as a set of default parameters for all categories and data types. The categories to be deployed (to support staged deployment) are also specified at this level.
- **Category.** Parameters that define a specific category of data types.
- **Type.** Parameters that define a specific data type.

A hierarchy and an inheritance-type functionality apply to the enterprise profile parameters; that is, the parameter values set at the profile level are inherited by the category and type levels. In general, the parameters set at the category and type levels override the inherited values.

The ADM enterprise profile parameters define the following deployment behavior or information:

- **Deployment Servers.** Different types of deployment items are generally deployed to different servers. The enterprise profile maintains a map between the item category or data types and the list of deployment servers.
- **Stop on Error.** A flag that indicates if the deployment process stops in case one or more items fail.
- **Degree of Parallelism.** The number of parallel deployment threads that can run for database and file categories. Always set to 1 for the repository category, because parallel deployment is not supported for repository types.
- **Target Environment Parameters.** The parameters that store deployment environment dependent parameters for each data category or data type. For example, they define the target directory for each file type.
- **Categories to Be Deployed.**
- **Category Dependency Parameters.** The parameters that define the dependency of a category on other category's status for an action to start.

For more information on configuring an ADM enterprise profile, see "Configuring the ADM Enterprise Profile" on page 49. The enterprise profile is read to the local ADM database when the deployment session is created using the create session command. After the session is created, the enterprise profile cannot be changed. To modify the profile, you must first acknowledge the deployment session, then create a new session for the ADM package.

About the ADM Target Environment

The Application Deployment Manager (ADM) target environment refers to the Siebel Business Application environment to which the application customizations, previously created, are migrated. This environment can be, for example, a test or production environment. The ADM framework automates the migration of application customizations to the target environment from the source environment. The orchestration environment manages and executes this migration.

The ADM target environment uses the following architecture components on the target Siebel Server to complete the ADM migration process:

- **Siebel Management Agents.** For more information on Siebel Management Agents and ADM, see "About the Siebel Management Agents and ADM" on page 24.
- **ADM Server Components.** ADM migration uses three server components during the migration process. For more information on these server components, see "About ADM Server Components" on page 25.

About the Siebel Management Agents and ADM

Siebel Management Agents are installed on the target environment Siebel Servers and communicate with the Siebel Management Server and its ADM deployment engine to facilitate the deployment of application customizations to the target environment.

Siebel Management Agents are controlled through Java Management Extension (JMX) technology and are implemented as Managed Bean (MBean) servers, which have multiple MBeans running. Each MBean is responsible for the deployment of one or more data categories. The MBean data category (i.e. deployment category) mapping is defined in the ADM registry. For each deployment unit, the Siebel Management Server looks up the name of the MBean, based on the category to which the deployment unit belongs.

Part of the MBeans capability is to implement the following ADM deployment functionality:

- Environment validation
- Deployment
- Activation
- Restore
- Restore activation
- Acknowledgement
- Status
- Logging

Siebel Management Agent Installation and Configuration

Siebel Management Agents are generally installed during the installation of the Siebel Server. For installation and configuration information of Siebel Management Agents, see *Siebel Installation Guide* for the operating system you are using. For information on administering Siebel Management Agents, see *Siebel System Administration Guide*. For ADM-specific configuration information on Siebel Management Agent, see "Configuring the Siebel Management Server and Siebel Management Agent after Installation" on page 39.

About ADM Server Components

The Application Deployment Manager (ADM) feature uses three ADM-specific server components to administer the process of deploying application customizations. The three server components are contained in one component group entitled Application Deployment Manager (alias ADM). The individual server components are:

- "Application Deployment Manager Object Manager (ADMObjMgr)"
- "Application Deployment Manager Processor (ADMProc)"
- "Application Deployment Manager Batch Processor (ADMBatchProc)"

The ADMObjMgr and ADMProc server components drive the process of deploying the database and repository items. They are responsible for initiating the deployment, activation, and status update steps by invoking corresponding ADM business service methods, which in turn invoke Siebel Enterprise Application Integration (EAI) business services, specialized business services, or Java business services.

The ADM component group and server components are configured and administered similarly to other Siebel application server components. For more information on the server components, see *Siebel System Administration Guide*.

Application Deployment Manager Object Manager (ADMObjMgr)

The Application Deployment Manager Object Manager (ADMObjMgr) receives and processes requests to invoke the business service methods associated with the deployment of application customizations, including creating asynchronous server requests to perform the processing. The Java MBean in the ADM agent communicates with this server component, using Siebel Java Data Bean.

This application object manager must be enabled on the target Siebel Server specified in the enterprise profile. During the deployment, the ADMObjMgr is the first entry point into the Siebel Server infrastructure from the ADM agent. It is necessary to have the agent communicate with the target Siebel Server on the same host for performance and security reasons.

Application Deployment Manager Processor (ADMProc)

The Application Deployment Manager Processor (ADMProc) server component processes the deployment or activation of database and repository deployment units. It is also responsible for sending the deployment status back to the Management Server through a Java MBean in the ADM agent. The only way that ADM can start a server task of this server component is by generating a server request. A Java Mbean in the Management Server Agent invokes the ADM Object Manager server component to submit this request through its SubmitRequest method.

ADM Processor tasks; that is, the processes to deploy the database and repository units, run only on the Siebel Servers that have the ADM Processor server component enabled. By enabling or disabling the ADM Processor server component, users control which Siebel Servers participate in the deployment of database and repository units. This control provides the following benefits:

- **Parallel deployment and load balancing.** The Siebel Server tasks are submitted as asynchronous server requests, which are distributed across all servers hosting the target ADM Processor component. You can control which servers host these tasks by enabling or disabling the ADM Processor component on individual Siebel Servers as required. However, deployment units with repository objects are always deployed in serial to make sure that multiple units are not deployed overlapping each other, which can leave the Siebel Repository in an unstable state.

- **Isolation of deployment from normal business.** One or more servers can be dedicated to ADM deployment, while other servers are still used for normal business operations by having Siebel Servers hosting the ADM component group without hosting any of the component groups for user Object Managers. Siebel Repository Files (.srf) or other deployment-related configuration changes can be made only on the dedicated servers without affecting the normal business operations on the other servers.

Application Deployment Manager Batch Processor (ADMBatchProc)

The Application Deployment Manager Batch Processor (ADMBatchProc) is a batch-mode server component implemented through a business service. It is used solely for exporting database data types to file from the Server Manager command line (srvrmgr program); that is, this server component implements the command-line version of the ADM GUI export view.

This server component and process can be useful for automating the migration of application customizations through a script or for command-line integration with a third-party tool. It is run by starting a server component job. For more information on using this packaging method, see “Creating ADM Deployment Units from Database Types Directly from Command-Line Interface” on page 74.

3 Overview of the ADM Deployment Process

This chapter contains an overview of the Application Deployment Manager (ADM) deployment process and contains the following topics:

About the Customizations Migration Process Using ADM

The Application Deployment Manager (ADM) feature automates the migration of Siebel application customizations between same-version and same-language environments. The migration process uses the ADM framework as a means of managing the customization migration process, validating the process in various stages, and activating the data customizations in the new environment. For more information on deploying customizations using the ADM framework, see "Deploying Customizations Using the ADM Framework" on page 28.

NOTE: You can migrate data between environments of different languages if no language is specified for the data being deployed.

The customizations migration process can also be run outside the ADM framework using the Siebel application ADM GUI or Siebel Server Manager command-line interface to facilitate the migration. (This process uses ADM functionality provided in the previous 7.x releases.) These options can be run only on a subset of application customizations and do not have the enhanced management features provided by the ADM framework. For more information on deploying customizations outside the ADM framework, see "Other Options for Deploying Customizations Using ADM" on page 32.

The decision to use either the ADM framework or the previous 7.x ADM functionality must be made based on the scope of the customizations migration. While the ADM framework provides more functionality and is generally recommended, it may require a greater investment in time and resources for some deployments.

What ADM Is Not Designed For

ADM is a feature designed to identify, package, and deploy user customizations to a same-version and same-language target Siebel Business Application. It is not designed for the following functions:

- A mass data migration tool
- A change management tool
- A systems management tool
- Installing, patching, or upgrading the Siebel Business Application

Deploying Customizations Using the ADM Framework

At the highest level, the process of migrating customizations using the Application Deployment Manager (ADM) framework involves the following phases:

1. **Packaging.** Bundling the customizations made in one environment in preparation for migration to another environment. For more information on this stage, see“About the Deployment Package Directory” on page 29 and “About the ADM Packaging Process” on page 30.

2. **Deploying.** Deploying the customizations package to the target environment. For more information on this stage, see “About the ADM Deployment Process” on page 30.

NOTE: The descriptions of these two phases assume that the ADM framework is installed and configured.

See Figure 3 for a high-level overview of the ADM process. For information about the ADM framework, see Chapter 2, "Overview of the ADM Architecture". For more information on the other options that describe how to migrate customizations outside the ADM framework, see "Other Options for Deploying Customizations Using ADM" on page 32.

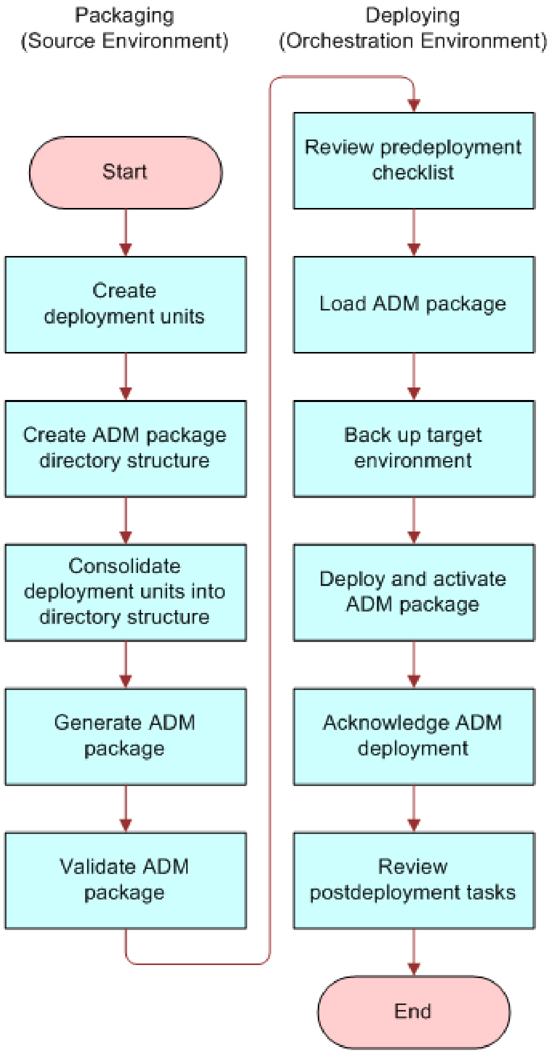

Packaging
(Source Environment)
Deploying
(Orchestration Environment)
Start
Create deployment units
Create ADM package directory structure
Consolidate deployment units into directory structure
Generate ADM package
Validate ADM package
Review predeployment checklist
Load ADM package
Back up target environment
Deploy and activate ADM package
Acknowledge ADM deployment
Review postdeployment tasks
End

Figure 3. High-level Overview of ADM Process

About the Deployment Package Directory

The Application Deployment Manager (ADM) deployment package directory is the location where the packages are loaded in preparation for deployment. During the deployment, the application customizations defined by the ADM package are copied from this directory to the target environment. For more information on the deployment process, see Chapter 6, "Deploying an ADM Package".

The ADM deployment package directory is specified while configuring Siebel Management Server. For information on configuring the Siebel Management Server, see "Configuring the Siebel Management Server and Siebel Management Agent after Installation" on page 39.

The ADM deployment package directory must be a network share location made available to the orchestration and target environments and must have read and write permissions for users who perform deployment operations, as well as for users under whom the Management Agent and Siebel Server processes run.

About the ADM Packaging Process

The Application Deployment Manager (ADM) packaging process prepares the application customizations for migration by consolidating the customization information into a single deployment package, the functional entity used by the ADM framework to stage and execute the deployment. The deployment package is a set of files representing the application customizations, held in a predefined package directory, together with a package descriptor file holding details of the package contents.

A *deployment unit* is the smallest deployable entity upon which the ADM framework operates, and physically represents one single file. Several application customizations can be represented within one deployment unit. Creating each deployment unit is dependent on the following categories of ADM data types: database, repository, and file. For information on consolidating ADM data types into deployment units, see "Consolidating ADM Data Types into Deployment Units" on page 64.

The ADM packaging process primarily uses the ADM Package Utility to create and validate the ADM deployment package. For more information on this utility, see "About the ADM Package Utility" on page 17.

For more information and detailed procedures on the ADM packaging process, see Chapter 5, "Creating an ADM Package."

About the ADM Deployment Process

The Application Deployment Manager (ADM) deployment process represents the actual, physical update of the application customizations to the target environment, as well as the activation of these new customizations. The deployment process involves the creation of a deployment instance, called a deployment session, which migrates the previously created and validated deployment package.

The ADM deployment process uses the ADM command-line interface to deploy and activate the deployment package. The ADM command-line interface includes several options and contingencies available during the deployment and activation process. For more information and detailed procedures on the ADM deployment process, see Chapter 6, "Deploying an ADM Package." For information on the ADM command flow, see "About the ADM Command Flow" on page 31.

About the ADM Command Flow

Application Deployment Manager (ADM) commands are processed through the Siebel Enterprise Server. See Figure 4 for a view of the ADM command flow.

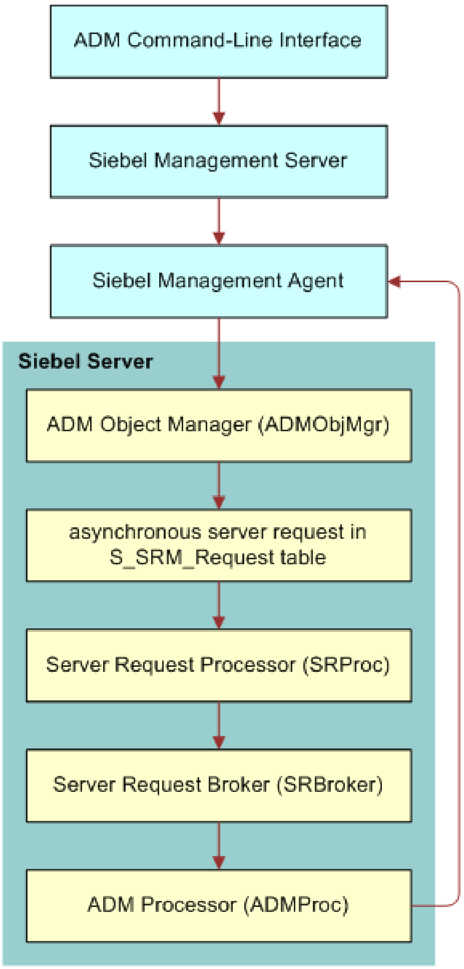

ADM Command-Line Interface
Siebel Management Server
Siebel Management Agent
Siebel Server
ADM Object Manager (ADMObjMgr)
asynchronous server request in S_SRM_Request table
Server Request Processor (SRProc)
Server Request Broker (SRBroker)
ADM Processor (ADMProc)

Figure 4. ADM Command Flow

Commands are entered in the ADM command-line interface (client environment), which executes the ADM deployment on the Siebel Management Server (orchestration environment). From the Management Server, the command is received by the Siebel Management Agent in the target Siebel Server and is passed to the ADM object manager (ADMObjMgr), where an asynchronous server request is created in the S_SRM_Request table. The Server Request Processor (SRProc) picks up the request and invokes the Server Request Broker (SRBroker) to process the request using Siebel Enterprise Application Integration (EAI) and send deployment status back to the Management Agent.

For more information on the ADM elements of the command flow, see the following topics:

- "About the ADM Command-Line Interface" on page 17

- “About the Siebel Management Server and ADM Deployment Engine” on page 19
- “About the Siebel Management Agents and ADM” on page 24
- “Application Deployment Manager Object Manager (ADMObjMgr)” on page 25
- “Application Deployment Manager Processor (ADMProc)” on page 26

For more information on SRProc and SRBroker, see *Siebel System Administration Guide*.

Other Options for Deploying Customizations Using ADM

The following options for deploying Application Deployment Manager (ADM) customizations are available for database data type customizations only. These options do not require the use of the ADM framework, as outlined in “Deploying Customizations Using the ADM Framework” on page 28. They represent the ADM deployment methods available in earlier releases of Siebel Business Applications:

- “About Deploying ADM Customizations Using ADM GUI and HTTP” on page 32
- “About Deploying ADM Customizations Using Export Files and ADM GUI” on page 32
- “About Deploying ADM Customizations Using Server Manager Command-Line Interface” on page 33

NOTE: These deployment options do not have the backup, activation, restore, or enhanced logging features of the ADM framework.

About Deploying ADM Customizations Using ADM GUI and HTTP

Deploying ADM customizations using the Siebel application ADM GUI is available to database types only. This deployment method requires both the source and target environment to be online and in communication using HTTP. The deployment is run from the source environment.

All ADM database types are available for migration using this method. However, the deployment is limited to just the copy operation of these run-time customizations. For more information on this deployment method, see “Deploying ADM Sessions Using the Application Deployment Manager GUI” on page 101.

About Deploying ADM Customizations Using Export Files and ADM GUI

Deploying ADM customizations using export files is available to certain database types only. This deployment method creates XML export files using the Siebel application ADM GUI. The XML files are exported from the source environment and imported into the target environment at the ADM GUI. For more information on this deployment method, see “Deploying ADM Sessions Using Export Files and the ADM GUI” on page 102.

About Deploying ADM Customizations Using Server Manager Command-Line Interface

Deploying ADM customizations using the Server Manager command-line interface (srvrmgr program) is available to certain database types only. This deployment method migrates export files created from the source environment, using the Siebel application ADM GUI, and migrates the export files using the Server Manager command-line interface and a preconfigured workflow process. For more information on this deployment method, see "Deploying ADM Sessions Using Command-Line Interface" on page 102.

4 Configuring and Administering the ADM Framework

This chapter contains topics on configuring the ADM framework—both postinstallation and during regular operations—as well as basic administration tasks. It includes the following topics:

About the ADM Framework Installation

The Application Deployment Manager (ADM) framework requires the installation of, and ADM-specific configurations for, the Siebel Management Agents and Siebel Management Server.

At installation, you must meet the following requirements:

- The correct versions of Java and Perl is installed on machines that host Siebel Management Agent or Management Server. For required versions of Java and Perl for ADM, see *Siebel System Requirements and Supported Platforms* on Siebel SupportWeb. You can check the version of Java by typing `java -version` at the DOS prompt. You can check the version of Perl by typing `perl -v` at a DOS prompt.
- The PATH environment variable is set to the path for the srvrmgr executable. For information on configuring the PATH environment variable, see "Configuring the PATH Environment Variable" on page 38.

You must install and configure at least one instance of Management Agent before you install and configure the Management Server. However, installing all the Siebel Servers and Management Agents before installing the Management Server is recommended.

By default, the Management Agent is automatically installed during a typical Siebel Server installation. In a custom Siebel Server installation, you can either include Management Agent in the installation or exclude it from being installed with the Siebel Server. If you exclude Management Agent from the Siebel Server installation, you must install it separately. Installing Management Agent separately is recommended for the following benefits:

- You can uninstall and reinstall the Management Agent separately, without having to also reinstall the Siebel Server.
- You can uninstall the Siebel Server without uninstalling the Management Agent.

For information on installing the Siebel Management Agents and Siebel Management Server, see *Siebel Installation Guide* for the operating system you are using.

The ADM framework must be configured after the installation. For information on these tasks, see "Process of Configuring the ADM Framework after Installing Siebel Management Server and Siebel Management Agent" on page 36. To verify a functioning ADM framework after installation, deploy the sample package, which is a task in "Process of Configuring the ADM Framework after Installing Siebel Management Server and Siebel Management Agent" on page 36.

Process of Configuring the ADM Framework after Installing Siebel Management Server and Siebel Management Agent

This topic describes the postinstallation configurations that you must complete after installing a Siebel Management Server and one or more Siebel Management Agents.

Each Siebel Server requires configuration to use the ADM framework. The ADM enterprise profile parameters and various language settings must also be modified. Finally, to verify a functioning ADM framework, deploy the sample package included with the installation.

The following topics describe the tasks to carry out:

- "Configuring the Siebel Server for ADM" on page 36
- "Configuring the PATH Environment Variable" on page 38
- "Configuring the Siebel Management Server and Siebel Management Agent after Installation" on page 39
- "Configuring the ADM Enterprise Profile after Installation" on page 40
- "Configuring ADM Workflows" on page 42
- "Configuring Language Settings for the ADM Framework" on page 42
- "Restarting the Servers after ADM Installation and Configuration" on page 43
- "Verifying the ADM Framework Setup" on page 43
- "Deploying a Sample ADM Package" on page 44

Configuring the Siebel Server for ADM

This topic describes how to configure the Siebel Server for Application Deployment Manager (ADM) after installing the Siebel Server. This configuration must be performed for each Siebel Server that has the ADM server components enabled. Each Siebel Enterprise Server must have at least one Siebel Server with the ADM component group enabled in order to use ADM for deployments to that Siebel Enterprise Server.

For information on configuring the Siebel Management Server and Siebel Management Agents after installation, see *Siebel Installation Guide* for the operating system you are using.

This topic is one part of the postinstallation tasks that you must perform to configure ADM. For more information, see "Process of Configuring the ADM Framework after Installing Siebel Management Server and Siebel Management Agent" on page 36.

To configure the Siebel Server for ADM

1. Verify that the ADM component group is enabled on at least one Siebel Server in the Siebel Enterprise Server and that all three ADM-specific server components have the status *Online/ Running*.

   If not, enable the ADM component group and restart the Siebel Server. For more information on this procedure, see *Siebel System Administration Guide*.

2. Verify that the Auxiliary System Management component group is enabled.

   If not, enable the Auxiliary System Management component group and restart the Siebel Server. For more information on this procedure, see *Siebel System Administration Guide*.

3. Create a directory at the root level of your drive, for example `D:\jar` (Windows) or `/jar` (UNIX), with read and write access privileges.

   NOTE: This step is necessary, because there is a restriction on the number of characters that can be specified for the Java CLASS PATH environment variable.

4. Navigate to `*MgmtAgentInstallDir*\lib` and copy the following files to the directory you created in Step 3:

   - siebelmgr.jar
   - admstatussvc.jar
   - Siebel.jar
   - SiebelJI_*xxx*.jar where *xxx* is the deployment language, for example, enu is a U.S. English deployment.

5. Log in to your Siebel application (for example, Call Center), and navigate to Administration - Server Configuration > Enterprises > Profile Configuration.

   This step displays the Profile Configuration list.

6. In the Profile field of the Profile Configuration list, query for the ADMJavaSubsys profile.

   The Profile Parameters list displays the parameters for the ADMJavaSubsys profile.

7. Select the JVM Classpath parameter and verify that it references each of the files that you copied to the directory in Step 4. Using the example, this parameter must reference:

   ```
   D:\JAR\siebelmgr.jar;D:\JAR\admstatussvc.jar;D:\JAR\Siebel.jar;D:\JAR\SiebelJI_xxx
   .jar
   ```

   where *xxx* is the deployment language.

   NOTE: For UNIX environments, use a colon (:) in place of a semi-colon (;) in the previous reference.

8 Select the JVM DLL Name parameter, and verify that the entry in the Value field references the location of the jvm.dll file in the JRE installation.

The JVM DLL Name parameter value depends on the operating system. See the following table for the values for the different operating systems.

Operating System	Parameter Value
Windows	JRE_HOME/bin/client/jvm.dll
Solaris	JRE_HOME/lib/sparc/client/libjvm.so
AIX	JRE_HOME/bin/j9vm/libjvm.so (IBM JRE)
Novell SUSE Linux	JRE_HOME/lib/i386/client/libjvm.so
Oracle Enterprise Linux	JRE_HOME/lib/i386/client/libjvm
HP-UX	JRE_HOME/lib/IA64N/server/libjvm.so

9 Navigate to Administration - Server Configuration > Enterprises > Parameters.

The Enterprise Parameters list appears.

10 In the Enterprise Parameters list, query to retrieve the record for the JVM Subsystem Name parameter.

11 Enter ADM Java Systems in the Value field for the JVM Subsystem Name parameter.

12 Navigate to Administration – Server Configuration > Enterprises > Synchronize.

The Synchronize view appears.

13 Click Synchronize.

NOTE: In the case of heterogeneous server environments, create a named subsystem for each server platform, and set the parameter ADM Java Systems at the ADM component level, with the named subsystem applicable to the environment.

Configuring the PATH Environment Variable

The directory path to the Server Manager (srvrmgr) executable must be set in the PATH environment variable at the installation of the ADM framework. The PATH environment variable can be configured globally (effective for all subsequent sessions) or for each session.

To set the Server Manager Path globally

1 Navigate to Settings > Control Panel > System > Advanced tab.

2 Click Environment Variables.

3 Select the Path system variable and click Edit.

4 In the Variable value field, enter the directory path to the srvrmgr executable. Use semicolons as the delimiter between directory names.

For information on configuring an environment variable globally, see *Siebel System Administration Guide*.

To set the Server Manager Path for each session

1. Open a command window.
2. Type the following command:

   ```
   set PATH="%PATH%;<srvrmgr folder>"
   ```

   where *srvrmgr folder* is the directory path to the srvrmgr executable. Use semicolons as the delimiter between directory names.

NOTE: You must update the Path environment variable before running the script. This will only affect the command shell in which it is run. If you open a new command window, then you will need to update the Path variable again.

Configuring the Siebel Management Server and Siebel Management Agent after Installation

This topic is part of the postinstallation tasks that you must perform to configure ADM. For more information, see "Process of Configuring the ADM Framework after Installing Siebel Management Server and Siebel Management Agent" on page 36.

On completion of the Siebel Management Server and Siebel Management Agent installations, a configuration wizard appears for the infrastructure component you installed. During the configuration, make sure to configure the settings for ADM.

When using the Siebel Management Server Configuration Wizard to configure Management Server and register one or two Management Agents, specify the default directory location of the ADM package, which is set in the admpkgr.bat file.

NOTE: The deployment package directory must be on a network share that is available to both the orchestration and target environments.

You must also execute the getserver.pl, makeagentconfig.pl, and admconfig.pl Perl scripts to register additional Management Agents and create the following files:

- deploy_*enterprise_name*.bat, located in the *MgmtSrvrInstallDir*\pref\system directory
- entprofile_*enterprise_name*.xml, located in the *MgmtSrvrInstallDir*\ADM directory

where *enterprise_name* is the name of the Siebel Enterprise Server.

For more information on configuring Siebel Management Server and Siebel Management Agent, see the chapter about configuring the Siebel Enterprise Server and its related components in *Siebel Installation Guide* for the operating system you are using.

Security configuration of the Management Server and Management Agent are also required for ADM. For information on the security configuration of the Management Server and Management Agent, see "About Security Configuration of Siebel Management Server and Siebel Management Agent" on page 47.

Configuring the ADM Enterprise Profile after Installation

This topic describes how to configure the initial ADM enterprise profile following the installation of Siebel Management Server and Siebel Management Agents. For more information, see "Process of Configuring the ADM Framework after Installing Siebel Management Server and Siebel Management Agent" on page 36.

For configuration updates to this file at later times and for more information on this file and its parameters, see:

- "Configuring the ADM Enterprise Profile" on page 49
- "About the ADM Enterprise Profile" on page 23

To configure the enterprise profile after installation

1. Navigate to `*MgmtSrvrInstallDir*\ADM` and open the entprofile_*enterprise_name*.xml file using the XML editor Microsoft Office InfoPath.
2. Verify that the value for AgentName corresponds to *xxx* as specified in the following files:
   - agentconfig.xml file, located in *MgmtAgentInstallDir*\pref\system

     `<entry key="registryhost" value="xxx">`
   - configuration.agent.xml, located in *MgmtSrvrInstallDir*\pref\system

     `<entry key="xxx" value="...">`
3. (Multiserver setup) Verify and update the SCBroker port number in the connect string for the server. The port for the SCBroker component is defined in the PortNumber component parameter.

   NOTE: Only one server must be defined, since all servers will connect to the same database.
4. Change the values in the DeployServer parameter.
   - (Single-server setup) Replace all the values for the DeployServer parameter with the value of AgentName as determined in Step 2, for example, `<DeployServer>`*xxx*`</DeployServer>`.
   - (Multiserver setup) List the deployment servers for the file, database, and repository data type categories in the DeployServer parameter with the value of the corresponding AgentNames, for example:

     `<DeployServer>`*xxx*`</DeployServer>` (for deploying on the server where the *xxx* agent is running)

     `<DeployServer>`*yyy*`</DeployServer>` (for deploying on the server where the *yyy* agent is running)

     NOTE: For the file category, every deployment server in the enterprise must be listed. For the database and repository categories, only 1 of the deployment servers in the Siebel Enterprise Server must be listed, since all servers will connect to the same database.

   NOTE: Entries for this parameter are case-sensitive.

5 Provide the path for the ADM backup directory in the database and repository sections by changing all occurrences of `<ADMBackupDir>\\net\CHANGE_ME\shared;\\CHANGE_ME\upload</ADMBackupDir>` to:

- For Windows, `<ADMBackupDir>\\`*`MachineName`*`\upload</ADMBackupDir>`

  where *`MachineName`* is the host name of the targeted machine.

  NOTE: The entry is case-sensitive.

- For UNIX, `<ADMBackupDir>/tmp/upload</ADMBackupDir>`

  The `/tmp/upload` path must be UNC qualified with full permissions.

The path must be accessible from the Management Server and Management Agent, and the ADM backup directory must be a shared location. The user under whom the Management Agent process runs must have read/write access to the ADM backup directory.

NOTE: Management Server is supported only on a Microsoft Windows platform, while Management Agent is supported on both the Windows and UNIX platforms.

6 To automatically run validation as part of all copy and activate commands during the deployment, set the value for the PerformValidation parameter to Yes:

```
<PerformValidation>Yes</PerformValidation>
```

If this value is set to Yes, then the validation automatically runs before any deployment or restore command. If it is set to No, then the validation will not be performed unless explicitly run as a separate command.

7 Navigate to *`MgmtSrvrInstallDir`*`\` and open the deploy_*enterprise_name*.bat file.

8 Verify that the value of the deployment package location corresponds to the following and that the path is fully qualified:

- For Windows, set `ADM_PACK_LOC=\\`*`MachineName`*`\packages`
- For UNIX, set `ADM_PACK_LOC=/mnt/packages`

The path must be accessible from the Management Server and Management Agent.

NOTE: Management Server is supported only on a Microsoft Windows platform, while Management Agent is supported on both the Windows and UNIX platforms.

9 Create a directory share, upload, and make it network accessible with full permissions.

For example, access the directory as follows:

`\\`*`MachineName`*`\upload`

This directory can be used as a backup location for ADM deployments. Siebel Management Agent users and Siebel Server users must have read and write access to this directory.

NOTE: Do not add the ADMBackupDir flag to file data types in the enterprise profile. The backup, by default, is taken from the ADMFile directory on the Siebel Management Agent.

Configuring ADM Workflows

This topic describes configuring ADM workflows by activating the workflows, and is a necessary task before running an ADM deployment. It describes one of the postinstallation tasks that you must perform following the installation of the ADM framework. For more information, see "Process of Configuring the ADM Framework after Installing Siebel Management Server and Siebel Management Agent" on page 36.

Activating a Siebel Business Application workflow is a standard workflow administration procedure. For more information on workflows and activating workflows, see *Siebel Business Process Framework: Workflow Guide*.

You can also activate ADM workflows directly from the ADM screens by using the following procedure.

To activate ADM workflows

1. From the application-level menu, choose Navigation > Site Map > Application Deployment Manager screen.
2. Select the Data Type Details view.
3. In the Data Types list, click the Menu button, and then click Activate ADM Workflows.

Configuring Language Settings for the ADM Framework

This topic describes setting language parameters to make sure the Application Deployment Manager (ADM) framework functions in the correct locale setting.

To configure language settings for the ADM framework

1. Navigate to `*MgmtSrvrInstallDir*\pref\system`, and edit the file configuration.xml.
2. Set the parameter value for the parameter entry key=locale to the correct language suffix, for example, `<entry key="locale" value="ja"/>`.

   NOTE: For information on language properties, see *MgmtSrvrInstallDir*`\resources`.

3. Navigate to *MgmtSrvrInstallDir*`\pref\system`, and edit the Siebel Management Agent configuration file (filename agentconfig.xml).
4. Set the parameter value for the parameter entry key=locale to the correct language suffix, for example, `<entry key="locale" value="ja"/>`.
5. Navigate to *MgmtSrvrInstallDir*`\ADM`, and search the enterprise profile (filename entprofile_*enterprise_name*.xml) for all Application Object Manager references. Make sure their suffixes have the correct language code, for example, change ADMObjMgr_enu to ADMObjMgr_jpn if using the Japanese language.

Restarting the Servers after ADM Installation and Configuration

Following the configuration tasks you undertook in the following topics, you must restart the Siebel Server, the Siebel Management Agent, and the Siebel Management Server in sequence:

- "Configuring the Siebel Server for ADM" on page 36
- "Configuring the Siebel Management Server and Siebel Management Agent after Installation" on page 39
- "Configuring the ADM Enterprise Profile after Installation" on page 40
- "Configuring Language Settings for the ADM Framework" on page 42

The *Siebel System Administration Guide* describes how to start and stop the Siebel Server, the Siebel Management Server, and Siebel Management Agents.

On both the Microsoft Windows and UNIX platforms, perform these tasks in the following order:

1. Stop the Siebel Server.
2. Stop the Siebel Management Agent.
3. Stop the Siebel Management Server.
4. Start the Siebel Server.
5. Start Siebel Management Agent.
6. Start Siebel Management Server.

NOTE: As indicated in the previous list, the different Siebel Management Agents on the Siebel Servers must be started before the Siebel Management Server is started to avoid an unnecessary large volume of errors in the log files. When shutting down, also shut down the Siebel Server before the agents.

Verifying the ADM Framework Setup

This topic describes how to verify that the ADM framework is set up properly.

To verify the ADM framework setup

1. Copy the deploy_*enterprise_name*.bat file from *MgmtSrvrInstallDir*\pref\system to *MgmtSrvrInstallDir*.

   The Management Server log shows that agents have come online with the Management Server.

2. Open the command prompt, and change to *MgmtSrvrInstallDir*.
3. Execute the following command:

   deploy_*enterprise_name* valent *username password*

   where:

- *enterprise_name* is the name of the Siebel Enterprise Server.
- *username* is the Siebel user name.
- *password* is the Siebel user password.

This command validates the enterprise profile after ADM postinstallation configuration.

Deploying a Sample ADM Package

This topic describes how you deploy a sample package to verify that ADM and the Siebel Management Server function correctly. For more information on the overall deployment process, see "Overview of the ADM Deployment Process" on page 27.

To deploy a sample package using ADM

1. Verify that the deploy_*enterprise_name*.bat file, which is located in the following directory, correctly references the packages directory for ADM:

   ```
   \\MgmtSrvrInstallDir\
   ```

   For example, the file must contain an entry similar to the following:

   ```
   set ADM_PACK_LOC=C:\SBA_80\MgmtSrvrInstallDir\ADM\packages
   ```

2. Open a command window, and navigate to the Siebel Management Server installation directory.
3. Execute the following command:

   ```
   deploy_enterprise_name load username password sample package_location
   ```

   where:

   - *enterprise_name* is the name of the Siebel Enterprise Server.
   - *username* is the Siebel user name.
   - *password* is the Siebel user password.
   - *package_location* is the location of the ADM package. This parameter is mandatory. An error is displayed if the package location is not specified.

   This command loads the package, sample, into the database.
4. Execute the following command:

   ```
   deploy_enterprise_name create username password sample
   ```

   This command creates an ADM deployment session for the package, sample.
5. Execute the following command:

   ```
   deploy_enterprise_name copy username password sample
   ```

   This command completes the deployment of the package, sample.
6. Execute the following command:

deploy_*enterprise_name* activate *username password* sample

This command activates the application customizations contained in the deployment package, sample.

7 Execute the following command:

deploy_*enterprise_name* status *username password* sample

This command displays the deployment status summary of the deployment package, sample.

NOTE: If you need to redeploy the sample package in case of errors, redo the copy command. If you want to redeploy, you must acknowledge the session, and create a new session, regardless of whether the first attempt was successful. If you want to redeploy to another Siebel Enterprise Server, you need to use the appropriate batch file, and start from the create command, because the package has already been loaded.

Sample output of successful sample package deployment using ADM:

The following is an example of the output you would receive for a successful deployment of the sample package using ADM:

```
C:\adm\mgmtsrvr>deploy_siebel load SADMIN SADMIN sample

Success         Package 'sample' has been successfully loaded into database.


C:\adm\mgmtsrvr>deploy_siebel create SADMIN SADMIN sample

Success         Session was created successfully for package 'sample'.


C:\adm\mgmtsrvr>deploy_siebel validate SADMIN SADMIN sample

Package Name   sample


All validation succeeded.


Validation rules of deployment types that are deployed to same server have been

merged. Please check log for details.


C:\adm\mgmtsrvr>deploy_siebel valent SADMIN SADMIN sample

Failure         option 'sample' not found.


C:\adm\mgmtsrvr>deploy_siebel copy SADMIN SADMIN sample -async

Success         Deployment for package 'sample' started.
```

```
C:\adm\mgmtsrvr>deploy_siebel status_detail SADMIN SADMIN sample
Package Name               sample
Enterprise Name            siebel
Session Number             466
Session Status             Deployment Started
Session Execution Status   Not Running

Group ID   Group Name                                  Unit ID   Unit Name
                                     Start Time           Last Modified        S
tatus                Execution Status   Server Name
941        File Group - Reports - Application Server   51444     file/AppServer/
reports/ENU/sample.rox               1/19/07 2:28:15 PM   1/19/07 2:28:35 PM   D
eployment Complete   Not Running        siebel
942        File Group - Webmaster                      51439     file/AppServer/
webmaster/enu/images/ebus.gif        1/19/07 2:28:40 PM   1/19/07 2:28:59 PM   M
igration Complete    Not Running        siebel
942        File Group - Webmaster                      51441     file/AppServer/
webmaster/enu/images/image_1.gif     1/19/07 2:28:40 PM   1/19/07 2:29:00 PM   M
igration Complete    Not Running        siebel
942        File Group - Webmaster                      51443     file/AppServer/
webmaster/enu/images/image_3.gif     1/19/07 2:28:40 PM   1/19/07 2:29:00 PM   M
igration Complete    Not Running        siebel
942        File Group - Webmaster                      51438     file/AppServer/
webmaster/enu/files/main.css         1/19/07 2:28:40 PM   1/19/07 2:29:00 PM   M
igration Complete    Not Running        siebel
942        File Group - Webmaster                      51440     file/AppServer/
webmaster/enu/images/globe77_d.gif   1/19/07 2:28:40 PM   1/19/07 2:29:00 PM   M
igration Complete    Not Running        siebel
942        File Group - Webmaster                      51442     file/AppServer/
```

```
webmaster/enu/images/image_2.gif     1/19/07 2:28:40 PM   1/19/07 2:29:00 PM   M
igration Complete    Not Running        siebel
943        Repository Group                             51437     repository/busv
c.sif                                1/19/07 2:28:36 PM   1/19/07 2:29:24 PM   D
eployment Complete   Not Running        siebel
943        Repository Group                             51436     repository/bcs.
sif                                  1/19/07 2:29:28 PM   1/19/07 2:29:42 PM   D
eployment Complete   Not Running        siebel
944        Database Group                               51435     database/42-5FN
ZY_Responsibility.xml                1/19/07 2:28:21 PM   1/19/07 2:29:14 PM   D
eployment Complete   Not Running        siebel

Total Number of Units     10
Total Units Completed     4
Total Units Not Started   0
Total Units Failed        0
Total Units Running       0

C:\adm\mgmtsrvr>deploy_siebel activate SADMIN SADMIN sample
Success         Command completed successfully.
```

About Security Configuration of Siebel Management Server and Siebel Management Agent

This topic describes the Siebel Management Server and Siebel Management Agent security configurations necessary for Application Deployment Manager (ADM).

During the installation and configuration of the Management Server a Siebel user (Siebel administrator) is specified with a corresponding username and password. The Siebel administrator runs the Management Server and Management Agent and may perform deployment operations.

Security responsibilities are also assigned during the installation and configuration of the Management Server. Three particular responsibilities pertain to ADM:

- **MgmtSrvr-Monitor.** Allows users to monitor the status of the deployment operations, like status or report, but does not allow them to execute ADM deploy and execute commands.

- **MgmtSrvr-Deploy&Execute.** Allows users to execute ADM deploy and execute commands.
- **MgmtSrvr-Admin.** Siebel Management Server administrative responsibilities.

Responsibilities can be modified by the Siebel administrator in the Administration - User screen of a Siebel application, such as Call Center.

User information and encrypted passwords may be stored in the following files:

- security.properties
- management.policy
- catalina.policy

NOTE: Do not edit the security.properties file unless the system user's password has changed. Changing the management.policy and catalina.policy files is not recommended.

For additional information about security configuration of Siebel Management Server and Siebel Management Agent, see *Siebel Security Guide*.

About ADM System Administration

The following topics describe general ADM system administration or configuration tasks that may be necessary when preparing for ADM deployments.

- "Modifying ADM Parameters on Siebel Management Server" on page 48
- "Configuring the ADM Enterprise Profile" on page 49

Modifying ADM Parameters on Siebel Management Server

This topic describes the task of modifying ADM parameters that configure the Siebel Management Server. This task involves changes to the configuration.adm.xml file. There are only two parameters that may require updated settings in this file:

- ADMRegistryFile
- DefaultProfileLocation

For example, the ADMRegistryFile parameter can be updated to point to a specific shared network location. ADM reads this location when commands are issued for deployment.

The other parameters in this file are properly set during the installation and generally do not require further configuration. For a brief description of the configuration.adm.xml file parameters, see Table 3.

NOTE: The package location defined in the deploy_*enterprise_name*.bat file supersedes the default package location defined in the configuration.adm.xml file. During deployment, the package location is taken from the deploy_*enterprise_name*.bat and not from configuration.adm.xml file.

To modify ADM parameters on Management Server

1. Navigate to the configuration.adm.xml file located in the following directory:

   `*MgmtSrvrInstallDir*\pref\system`

2. Using an XML or text editor, open the configuration.adm.xml file.
3. Modify the desired parameters.

   See Table 3 for a brief description of the configuration.adm.xml file parameters.

4. Restart the ADM agent for the configurations to take effect.

Table 3. ADM Parameters Available in the configuration.adm.xml File

Parameter Name	Description
ADMRegistryFile	Default ADM registry location
ConnectStr	Connect string to local ADM database
DBEngExe	ADM local database engine executable
DBName	ADM local database name
DBStartCmdArgs	Local database filename
DBStopCmd	Stop local database engine command
DefaultPackageLocation	Default ADM package location
DefaultProfileLocation	Default ADM profile location
DriverName	JDBC driver Java package name
SQLFilePath	File for externalized SQL statements
XMLFileEncoding	Encoding of XML files used and generated by ADM

Configuring the ADM Enterprise Profile

Configuring the ADM enterprise profile is a one-time setup task that must be performed before your initial deployment. The ADM enterprise profile requires subsequent updating only if changes are made to your environment or if deployment behavior changes are necessary.

The ADM enterprise profile is an XML document, by default named entprofile.xml, located in the \MgmtSrvrInstallDir\ADM directory. This file defines several parameters that control the ADM deployment behavior, including information on the ADM data types for deployment. Most of these parameters are set at installation. For more information on configuring the ADM enterprise profile at installation, see "Configuring the ADM Enterprise Profile after Installation" on page 40. For more information on the ADM enterprise profile, see "About the ADM Enterprise Profile" on page 23. The preferred method for examining or modifying the contents of the enterprise profile is through Microsoft Infopath.

To configure the ADM enterprise profile

1. Navigate to the entprofile.xml file in the \MgmtSrvrInstallDir\ADM\ directory.
2. Using an XML editor, for example, Microsoft Office InfoPath, or a text editor, open the entprofile.xml file.
3. Locate the environment and deployment-specific parameters, and enter the values for your environment.

   For information on specific parameters, see Table 4.

The changes made to the enterprise profile are reflected when you create a new session.

Table 4. Listing of ADM Enterprise Profile Parameters

Parameter	Default	Description
PerformBackup	Y	When set to Y, a backup of the target system's existing data is performed automatically before the deployment. This flag can be set at all three levels. Setting this flag at a lower level allows the backup operation to skip individual units or groups. Turning off the backup operation can increase performance of the deployment.
DegParallelism	1	This parameter must be set to a nonzero number and is set at the category level only. For the repository category, set only to 1. For the database category, parallelism represents the number of database items deployed concurrently. For example, if the value is set to 2, two deployment units deploy concurrently. For the file category, parallelism represents the number of file servers deployed concurrently. This setting is dependent on the number of servers and the number of units inside the File category of the package. For example, if the File category consists of 5 units and there are 2 servers, the maximum value for the degree of parallelism is 10 (5 multiplied by 2).
StopOnError	No	This parameter is set at the profile level. If set to Yes, the deployment stops when an error occurs. If set to No, the deployment continues to the next unit or group even though an error occurs. ADM does not check dependencies between separate units—especially database items. If one unit fails and another database unit is dependent on the failed unit, then the second unit also fails. Database items that have been related though the ADM UI are treated as one unit in this case (one file).

Table 4. Listing of ADM Enterprise Profile Parameters

Parameter	Default	Description
BackupDir	Not applicable	This parameter contains the path of the ADM backup directory, which contains .sif files. The ADM backup directory must be a shared location.
DeployServer	Not applicable	This parameter contains the paths of the deployment servers in the target environment. **NOTE:** Entries for this parameter are case-sensitive.

About ADM Configurations for New Data Types

Application Deployment Manager (ADM) can be extended to support additional database and file data types for deployment. Creating new file data types for deployment requires a new entry in the ADM registry. Creating new database data types for deployment requires a new entry in the ADM registry, as well as a new integration object and a new content object.

NOTE: Adding repository data types is unnecessary, since all repository objects are automatically covered.

The following topics describe ADM system administration or configuration tasks that may be necessary when preparing for ADM deployments of new data types.

- "Configuring the ADM Registry" on page 51
- "Restoring the ADM Registry" on page 52
- "Creating Integration Objects for ADM" on page 52
- "Creating Content Objects for ADM" on page 53
- "Configuring ADM Parent Child Relationships on Integration Objects" on page 54

For information on specific data type attributes used for creating new data types, see Appendix B, "ADM Support for Additional Data Types". For other deployment and activation details for ADM data types, see Appendix A, "ADM Supported Data Types".

Configuring the ADM Registry

Configuring the Application Deployment Manager (ADM) registry must only occur when absolutely necessary, and extreme caution must be exercised in order to prevent the corruption of the ADM function.

Scenarios that will require you to modify the ADM registry include the following:

- You are adding a new data type (database/EAI or file).
- You are changing an ADM data type name.

- You need to edit the validation rules to include more components, or valid states for components, as part of the validation rules.

The ADM registry is an XML document, by default named adm_registry.xml, located in the \MgmtSrvrInstallDir\ADM directory. This file defines the ADM deployment behavior. For more information on this file, see "About the ADM Registry" on page 22.

To configure the ADM registry file

1. Back up the existing adm_registry.xml file from the \MgmtSrvrInstallDir\ADM\ directory to another location.
2. Navigate to the adm_registry.xml file in the \MgmtSrvrInstallDir\ADM\ directory.
3. Using an XML editor, for example, Microsoft Office InfoPath, or a text editor, open the adm_registry.xml file.
4. Locate the parameters you want to modify and enter the values you want to include.

   For information on the file structure and parameters of the ADM registry file, see "About the ADM Registry" on page 22.

For information on restoring the ADM registry, see "Restoring the ADM Registry" on page 52.

Restoring the ADM Registry

The ADM registry can be restored from the backup you created prior to configuring the ADM registry file. For more information on how to back up the ADM registry, see "Configuring the ADM Registry" on page 51. For more information on the ADM registry, see "About the ADM Registry" on page 22.

To restore the ADM registry file from backup

1. Shut down the Management Server.
2. Replace the existing adm_registry.xml file in the \MgmtSrvrInstallDir\ADM\ directory with the backup file you had placed in another location.
3. Restart the Management Server.

Creating Integration Objects for ADM

Creating an integration object for Application Deployment Manager (ADM) allows you to migrate new database data types using ADM.

NOTE: The preconfigured data types do not require any configuration in Siebel Tools.

The integration object specifies the format and structure of the set of data that you want to migrate.

Use the EAI Object Wizard in Siebel Tools to create integration objects. For more information on the integration objects and on using the EAI Object Wizard, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

NOTE: Intermediate-level knowledge of the Siebel EAI architecture and integration is a prerequisite before performing these tasks. For more information on Siebel EAI, see *Overview: Siebel Enterprise Application Integration*.

To create an ADM integration object

1. Identify the business object that corresponds to the data type you want to migrate using ADM. This business object must have only one primary business component.
2. In Siebel Tools, choose File > New Object.
3. From the New Object screen, select Integration Object.

   The EAI Object Wizard guides you through the process of setting up an integration object.
4. Remove any unwanted fields or integration components from the new integration object.
5. Adjust the user keys of the integration object. For more information on user keys for integration objects, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.
6. Validate the integration object by selecting the integration object and clicking Validate.
7. Review the report, and modify your integration object as needed.

Creating Content Objects for ADM

Creating a content object for Application Deployment Manager (ADM) allows you to migrate new database data types using ADM.

NOTE: The preconfigured data types do not require any configuration in Siebel Tools.

Before creating a content object, make sure you have created an integration object for the data you want to migrate. See "Creating Integration Objects for ADM" on page 52.

The content object specifies the data that will be migrated from one Siebel application to another. Each content object is composed of a business object and an integration object.

Content objects are also used by Content Center. The following procedure references properties necessary for ADM functionality. For Content Center functionality, additional properties must be set for the content object. For more information on Content Center, see *Siebel Applications Administration Guide*.

NOTE: One content object can be used by both Content Center and ADM at the same time.

To create content object for ADM

1. In Siebel Tools, navigate to Content Objects type.
2. Create a new record.

3. Add the data type business object to the content object record.
4. Add the data type integration object to the content object record.
5. Create child records. On the child record, enter the name of a view that does not have any visibility setting (admin view) by selecting from the picklist.

   The child record determines which view will be used during the preview functionality in the ADM GUI. This view must correspond to the business object of the data area.

Configuring ADM Parent Child Relationships on Integration Objects

Many Application Deployment Manager (ADM) data types have hierarchical structures of different objects that consist of child and parent relationships. In some cases, you might want to control how these hierarchies are deployed by removing the child entities from ADM deployments.

For example, the responsibility data type also migrates users as well as responsibilities, because responsibilities are associated with users. The users need to be included with the responsibility. However, in this example, you might not want to migrate users between environments, because this migration deploys development users to a production environment.

You must disable the child integration component of the respective data type integration object to deploy only parent records of hierarchical relationships. Perform the following procedure to configure this behavior.

NOTE: Intermediate-level knowledge of the Siebel EAI architecture and integration is a prerequisite before performing these tasks. For more information on Siebel EAI, see *Overview: Siebel Enterprise Application Integration*.

To configure an ADM integration object to activate and deactivate the child integration component

1. In the source environment, select the integration object you want to update in Siebel Tools.

   NOTE: The project to which the integration object belongs must be locked.

2. Click Synchronize in the Integration Objects list.

   The Integration Object Synchronize wizard appears.

3. Click the plus sign (+) to list all related integration components.
4. Select the checkboxes next to the integration components you want to include in the integration object. Clear the checkboxes next to the integration components you do not want to include.
5. Complete the wizard, and click Finish to synchronize the Siebel integration object and the Siebel business object.
6. Right-click your integration object, and select Validate to validate your integration object.

For more information on using the synchronization wizard and configuring integration objects, see *Integration Platform Technologies: Siebel Enterprise Application Integration*.

ADM Run-Time Requirements

The proper functionality of Application Deployment Manager (ADM) depends on run-time requirements for the following items:

- "Siebel Objects That Must Not Be Modified Or Removed" on page 55
- "Server Components That Must Be Enabled And Online" on page 56

Siebel Objects That Must Not Be Modified Or Removed

If certain Siebel objects are modified or removed, Application Deployment Manager (ADM) will fail to function properly, and support for restore from backup will no longer be available. The following Siebel objects must not be modified or removed during the deployment:

- "Workflows"
- "Business Services"
- "Responsibilities"

Workflows

Do not modify or remove the following workflows:

- ADM Deployment
- ADM Restore
- UDA Acknowledgement
- UDA Batch Deployment
- UDA Batch Import
- UDA Data Query
- UDA Deployment
- UDA Environment Verification
- UDA HTTP Transport
- UDA Import File
- UDA Target Workflow

Business Services

Do not modify or remove the following business services:

- ADM JDB Service
- XML Converter
- ADM Service

- ADM Status Report Business Service
- Access Controlled BS Cache Clear Service
- Access Controlled Task Cache Clear Service

Responsibilities

Do not modify or remove the following responsibilities:

- MgmtSrvr-Admin
- MgmtSrvr-Deploy&Execute
- MgmtSrvr-Monitor

Server Components That Must Be Enabled And Online

Specific server components must be enabled and online to run an Application Deployment Manager (ADM) deployment. See Table 5 for server component requirements to run an ADM deployment.

Table 5. Server Component Requirements for ADM Deployment

Component Group	Component	Export	Deploy from File	HTTP Deployment	Framework Deployment
Application Deployment Manager (ADM)	ADMBatchProc	Optional	Optional	Optional	Not needed
	ADMObjMgr				Required
	ADMProc				
Auxiliary System Management (SystemAux)	FSMSrvr	Required	Required	Required	Required
	SRProc				
	SvrTblCleanup				
	SvrTaskPersist				
	AdminNotify				

Table 5. Server Component Requirements for ADM Deployment

Component Group	Component	Export	Deploy from File	HTTP Deployment	Framework Deployment
Enterprise Application Integration (EAI)	BusIntBatchMgr	Optional	Optional	Required	Optional
	BusIntMgr				
	CustomAppObjMgr				
	EIM				
	EAIObjMgr				
	JMSReceiver				
	MqSeriesSrvRcvr				
	MqSeriesAMIRcvr				
	MSMQRcvr				
	SMQReceiver				
System Management (System)	ServerMgr	Required	Required	Required	Required
	SRBroker				
	SCBroker				
	SiebSrvr				
	SrvrSched				
Workflow Management (Workflow)	GenTrig	Optional	Optional	Required	Optional
	WorkMon				
	WfProcBatchMgr				
	WfProcMgr				
	WorkActn				
	WfRecvMgr				

5 Creating an ADM Package

This chapter covers the process of creating an Application Deployment Manager (ADM) package for deployment using the ADM framework. It includes the following topics:

- "About the ADM Package Utility Modes and Switches" on page 60
- "Copying the ADM Package Utility to a UNIX Environment" on page 62
- "Process of Creating an ADM Package" on page 63
- "Consolidating ADM Data Types into Deployment Units" on page 64
- "Process of Creating ADM Deployment Units from Database Types" on page 64
- "Creating ADM Deployment Units from Repository Types" on page 75
- "Consolidating ADM Deployment Units from File Types" on page 78
- "Creating the ADM Package Directory Structure" on page 78
- "Consolidating ADM Deployment Units into the Deployment Package Directory" on page 79
- "Generating the ADM Package" on page 80
- "Validating the ADM Package" on page 80

For more information on the overall ADM process or deploying an ADM Package, see:

- Chapter 3, "Overview of the ADM Deployment Process"
- Chapter 6, "Deploying an ADM Package"

See Figure 5 for a high-level overview of the ADM packaging process as it fits in the overall ADM process.

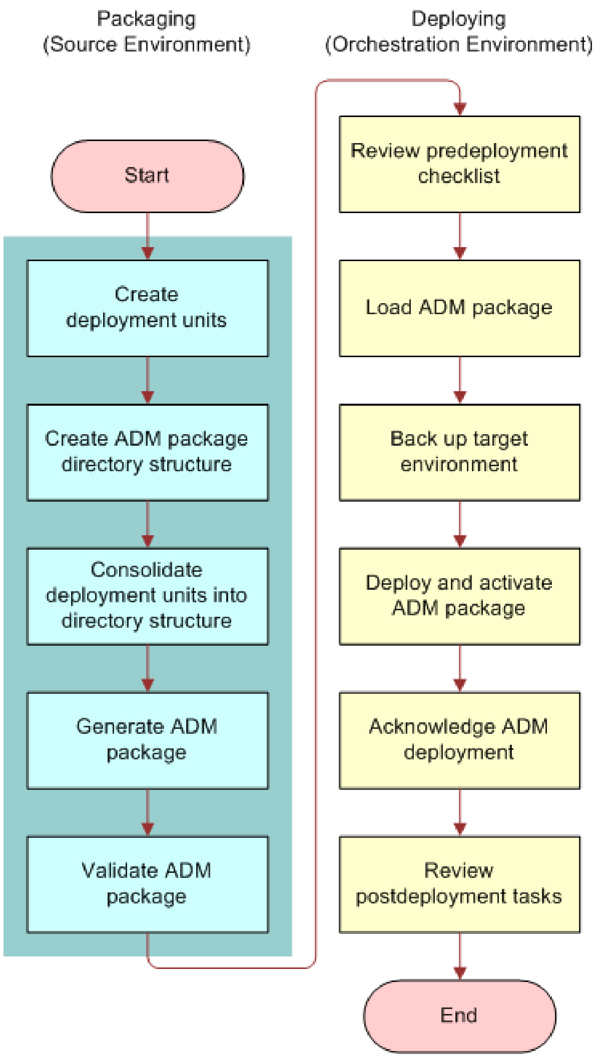

Packaging
(Source Environment)
Deploying
(Orchestration Environment)
Start
Create deployment units
Create ADM package directory structure
Consolidate deployment units into directory structure
Generate ADM package
Validate ADM package
Review predeployment checklist
Load ADM package
Back up target environment
Deploy and activate ADM package
Acknowledge ADM deployment
Review postdeployment tasks
End

Figure 5. High-level Overview of ADM Packaging Process

About the ADM Package Utility Modes and Switches

The ADM Package Utility creates and validates the ADM deployment package required by the ADM framework to deploy Siebel application customizations to a target environment. For more information on this utility, see “About the ADM Package Utility” on page 17.

There are three ADM Package Utility modes:

- **Init.** Creates an empty ADM package directory structure. Deployment files must be manually copied into this directory structure before using the other ADM Package Utility functions. For more information on this task, see "Creating the ADM Package Directory Structure" on page 78.
- **Generate.** Creates the ADM package descriptor file. For more information on this task, see "Generating the ADM Package" on page 80.
- **Validate.** Validates an existing ADM package. The package directory structure must be in place and the generate function executed before running the validate function. For more information on this task, see "Validating the ADM Package" on page 80.

NOTE: If the mode is not stated in the admpkgr command, generate mode is the default mode.

The ADM Package Utility uses several switches, as described in Table 6, depending on the packaging option required. To modify the switches, edit the admpkgr.bat file.

TIP: For more information on copying the ADM Package Utility to other environments, see "Copying the ADM Package Utility to a UNIX Environment" on page 62.

Table 6. ADM Package Utility Switches

Switch	Long Name	Required	Default	Description
-r	-registryfile	No	Not applicable	ADM registry file path.
-g	-language directory	No	ENU	Used in init mode to create language subdirectories.
-h	-help	Not applicable	Not applicable	Lists help for ADM package commands.
-i	-ignoreerrors	No	None	Ignores errors while packaging. This flag allows for the creation of the package descriptor file even if the validation steps fail. However, if an error prevents the creation of the package descriptor file, this flag is ignored.
-l	-logdirectory	No	Current directory	Location of ADM Package Utility log files.
-p	-package directory	No	Current directory	Location of ADM Package folder containing ADM deployment units.
-v	-version	Not applicable	Not applicable	Prints ADM Package Utility version.

The ADM Package Utility returns the values described in Table 7, after execution in each of the three modes.

Table 7. ADM Package Utility Return Codes

Mode	Result	Return Code Value
Init	Errors (directory not created)	3
	Warnings	1
	Success	0
Generate	Errors (no descriptor file)	3
	Errors (with descriptor file)	2
	Warnings	1
	Success	0
Validate	Errors	3
	Warnings	1
	Success	0

Copying the ADM Package Utility to a UNIX Environment

This topic describes copying the Application Deployment Manager (ADM) Package Utility to a UNIX environment. By default, the ADM package utility is installed with the Management Server or Siebel Tools.

NOTE: The Unix environment must have a compatible JRE installed and included in the PATH environment variable.

To copy the ADM Package Utility to a UNIX environment

1. Navigate to the ADMPKGR directory of a Siebel Tools environment, for example, *Siebel_Tools_Root/*ADMPKGR.
2. Copy the ADMPKGR directory to the UNIX environment using FTP.
3. Open the admpkr.ksh file in the following directory: `ADMPKGR/BIN/admpkr.ksh.`
4. Set the parameter, PACK_LOC, with a value describing the directory where deployment packages are present on the UNIX environment.
5. Navigate to the `ADMPKGR/BIN` directory on the UNIX environment, and execute ADM Package Utility commands using the admpkgr.ksh file.

Process of Creating an ADM Package

The process of creating an Application Deployment Manager (ADM) package is the first phase in migrating application customizations to a new environment using the ADM framework. For more information on the overall migration process, see Chapter 3, "Overview of the ADM Deployment Process." For more information on deploying an ADM package, see Chapter 6, "Deploying an ADM Package."

To create an ADM package, perform the following tasks:

1. "Reviewing the ADM Data Type Requirements" on page 63

   This task helps you determine how to package the application customizations.

2. "Consolidating ADM Data Types into Deployment Units" on page 64

   This task consolidates the application customizations in a format that can be packaged.

3. "Creating the ADM Package Directory Structure" on page 78

   This task creates the directory structure used by the ADM framework.

4. "Consolidating ADM Deployment Units into the Deployment Package Directory" on page 79

   This task consolidates the ADM deployment units into the ADM package directory structure.

5. "Generating the ADM Package" on page 80

   This task generates the necessary ADM package descriptor files. For information about ADM package descriptor files, see "About the ADM Package Descriptor File" on page 81.

6. "Validating the ADM Package" on page 80

   This task validates the ADM package contents in preparation for deployment.

Reviewing the ADM Data Type Requirements

This task is a step in "Process of Creating an ADM Package" on page 63. Reviewing the data type requirements prior to creating ADM packages helps you determine how to organize your data into deployment units for a more efficient deployment of your customizations. For more recommendations on facilitating a successful deployment, see:

- "Managing ADM Deployment Dependencies" on page 109
- "Managing Your ADM Package Content" on page 110
- "Minimizing Downtime When Deploying ADM Packages" on page 110

Consolidating ADM Data Types into Deployment Units

This task is a step in "Process of Creating an ADM Package" on page 63. Consolidating Application Deployment Manager (ADM) data types into deployment units is a task based on the category of ADM data type: database, repository, or file. Consolidating the units prepares the application customizations for the packaging stage of the process. For more information on specific data types, see Appendix A, "ADM Supported Data Types."

See the following topics for descriptions of consolidating ADM data types into deployment units based on ADM data category:

- "Process of Creating ADM Deployment Units from Database Types" on page 64
- "Creating ADM Deployment Units from Repository Types" on page 75
- "Consolidating ADM Deployment Units from File Types" on page 78

Process of Creating ADM Deployment Units from Database Types

Creating Application Deployment Manager (ADM) deployment units from database types requires accessing the source Siebel application ADM GUI screens. From these screens you create an ADM deployment project, which contains one or more of the ADM data types. At this point, the ADM data types can contain relationships and be filtered for specific data. The project is then exported to file by creating a deployment session. The exported file serves as the deployment unit for the ADM database types.

NOTE: You can also create ADM database deployment units directly, bypassing the ADM GUI and using the Server Manager command-line interface. For information on this task, see "Creating ADM Deployment Units from Database Types Directly from Command-Line Interface" on page 74.

For information on creating deployment units for repository or file types, see "Creating ADM Deployment Units from Repository Types" on page 75 or "Consolidating ADM Deployment Units from File Types" on page 78.

To create a deployment unit from one or more ADM database types, perform the following tasks:

1. If necessary, "Creating ADM Data Type Relationships" on page 65.
2. "Creating ADM Deployment Projects" on page 66.
3. "Creating ADM Deployment Filters" on page 70
4. "Enabling the ADM Deployment Project" on page 71
5. "Creating ADM Deployment Sessions" on page 71
6. Deploying the ADM session to file by using the ADM GUI or Server Manager command-line interface. For information on these tasks:
   - "Deploying ADM Sessions to File Using ADM GUI" on page 72
   - "Deploying ADM Sessions to File Using Command-Line Interface" on page 73

About ADM Data Type Relationships

Creating Application Deployment Manager (ADM) data type relationships is only necessary if certain data types must be deployed before other data types. Creating this relationship makes sure that related data types are deployed as a single transaction to the target system. If any of the related data types are not applied correctly, then all related data types are not migrated to the target system.

ADM data type relationships are relationships between data types; that is, there are also relationships between different entities internally within data types. These relationships are defined within the Integration Object for a data type. For descriptions on creating or removing data type relationships, see "Creating ADM Data Type Relationships" on page 65.

Example of ADM Data Type Relationship

An example of an ADM data type relationship is as follows: a List of Values (LOV) with a State Model configured on that LOV. Therefore, the LOV must be governed by a State Model.

To set up this relationship in ADM add the LOV data type as a child record to the State Model data type.

When this relationship is set up, the records of the LOV (LOV Type record and LOV value records) are inserted into the database first followed by the State Model records. If there is an error in moving either the State Model or the LOV records, then both are not allowed to exist separately on the target system. If the State Model encounters a database error, then the previously inserted LOV records are removed (rolled back) so that the errors are corrected and the session can be retried.

Creating ADM Data Type Relationships

Creating Application Deployment Manager (ADM) data type relationships is a step in "Process of Creating ADM Deployment Units from Database Types" on page 64. For more information on ADM data type relationships, see "About ADM Data Type Relationships" on page 65.

Set up parent-child relationships between data types using the Data Type Explorer view.

NOTE: The deployment of child data types will be attempted before the parent data types.

To add a child relationship to an ADM data type

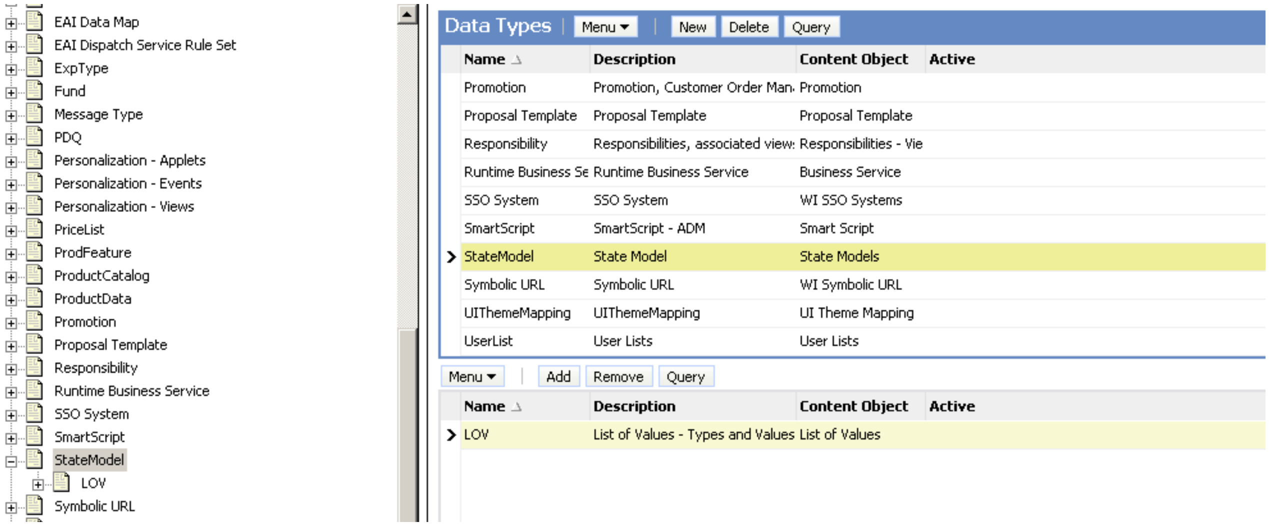
1. Navigate to the Application Deployment Manager screen > Data Type Explorer view.
2. In the Data Types list or Data Type explorer list, select the data type of interest.
3. Click the Add button to access the data type list.
4. Select the child data type of interest from this list.

   The selected child data type appears under the parent data type in the Data Type Explorer view.

To remove a child relationship from an ADM data type

1. Navigate to the Application Deployment Manager screen > Data Type Explorer view.
2. In the Data Types list or Data Type explorer list, select the data type of interest.
3. In the child data type list, select the child data type you want to remove, and click the Remove button.

Figure 6 shows an example of creating an ADM data type child relationship between State Model and LOV data types.



- EAI Data Map
- EAI Dispatch Service Rule Set
- ExpType
- Fund
- Message Type
- PDQ
- Personalization - Applets
- Personalization - Events
- Personalization - Views
- PriceList
- ProdFeature
- ProductCatalog
- ProductData
- Promotion
- Proposal Template
- Responsibility
- Runtime Business Service
- SSO System
- SmartScript
- StateModel
  - LOV
- Symbolic URL

Name	Description	Content Object	Active
Promotion	Promotion, Customer Order Man	Promotion	
Proposal Template	Proposal Template	Proposal Template	
Responsibility	Responsibilities, associated view	Responsibilities - Vie	
Runtime Business Se	Runtime Business Service	Business Service	
SSO System	SSO System	WI SSO Systems	
SmartScript	SmartScript - ADM	Smart Script	
StateModel	State Model	State Models	
Symbolic URL	Symbolic URL	WI Symbolic URL	
UIThemeMapping	UIThemeMapping	UI Theme Mapping	
UserList	User Lists	User Lists	

Name	Description	Content Object	Active
LOV	List of Values - Types and Values	List of Values	

Figure 6. Creating a Child Data Type Relationship

Creating ADM Deployment Projects

Creating Application Deployment Manager (ADM) deployment projects is a step in "Process of Creating ADM Deployment Units from Database Types" on page 64. A *project* is the template that can be reused to export the same data types on a regular basis. A project can contain many data types.

To create an ADM deployment project

1. Navigate to the Application Deployment Manager screen > Deployment Projects view.
2. In the Deployment Projects list, click the menu button and then New Record.

3 Fill in the project record fields as necessary. See the following table for more information on some of these fields.

Field	Description
Name	Name of the deployment project.
Target System	Target system for deploying the migration information. This must be the URL to the target EAI object manager. **NOTE:** This parameter is required only for HTTP online deployment.
Target User Name	User name required to access the target system. **NOTE:** This parameter is required only for HTTP online deployment.
Active Flag	Set by default, this flag signifies the project is available for deployment as a deployment session. Clearing this field makes sure that this project is not deployed.
Export to File	Make sure to check this field to create the export file.
Session Configurable	Check this field if further configurations are required when accessing this project as a deployment session. If this value is unchecked, the project field, Export to File, is read-only, as are the project data type fields, Active and Deployment Filter.

4 In the Deployment Project data type list, populate the new project with data types by clicking the menu button then New Record. For each record:

a Select a previously created data type from the Data Type Name drop-down list.

NOTE: If the data type of interest is not available, it may be set as Inactive. Make sure the Active field for the data type is checked in the Data Type Details view.

b Fill in the other data type record fields as necessary. See the following table for more information on some of these fields.

Field	Description
Active	Clear the active field to inactivate a data type within a project. Use this feature to migrate a deployment project without migrating the data from the inactivated data types.

Field	Description
Deployment Filter	See "Creating ADM Deployment Filters" on page 70 for more information on this field.
Deployment Mode	The type of deployment mode for the data type. There are four possible selections based on the data type: ■ Synchronize. The business object instance in the database is synchronized with the object being imported; that is, the deployment can result in inserts, updates, or deletes. **NOTE:** Child data will be deleted to match the imported object. ■ Upsert. The business object instance in the database is updated with the object that is imported; that is, the deployment can result in inserts or updates (but not deletes). ■ Insert. Available only to new objects (root components) that are imported (not a generic choice but supported for special cases only, for example, iHelp data type). ■ Custom. Available only for deployment mode defined by custom (non-EAI) data types, for example, C/OM Workspace Objects.

c Click the menu button and then Save Record.

5 Add other data types as required to the draft deployment project.

NOTE: If a relationship is set up between two data types, only the parent data type is visible in the picklist. However, after adding the data type to the project item list, the parent data type record can be expanded to reveal the child data types. Make sure to save the record if you cannot expand the parent data type.

6 From the Deployment Projects list, click the menu button and then Save Record.

About ADM Deployment Filters

Application Deployment Manager (ADM) deployment filters allow the selection of specially selected records of a particular data type for export. Creating ADM deployment filters is a part of creating ADM deployment projects. A deployment filter must generally be specified for all deployment units. For information on creating deployment filters in deployment projects, see "Creating ADM Deployment Filters" on page 70.

You can also create ADM deployment filters based on predefined queries (PDQs). For information on this task, see "Creating ADM Deployment Filters with PDQs" on page 70.

ADM deployment filters can also be set or modified when creating deployment sessions. See "Creating ADM Deployment Sessions" on page 71 for more information on this task.

Each data type is represented by a Siebel integration object. The deployment filter applies to the primary business component defined within the integration object, and can be set on any field defined on the primary business component and present in the integration object. For detailed information on business components, see *Configuring Siebel Business Applications*.

Filtering is also possible on the child business components to exclude or include certain records. See the end of this topic for an example.

The format for the deployment filter value entered into the Filter field of the data type record is as follows:

`[`*`Field_Name`*`] `*`operator`*` '`*`Filter_Criterion`*`'`

where:

- *`Field_Name`* is the name of the field on which to filter records. For a list of available fields for each data type, click the Select button in the Field List field. Make sure the field name is enclosed in brackets.
- *`operator`* is a standard Siebel query operator that defines the filter, for example, `like` or =. For a list of Siebel query operators, see *Using Siebel Tools*.
- *`Filter_Criterion`* is the criterion by which to filter the field name. Use an asterisk (*) as the wildcard character for part of the criterion. For example, to filter by names beginning with an *A* use *'A*'* for the criterion. The filter criterion is case-sensitive and must be enclosed in quotes. Use multiple filters for one data type by using the logical AND or OR operators.

NOTE: You cannot use a function as a filter criterion. For example, a filter such as `[Name] = GetProfileAttr ('VODImpExpSpec')` is not supported.

Some deployment filter examples follow:

- `[Name] like 'ABC*' or [Name] = 'My StateModel'`

  This example filters the data type State Model using the Name field. The primary business component for State Models is State Model, the field containing the state model name is Name.

- `[Activation Date]>'12/29/2001 14:58:29'`

  This example filters the data type Assignment Rules using the Activation Date field. All assignment rules with a start date later than 12/29/2001 are filtered for migration.

- `[Value]='ACCOUNT_STATUS' AND [List Of Values Child (UDA).Language] ='ENU'`

  This example filters the data type LOV using the Value field and the child business component List of Values Child (UDA).Language. The filter allows deployment of only the English language LOVs for the ACCOUNT_STATUS LOV type.

Creating ADM Deployment Filters

Creating Application Deployment Manager (ADM) data filters is an optional, but highly recommended, task of the ADM Server Manager GUI set-up process. This task is a step in "Process of Creating ADM Deployment Units from Database Types" on page 64.

The deployment filter allows the migration of specially selected records of a particular data type. It is recommended to use a filter for each data type. For more information about deployment filters, see "About ADM Deployment Filters" on page 68.

You can also create ADM deployment filters based on predefined queries (PDQs). For information on this task, see "Creating ADM Deployment Filters with PDQs" on page 70.

To create an ADM deployment filter

1. Navigate to the Application Deployment Manager screen > Deployment Projects view.
2. In the Deployment Projects list, select the project of interest.
3. In the Deployment Project data types list, select the data type of interest.
4. In the Deployment Filter field, create a search expression to filter only those items of a data type matching the condition for migration.

   NOTE: For more information on the format for deployment filters, see "About ADM Deployment Filters" on page 68.

5. To make sure the filter is accurate, click the Validate Filter button.

Creating ADM Deployment Filters with PDQs

Creating ADM deployment filters with predefined queries (PDQs) provides an alternate means of identifying specific records of a particular data type for deployment. This task is a step in "Consolidating ADM Data Types into Deployment Units" on page 64. For more information on creating deployment filters alone, see "Creating ADM Deployment Filters" on page 70 and "About ADM Deployment Filters" on page 68.

To create ADM deployment filters with PDQs

1. In the corresponding view of the data you want to deploy, query for the records of interest.
2. From the application-level menu, select Query, and then Save Query As. This will save the query as a PDQ.

   For example, navigate to the Administration - Assignment screen > Assignment Rules List view, query for all assignment rules starting with the letter *A,* and save the query.

   For information on querying and PDQs, see *Siebel Fundamentals*.

3. Navigate to the Application Deployment Manager screen > Deployment Projects view.
4. In the Deployment Projects list, select the project of interest.

5. In the Deployment Project data types list, select the data type of interest.

   In this example, select the Assignment Rules data type (AssignRule).

6. In the Deployment Filter field, select the saved PDQ from the drop-down list.

   In this example, select the PDQ saved from the Assignment Rules List view.

   The Deployment Filter field is populated with the PDQ, and the PDQ field contains the name of the saved query.

7. To make sure the filter is accurate, click the Validate Filter button.

Enabling the ADM Deployment Project

Enabling the Application Deployment Manager (ADM) deployment project is a step in "Process of Creating ADM Deployment Units from Database Types" on page 64.

Enabling the project:

- Validates data filters created on the data types.
- Enables the project for use in deployment sessions.
- Locks the project values; that is, none of the project values can change except the Active field. (If you deactivate the project value by clearing the Active field, you cannot deploy the project.)

NOTE: After a project is enabled, you cannot update it, but you can deactivate it to prevent further usage. To update an enabled project, copy the project, and provide a different name.

To enable the ADM deployment project

1. Navigate to the Application Deployment Manager screen > Deployment Projects view.
2. In the Deployment Projects list, select the draft deployment project of interest. (The Status field of the draft deployment project record appears as Draft.)
3. Click the Enable button to activate the project.

   ADM populates the Status field with Enabled and the Publication Date/Time field with the date and time of the project activation.

Creating ADM Deployment Sessions

Creating deployment sessions using the Application Deployment Manager (ADM) is a step in "Process of Creating ADM Deployment Units from Database Types" on page 64.

Before creating a deployment session, make sure that the required configurations are complete and a deployment project exists. See "Creating ADM Deployment Projects" on page 66 for information on creating deployment sessions and other required configurations.

To create a deployment session

1. Navigate to the Application Deployment Manager screen > Deployment Sessions view.
2. In the Deployment Sessions list, click the menu button and then New Record.
3. In the Project Name field drop-down list, select the deployment project of interest.

   ADM populates several fields in the record with information from the deployment project.

   NOTE: Only enabled deployment projects are available for addition to a deployment session. See "Enabling the ADM Deployment Project" on page 71 for more information on this task.

4. If necessary, clear the Deployment Lock field to change the Deploy button to read-only status. (Click the Refresh button to see this change.)

   When the Deployment Lock field is selected, the individual listed in the Locked By field is the only person who can deploy the deployment session. (In normal operations, the Deployment Lock flag is always checked.)

   NOTE: The Deployment Lock flag is useful in complex environments where multiple individuals work together. For example, one person can create and unlock the session, which allows another person to log in, lock, and deploy that session.

5. If necessary, change the values in the fields of the deployment project's data types.

   These fields are accessible only if the deployment project field Session Configurable was checked during the creation of the deployment project. For more information on these fields, see "Creating ADM Deployment Projects" on page 66.

6. To refresh the data in the Deployment Sessions view, click the Refresh button.

Deploying ADM Sessions to File Using ADM GUI

Deploying sessions using the Application Deployment Manager (ADM) GUI is one means of creating an ADM deployment unit and is a step in "Process of Creating ADM Deployment Units from Database Types" on page 64. For other options, see "Deploying ADM Sessions to File Using Command-Line Interface" on page 73.

Before deploying a session, make sure that the required configurations are complete and a deployment session exists. See "Creating ADM Deployment Sessions" on page 71 for information on creating deployment sessions and other required configurations.

To deploy an ADM session to file

1. Access the source system; that is, the system from which you want to migrate the data.
2. Navigate to the Application Deployment Manager screen > Deployment Sessions view.
3. Select the deployment session of interest. Make sure the Export to File and the Deployment Lock fields are checked.
4. Click the Deploy button.

5. Enter the network path for the export directory, then click the Export button. (Do not specify a filename. ADM automatically generates a filename based on the items within the session.)

   For example, on Windows, if the network administrator sets up a shared directory called stage_1 on the server OMEGA, then the network path supplied to ADM must be `\\OMEGA\stage_1`.

   Click the Refresh button to update the status of the deployment.

   NOTE: The network path is relative to the Siebel Server. It is recommended to set up a common area for file operations, which is accessible from all the Siebel Servers. Additional setup tasks may be necessary depending on the operating system on which the Siebel Server is running.

6. Navigate to the network path entered in the previous step to review and access the export file.

NOTE: The volume of data and the number of items within a session contribute to the processing time.

Deploying ADM Sessions to File Using Command-Line Interface

Deploying Application Deployment Manager (ADM) sessions using the Siebel Server Manager command-line interface program (srvrmgr) is one means of creating an ADM deployment unit and is a step in "Process of Creating ADM Deployment Units from Database Types" on page 64. For other options, see "Deploying ADM Sessions to File Using ADM GUI" on page 72.

Before deploying sessions using the command-line interface, make sure that the required configurations are complete and the deployment sessions exist. See "Creating ADM Deployment Sessions" on page 71 for information on creating deployment sessions and other required configurations. Also make sure ADM batch workflows are deployed and activated.

To export a deployment session file using Server Manager command-line interface

1. In the source system, make sure that the deployment sessions intended for srvrmgr deployment have the field, Export to File, checked.

   See "Creating ADM Deployment Sessions" on page 71 for more information on deployment sessions.

2. In the source system, make sure the Deployment Lock value is checked and your Server Manager user ID is in the Locked By field for each session.

   NOTE: You can migrate only deployment sessions that are locked with your user ID.

3. Start Server Manager command-line interface (srvrmgr program) with your user ID.

   For more information on this procedure, see *Siebel System Administration Guide*.

4. Verify that the server component Workflow Process Manager (WfProcMgr) is online (status is Running) by entering the following command:

```
list comp WfProcMgr for server Siebel_Server_Name
```

If it is not running and online, enable the Workflow component group and restart the Siebel Server. For more information on these procedures, see *Siebel System Administration Guide*.

5 Enter the following command at the Server Manager command-line interface prompt:

```
start task for comp WfProcMgr server Siebel_Server_Name with ProcessName="UDA Batch
Deployment", RowID="Session_ID"
```

where:

- *Siebel_Server_Name* is the name of the Siebel Server.
- UDA Batch Deployment is the name of the workflow process. This value is case-sensitive.
- *Session_ID* is the session ID of the deployment session created earlier through the ADM GUI.

Creating ADM Deployment Units from Database Types Directly from Command-Line Interface

Creating ADM deployment units from individual database types can be performed from the Siebel Server Manager command-line interface (srvrmgr), bypassing the ADM GUI tasks of creating projects and deployment sessions. For more information on this process, see "Process of Creating ADM Deployment Units from Database Types" on page 64.

This task starts a component job for the server component Application Deployment Manager Batch Processor (alias ADMBatchProc). For more information on ADMBatchProc, see "Application Deployment Manager Batch Processor (ADMBatchProc)" on page 26.

For more information on administering the server components and using the Server Manager command-line interface, see *Siebel System Administration Guide*.

To create ADM deployment units directly from command-line

1 Start Server Manager command-line interface (srvrmgr program) with your user ID.

2 Verify that the server component Application Deployment Manager Batch Processor (alias ADMBatchProc) is online.

If not, enable the Application Deployment Manager (ADM) component group and restart the Siebel Server. For more information on these procedures, see *Siebel System Administration Guide*.

3 Enter the following command at the Server Manager command-line interface prompt:

```
start task for comp admbatchproc with admpath=file_location,
admdatatype=database_type, admfilter=filter_argument, admmethod=method_name,
admprefix=export_file_name
```

where:

- *admpath* is the location for the resultant export files.

  - *admdatatype* is the name of the data type.

  - *admfilter* is the filter to be applied on the data type, for example, `'[List Of Values Parent (UDA).Value]="NM"'`. For more information on creating filters, see "About ADM Deployment Filters" on page 68.

  - *admmethod* is one of three methods: upsert, synchronize, or custom. For more information on these methods, see Deployment Mode in Step 4 on page 67 of "Creating ADM Deployment Projects" on page 66.

  - *admprefix* is the prefix for the name of the exported XML file.

4 Use the resultant XML file as the deployment unit in the remainder of the packaging process.

Creating ADM Deployment Units from Repository Types

Creating Application Deployment Manager (ADM) deployment units from repository data types requires the use of Siebel Tools. For more information on using Siebel Tools or configuring Siebel applications with Siebel Tools, see the following documentation:

- *Using Siebel Tools*

- *Configuring Siebel Business Applications*

For information on creating deployment units for database types or file types, see "Consolidating ADM Data Types into Deployment Units" on page 64.

You can create deployment units for repository data types by:

- **Generating a mid-level release.** This task exports the repository objects after a certain date and is used for larger-scale deployments. For information on this task, see "Generating ADM Deployment Units Using a Mid-Level Release" on page 76.

- **Generating a hotfix.** This task exports only selected repository objects and is generally used for smaller-scale deployments. For information on this task, see "Generating ADM Deployment Units Using a Hotfix" on page 76.

- **Using the command-line consoleapp.exe program.** This program creates the necessary files used as deployment units. For information on this task, see "Using Consoleapp.exe to Create Repository Deployment Units" on page 77.

The subsequent .sif file, created by the previous tasks, is the deployment unit used in the ADM packaging process. In addition to the .sif file, an XML unit descriptor file is also created that includes information on the repository objects and is used for validation during the packaging task. Both files are necessary for the ADM deployment package.

NOTE: To migrate database schema changes, do not use Siebel Tools. Instead, use the Repository Migration tool in the Database Configuration Wizard. For more information on Repository Migration, see *Going Live with Siebel Business Applications*.

Additional care must be taken when managing the deployment of repository data. For information on synchronizing repository updates, see "Synchronizing Repository Updates" on page 111.

Generating ADM Deployment Units Using a Mid-Level Release

Generating ADM deployment units from repository data types uses the Siebel Tools mid-level release feature. This task creates a .sif file that is used as the deployment unit necessary for the ADM packaging process.

Generating a mid-level release uses the Siebel Tools GUI to complete the task as described in the following procedures.

To generate ADM deployment units using mid-level release at the Siebel Tools GUI

1. Log in to Siebel Tools, and make sure the changes are checked in.
2. From the menu bar, navigate to View > Options > General Tab.
3. Set the date in the Changed Date form.

   This date determines the objects changed after the set date and includes them in the mid-level release.
4. From the menu bar, navigate to Tools > Generate Mid-Level Release.
5. In the Generate Mid-level Release dialog, name the release, and click the Generate List button.
6. Review the repository objects included in the mid-level release and, if necessary, remove any objects from the list by using the Delete key.
7. Select Export to create the .sif file used as the ADM deployment unit.
8. Navigate to the Siebel Tools installation directory and the directory with the mid-level release label to locate the .sif file with the mid-level release label. Copy the generated .sif file along with the XML unit descriptor file, which will be used in the ADM package.

Generating ADM Deployment Units Using a Hotfix

Generating ADM deployment units from repository data types uses the Siebel Tools hotfix feature. This task creates a .sif file that is used as the deployment unit necessary for the ADM packaging process.

Generating a hot fix uses the Siebel Tools GUI to complete the task as described in the following procedure.

To generate ADM deployment units using a hotfix at the Siebel Tools GUI

1. Log in to Siebel Tools, and make sure the changes are checked in.
2. On each object that you want to export, right-click and select Add to Hotfix. You can select multiple objects to add to the Hotfix.
3. In the Hotfix Release dialog, name the release and review the repository objects included in the mid-level release.

4 If necessary, remove any objects from the list by using the Delete key.

5 Select Export to create the .sif file used as the ADM deployment unit.

6 Navigate to the Siebel Tools installation directory and the ADM directory to locate the .sif file with the mid-level release label. Copy the generated .sif file along with the XML unit descriptor file, which will be used in the ADM package.

Using Consoleapp.exe to Create Repository Deployment Units

Using the consoleapp.exe program is one means of creating repository deployment units. For more information on consoleapp.exe and its use, see *Using Siebel Tools*. This task allows you to export a predefined list of objects from Siebel Tools.

To use the Siebel Tools GUI to create repository deployment units, see "Creating ADM Deployment Units from Repository Types" on page 75.

To generate ADM deployment units using consoleapp.exe program

- Launch the Siebel Tools command-line executable consoleapp.exe using either a full set of arguments or a reference to the arguments contained in a file.

  For more information on using this utility, see *Using Siebel Tools*.

Argument set at launch:

The following argument is set when launching the consoleapp.exe program:

```
consoleapp.exe configuration_file.cfg language username password "Siebel Tools
Export Support for ADM" “Export:ArgumentList"
```

For example,

```
consoleapp.exe

"D:\Tools\bin\enu\tools.cfg"

ENU SADMIN SADMIN

"Siebel Tools Export Support for ADM"

"Export:

Repository=Siebel Repository,LogFile=D:\nm_export.log,

ExportFile=D:\export_data.sif,

DescriptorFile=D:\export_desc.xml,

Object_1=Account List Applet,Type_1=Applet,ExportCount=1"
```

File reference at launch:

```
consoleapp.exe configuration_file.cfg language username password /f
export_argument_file.xml
```

For example,

```
consoleapp.exe "D:\tools.cfg"

enu SADMIN SADMIN /f "D:\exportArgfile.xml“
```

Consolidating ADM Deployment Units from File Types

File types use the native file format as the deployment unit and must be manually collected in preparation for packaging. Review ADM file-type data types in Appendix A, “ADM Supported Data Types” for information on available file types for migration, their file locations, and any other considerations. For information on creating the deployment units for database types or repository types, see “Consolidating ADM Data Types into Deployment Units” on page 64.

Creating the ADM Package Directory Structure

This task is a step in “Process of Creating an ADM Package” on page 63. Creating the Application Deployment Manager (ADM) package directory is a task that uses the ADM Package Utility to create the ADM package directory structure based on the applicable ADM registry file. For information on the ADM Package Utility or the ADM registry file, see:

- “About the ADM Package Utility Modes and Switches” on page 60
- “About the ADM Registry” on page 22

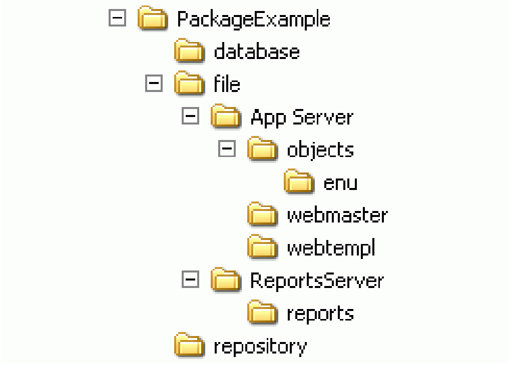
To create the ADM package directory structure

1. Navigate to an accessible location that contains the ADM package directory structure. It must be accessible with at least read privileges by the user IDs running the Siebel Server process and the ADM agents process.
2. If necessary, set optional switches, as described in Table 6 on page 61, in the admpkgr.bat file.
3. Execute the ADM Package Utility with the following command:

   ```
   admpkgr.bat init package_name
   ```

   where *package_name* is the name of the ADM package folder that contains the application customizations to be deployed.

See Figure 7 for an example of the directory structure created with the ADM Package Utility.



```
⊟ PackageExample
     database
  ⊟ file
     ⊟ App Server
        ⊟ objects
             enu
           webmaster
           webtempl
     ⊟ ReportsServer
           reports
     repository
```

Figure 7. Example of ADM package directory structure created with the ADM Package Utility

Consolidating ADM Deployment Units into the Deployment Package Directory

This task is a step in "Process of Creating an ADM Package" on page 63. Consolidating the Application Deployment Manager (ADM) deployment units into the deployment package directory requires first manually creating the deployment unit files as defined in "Consolidating ADM Data Types into Deployment Units" on page 64.

To consolidate ADM deployment units into the ADM deployment package directory

1. Locate all the deployment units that make up the application customizations you want to migrate. Deployment units have the following format, based on type:

   - **Database types.** *session_name*.xml and XML unit descriptor file
   - **Repository types.** *export_name*.sif and XML unit descriptor file
   - **File types.** .srf, .gif, .css, .html, and so on

2. Navigate to the root directory created and named in the task "Creating the ADM Package Directory Structure" on page 78.

3. Place the database-type deployment units in the `database` subdirectory.

4. Place the repository-type deployment units in the `Repository` subdirectory.

5. Place the file-type deployment units in the `file` directory and their respective subdirectories as follows:

   - SRF files in the language-specific subdirectory of the `objects` directory
   - Browser script files, image files, and cascading style sheets in the `webmaster` subdirectory
   - Web template files in the `webtempl` subdirectory
   - Report files in the reports subdirectory of the `ReportServer` directory

Generating the ADM Package

This task is a step in "Process of Creating an ADM Package" on page 63. Generating the Application Deployment Manager (ADM) package is a task that uses the ADM Package Utility to run the generate command. The generate command does the following:

- Traverses the package directory structure and validates the contents
- Calculates a checksum for future integrity checks
- Creates a package descriptor file
- Reports any warning or error messages

For information about the ADM package descriptor file, see "About the ADM Package Descriptor File" on page 81.

To generate the ADM package

- Execute the ADM Package utility with the following command:

  ```
  admpkgr generate package_name
  ```

  where *package_name* is the name of the ADM package folder that contains the application customizations to be deployed.

The location of the package folder is set in admpkgr.bat file. You can edit this file and set the default package folder location or make other configurations by setting the switches described in Table 6 on page 61.

Validating the ADM Package

This task is a step in "Process of Creating an ADM Package" on page 63. Validating the Application Deployment Manager (ADM) package is a task that uses the ADM Package Utility to run the validate command. The validate command reviews the package directory structure and contents. This task is generally necessary only if the package descriptor file was manually updated. For information on updating the descriptor file, see "Editing the ADM Package Descriptor File" on page 81.

To validate the ADM package

- Execute the ADM Package Utility with the following command:

  ```
  admpkgr validate package_name
  ```

  where *package_name* is the name of the ADM package folder that contains the application customizations to be deployed.

The location of the package folder is set in admpkgr.bat file. You can edit this file and set the default package folder location or make other configurations by setting the switches described in Table 6 on page 61.

About the ADM Package Descriptor File

The package descriptor file contains information that is grouped by data type: database, repository, and file. Individual deployment unit details are listed in chronological order under each data type. If there are no units for a particular data type in the package, that data type group will not be created. Each unit has details such as:

- **PackageNameAtCreation.** The package name at creation.
- **SchemaVersion.** The aggregate SchemaVersion of all the database units.

The descriptor file has an attribute, ActivateOnly, which can be manually set to TRUE to make the package ActivateOnly. By default this attribute is set to FALSE.

Editing the ADM Package Descriptor File

You can edit the ADM package descriptor file to control the sequence of the deployment of the units under each data type. You can edit the package descriptor file using Microsoft Office InfoPath. For more information on the ADM package descriptor file, see "About the ADM Package Descriptor File" on page 81.

6 Deploying an ADM Package

This chapter covers the process of deploying and activating an Application Deployment Manager (ADM) package. Deploying and activating ADM packages represents the migration of application customizations into a new environment. This chapter includes the following topics:

- "About the ADM Command-Line Interface Syntax" on page 84
- "Process of Deploying an ADM Package" on page 85
- "Reviewing the ADM Predeployment Checklist" on page 86
- "Loading an ADM Package" on page 87
- "Deploying and Activating an ADM Package" on page 90
- "Administering the ADM Deployment Execution" on page 93
- "Acknowledging an ADM Deployment" on page 100
- "Other ADM Deployment Options" on page 101
- "Reviewing Postdeployment Tasks" on page 103

For more information on the overall ADM process or creating an ADM package, see:

- Chapter 3, "Overview of the ADM Deployment Process"
- Chapter 5, "Creating an ADM Package"

See Figure 8 for a high-level overview of the ADM deploying process as it fits in the overall ADM process.

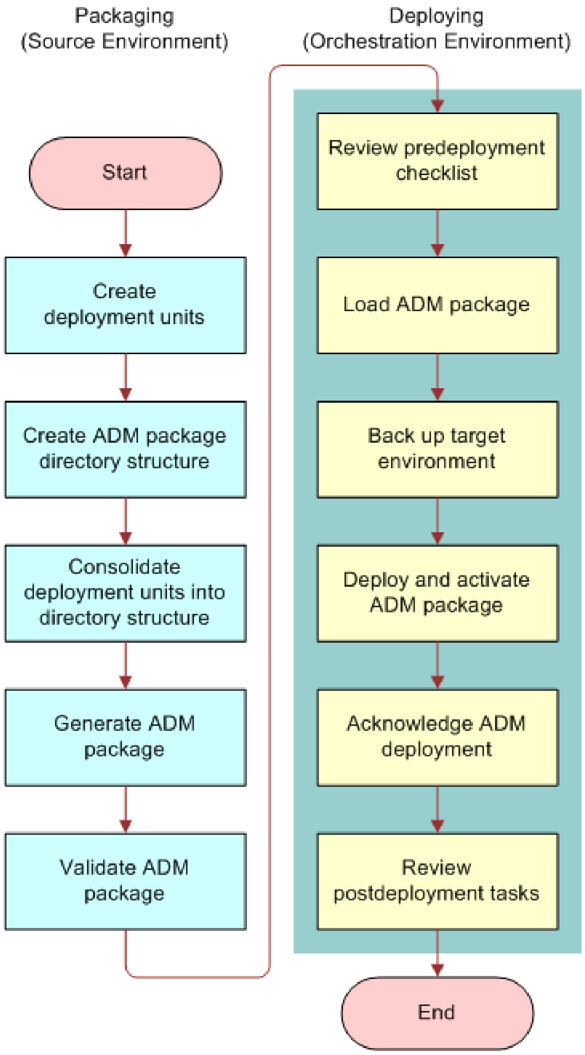

Packaging
(Source Environment)
Deploying
(Orchestration Environment)
Start
Create deployment units
Create ADM package directory structure
Consolidate deployment units into directory structure
Generate ADM package
Validate ADM package
Review predeployment checklist
Load ADM package
Back up target environment
Deploy and activate ADM package
Acknowledge ADM deployment
Review postdeployment tasks
End

Figure 8. High-level Overview of ADM Deploying Process

About the ADM Command-Line Interface Syntax

The Application Deployment Manager (ADM) command-line interface is a Java-based tool executed by running a batch command (.bat) file with the appropriate commands and parameters. The command file is named for the target Siebel Enterprise Server; that is, the batch filename is of the format deploy_*enterprise_name*.bat. For more information on ADM command-line interface, see "About the ADM Command-Line Interface" on page 17.

The general syntax for running this batch file is as follows:

```
deploy_enterprise_name function username password package_name options
```

where:

- *enterprise_name* is the name of the Siebel Enterprise Server.
- *function* is the command used to administer the ADM deployment, such as copy, validate, and so on. These commands are further described in subsequent topics.
- *username* is the Siebel user name.
- *password* is the Siebel password.
- *package* is the name of the ADM deployment package of interest.
- *options* are the optional flags that supplement the function command. For more information on these flags, see Table 8 on page 90 and Table 9 on page 95.

For further assistance, use the help parameter for information on the ADM batch file; that is, run:

```
deploy_enterprise_name -help
```

Process of Deploying an ADM Package

The process of deploying an Application Deployment Manager (ADM) package is the secondary phase in migrating application customizations to a target environment using the ADM framework. For more information on the overall migration process, see Chapter 3, "Overview of the ADM Deployment Process." For more information on creating an ADM package, see Chapter 5, "Creating an ADM Package."

To complete the deployment of an ADM package, perform the following tasks:

1. "Reviewing the ADM Predeployment Checklist" on page 86

   This task confirms the readiness of the package and the environment prior to deployment.

2. "Loading an ADM Package" on page 87

   This task loads the ADM package into the ADM framework.

3. "Backing Up Before Deploying ADM Packages" on page 89

   This task backs up the target environment before deployment.

4. "Deploying and Activating an ADM Package" on page 90

   This task represents the actual deployment of customizations data to the new environment. At this stage, there are several options or contingencies available, which are described in "Administering the ADM Deployment Execution" on page 93.

5. "Acknowledging an ADM Deployment" on page 100

   This task acknowledges that the deployment is complete. When the package has been acknowledged, you cannot restore the original application customizations in the target environment or perform any more operations for that deployment session.

6. “Reviewing Postdeployment Tasks” on page 103

   This task completes the deployment process for those data types that require additional action outside the ADM framework.

NOTE: Steps 1 and 2 need to be performed only once for each package on each Siebel Server. For example, you can deploy the same package on the same Siebel Server to different Siebel Enterprise Servers, but it is necessary to run the checklist and load the package only once.

Reviewing the ADM Predeployment Checklist

Reviewing the Application Deployment Manager (ADM) predeployment checklist is a step in “Process of Deploying an ADM Package” on page 85. This topic lists the items you must review before executing an ADM package deployment.

Review the following tasks before executing an ADM package deployment:

- “Reviewing ADM Data Types” on page 86
- “Validating the ADM Enterprise Profile” on page 86
- “Choosing to Use a Staged or Full Deployment” on page 87

Reviewing ADM Data Types

This task involves reviewing the ADM data types you are deploying. Some of these data types have specific or additional deployment or activation related steps. For more information on individual data types and their requirements, see Appendix A, “ADM Supported Data Types.”

Validating the ADM Enterprise Profile

This task describes validating the Application Deployment Manager (ADM) enterprise profile, which includes checks for correct formatting and if the target servers match those listed in the serverinfosection in the enterprise profile. It also checks if the ADM management server is configured to locate these servers.

The ADM enterprise profile is referenced during the deployment task and must be valid. For more information on the ADM enterprise profile, see “About the ADM Enterprise Profile” on page 23. For information on configuring this file, see“Configuring the ADM Enterprise Profile” on page 49.

To validate the enterprise profile

1. At the ADM command-line interface prompt, enter:

   ```
   deploy_ent_profile_name valent username password
   ```

   where:

   - *ent_profile_name* is the XML ADM enterprise profile name. The default name for this file is entprofile.xml.

- *username* is the Siebel user name.
- *password* is the Siebel password.

2 After running this command, the ADM command-line interface returns either an error or success message indicating the validity of the ADM enterprise profile.

Choosing to Use a Staged or Full Deployment

This task involves deciding whether to use a *staged* or *full* deployment. A full deployment migrates the customizations to all Siebel Servers in a Siebel Enterprise Server in a single deployment of the package without interruption.

A staged deployment migrates application customizations to a subset of Siebel Servers in a Siebel Enterprise Server in multiple deployments staggered across different servers, which minimizes downtime.

Loading an ADM Package

Loading an Application Deployment Manager (ADM) package is a step in "Process of Deploying an ADM Package" on page 85. This task loads the ADM package data into the local Management Server database in preparation for deployment execution.

Loading the ADM package also validates the integrity of the ADM package; that is, it makes sure that the package has not been altered since packaging and that the target environment is ready for this package.

After loading an ADM package into the Management Server, the ADM command-line interface provides additional commands to review and administer this package. For information on these commands, see "Reviewing the ADM Package on the Management Server" on page 88.

To load an ADM package into the management server

1 At the ADM command-line interface prompt, enter:

```
deploy_enterprise_name load username password package_name package_location
```

where:

- *enterprise_name* is the name of the Siebel Enterprise Server.
- *username* is the Siebel user name.
- *password* is the Siebel password.
- *package_name* is the name of the XML ADM package file.
- *package_location* is the location of the ADM package. This parameter is mandatory. An error is displayed if the package location is not specified.

The output displays a "Success" or "Failure" message to indicate whether the package has been loaded into the management server.

2 If necessary, review and administer the ADM package loaded into they ADM management server database by using the commands described in "Reviewing the ADM Package on the Management Server" on page 88.

Reviewing the ADM Package on the Management Server

This topic describes the Application Deployment Manager (ADM) commands that review and administer the ADM packages loaded into the Siebel Management Server database. Using the ADM command-line interface, you can:

- List ADM packages in the Siebel Management Server
- List a specific ADM package in the Siebel Management Server
- List an ADM package and its contents
- Delete an ADM package from the Siebel Management Server

To list ADM packages in the Management Server

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* list *username password*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.

  The output lists the ADM packages loaded in the Siebel Management Server database.

To list an ADM package's contents

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* list_detail *username password package_name*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

  The output lists the deployment units in the specified ADM package loaded in the Siebel Management Server database.

To delete an ADM package from the ADM framework

- At the ADM command-line prompt, enter:

```
deploy_enterprise_name delete username password package_name
```

where:

- *enterprise_name* is the name of the Siebel Enterprise Server.
- *username* is the Siebel user name.
- *password* is the Siebel password.
- *package_name* is the name of the XML ADM package file.

The ADM package and any associated sessions are removed from the Siebel Management Server database.

Backing Up Before Deploying ADM Packages

This task is a step in "Process of Deploying an ADM Package" on page 85. Executing the backup command backs up the target environment's current application customizations that will be updated based on the ADM package and enterprise profile referenced in the command. This process occurs automatically during the deployment command, but running it separately reduces the deployment time.

NOTE: The flag to configure the automatic backup during the deployment command is set in the enterprise profile.

To back up target environment before deploying an ADM package

- At the ADM command-line prompt, enter:

  ```
  deploy_enterprise_name backup username password package_name
  ```

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

  This command can also use the following optional flags: -async and -ignore _validation_errors. For more information on these flags, see Table 8 on page 90.

Deploying and Activating an ADM Package

Deploying and activating an Application Deployment Manager (ADM) package is a step in "Process of Deploying an ADM Package" on page 85. This task represents the migration and activation of the customizations for use in the target environment. This task uses the ADM command-line interface program. For information on this tool, see "About the ADM Command-Line Interface Syntax" on page 84.

Perform the commands in the following tasks to deploy and activate an ADM package:

1. "Creating an ADM Deployment Session" on page 91
2. "Validating an ADM Deployment Environment" on page 91
3. "Deploying an ADM Deployment Session" on page 92
4. "Activating an ADM Deployment Session" on page 92
5. (Optional) "Deploying and Activating an ADM Deployment Session" on page 93

Several optional command-line flags further define the deployment and activation activities and are described in Table 8 on page 90.

Table 8. Optional Command-Line Flags While Deploying and Activating ADM Packages

Flag	Description
-async	Performs the command in asynchronous mode. The copy command returns immediately after issuing the command, but the actual deployment runs in the background. To check the status of the deployment, use the status_detail command. For more information on this command, see "Reviewing an ADM Deployment's Status" on page 97.
-activateonly	Specifies that this package is for activation only.
-ignore_validation_errors	Ignores validation errors output and continues the command.
-syncstatus	Updates the status of the deployment execution if any system errors occur that cause the status reporting on the ADM framework to lose the connection with the target Siebel Enterprise Server.
-t	Acts as a category filter on the ADM package when used with the arguments file, database, or repository.
-fileservers	Deploys the package to only a subset of Siebel Servers within any Siebel Enterprise Server when used with a staged deployment on file servers. Takes the agent name, defined in the enterprise profile, as the argument.

Additionally, there are several administrative commands that might be necessary to run during the deployment task, including starting and stopping the deployment, restoring the previous application customizations, and reviewing deployment status and reports. For more information on these commands, see "Administering the ADM Deployment Execution" on page 93.

Creating an ADM Deployment Session

This command creates an ADM deployment session record, which represents the application customizations that are ready for migration based on the ADM package and enterprise profile referenced at that particular time.

To create an ADM deployment session

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* create *username password package_name*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

  This command can also use the following optional flags: -activateonly and -ignore _validation_errors. For more information on these flags, see Table 8 on page 90.

Validating an ADM Deployment Environment

This command validates the run-time deployment environment by checking the following:

- The schema version of the source environment (source of files) is compatible with the target environment (major schema version must match).
- The ADM agents designated to receive the commands are online.
- The Siebel Servers in the Siebel Enterprise Server referenced in the profile are online.

  NOTE: The ADMProc server component must be enabled and running on each Siebel Server.

To validate an ADM deployment environment

- At the ADM command-line prompt, enter:

  `deploy_`*`enterprise_name`* `validate` *`username password package_name`*

  where:

  `enterprise_name` is the variable for the name of the Siebel Enterprise Server.

  `username` is the variable for Siebel user name.

  `password` is the variable for Siebel password.

  `package_name` is the variable for the name of the XML ADM package file.

  Only errors are returned to the command-line interface.

Deploying an ADM Deployment Session

This command updates the target environment with the application customizations defined by the ADM package and represented by the previously created deployment session.

NOTE: To redeploy the same ADM package, you must acknowledge that session and create a new session to deploy the ADM data. For information on these tasks, see "Acknowledging an ADM Deployment" on page 100 and "Creating an ADM Deployment Session" on page 91.

To deploy an ADM deployment session

- At the ADM command-line prompt, enter:

  `deploy_`*`enterprise_name`* `copy` *`username password package_name`*

  where:

  - *`enterprise_name`* is the name of the Siebel Enterprise Server.
  - *`username`* is the Siebel user name.
  - *`password`* is the Siebel password.
  - *`package_name`* is the name of the XML ADM package file.

  This command can also use the following optional flags: `-t`, `-fileserver`, -async, and -ignore _validation_errors. For more information on these flags, see Table 8 on page 90.

Activating an ADM Deployment Session

This command activates, if possible, the new application customizations in the target environment.

To activate an ADM deployment session

- At the ADM command-line prompt, enter:

  `deploy_`*`enterprise_name`* `activate` *`username password package_name`*

  where:

- *enterprise_name* is the name of the Siebel Enterprise Server.
- *username* is the Siebel user name.
- *password* is the Siebel password.
- *package_name* is the name of the XML ADM package file.

This command can also use the following optional flags: `-t`, `-fileserver`, -async, and -ignore _validation_errors. For more information on these flags, see Table 8 on page 90.

Deploying and Activating an ADM Deployment Session

This combination command can be used as an alternative to running the deploy (copy) and activate commands separately. It updates the target environment with the application customizations defined by the ADM package and represented by the previously created deployment session. This command also activates, if possible, the new application customizations in the target environment.

To deploy and activate an ADM deployment session

- At the ADM command-line prompt, enter:

  ```
  deploy_enterprise_name copy_activate username password package_name
  ```

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

  This command can also use the following optional flags: `-t`, `-fileserver`, -async, and -ignore _validation_errors. For more information on these flags, see Table 8 on page 90.

Administering the ADM Deployment Execution

Administering the Application Deployment Manager (ADM) execution can involve running some of the following tasks:

- "Stopping and Restarting an ADM Deployment Session" on page 94
- "Backing Up Before Restoring an ADM Session" on page 94
- "Creating an ADM Restore Session" on page 95
- "Reviewing an ADM Deployment's Status" on page 97

Stopping and Restarting an ADM Deployment Session

Stopping an ADM deployment is an administrative task that may be necessary if you are running a staged deployment or for other nondeployment reasons.

Restarting a stopped deployment involves reinvoking the original deployment command as described in "Deploying an ADM Deployment Session" on page 92 or "Deploying and Activating an ADM Deployment Session" on page 93. The ADM framework continues to deploy only the units of the deployment session that were not fully deployed initially.

To stop an ADM deployment session

- At the ADM command-line prompt, enter:

  ```
  deploy_enterprise_name stop username password package_name
  ```

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the XML ADM package file.

Backing Up Before Restoring an ADM Session

This command backs up the target environment's current application settings that will be updated by the ADM package referenced in the command. This process normally occurs automatically during the restore command (as specified in the enterprise profile), but running it separately reduces the deployment time.

To back up target environment before restoring an ADM package

- At the ADM command-line prompt, enter:

  ```
  deploy_enterprise_name backup username password package_name
  ```

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

  This command can also use the following optional flags as described in Table 8 on page 90 and Table 9 on page 95.

Creating an ADM Restore Session

Restoring an Application Deployment Manager (ADM) session is a task to restore objects affected by the deployment to their predeployment versions, as well as activate these previous versions. The restore task is subject to various limitations and must be performed while a valid ADM deployment session is active and before the deployment session is acknowledged.

This task involves the following commands to complete:

1. "Switching to an ADM Restore Session" on page 95
2. "Restoring the ADM Deployment Copy" on page 96
3. "Restoring the ADM Deployment Activation" on page 96
4. (Optional) "Restoring the ADM Deployment Copy and Activation" on page 96

These commands can use the following optional flags: `-t, -fileserver`, -async, and -ignore `_validation_errors` defined in Table 8 on page 90. Several optional command-line flags that further define the restore task are described in Table 9.

Table 9. Optional Command-Line Flags While Restoring an ADM Session

Flag	Description
`-l`	Group or unit level for a manual restore
`-id`	Group or unit ID for a manual restore
`-name`	Group or unit name for a manual restore

Switching to an ADM Restore Session

This command changes an existing ADM deployment session into a restore session; that is, the command readies the ADM framework to restore the original application customizations updated by the recently deployed session.

To start an ADM restore session

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* restore_session *username password package_name*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

Restoring the ADM Deployment Copy

This command restores the back-up copy of the application configuration on the target environment and represents the actual restore of the configuration to their prior state before the ADM session deployment.

To restore the ADM deployment copy

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* restore_copy *username password package_name*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

Restoring the ADM Deployment Activation

This command activates the restored copy of the application customizations on the target environment.

To restore the ADM deployment activation

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* restore_activate *username password package_name*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

Restoring the ADM Deployment Copy and Activation

You can use this combination command as an alternative to running the restore_copy and restore_activate commands separately. It restores the backup copy of the application customizations on the target environment and represents the actual restore of the customizations to their prior state before the ADM session deployment. This command also activates the restored copy of the application customizations on the target environment.

To restore the ADM deployment copy and activation

- At the ADM command-line prompt, enter:

deploy_*enterprise_name* restore_copy_activate *username password package_name*

where:

- *enterprise_name* is the name of the Siebel Enterprise Server.
- *username* is the Siebel user name.
- *password* is the Siebel password.
- *package_name* is the name of the XML ADM package file.

Reviewing an ADM Deployment’s Status

This topic describes the Application Deployment Manager (ADM) command-line interface commands that provide a status of the ADM deployment at either the server, group, or unit level or provides feedback on the history of the deployment.

The following topics describe the various status reporting commands of the ADM command-line interface:

- “Reviewing the Status of an ADM Deployment Session” on page 97
- “Creating an ADM Deployment Report” on page 98
- “Creating a Destination Map for File Directories” on page 99

Reviewing the Status of an ADM Deployment Session

This command or combination of commands provides feedback on the progress of the ADM deployment session at various levels.

To review the status summary of an ADM deployment session

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* status *username password package_name*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

  This command displays a summary of the deployment status of a package.

To review the status detail of an ADM deployment session

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* status_detail *username password package_name*

  This command displays the following deployment status details of a package:

  - Package name
  - Siebel Enterprise Server name
  - Session number
  - Session status
  - Session execution status
  - Total number of units
  - Total units completed
  - Total units not started
  - Total units failed
  - Total units running

To review the errors encountered during an ADM deployment session

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* status_error *username password package_name*

  This command displays the error status of the package.

To review the history of deployment sessions based on an ADM package

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* history *username password package_name*

  This command displays the history of all the commands run on that particular package during that particular session.

Creating an ADM Deployment Report

This command creates a report, in the form of an HTML file, detailing the performance of the ADM deployment session, including success or failure information for each database unit.

To create an ADM deployment session report

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* report *username password package_name*

  where:

- *enterprise_name* is the name of the Siebel Enterprise Server.
- *username* is the Siebel user name.
- *password* is the Siebel password.
- *package_name* is the name of the XML ADM package file.

The ADM framework has several logging features that document the ADM deployment process in case of incorrect deployment or other issues. For more information on troubleshooting or logging, see Chapter 8, “ADM Logging Capabilities and Troubleshooting.”

Creating a Destination Map for File Directories

This command creates a report that lists the files that are deployed to the target environment. It also lists the server to which each files was deployed. The default filename for this report is mapfile.xml, which is located in the *Siebel_Management_Server_Root*\log directory. The filename and location can be changed by modifying the value of the ADM_MAP_FILE parameter in the deploy.bat file.

This command uses several switches, as described in Table 10. To modify the switches, edit the deploy.bat file.

Table 10. Switches Used While Creating a Destination Map

Switch	Description	Mandatory or Optional
-f	Function, uses genmap as its value.	Mandatory
-a	All, lists all files that are deployed (including those that failed).	Optional
-filename	Filename, specifies XML output filename. Default filename is mapfile.xml.	Mandatory
-p	Package, specifies the package name.	Mandatory
-en	Enterprise name, specifies the Siebel Enterprise Server name.	Mandatory

To create a destination map of only those files that are successfully deployed

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* genmap *username password package_name*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.

- *password* is the Siebel password.
- *package_name* is the name of the XML ADM package file.

To create a destination map of all files that are deployed (including those that failed)

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* genmap_all *username password package_name*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

Acknowledging an ADM Deployment

Acknowledging an Application Deployment Manager (ADM) package is a step in "Process of Deploying an ADM Package" on page 85. This task closes the deployment session while deleting temporary and backup files used during the deployment.

You can acknowledge a full or partially deployed ADM deployment session, after which, the only commands available to the ADM deployment session are those that review the history and status of the ADM deployment session.

After an ADM deployment session is acknowledged:

- The ADM deployment session cannot be redeployed.
- The restore operation cannot be invoked.

To acknowledge an ADM deployment session

- At the ADM command-line prompt, enter:

  deploy_*enterprise_name* acknowledge *username password package_name*

  where:

  - *enterprise_name* is the name of the Siebel Enterprise Server.
  - *username* is the Siebel user name.
  - *password* is the Siebel password.
  - *package_name* is the name of the XML ADM package file.

Other ADM Deployment Options

The following alternative ADM deployment options are available for database data types only and represent the ADM functionality of previous Siebel Application releases. These deployment options require the creation of an ADM session or an export file representing that ADM session.

NOTE: These deployment options do not have the packaging, backup, activate, or restore functionality of a deployment using the ADM framework.

The alternative ADM deployment options are:

- "Deploying ADM Sessions Using the Application Deployment Manager GUI" on page 101
- "Deploying ADM Sessions Using Export Files and the ADM GUI" on page 102
- "Deploying ADM Sessions Using Command-Line Interface" on page 102

Deploying ADM Sessions Using the Application Deployment Manager GUI

Deploying sessions by using the ADM GUI is one of the alternate means of migrating ADM database data types from one Siebel application environment to another. For other alternative deployment options, see "Other ADM Deployment Options" on page 101.

Before using this deployment option, make sure an ADM session exists in the source environment. To create an ADM session, follow tasks 1 through 5 as outlined in "Process of Creating ADM Deployment Units from Database Types" on page 64.

NOTE: Make sure the parameters Target System and Target User Name are provided when creating the ADM Project.

TIP: A best practice when using this deployment method is to use a project filter for each database item being deployed.

To migrate a deployment session

1. Navigate to the Application Deployment Manager screen > Deployment Sessions view.
2. Select the deployment session of interest. Make sure the Deployment Lock field is checked.
3. Click the Deploy button.
4. Enter the password for the target system user, and then click the Deploy button.

   The Status field of the deployment session record changes to Submitted. Click the Refresh button to further update the status of the deployment.

   NOTE: The volume of data and the number of items within a session contribute to the processing time.

5. Check the details of the deployment by reviewing the ADM log files and the EAI queue.

   For more information about ADM log files, see Chapter 8, "ADM Logging Capabilities and Troubleshooting."

Deploying ADM Sessions Using Export Files and the ADM GUI

Deploying sessions using export files is one of the alternate means of migrating ADM database data types from one Siebel application environment to another. For other alternative deployment options, see "Other ADM Deployment Options" on page 101.

Before using this deployment option, make sure an ADM session exists and the deployment unit XML file has been exported as outlined in the tasks 1 through 6 of "Process of Creating ADM Deployment Units from Database Types" on page 64.

To deploy an ADM session by importing an ADM XML deployment session file

1. Access the target system; that is, the system to which you want to migrate data.
2. Make sure the source file is accessible.
3. Navigate to the Application Deployment Manager screen > Deployment Sessions view.
4. In the Deployment Sessions list, click the menu button and then Deploy from File.
5. Enter the network path and filename for the import file, then click the Import button.

   NOTE: The import process requires the filename as input. If the export session generated multiple files, the import step must be executed for each file.

6. Check the details of the deployment by reviewing the log files and EAI queue of the target application object manager.

   For more information about ADM log files, see Chapter 8, "ADM Logging Capabilities and Troubleshooting."

Deploying ADM Sessions Using Command-Line Interface

Deploying sessions using the Siebel Server Manager command-line interface program (srvrmgr) is one of the alternate means of migrating ADM database data types from one Siebel application environment to another. For other alternative deployment options, see "Other ADM Deployment Options" on page 101.

Before using this deployment option, make sure an ADM session exists and the deployment unit XML file has been exported as outlined in the tasks 1 through 6 of "Process of Creating ADM Deployment Units from Database Types" on page 64.

This procedure runs a workflow process, named UDA Batch Import, on the target environment. This workflow process imports the data into the target environment, based on a session ID provided, using Siebel EAI as the transport. The Server Manager command line can be used to generate export files multiple times and can provide basic automation for the export step.

NOTE: The Workflow Process Manager component (WfProcMgr) must be running on the target Siebel Server in order for this procedure to run correctly.

To import deployment session file using Server Manager command-line interface

1. Make sure the export deployment file exists in a shared network location.

   NOTE: You do not need to create a deployment session in the target environment to import a deployment session file.

2. Start Server Manager command-line interface program (srvrmgr) with your user ID in the target system.

   For more information on this procedure, see *Siebel System Administration Guide*.

3. Verify that the server component Workflow Process Manager (alias WfProcMgr) is online.

   If not, enable the Workflow Management component group (alias Workflow) and restart the Siebel Server. For more information on these procedures, see *Siebel System Administration Guide*.

4. Enter the following command at the Server Manager command-line interface prompt:

   ```
   start task for comp WfProcMgr server Siebel_Server_Name with ProcessName="UDA Batch
   Import", RowID="Session_ID"
   ```

   where:

   - `Siebel_Server_Name` is the name of the Siebel Server.
   - `UDA Batch Import` is the name of the workflow process. This value is case-sensitive.
   - `Session_ID` is the session ID of the deployment session used to generate the export files.

   NOTE: This session ID does not exist anywhere in the target database. It is the session ID used to export files. This value maps to an .ini file generated during the export.

5. Check the details of the batch deployment by reviewing the source system ADM log files and EAI queue. For more information about ADM log files, see Chapter 8, "ADM Logging Capabilities and Troubleshooting."

NOTE: One import step reads all the previously generated files by the export session executed on the source system.

Reviewing Postdeployment Tasks

Reviewing postdeployment tasks is the final step in "Process of Deploying an ADM Package" on page 85. The deployment of some ADM data types requires postdeployment tasks such as activation outside of the ADM framework. For more information on specific ADM data types, see Appendix A, "ADM Supported Data Types."

7 Guidelines for Deploying ADM

This chapter contains Application Deployment Manager (ADM) deployment guidelines and business case scenarios. It includes the following topics:

Business Case Scenarios for Application Deployment Manager

The following four business cases provide potential scenarios for using Application Deployment Manager (ADM), based on the frequency of the target environment updates and also by the type or amount of application customizations being migrated. The scenarios are as follows:

Scenario for a Minor Customization Update Using ADM

This topic gives one example of how Application Deployment Manager (ADM) may be used. You may use ADM differently, depending on your business model. The following scenario illustrates the use of ADM to deploy a simple change.

A Siebel Business Application administrator reviews the current production deployment on a daily basis and schedules an update to the deployment once a week. This weekly update includes minor changes and enhancements. Several users of the production application have requested the following:

- Updates to three List of Values (LOVs) and the addition of five new LOVs

- Updates to 10 responsibilities and views and the addition of two new responsibilities and views
- Updates to 50 assignment rules and the addition of 10 new assignment rules

NOTE: These minor updates and enhancements represent no new changes in business processes and do not require schema changes or compiled repository objects.

The Siebel Business Application administrator uses ADM to create one single ADM package containing the updates and enhancements. The ADM package contains the customizations data for each ADM data type. Starting first with a refreshed test environment, the administrator deploys and activates the ADM package. In this scenario, the target Siebel Enterprise Server requires minimal or no downtime.

After reviewing the customizations updates in the test environment and verifying the correct application of the customizations, the administrator deploys and activates the ADM package to the production environment. The new and updated LOVs, responsibilities, views, and assignment rules are now available to the business users of the Siebel application with minimal downtime. This type of deployment can take place during working hours. However, it is always recommended that such changes be made when the system is offline or during periods of low load. The length of downtime is best determined during the deployments to the test environment.

Scenario for a Minor Functional Update Using ADM

This topic gives one example of how Application Deployment Manager (ADM) may be used. You may use ADM differently, depending on your business model.

A Siebel Business Application administrator reviews the current production deployment on a weekly or monthly basis and reviews the update requests from users. These requests include minor modifications to existing business processes, the addition of new business processes, and minor customizations updates as described in "Scenario for a Minor Customization Update Using ADM" on page 105. These updates include:

- Updates to three List of Values (LOVs) and the addition of five new LOVs
- Updates to 10 responsibilities and views and the addition of two new responsibilities and views
- Updates to 50 assignment rules and the addition of ten new assignment rules
- Updates to four workflow policies and the addition of one new policy
- Updates to 20 SmartScripts and the addition of thirty new scripts
- Updates to 10 repository objects, three SRF files, 20 browser scripts, and three report files

NOTE: These updates and enhancements represent minor changes in the business processes and compiled repository objects, but do not require schema changes.

The Siebel Business Application administrator uses ADM to create one ADM package containing the updates and enhancements and another ADM package containing the SRF file:

- **ADM package 1.** The LOV, responsibilities, views, assignment rules; workflow policy and SmartScript updates; repository objects, browser scripts, and report files.
- **ADM package 2.** The SRF file.

Starting first with ADM package 2, shut down the target environment before deployment. Restart the environment after the deployment of ADM package 2, and then deploy ADM package 1, which should not require a shutdown before or a restart after deployment. However, you may need to restart the environment after the activation of some of the customizations. Any dependencies between deployment units in ADM package 1 must be controlled through the category level deployment feature. For more information on individual data types and their requirements, see Appendix A, "ADM Supported Data Types."

After reviewing the customizations updates in the test environment and verifying the correct application of the customizations, the administrator deploys and activates the ADM package to the production environment in a similar fashion. The new and updated changes are now available to the business users of the Siebel application with minimal downtime. Because this type of deployment requires downtime, it is recommended that it take place outside working hours. The length of downtime is best determined during the deployments to the test environment.

Scenario for a Major Functional Update Using ADM

This topic gives one example of how Application Deployment Manager (ADM) may be used. You may use ADM differently, depending on your business model.

A Siebel Business Application administrator reviews the current production deployment on a quarterly basis based on user feedback and management reviews of current business practices. At this time, a major functional update is planned to implement new features and business processes. This internal release includes customizations and compiled repository objects as described in "Scenario for a Minor Functional Update Using ADM" on page 106 in addition to changes to the schema. A repository migration is not used for this limited repository update.

In this scenario, a Siebel Business Application administrator creates multiple ADM packages similar to the "Scenario for a Minor Functional Update Using ADM" on page 106. In a refreshed test environment, the Siebel administrator logs all users off the system, deploys the ADM package containing the schema changes, shuts down the Siebel Enterprise Server, then synchronizes the database with the repository schema. Restarting the target Siebel Enterprise Server, the administrator then deploys and activates the remaining ADM customizations. For information on running a synchronization on a new or updated schema, see *Going Live with Siebel Business Applications*.

After reviewing the customizations updates in the test environment and verifying the correct application of the customizations, the administrator deploys and activates the ADM package to the production environment in a similar fashion.

NOTE: You must run DDLSync after any schema changes are migrated from one environment to another.

Run this scenario carefully in a test or staged environment prior to production to determine the downtime impact of this update.

Scenario for a Major Functional Update Including Full Repository Changes Using ADM

This topic gives one example of how Application Deployment Manager (ADM) may be used. You may use ADM differently, depending on your business model.

A Siebel Business Application administrator reviews the current production deployment on an annual basis primarily based on the management or executive reviews of the current business practices. At this time, a major functional release is planned to implement the new features and business processes. This major internal release includes the customizations and compiled repository objects described in "Scenario for a Minor Functional Update Using ADM" on page 106, in addition to changes to the schema and a full repository update. In this scenario, due to the large number of repository updates, the repository migration feature is used instead of an ADM package containing repository updates. For information on running a repository migration, see *Going Live with Siebel Business Applications*.

In this scenario, in a refreshed test environment, a Siebel Business Application administrator runs a repository migration to update the new repository. The administrator then performs other customizations migrations using ADM as described in "Scenario for a Major Functional Update Using ADM" on page 107.

After reviewing the customization updates in the test environment and verifying the correct application of the customizations, the administrator deploys and activates the ADM package to the production environment in a similar fashion.

Carefully run this scenario in test or a staged environment prior to production to determine actual downtime impact of this update.

About ADM Roles

Using Application Deployment Manager (ADM) involves the system user and end user roles. The system user, usually the Siebel administrator, runs Siebel Management Server and Siebel Management Agent and can perform deployment operations. Assign the following responsibilities to this user:

- MgmtSrvr-Monitor
- MgmtSrvr-Deploy&Execute
- MgmtSrvr-Admin

For additional information on these responsibilities pertaining to ADM, see "About Security Configuration of Siebel Management Server and Siebel Management Agent" on page 47.

The end user can perform deployment operations, and in that case, should be assigned the MgmtSrvr-Monitor and MgmtSrvr-Deploy&Execute responsibilities. The end user is created in the same manner that a new Siebel user is created; that is, using UI/import. For more information on creating a new Siebel user, see the chapter about user administration in *Siebel Security Guide*.

Selecting ADM Over Other Data Migration Options

Siebel Business Applications provide several data migration tools, including Application Deployment Manager (ADM), to migrate data based on the business scenario and volume of data to be migrated.

ADM is designed to migrate application customizations to same-version and same-language environments. For more information on this feature, see "About the Customizations Migration Process Using ADM" on page 27.

NOTE: You can migrate data between environments of different languages if no language is specified for the data being deployed.

The ADM deployment framework provides the following advantages for customizations migration:

- Manages change management requests
- Tracks changes through the internal ADM database
- Maintains, tracks, and reuses existing packages
- Provides low-volume migration along with support for a combination of database, repository, and file data types

ADM cannot be substituted for the specialized operations performed by using some of the other Siebel Business Application migration or deployment tools:

- Siebel EAI is used for migrating small volumes of data for mostly daily or weekly changes. The ADM framework is built on Siebel EAI. For more information on Siebel EAI, see *Overview: Siebel Enterprise Application Integration*.
- Siebel EIM is used for migrating bulk data between databases using migration tables and is used for mostly large volumes. For more information on Siebel EIM, see *Siebel Enterprise Integration Manager Administration Guide*.
- Repository Migration option in the Siebel Database Configuration Wizard is used for migrating repository data and schema changes and is designed for large volumes. For more information on the Repository Migration option, see *Going Live with Siebel Business Applications*.

Managing ADM Deployment Dependencies

Review the following guidelines and recommendations when managing Application Deployment Manager (ADM) data type dependencies during an ADM deployment:

- ADM supported data types might have a preconfigured dependency or prerequisites on the availability of certain data before carrying out an ADM deployment. For more information on the data type dependency, see Appendix A, "ADM Supported Data Types" and Table 13 on page 131.
- Research and understand the dependencies included with business processes used by your organization. Addressing all deployment dependencies during an ADM deployment reduces errors and system downtime.

- Formalize dependencies by creating relationships in the ADM GUI when preparing data types for packaging. For more information on data type relationships, see "Creating ADM Data Type Relationships" on page 65.

Managing Your ADM Package Content

Review the following guidelines and recommendations when managing your Application Deployment Manager (ADM) package content:

- Follow the documented naming conventions throughout the ADM deployment process,
  - During the creation of the ADM project
  - When storing the exported database files in directories
  - When storing files in directories with versions or tracked through a version tracking system
- Follow a documented deployment strategy including procedures, checklists, and naming conventions. Historical package data is stored in the ADM database for reference.
- Create the ADM package folder structure using the ADM Packaging Utility rather than creating it manually.
- Use a stored file system or document tracking system when creating deployment units involves teams of users to make sure the package content is current and correct.
- Validate packages before deployment.

Minimizing Downtime When Deploying ADM Packages

Use the following recommendations to minimize your Siebel Business Application downtime when deploying ADM packages:

- Group data types and customizations based on the system impact.

  For information on data type system impact, see Appendix A, "ADM Supported Data Types." For example, group data types that have no system impact or do not require activation in the same package.
- Test your ADM deployment between a development environment and a test environment. Create benchmark downtimes, if necessary, for specific deployment packages.
- Use staged deployment. Staging the ADM deployment to a subset of Siebel Servers in a Siebel Enterprise Server rather than to the entire Siebel Enterprise Server minimizes downtime.
- Identify the type of release to reduce downtime.

  For the types of releases, see "Business Case Scenarios for Application Deployment Manager" on page 105.

Synchronizing Repository Updates

Perform the following tasks to make sure the repository data in the source and target environments are synchronized:

- Run DDLSync, using the Siebel Database Configuration Wizard, after any database schema changes are migrated from one environment to another. For information on the Database Configuration Wizard, see *Siebel Database Upgrade Guide*.
- Perform a Repository Migration, using the Siebel Database Configuration Wizard, if any discrepancies exist between repositories after a compiled repository objects migration. For information on the Database Configuration Wizard and Repository Migration, see *Going Live with Siebel Business Applications*.
- Compile the Siebel Repository File (SRF) after changes are made to the repository. For information on compiling the SRF, see *Using Siebel Tools*.

  NOTE: Compiling the SRF generates browser scripts that must be copied to the `file/App Server/webmaster deployment` package directory.

Integrating ADM with Change Management Systems

While Application Deployment Manager (ADM) can be used as a stand-alone system to deploy application customizations, its granular design also facilitates integration with change management systems (CMS) to support automation.

CMS can be used as an external system to automate deployments by invoking ADM commands as needed. This can be achieved through any of the following options:

- A script (Perl/shell)
- A batch scheduling system

8 ADM Logging Capabilities and Troubleshooting

This chapter contains an overview of the Application Deployment Manager (ADM) logging capabilities and options for troubleshooting ADM deployments. It includes the following topics:

- "About ADM Framework Logging Options" on page 113
- "Configuring ADM Framework Log Files" on page 117
- "Reviewing ADM Log Files" on page 119
- "Troubleshooting ADM Error Messages" on page 120

About ADM Framework Logging Options

The Application Deployment Manager (ADM) feature contains logging options throughout the deployment framework and records application behavior during most phases of the ADM deployment and for most ADM framework entities. For more information on ADM logging options, see:

- "About ADM Log Files on Siebel Management Server" on page 114
- "About ADM Log Files on Siebel Management Agents" on page 115
- "About ADM Log Files on Siebel Servers" on page 115
- "About ADM Server Components Log Files" on page 116
- "About SRProc and SRBroker Log Files" on page 117
- "About ADM Package Utility Log Files" on page 117

Table 11 contains an overview of ADM log files.

Table 11. ADM Log Files

Log Filename	ADM Framework Entity	Log File Content and Locations
ManagementServer.log	Siebel Management Server	See "About ADM Log Files on Siebel Management Server" on page 114
ADMDeployment.log		
packagename_sessionnumber.log		

Table 11. ADM Log Files

Log Filename	ADM Framework Entity	Log File Content and Locations
Agent.log	Siebel Management Agent	See "About ADM Log Files on Siebel Management Agents" on page 115.
ADMDispatchMbean.log		
ADMMbean.log		
ADMMbean_*packagename_sessionID_deploymentunit*.log		
packagename_sessionID_deploymentunit.log		
packagename_sessionnumber_packagename_category.log	Siebel Server	See "About ADM Log Files on Siebel Servers" on page 115.
*packagename_sessionnumber_packagename_category*_backup.log		
ADMObjMgr_*TaskID*.log	ADM Server Component	See "About ADM Server Components Log Files" on page 116.
ADMProc_*TaskID*.log		
ADMBatchProc_*TaskID*.log		
ADMProc_*PackageName_datatypename*.log		
SRProc.log	System Management Server	See "About SRProc and SRBroker Log Files" on page 117.
SRBroker.log		
pkgr_packagename.log	ADM Package Utility	See "About ADM Package Utility Log Files" on page 117.

About ADM Log Files on Siebel Management Server

This topic describes the Siebel Management Server log files and Application Deployment Manager (ADM)-specific log files that reside on the Siebel Management Server. The Siebel Management Server maintains the following log files of interest to an ADM deployment:

- **ManagementServer.log.** Master log file that records activities on the Siebel Management Server, deployment activities, errors, status for database and repository items, and deployment summaries for file items. This log also provides information on bean, Management Server, and Management Agent registrations.
- **ADMDeployment.log.** Primary log file for all ADM operations. It records activities on the Management Server, such as deployment activities, errors, status for database and repository items, and deployment summaries.

- ***packagename_sessionnumber*.log.** ADM session log file that records the detailed status and activities of the package content. For example, if a file group is successfully deployed to an individual Siebel Server, this log file records the operation, timestamp, and Siebel Server for each individual file.

These log files are located in the following Siebel Management Server directory: `*Siebel_Management_Server_Root*\*MgmtSrvrInstallDir*\log`.

The amount of information recorded is based on the log level settings. For information on configuring Siebel Management Server log files, see "Configuring ADM Log Files on Siebel Management Server and Siebel Management Agent" on page 118.

About ADM Log Files on Siebel Management Agents

This topic describes the Siebel Management Agent log files that record data defining an Application Deployment Manager (ADM) deployment. The Siebel Management Agent maintains the following log files of interest to an ADM deployment:

- **Agent.log.** Top-level Management Agent log file that records information on Mbean registration and management. This log file is located in the following Siebel Management Agent Directory: `\Management_Agent_Home\log`.
- **ADMDispatchMbean.log.** Log file that records the notifications sent from the Management Agents to the Management Server. This log file is located in the following Siebel Management Agent directory: `Siebel_Management_Agent_Root\agent\log`.
- **ADMMbean.log.** Log file that records information on how the Mbeans on the Management Agents handle ADM tasks. This log file is located in the following Siebel Management Agent directory: `Siebel_Management_Agent_Root\agent\log`.
- **ADMMbean_*packagename_sessionID_deploymentunit*.log.** Log file that records the deployment status details of the deployment unit. For file type deployment, it records the progress of the file copy. For database and repository type deployments, it records the response of the business services invoked during deployment of the units. This log file is located in the following Siebel Management Agent directory: `Siebel_Management_Agent_Root\agent\log`.
- ***packagename_sessionID_deploymentunit*.log.** Log files that contains the detailed description of activities and status for the deployment units after they reach Siebel Management Agent. These log files also list the Request ID for the deployment units, and are used by the business services during the deployment process. These log files are located in the Siebel Server log directory.

For information on configuring Siebel Management Agent log files, see "Configuring ADM Log Files on Siebel Servers" on page 119.

About ADM Log Files on Siebel Servers

This topic describes the Application Deployment Manager (ADM) log files that are generated on the Siebel Server. The Siebel Server maintains the following log files of interest to an ADM deployment:

- ***packagename_sessionnumber_unitname_category*.log.** Log file that contains logging information of the unit being deployed and states the status of the deployment.
- ***packagename_sessionnumber_unitname_category*_backup.log.** Log file that records information about the backup path and its status. This file is generated only when the backup is set to `TRUE` in the enterprise profile.

These log files are located in the log directory for each individual Siebel Server. The log directory location on Windows is `SIEBSRVR_ROOT\log`. The log directory on UNIX is `SIEBSRVR_ROOT/enterprises/EnterpriseServerName/SiebelServerName/log`.

For information on configuring ADM log files for Siebel Server, see "Configuring ADM Log Files on Siebel Servers" on page 119.

About ADM Server Components Log Files

The Application Deployment Manager (ADM) server components, like all Siebel Server components, write component behavior data to log files. The Siebel Server components maintain the following log files of interest to an ADM deployment:

- **ADMObjMgr_*TaskID*.log.** Log file that displays the creation of asynchronous server requests. For example, it shows the creation of a record in the S_SRM_REQUEST table, which contains a record for the request with ROW_ID that matches the request ID in the log file.
- **ADMProc_*number*.log.** Log files that records information on the Siebel Server about current tasks performed by the ADM deployer code, including any errors or data conflicts, as well as EAI errors. This information is very useful in troubleshooting ADM server-side problems. These log files are created only when the ADMProc component's event log level (evtloglvl) is set higher than 3 (default value is 1).
- **ADMBatchProc_*number*.log.** Information is written to these files only when ADM srvrmgr commands are used to export deployment data. These files are created when ADMBatchProc component's event log level (evtloglvl) is set higher than 3 (default value is 1).
- **ADMProc_*PackageName_datatypename*.log.** Log file that records the objects deployed and any actions taken. The objects are recorded with the data type, integration object name or business object name, and the new user key values of the root level object. The action recorded indicates whether the object is updated or created. This log file is created for each package session and for each unit. Multiple executions of the same unit update the same log file. This log file is created or updated only by the migration (or import) step of the ADM Processor server component. In case the migration is rolled back and no object is migrated, this log file must not record any object migration.

To show warnings and diagnostic information, you must increase log levels for the server components. For information on configuring ADM server component logging, see "Configuring ADM Server Component Log Files" on page 119.

For more information on the ADM server components, see "About ADM Server Components" on page 25. For more information on Siebel Server component log files, their naming conventions, locations, and configuration details, see *Siebel System Monitoring and Diagnostics Guide*.

About SRProc and SRBroker Log Files

The Server Request Broker (SRBroker) and Server Request Processor (SRProc) are system management server components that process asynchronous and synchronous server requests from Siebel Server components. The following log files track the server requests through the ADM deployment process:

- **SRProc.log.** Log file that displays the processing of requests from S_SRM_REQUEST to ADMProc through SRBroker.
- **SRBroker.log.** Log file that displays request submissions to ADMProc and completion status.

These log files can be located on any server hosting the ADM Processor component. For more information on SRProc and SRBroker, see *Siebel System Administration Guide* and *Siebel System Monitoring and Diagnostics Guide*.

About ADM Package Utility Log Files

This topic describes the Siebel Package Utility log files that record data defining the creation of an Application Deployment Manager (ADM) package. For more information on the ADM Package Utility, see "About the ADM Package Utility" on page 17.

The log files have the following naming convention: pkgr_*packagename*.log and reside in the directory set by the command utility switch, `-l`. For more information on this switch, see "About the ADM Package Utility Modes and Switches" on page 60.

The ADM Package Utility generates a log file for each command execution. When more than one log file is generated with the same package name and in the same directory, the existing log files with that package name are renamed with a number in sequence appended to it. The most recently generated log file has no sequence number appended to the filename. To read the log files in chronological order, open the log file with the highest number.

Each log file displays the errors, warnings, and success status of the command execution. The following is an example of an item displayed in the log file:

```
2004-11-04 04:30:13 (SBL-ADM-00789) Package folder does not contain any valid item.
```

Configuring ADM Framework Log Files

The Application Deployment Manager (ADM) logging capabilities can be configured to control the amount of data logged or number of log files retained during an ADM deployment. For more information on configuring ADM logging options, see:

- "Configuring ADM Log Files on Siebel Management Server and Siebel Management Agent" on page 118
- "Configuring ADM Log Files on Siebel Servers" on page 119
- "Configuring ADM Server Component Log Files" on page 119

Configuring ADM Log Files on Siebel Management Server and Siebel Management Agent

This topic describes configuring log files used by ADM on the Siebel Management Server and Siebel Management Agent. For more information on these log files, see "About ADM Log Files on Siebel Management Server" on page 114. The following configurations are available for the log files:

- **Log level.** The level or amount of information recorded in the log file.
- **Log file size.** The size of the log file before it is closed.
- **Log file cycling.** The number of log files saved and recorded in order. Each new log file is appended with a number in sequence. To read the log files in chronological order, open the log file with the highest number. The value for this property indicates the number of log files maintained; that is, if the number of log files saved equals the value set for this property, the oldest log file is deleted when the next log file is created.

NOTE: The default log level is set to INFO for Siebel Management Server and Management Agent, as well as for ADM Siebel Server components. Increasing these log levels to FINE slows down the deployment.

To configure the Siebel Management Server and Siebel Management Agent logging behavior

1. Navigate to the .properties file:
   - On the Siebel Management Server, navigate to the CMSLogging.properties file in the following location: `\`*`MgmtSrvrInstallDir`*`\config\CMSLogging.properties`.
   - On the Siebel Management Agent environment, navigate to the AgentLogging.properties file in the following location:
     `Siebel_Management_Agent_Root``\agent\config\AgentLogging.properties`
2. To configure the log level, set the com.siebel.adm.mgmtsrvr.deploy.deploymgr.level property in the .properties file to the appropriate level. See the following table for the values.

Log Level	Description
FINE	Log level providing tracing information, which is useful for debugging. This setting is the default value.
INFO	Log level providing informational messages.
WARNING	Log level providing information on potential problems during the deployment. This setting is recommended for normal execution.
SEVERE	Log level providing information on a serious problem during the deployment.

   An inheritance behavior is based on the log level setting; that is, if you select INFO level, the WARNING and SEVERE level log details are also recorded.

3. To configure the log file cycling, set the `java.util.logging.FileHandler.count` property in the CMSLogging.properties file to an integer value greater than 0.

4. To configure the log file size, set the `java.util.logging.FileHandler.limit` property to a value in bytes. By default, there is no limit on the log file size.

Configuring ADM Log Files on Siebel Servers

Configuring ADM log files on Siebel Servers requires setting the log levels for Siebel Server event types. For more information on configuring Siebel Server log files, see *Siebel System Monitoring and Diagnostics Guide*.

Configuring ADM Server Component Log Files

Configuring the ADM server component log files requires setting the log levels for component event types. For more information on configuring Siebel Server component log files, see *Siebel System Monitoring and Diagnostics Guide*.

Reviewing ADM Log Files

This topic describes two specific occasions for reviewing the log files:

- "After Installing the ADM Framework" on page 119
- "After Executing an ADM Deployment" on page 119

After Installing the ADM Framework

Reviewing the following ADM log files after installing the ADM framework will help you determine the success or failure of the installation:

- **ManagementServer.log.** Displays `state=online` with a successful installation.
- **Agent.log.** Displays no error messages with a successful installation.

After Executing an ADM Deployment

Reviewing the following ADM log files after executing an ADM deployment will help you determine the success or failure of the deployment:

- ***packagename_sessionnumber*.log.** Displays `status=MIGRATION_COMPLETE` to indicate a successful deployment.
- **ADMMbean_*packagename_sessionID_deploymentunit*.log.**
  - For file category, displays `DEPLOYMENT_COMPLETE` to indicate a successful deployment.
  - For database and repository categories, displays `Deployment request successfully submitted` to indicate a successful deployment.
- ***packagename_sessionnumber_packagename_category*.log.**
- **ADMProc_*TaskID*.log.** Displays `Deployment failed` to indicate a failed deployment.

Troubleshooting ADM Error Messages

This topic provides troubleshooting tips and information about issues that may occur while using the Application Deployment Manager (ADM) feature.

ADM Packager (Admpkgr) Is Not Functioning Correctly

If the ADM Packager is not functioning correctly, make sure that:

- The package location given in the command has read and write permission for the user.
- The package name in the command is spelled correctly and uses the appropriate capitalization. The package name is case-sensitive.
- No previous descriptor files exist in the package location. Delete old descriptor files before creating new ones.

ADM Packager (Admpkgr) Error: Definition of Data Type Not Found

If the ADM Packager reports an error stating that the definition of a particular data type was not found, check the structure of the package. Also check the adm_registry.xml file to determine if the data type in question is supported by ADM.

Create Command Error: Session Already Exists for the Current Package

If you receive an error message stating that a session already exists for the current package after you run a create command, then you must first acknowledge the previous session, then run the create command again.

Create Command Error: Server Name Not in Deploy Server List or Enterprise Profile

If you receive the error message:

```
The specified server is specified in the target server but not in the deploy server
list or <enterprise_target_information>.
```

Make sure that the server name in entprofile.xml matches the server name in agentconfig.xml. The servers in the list of deployment servers must also match the agents under <enterprise_target_information> in entprofile.xml. Values are case-sensitive.

Backup Command Hangs

If nothing happens after you run the backup command, make sure that:

- The package location is specified in deploy.bat.
- The package location has read and write permissions.
- The package is present and the descriptor file matches the package.

- Re-create the descriptor file and try again.

Copy Command Error: Multiple Matches for Integration Component User Key

If you receive the error message:

```
Multiple matches found for instance of integration component '<x>' using search
specification '<x>' in the business component '<x>', based on user key '<x>'. (SBL-
EAI-04390)
```

Clean the duplicate records in the target environment before executing the deployment.

Copy Command Error: Server Not Online

If you receive an error that the server is not online, make sure the servers in the list of deployment servers are online. Also make sure that ADMProc and ADMObjMgr server components are running.

Copy Command Error: Validation Fails

If the validation fails, make sure that the agent name under <targetserverdetails> in entprofile.xml matches the deployment server for file, database, and repository categories. Also make sure that the agent name in agentconfig.xml matches the agent details in entprofile.xml. Values are case-sensitive.

Copy Command Hangs

If nothing happens after you run the copy command, make sure that SRBroker and SRProc system management server components are online and that ADMJavaSubsys profile is configured properly. Locate the log files for this package deployment (ADMProc*.log) in the Siebel Server log directory and check for error messages.

Copy Fails Due to a FileSystem Error

If the copy fails and the log files for this package deployment (ADMProc*.log) in the Siebel Server log directory indicate a filesystem error, make sure the filesystem path exists and has read and write access from the server.

Enterprise Name Error

If you receive the following error message in the ManagementServer.log or Agent.log the Siebel Enterprise Server name was not set properly in the Management Server:

```
ManagementServer SEVERE 2006-10-03 03:27:03 Cannot create MBean
[Siebel:name=ServerAdmin] of class
[com.siebel.management.serveraccess.ServerAdmin](SBL-CMS-00201)
java.lang.IllegalArgumentException: (SBL-CMS-00953) Name server name is null or
empty.
```

You must change the Siebel Enterprise Server name in Management Server and Management Agent.

To change the enterprise name in Management Server and Management Agent

1 Make sure the srvrmgr path is present in startagent.bat.

   For Windows, the srvrmgr is generally in $Siebel_Root$\bin for the release version and in $Siebel_Root$\bin\w32ud for the debug version.

   If the srvrmgr path is not present, add it to startagent.bat.

2 Change the Siebel Enterprise Server name in deploy.bat by entering the following command:

   ```
   set ADM_ENTERPRISE= <enterprise_name>
   ```

3 Change the Siebel Enterprise Server name in the Header File of entprofile.xml. For example:

   ```
   Description="This profile is used to describe the target enterprise related
   parameters." EnterpriseName=":<enterprise_name>"
   ```

4 Change the Siebel Enterprise Server name in the EnterpriseTargetServerInformation section of entprofile.xml. For example:

   ```
   <EnterpriseTargetServerInformation>
   <ServerInfo AgentName="sdchs20i059" SiebelServerConnectString="siebel://
   sdchs20i059:2321/:<enterprise_name>/ADMObjMgr_enu"
   ServerInstallDir="d:\sebl_builds\sbl1202\"/>
   </EnterpriseTargetServerInformation>
   ```

5 Change the Siebel Enterprise Server name for the agent in configuration.agent.xml. For example:

   ```
   <entry key="sdchs20i059"
   value="sdchs20i059:siebelServer::<enterprise_name>:service:jmx:rmi://sdchs20i059/
   jndi/rmi://sdchs20i059:1299/jmx/siebel/agent" />
   ```

6 Change the Siebel Enterprise Server name for the entry key in configuration.globals.xml. For example:

   ```
   <entry key="<enterprise_name>.management.serveraccess.nameServer"
   value="string:sdchs20i059:2320" />
   ```

7 Restart the Management Server and Management Agent for the changes to take effect.

Deployment Filter Can Return Zero Records While Exporting Database Data Types to Deployment Units

In situations where matching data is not present, the administrator might be creating filters in order to export the project when data becomes available. Thus, setting deployment filters that return zero records is not prohibited.

Cannot Run Two Management Servers on the Same Machine

Multiple Management Servers cannot be run from the same machine at the same time. You can install multiple Management Servers on the same machine, but only one can run at a given time.

This limitation exists due to a conflict in port numbers. The Management Server, Tomcat server, and local database occupy certain ports on the machine. If you have two instances running, then these ports conflict. You can configure Management Server and Tomcat server to listen on different ports, but the database port cannot be changed.

Cannot Add More Than One File System for Each Siebel Enterprise Server

In a multiserver deployment, set FileSystem only at the enterprise level, and point it to the same location. Release 8.0 supports a multipartitioned FileSystem, but there must still be only one enterprise-shared value for FileSystem.

CAUTION: If FileSystem is not set correctly, the deployment of ADM data types that use EAI as the deployment method might fail.

A ADM Supported Data Types

This appendix provides deployment and activation details and limitation for Application Deployment Manager (ADM) data types. The ADM data types described in this appendix represent Oracle's Siebel application customizations that are available for migration using the ADM framework. This appendix has the following topics:

How to Use This Appendix

This appendix contains details on each supported ADM data type and the data types' behavior and dependencies during the ADM packaging, deployment, and activation processes. Table 12 on page 127 and Table 13 on page 131 list this information.

A subset of data types requires additional consideration or configuration during the deployment process and are described in "ADM Data Type Details" on page 135.

Use this collected information on supported ADM data types to determine a migration strategy or when setting IT or infrastructure update policies. This information can help form a best course of action when performing complex ADM deployments with many dependencies.

You can also use this appendix to categorize the development data types based on the system impact or deployment behavior. This information is useful when combined with your internal release strategy. For example, data types that do not require activation can be deployed and made available immediately, with no downtime.

About the ADM Data Type Summary Table

This topic describes the column headings in Table 12 on page 127 that provide a summary definition of the content and functionality of individual Application Deployment Manager (ADM) data types. For information on how to use this information during the ADM packaging and deployment phases, see "How to Use This Appendix" on page 125. The following list explains the terms used in Table 12:

- **Data Type.** Name of the ADM data type, which represents the customized application data.

- **Deployment Alias.** Alternate name or alias of the ADM data type used during deployment.
- **Category.** High-level category of the ADM data types. There are three categories: database types, repository types, and file types.
- **B (Backup).** Indicates if the ADM data type is available for backup on the target environment.
- **A (Activate).** Indicates if the ADM data type is available for activation on the target environment. If it is not available, then the data type does not require activation to be deployed and will become active as soon as the copy process is complete.
- **R (Restore).** Indicates if the ADM data type can be restored on the target environment.
- **S (System Impact).** Indicates impact to the system, if any, when items of this type are deployed and activated. The ADM Package Utility aggregates the values for each data type in an ADM package and enters the greatest impact into the package descriptor file. There are six possible values:
  - **No impact.** The data type can be deployed without any impact to the availability of the system.
  - **User reconnect.** The user must log in again to access any newly deployed items for this data type.
  - **Manual activation.** The data type does not support activation through the ADM framework; therefore, manual steps must be taken to complete the deployment.
  - **Web server restart.** Restart Web servers to fully deploy and activate the items for this data type.
  - **Rolling server restart.** Restart Siebel Servers to fully deploy and activate the items for this data type. However, it is possible to limit the impact on server availability by performing the deployment (and server restart) to a subset of the Siebel Servers at a time; that is, in a rolling fashion.
  - **Siebel Enterprise Server restart.** Restart the entire target Siebel Enterprise Server to fully deploy and activate items for this data type.

Summary of ADM Data Types

The following table presents summary data for each of the supported Application Deployment Manager (ADM) data types. For more information on each of the column headings, see "About the ADM Data Type Summary Table" on page 125.

NOTE: B indicates Backup, A indicates Activate, R indicates Restore, and S indicates System Impact.

Table 12. ADM Supported Data Types

Data Type	Deployment Alias	Category	B	A	R	S
Access Controlled Business Service	Access Controlled Business Service	Database	Yes	Yes	Yes	No impact
Access Controlled Task	Access Controlled Task	Database	Yes	Yes	Yes	No impact
Access Group	AccessGroup	Database	Yes	No	Yes	No impact
Account Promotion	AccntPromo	Database	Yes	No	Yes	No impact
ADM Abs Admin Service Region	ADM Abs Admin Service Region	Database	Yes	No	Yes	No impact
ADM Abs Admin Time Window Map	ADM Abs Admin Time Window Map	Database	Yes	No	Yes	No impact
ADM Comm Inbound Group	ADM Comm Inbound Group	Database	Yes	No	Yes	No impact
ADM Comm Package	ADM Comm Package	Database	Yes	No	Yes	No impact
ADM CommSrv CM Adapter Administration	ADM CommSrv CM Adapter Administration	Database	Yes	No	Yes	No impact
ADM Project	ADM Project	Database	Yes	No	Yes	No impact
ADM Search Category Available Flds	ADM Search Category Available Flds	Database	Yes	No	Yes	No impact
ADM Search Connector	ADM Search Connector	Database	Yes	No	Yes	No impact
ADM Search Engine	ADM Search Engine	Database	Yes	No	Yes	No impact
ADM Search Parent Category	ADM Search Parent Category	Database	Yes	No	Yes	No impact
ADM Shift Exception	ADM Shift Exception	Database	Yes	No	Yes	No impact
ADM Shift Schedule	ADM Shift Schedule	Database	Yes	No	Yes	No impact
Aggregate Discount	AggrDiscount	Database	Yes	Yes	Yes	No impact
Aggregate Discount Sequence	AggrDiscSeq	Database	Yes	Yes	Yes	No impact
Assignment Group	AssignGroup	Database	Yes	No	Yes	No impact
Assignment Rule	AssignRule	Database	Yes	No	Yes	No impact
Attribute Adjustments	AttrAdjustments	Database	Yes	Yes	Yes	No impact

Table 12. ADM Supported Data Types

Data Type	Deployment Alias	Category	B	A	R	S
Audit Trail Admin	Audit Trail Admin	Database	Yes	Yes	Yes	No impact
Business Rule	Business Rule	Database	Yes	Yes	Yes	No impact
Content Fixup	Content Fixup	Database	Yes	No	Yes	No impact
Correspondence Template	Correspondence Template	Database	Yes	No	Yes	No impact
Cost List	CostList	Database	Yes	No	Yes	No impact
Data Map	DataMapObject	Database	Yes	No	Yes	No impact
Discount Matrices	Discount and EC Matrix	Database	Yes	Yes	Yes	No impact
EAI Data Map	EAI Data Map	Database	Yes	Yes	Yes	No impact
EAI Dispatch Service Rule Set	EAI Dispatch Service Rule Set	Database	Yes	Yes	Yes	No impact
ExpLOV: Expense Type LOV	ExpLOV	Database	Yes	Yes	Yes	No impact
ExpType: Expense Type	ExpType	Database	Yes	No	Yes	No impact
Fund	Fund	Database	Yes	No	Yes	No impact
Host Table	Host Table	Database	Yes	No	Yes	No impact
iHelp	iHelp	Database	Yes	Yes	Yes	No impact
iHelp - Header	iHelp - Header	Database	Yes	Yes	Yes	No impact
List of Values	LOV	Database	Yes	Yes	Yes	No impact
Message Types, Administration - Order Management	MessageType	Database	Yes	Yes	Yes	No impact
Personalization—Actions	Personalization - Actions	Database	Yes	No	Yes	No impact
Personalization - Applets	Personalization - Applets	Database	Yes	Yes	Yes	No impact
Personalization—Event Defs	Personalization - Event Defs	Database	Yes	No	Yes	No impact
Personalization - Events	Personalization - Events	Database	Yes	Yes	Yes	No impact
Personalization—Rules	Personalization - Rules	Database	Yes	No	Yes	No impact
Personalization - Views	Personalization - Views	Database	Yes	Yes	Yes	User reconnect

Table 12. ADM Supported Data Types

Data Type	Deployment Alias	Category	B	A	R	S
Predefined Query	PDQ	Database	Yes	No	Yes	No impact
Price List	PriceList	Database	Yes	Yes	Yes	No impact
Product Feature	ProdFeature	Database	Yes	No	Yes	No impact
Product Line	ProdLine	Database	Yes	No	Yes	No impact
Product Catalog	ProductCatalog	Database	Yes	No	Yes	No impact
Products	ProductData	Database	Yes	No	Yes	No impact
Promotion	Promotion	Database	Yes	No	Yes	No impact
Proposal Template	Proposal Template	Database	Yes	No	Yes	No impact
Reports - Actuate Encyclopedia	Reports - Actuate Encyclopedia	File	No	No	Yes	No impact
Reports - Application Server	Reports - Application Server	File	Yes	No	Yes	No impact
Reports - Report Server	Reports - Report Server	File	Yes	No	Yes	No impact
Responsibility	Responsibility	Database	Yes	No	Yes	No impact
Run-Time Business Service		Database	Yes	No	Yes	
SmartScript	SmartScript	Database	Yes	Yes	Yes	No impact
Siebel Repository File	SRF	File	Yes	No	Yes	Rolling server restart
SSO System	SSO System	Database	Yes	No	Yes	User reconnect
State Model	StateModel	Database	Yes	No	Yes	No impact
Symbolic URL	Symbolic URL	Database	Yes	No	Yes	User reconnect
UI Theme	UIThemeMapping	Database	Yes	No	Yes	No impact
User List	UserList	Database	Yes	No	Yes	No impact
View	View	Database	Yes	No	Yes	No impact
Volume Discount	VolumeDiscount	Database	Yes	Yes	Yes	No impact
Web Application	Web Application	Database	Yes	No	Yes	No impact
Web Service - Inbound	Web Service - Inbound	Database	Yes	Yes	Yes	No impact

Table 12. ADM Supported Data Types

Data Type	Deployment Alias	Category	B	A	R	S
Web Service - Outbound	Web Service - Outbound	Database	Yes	Yes	Yes	No impact
Web Template	Web Template	File	Yes	No	Yes	Siebel Enterprise Server restart
Webmaster	Webmaster	File	Yes	Yes	Yes	User reconnect
Workflow Action Type	Workflow Action Type	Database	Yes	Yes	Yes	No impact
Workflow Group	Workflow Group	Database	Yes	Yes	Yes	No impact
Workflow Policy	Workflow Policy	Database	Yes	Yes	Yes	No impact
Workflow and Taskflow	Workflow and Taskflow	Repository	Yes	Yes	Yes	No impact
Workload Rule	Workload Rule	Database	Yes	No	Yes	No impact
Workspace Projects	Workspace Projects	Database	No	No	No	Manual activation

About the ADM Data Type Dependency Table

This topic describes the column headings in Table 13 on page 131 that provide general information on deployment dependencies of individual Application Deployment Manager (ADM) data types. For information on how to use this information during the ADM packaging and deployment phases, see “How to Use This Appendix” on page 125.

- **Data Type.** The ADM data type of interest. For more information on this data type, see Table 12 on page 127.
- **Run-Time Customization Dependency.** The names of objects that must exist before the specific data type item can be imported. This is a deployment dependency and not a dependency for the functionality.
- **Repository Item Dependency.** The names of repository objects that must exist before the specific data type item can be imported. This is a deployment dependency and not a dependency for the functionality.
- **SRF Dependency.** Describes if there are any Siebel Repository file dependencies for the specific data type before deploying ADM data types.

ADM Data Type Dependencies

The following table presents deployment dependency data for each of the supported Application Deployment Manager (ADM) data types. For more information on each of the column headings, see "About the ADM Data Type Dependency Table" on page 130.

Table 13. ADM Data Type Dependencies

Data Type	Run-Time Customization Dependency	Repository Item Dependency	SRF Dependency
Access Controlled Business Service	Business Services, Responsibilities	None	None
Access Controlled Task	Tasks, Responsibilities	None	None
Access Group	User lists	None	None
Account Promotion	Products, Promotions	None	None
ADM Abs Admin Service Region	Not applicable	Not applicable	Not applicable
ADM Abs Admin Time Window Map	Not applicable	Not applicable	Not applicable
ADM Comm Inbound Group	Not applicable	Not applicable	Not applicable
ADM CommSrv CM Adapter Administration	Not applicable	Not applicable	Not applicable
ADM Project	Data types	None	None
ADM Search Category Available Flds	None	Business components	Integration object and content object
ADM Search Connector	None	Business components	Integration object and content object
ADM Search Engine	None	Business components	Integration object and content object
ADM Search Parent Category	None	Business components	Integration object and content object
ADM Shift Exception	Not applicable	Not applicable	Not applicable
ADM Shift Schedule	Not applicable	Not applicable	Not applicable
Aggregate Discounts	Products	None	None
Aggregate Discount Sequence	Aggregate Discounts	None	None

Table 13. ADM Data Type Dependencies

Data Type	Run-Time Customization Dependency	Repository Item Dependency	SRF Dependency
Assignment Group	None	Assignment Object	None
Assignment Rule	None	Assignment Object	None
Attribute Adjustments	Product Class, Product Attributes, Variable Maps	None	None
Audit Trail Admin	None	None	None
Business Rule			
Content Fixup	None	None	None
Correspondence Template	Microsoft Word document files	Business components	Integration object and content object
Cost List	Products	None	None
Data Map	None	Business objects, business components	None
Discount Matrices	Product, Volume Discount, Attribute Adjustments, Product Class, Product Lines	None	None
EAI Data Map	None	None	Integration object
EAI Dispatch Service Rule Set	None	None	
ExpLOV: Expense Type LOV	None	None	Integration object
ExpType: Expense Type	List of Values	None	Integration object
Fund	Period	None	None
Host Table	None	None	None
iHelp	Access Groups, Responsibilities	None	Screens
iHelp - Header	Access Groups, Responsibilities	No	Screens
List of Values	None	None	None
Message Types, Administration - Order Management	None	None	None

Table 13. ADM Data Type Dependencies

Data Type	Run-Time Customization Dependency	Repository Item Dependency	SRF Dependency
Personalization—Actions	None	None	None
Personalization - Applets	None	None	None
Personalization—Event Defs	Depends on personalization profile attributes (can be set at run-time)	None	None
Personalization - Events	Depends on personalization profile attributes (can be set at run-time)	None	None
Personalization—Rules	Depends on personalization profile attributes (can be set at run-time)	None	None
Personalization - Views	None	None	None
Predefined Query	None	None	None
Product Catalog	Catalog Category, Products, Access Group	None	None
Price List	Products, Volume Discounts, Attribute Adjustments	None	None
Product Feature	Product Line	None	None
Product Line	Products	None	None
Products	Product Line, Product Class	None	None
Promotion	None	None	None
Proposal Template	Microsoft Word files, other literature files	Applet repository data, business components	Integration object and content object
Reports - Actuate Encyclopedia	None	None	None
Reports - Application Server	None	Language files	None

Table 13. ADM Data Type Dependencies

Data Type	Run-Time Customization Dependency	Repository Item Dependency	SRF Dependency
Reports - Report Server	None	Language files	None
Responsibility	Views, Users	None	Views being deployed
Siebel Repository File			
SmartScript	None	None	Any references to custom objects
SSO System	Symbolic URL	None	None
State Model	List of Values	None	None
Symbolic URL	Web Application Administration, Host Administration, Fixup Administration	None	Target of the symbolic URL does not display unless the symbolic URL name is used as a calculated field value on the underlying business component.
UI Theme	None	None	None
User List	Users	None	None
View	None	None	Views being deployed
Volume Discount	None	None	None
Web Application	None	None	None
Webmaster	None	None	None
Web Service - Inbound	None	None	Business Service
Web Service - Outbound	None	None	Business Service
Web Template	None	None	None
Workflow Action Type	None	None	None
Workflow Group	None	None	None
Workflow Policy	None	None	None

Table 13. ADM Data Type Dependencies

Data Type	Run-Time Customization Dependency	Repository Item Dependency	SRF Dependency
Workflow and Taskflow	None	A few objects require a particular order when imported to avoid invalid ROW_ID references.	None
Workload Rule	None	Assignment object	None
Workspace Projects	None	None	None

ADM Data Type Details

The Application Deployment Manager (ADM) process can vary slightly or have other limitations or configurations associated with it depending on the data type selected for migration. Review the following subset of data types for further details to configure or consider during the ADM deployment and activation process. The following ADM data types have additional ADM details:

- "List of Values" on page 135
- "Personalization—Actions" on page 136
- "Personalization—Event Defs" on page 136
- "Personalization—Rules" on page 136
- "Reports - Actuate Encyclopedia" on page 136
- "Run-Time Business Service" on page 137
- "SmartScript" on page 137
- "UI Theme" on page 137
- "Webmaster" on page 137
- "Workflow Policy" on page 139
- "Workflow and Taskflow" on page 139
- "Workspace Projects" on page 139

List of Values

The List of Values (LOV) data type has the following behavior when deployed using ADM: the hierarchy in the List of Values Explorer view is different from the hierarchy in the List of Values list view. The list view defines relations between different LOV types. Use the Parent LIC field in List of Values list view to define relations within the same type.

Personalization—Actions

The Personalization - Actions data type has the following behavior when deployed using ADM: this data type is imported together with, but before, Personalization - Events data type. Both data types require the same activation and restore activation, which are defined only for Personalization - Events.

Personalization—Event Defs

The Personalization - Events Def data type has the following behavior when deployed using ADM: this data type is imported together with, but before, the Personalization - Events data type. Both data types require the same activation and restore activation, which are defined only for Personalization - Events.

Personalization—Rules

The Personalization - Rules data type has the following behavior when deployed using ADM: this data type is imported together with, but before, the Personalization - Applets data type. Both data types require the same activation and restore activation, which are defined only for Personalization - Applets.

Reports - Actuate Encyclopedia

The Reports - Actuate Encyclopedia data type requires the encryption of the report server password when deploying with ADM. Perform the following task to encrypt the report server password.

To encrypt Actuate report server password

1. Encrypt the Actuate report server password by running the following command:

   ```
   java -classpath d:\management_server\lib\siebelmgr.jar
   com.siebel.management.util.Decoder d:\management_server\security
   d:\management_server\security\.key password
   ```

2. In the enterprise profile XML file, locate the following two sections and update them with the following values:

   - <ReportServerPassword=*password*>
   - <ReportServerPassword Encrypted=YES>

3. Encrypt your management server password with the same command as in Step 1:

   ```
   java -classpath d:\management_server\lib\siebelmgr.jar
   com.siebel.management.util.Decoder d:\managment_server\security
   d:\management_server\security\.key management_server_password
   ```

4. Copy the password displayed on the console and paste it in the Siebel Management Server's security.properties file.

   For example, `com.siebel.management.security.password=jT1qU9LE5bU=`

5. Add the location of the key file to the security.properties file.

   For example, `com.siebel.management.security.keyFileName=d:`*`/management_server/`*`security/.key`

6. Restart the Siebel Management Server.

Run-Time Business Service

The Run-Time Business Service data type has the following condition when packaging this data with ADM: the Run-Time Business Service data is only allowed in Synchronize deployment mode.

SmartScript

The SmartScript data type has the following limitations when deploying with ADM:

- The new version of existing SmartScripts will be activated when all users have stopped using old version.
- Make sure SmartScripts deployed using ADM are valid. For information on how to validate SmartScripts, see "Validating the SmartScript Data Type" on page 137.

Validating the SmartScript Data Type

SmartScripts deployed using ADM must be valid. If you export unverified SmartScripts, the generated XML file is incorrect. For more information on SmartScripts, see *Siebel SmartScript Administration Guide*.

To validate SmartScripts

1. Navigate to the Scripts view in the Administration - SmartScript screen, and select a SmartScript.
2. From the Menu button, select Verify.
3. The Wizard starts and displays the Verify Intro view.

   For more information on this task, see *Siebel SmartScript Administration Guide*.

UI Theme

To access the UI Theme data type when packaging it with ADM, so the following:

- Clicking on UI Themes data type link from Deployment Projects, Data Type list takes you to the ISS Product Administration View. From this view, you can access UI Theme records in the pick applets of the Base Theme or Product Theme fields in the Administration - Product > User Interface list applet.

Webmaster

The Webmaster data type has the following limitations when deploying with ADM:

- The value of the Webmaster destination root parameter (DestinationRoot) in the ADM enterprise profile (`MgmtSrvrInstallDir\ADM\entprofile_`*`enterprise_name`*`.xml`) is listed as `\\CHANGE_ME\upload`. It is recommended that you change this path to reflect the Uniform Naming Convention (UNC) path of the shared directory on the file server to which the Webmaster files can be copied during the copy operation. The activation operation deploys the files from this shared location to the Web server plugin directory eappweb/public/*lang*.
- The value of the WebServerHosts parameter must include the Web server port number if the default port is not used.

  For example: `<WebServerHosts>http://`*`web_server`*`:16661</WebServerHosts>`
- The language value code appended to the application contained in the application path parameter must be replaced with the correct language code if not ENU. For more information on language settings for the ADM Framework, see "Configuring Language Settings for the ADM Framework" on page 42.
- Make sure you set the application object manager (AOM) parameter, EnabledADMSupport, to True when deploying and activating the Webmaster data type. For more information on setting the AOM parameter for Webmaster, see "Setting the AOM Parameter for the Webmaster Data Type" on page 138.

Setting the AOM Parameter for the Webmaster Data Type

The application object manager (AOM) parameter, EnabledADMSupport, must be set to True when deploying and activating the Webmaster data type. Set this parameter from either the Server Manager command line or the application GUI. For information on parameter administration, see *Siebel System Administration Guide*.

To set the AOM parameter, EnableADMSupport, to True from the application GUI

1. Navigate to the Administration - Server Configuration screen > Servers view.
2. In the Siebel Servers list, select the Siebel Server of interest.
3. Select the Components view tab.
4. In the Components list, select the AOM of interest, for example, Call Center Object Manager (alias SCCObjMgr).
5. Select the Parameters view tab below the Components list.
6. From the Parameters list, select or query for EnableADMSupport.
7. Update the parameter value to True.
8. Navigate to Administration - Server Configuration screen > Enterprises > Synchronize view, and click Synchronize.
9. Restart the Siebel Server.

   For information on restarting the Siebel Server, see *Siebel System Administration Guide*.

Workflow Policy

The Workflow Policy data type requires additional attention when activating it after an ADM deployment. Perform the following task to make sure additional configurations are satisfied.

To make sure Workflow Policy data type is ready for activation

1. Make sure the Workflow Management (alias Workflow) component group is enabled. For information on this procedure, see *Siebel System Administration Guide*.
2. Update the following parameters for the Generate Triggers (alias GenTrig) server component:
   - PrivUserName to Table Owner Name (for example, SIEBEL)
   - PrivUserPass to table password (for example, db2)

   NOTE: PrivUserName is the Table Owner Name.
3. Update the named subsystem ADMJavaSubsys:
   a. Navigate to the Administration - Server Configuration screen > Enterprises > Profile Configuration view.
   b. In the Profiles list, select ADMJavaSubsys.
   c. In the Profile Parameters list, select the JVM class path parameter, and add siebelmgrclient.jar to the class path.

   NOTE: The number of characters in the class path must not exceed 100 characters.
4. Synchronize batch server component ADM Processor (ADMProc).

   For information on this task, see *Siebel System Administration Guide*.
5. Make sure the Workflow Policy Activation Server (WfPolicyActSrvr) in the ADM enterprise profile is the name of the correct server.
6. Restart Oracle’s Siebel Server, Siebel Management Agent, and Siebel Management Server.

Workflow and Taskflow

The Workflow and Taskflow repository data type has the following limitations when deploying with ADM:

- Make sure Workflow and Taskflow are in a completed state before deploying.
- Use the hotfix method to package a small number (5-10) of repository objects. For information on this task, see “Generating ADM Deployment Units Using a Hotfix” on page 76.
- Use the mid-level release method to package a large number (up to 450) of repository objects. For information on this task, see “Generating ADM Deployment Units Using a Mid-Level Release” on page 76.

Workspace Projects

The Workspace Projects data type has the following variations to general ADM deployment behavior:

- The deployment mode selection in the ADM Deployment Project view is set to custom for Workspace Projects, because this data type implements its own migration method (similar to synchronize) to make sure that the entire Workspace Project is fully migrated.
- The Workspace Project data type has its own syntax for the deployment filter field:

  ```
  [Name] = “Any joint workspace name" [Full] = “Y/N"
  ```

  where:

  - `[Name]` is required.
  - `[Full]` is optional. The filter field defaults to N if no value is specified.

  The optional Y/N flag indicates a full structure export or single object export for the objects included in the Workspace Project. In other words, Y indicates a full structure export, and N indicates a single object export.

  For example, if a Workspace Project named Vehicle Products includes a Product called Car, then if [Full] is set to Y, the associated product attributes like colors, options, accessories, and so on, are also exported, but if set to N, they are not exported. Also, only one Workspace Project name can be specified in the filter and be exported for each entry.

B ADM Support for Additional Data Types

This appendix provides data type attributes for use in extending Application Deployment Manager (ADM) to support additional database and file data types for deployment. The ADM data types described in this appendix represent Oracle's Siebel application customizations that are available for migration using the ADM framework. Refer to Appendix A, "ADM Supported Data Types" for other deployment and activation details for ADM data types. This appendix has the following topics:

- "About Data Type Attributes for Adding New ADM Data Types" on page 141
- "Data Type Attributes for Adding New ADM Data Types" on page 146

About Data Type Attributes for Adding New ADM Data Types

This topic describes the data type attribute categories in "Data Type Attributes for Adding New ADM Data Types" on page 146 that provide data type information to consider when adding new Application Deployment Manager (ADM) data types. The following terms are used in "Data Type Attributes for Adding New ADM Data Types" on page 146:

- "Parent Data Type" on page 142
- "Registry Parameters" on page 142
- "Enterprise Profile Parameters" on page 142
- "Backup Method" on page 142
- "Deploy Method" on page 142
- "Deploy Validation Method" on page 143
- "Activate Method" on page 143
- "Activate Validation Method" on page 143
- "Steps Automated by Activate or Restore Activate Methods" on page 144
- "Restore Method" on page 144
- "Delete New Objects Parameter" on page 144
- "Restore Activate Method" on page 145
- "Restore Activate Validation Method" on page 145
- "Max Units Per Package" on page 145
- "Manual Steps" on page 145
- "Additional Downtime Impact" on page 146
- "Deployment Prerequisites" on page 146

- “Notes” on page 146

Parent Data Type

Name of the parent data type for hierarchical database types.

Registry Parameters

Parameters used to provide additional information about a data type. The information is provided in a name-value pair in the registry file; for example, TargetServerType=ReportServer for `<parameter name="TargetServerType" value="ReportServer"/>`. Each data type can have any number of parameters defined.

NOTE: The parameters are read by the code and used according to the data type's requirement. You cannot add a new parameter and expect the ADM framework to consider it without code changes.

Enterprise Profile Parameters

Name of parameters specified in the entprofile.xml file for a particular data type.

Backup Method

A backup method is used to back up data in the target environment, which will be affected during the copy operation of the package. The backed up data can later be used in the restore operation to revert the changes. There are four possible values:

- **Not Applicable.** There is no way to back up this data type. In this case, deployment will copy new data without backing up the old data, and there would be no way to restore the original data after deployment is executed. However, there are a few exceptions in which the restore operation can revert the changes to a version in the target environment without having performed a backup operation prior to deployment.
- **Business Service.** The specified custom Siebel Business service will be called, and the method in the Method attribute will be invoked.
- **EAI.** The default implementation for database types.
- **Mbean.** The default implementation for file types.

Deploy Method

The method that performs the copy operation. There are four possible values:

- **Business Service.** The specified custom Siebel Business service will be called, and the method in the Method attribute will be invoked.
- **EAI.** The default implementation for database types.
- **Mbean.** The default implementation for file types.

- **JavaAPI.** A custom java implementation, with the class and method as arguments.

NOTE: The way the target data is affected for database items depends on the deployment mode used during export.

Deploy Validation Method

Validation performed before executing the copy operation. The following methods can be invoked:

- **CheckSchemaVersion.** The method called to check the database schema version of the source database, present in the exported XML file, against the database schema version of the target database. The validation is successful if the versions match. The validation fails if the versions do not match.
- **ServerComponentStatus.** The method called to check the state of a specified Siebel Server component on the target server against a given state (running, offline, or online), for example,

  ```
  <component>
      <name>ADMObjMgr</name>
      <status>Online</status>
  </component>
  ```

  where name is the component name, and status is the status of the component. If the states match, the validation is successful. If the states do not match, the validation fails. In this example, if ADMObjMgr is online on the target server, then the validation is successful.

Activate Method

The method that makes the deployed data available to users in the target environment. There are two possible values:

- **Not Applicable.** Either there is no activation step, or the activation step is not automated.
- **Business Service.** The specified custom Siebel Business service will be called, and the method in the Method attribute will be invoked.

NOTE: Not all data types need activation. Some data types are immediately available to the users after deployment.

Activate Validation Method

Validation performed before executing the activate operation. The following methods can be invoked:

- **Not Applicable.**

- **ServerComponentStatus.** The method called to check the state of a specified Siebel Server component on the target server against a given state (running, offline, or online), for example,

  ```
  <component>
      <name>ADMObjMgr</name>
      <status>Online</status>
  </component>
  ```

  where name is the component name, and status is the status of the component. If the states match, the validation is successful. If the states do not match, the validation fails. In this example, if ADMObjMgr is online on the target server, then the validation is successful.

Steps Automated by Activate or Restore Activate Methods

Steps that are automatically executed with the activate or restore activate methods. Necessary steps that are not automated must be manually performed.

Restore Method

The method used to restore the target system to the same functional level as before the deployment operations were invoked. There are four possible values:

- **Business Service.** The specified custom Siebel Business service will be called, and the method in the Method attribute will be invoked.
- **EAI.** The default implementation for database types.
- **Mbean.** The default implementation for file types.
- **JavaAPI.** A custom java implementation, with the class and method as arguments.

NOTE: The restore implementation is not necessarily an "undo" mechanism, even though it restores the original data to the target system. For example, the version of versioned objects may increment. Additional actions may be required to fully restore the target system to its original state prior to deployment.

Delete New Objects Parameter

If this ADM registry parameter is set to Yes during a restore operation, then all top-level objects newly created by the deployment operation are deleted from the target system. If this parameter is set to No, then the new objects are not deleted, but remain in the target system.

If delete is not appropriate for a specific object type, it may be necessary to modify the objects to render it inactive. For example, you may need to mark an object as inactive to, in effect, achieve the functional state before deployment.

If an object is created during deployment, and is activated after deployment, then a custom restore method must be performed to implement the additional steps needed for completely deactivating the object.

If the newly created top-level objects should not or cannot be deleted during a restore operation, then the value of the parameter must be set to "No", and a comment should be entered in the registry to clearly describe why.

Restore Activate Method

The method that makes the restored data available to users in the target environment. There are two possible values:

- **Not Applicable.** Either there is no restore activate step, or the restore activate step is not automated.
- **Business Service.** The specified custom Siebel Business service will be called, and the method in the Method attribute will be invoked.

NOTE: Not all data types need activation. Some data types are immediately available to the users after restore.

Restore Activate Validation Method

The validation method performed before executing the restore activate operation. The following methods can be invoked:

- **Not Applicable.**
- **ServerComponentStatus.** The method called to check the state of a specified Siebel Server component on the target server against a given state (running, offline, or online), for example,

```
<component>
    <name>ADMObjMgr</name>
    <status>Online</status>
</component>
```

  where name is the component name, and status is the status of the component. If the states match, the validation is successful. If the states do not match, the validation fails. In this example, if ADMObjMgr is online on the target server, then the validation is successful.

Max Units Per Package

The maximum number of units allowed in an ADM package due to limitations with the product. The ADM Package Utility enforces the limit.

Manual Steps

Manual steps required for deploying the data type. These steps include clicking buttons, restarting a service, clearing the cache, and so on.

Additional Downtime Impact

Other impacts to system availability or downtime during deployment, in addition to those listed in Table 12 on page 127.

Deployment Prerequisites

Prerequisites for successful deployment of the data type.

Notes

Special considerations, limitations, and comments for the deployment of the data type.

Data Type Attributes for Adding New ADM Data Types

This topic contains data type attributes that you may use toward adding new Application Deployment Manager (ADM) data types for deployment. For more information on each of the data type attribute categories, see "About Data Type Attributes for Adding New ADM Data Types" on page 141. This topic includes information for the following ADM data types:

- "Access Controlled Business Service Database Data Type" on page 148
- "Access Controlled Task Database Data Type" on page 149
- "AccessGroup Database Data Type" on page 150
- "AccntPromo Database Data Type" on page 151
- "ADM Abs Admin Service Region Database Data Type" on page 152
- "ADM Abs Admin Time Window Map Database Data Type" on page 153
- "ADM Comm Inbound Group Database Data Type" on page 154
- "ADM CommSrv CM Adapter Administration Database Data Type" on page 155
- "ADM Project Database Data Type" on page 156
- "ADM Search Category Available Flds Database Data Type" on page 157
- "ADM Search Connector Database Data Type" on page 158
- "ADM Search Engine Database Data Type" on page 159
- "MessageType Database Data Type" on page 185
- "PDQ (Predefined Query) Database Data Type" on page 186
- "Personalization—Actions Database Data Type" on page 187
- "Personalization—Applets Database Data Type" on page 188
- "Personalization—Event Defs Database Data Type" on page 189
- "Personalization—Events Database Data Type" on page 190
- "Personalization—Rules Database Data Type" on page 191
- "Personalization—Views Database Data Type" on page 192
- "PriceList Database Data Type" on page 193
- "ProdFeature Database Data Type" on page 194
- "ProdLine Database Data Type" on page 195
- "ProductCatalog Database Data Type" on page 196

- "ADM Search Parent Category Database Data Type" on page 160
- "ADM Shift Exception Database Data Type" on page 161
- "ADM Shift Schedule Database Data Type" on page 162
- "AggrDiscount Database Data Type" on page 163
- "AggrDiscSeq Database Data Type" on page 164
- "AssignGroup Database Data Type" on page 165
- "AssignRule Database Data Type" on page 166
- "AttrAdjustments Database Data Type" on page 167
- "Audit Trail Admin Database Data Type" on page 168
- "Business Rule Database Data Type" on page 169
- "Content Fixup Database Data Type" on page 170
- "Correspondence Template Database Data Type" on page 171
- "CostList Database Data Type" on page 172
- "DataMapObject Database Data Type" on page 173
- "Discount and EC Matrix Database Data Type" on page 174
- "EAI Data Map Database Data Type" on page 175
- "EAI Dispatch Service Rule Set Database Data Type" on page 176
- "ExpLOV Database Data Type" on page 177
- "ExpType Database Data Type" on page 178
- "Fund Database Data Type" on page 179
- "Host Table Database Data Type" on page 180
- "iHelp Database Data Type" on page 181
- "ProductData Database Data Type" on page 197
- "Promotion Database Data Type" on page 198
- "Proposal Template Database Data Type" on page 199
- "Reports—Actuate Encyclopedia File Data Type" on page 200
- "Reports—Application Server File Data Type" on page 201
- "Reports—Report Server File Data Type" on page 202
- "Responsibility Database Data Type" on page 203
- "SmartScript Database Data Type" on page 204
- "SRF (Siebel Repository File) File Data Type" on page 205
- "StateModel Database Data Type" on page 206
- "Symbolic URL Database Data Type" on page 207
- "UIThemeMapping Database Data Type" on page 208
- "UserList Database Data Type" on page 209
- "View Database Data Type" on page 210
- "VolumeDiscount Database Data Type" on page 211
- "Web Application Database Data Type" on page 212
- "Webmaster File Data Type" on page 213
- "Web Service—Inbound Database Data Type" on page 214
- "Web Service—Outbound Database Data Type" on page 215
- "Web Template File Data Type" on page 216
- "Workflow and Taskflow Repository Data Type" on page 217
- "Workflow Action Type Database Data Type" on page 218

- "iHelp—Header Database Data Type" on page 182
- "JointWorkspace (Workspace Projects) Database Data Type" on page 183
- "LOV (List of Values) Database Data Type" on page 184
- "Workflow Group Database Data Type" on page 219
- "Workflow Policy Database Data Type" on page 220
- "Workload Rule Database Data Type" on page 221

Access Controlled Business Service Database Data Type

Attributes for the Access Controlled Business Service database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Access Controlled Business Service data type attributes, see Table 14. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 14. Attributes for the Access Controlled Business Service Database Data Type

Data Type Attribute	Access Controlled Business Service
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=Access Controlled BS Cache Clear Service; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=Access Controlled BS Cache Clear Service; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	User reconnect in order to see changes
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Access Controlled Task Database Data Type

Attributes for the Access Controlled Task database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Access Controlled Task data type attributes, see Table 15. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 15. Attributes for the Access Controlled Task Database Data Type

Data Type Attribute	Access Controlled Task
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=Access Controlled Task Cache Clear Service; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=Access Controlled Task Cache Clear Service; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	User reconnect in order to see changes
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

AccessGroup Database Data Type

Attributes for the AccessGroup database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the AccessGroup data type attributes, see Table 16. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 16. Attributes for the AccessGroup Database Data Type

Data Type Attribute	AccessGroup
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

AccntPromo Database Data Type

Attributes for the AccntPromo database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the AccntPromo data type attributes, see Table 17. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 17. Attributes for the AccntPromo Database Data Type

Data Type Attribute	AccntPromo
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ADM Abs Admin Service Region Database Data Type

Attributes for the ADM Abs Admin Service Region database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Abs Admin Service Region data type attributes, see Table 18. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 18. Attributes for the ADM Abs Admin Service Region Database Data Type

Data Type Attribute	ADM Abs Admin Service Region
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ADM Abs Admin Time Window Map Database Data Type

Attributes for the ADM Abs Admin Time Window Map database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Abs Admin Time Window Map data type attributes, see Table 19. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 19. Attributes for the ADM Abs Admin Time Window Map Database Data Type

Data Type Attribute	ADM Abs Admin Time Window Map
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ADM Comm Inbound Group Database Data Type

Attributes for the ADM Comm Inbound Group database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Comm Inbound Group data type attributes, see Table 20. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 20. Attributes for the ADM Comm Inbound Group Database Data Type

Data Type Attribute	ADM Comm Inbound Group
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ADM CommSrv CM Adapter Administration Database Data Type

Attributes for the ADM CommSrv CM Adapter Administration database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM CommSrv CM Adapter Administration data type attributes, see Table 21. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 21. Attributes for the ADM CommSrv CM Adapter Administration Database Data Type

Data Type Attribute	ADM CommSrv CM Adapter Administration
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ADM Project Database Data Type

Attributes for the ADM Project database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Project data type attributes, see Table 22. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 22. Attributes for the ADM Project Database Data Type

Data Type Attribute	ADM Project
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	All data types included in the exported project must be existing on the target system.
Notes	None

ADM Search Category Available Flds Database Data Type

Attributes for the ADM Search Category Available Flds database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Search Category Available Flds data type attributes, see Table 23. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 23. Attributes for the ADM Search Category Available Flds Database Data Type

Data Type Attribute	ADM Search Category Available Flds
Parent Data Type	ADM Search Parent Category
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	ADM, Workflow, and EAI components must be enabled on the target server.
Notes	None

ADM Search Connector Database Data Type

Attributes for the ADM Search Connector database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Search Connector data type attributes, see Table 24. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 24. Attributes for the ADM Search Connector Database Data Type

Data Type Attribute	ADM Search Connector
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	ADM, Workflow, and EAI components must be enabled on the target server.
Notes	None

ADM Search Engine Database Data Type

Attributes for the ADM Search Engine database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Search Engine data type attributes, see Table 25. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 25. Attributes for the ADM Search Engine Database Data Type

Data Type Attribute	ADM Search Engine
Parent Data Type	ADM Search Connector
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	ADM, Workflow, and EAI components must be enabled on the target server.
Notes	None

ADM Search Parent Category Database Data Type

Attributes for the ADM Search Parent Category database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Search Parent Category data type attributes, see Table 26. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 26. Attributes for the ADM Search Parent Category Database Data Type

Data Type Attribute	ADM Search Parent Category
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	ADM, Workflow, and EAI components must be enabled on the target server.
Notes	None

ADM Shift Exception Database Data Type

Attributes for the ADM Shift Exception database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Shift Exception data type attributes, see Table 27. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 27. Attributes for the ADM Shift Exception Database Data Type

Data Type Attribute	ADM Shift Exception
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ADM Shift Schedule Database Data Type

Attributes for the ADM Shift Schedule database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ADM Shift Schedule data type attributes, see Table 28. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 28. Attributes for the ADM Shift Schedule Database Data Type

Data Type Attribute	ADM Shift Schedule
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

AggrDiscount Database Data Type

Attributes for the AggrDiscount database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the AggrDiscount data type attributes, see Table 29. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 29. Attributes for the AggrDiscount Database Data Type

Data Type Attribute	AggrDiscount
Parent Data Type	AggrDiscSeq
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=ADM on Aggregate Discount; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=ADM on Aggregate Discount; Method=Activate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

AggrDiscSeq Database Data Type

Attributes for the AggrDiscSeq database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the AggrDiscSeq data type attributes, see Table 30. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 30. Attributes for the AggrDiscSeq Database Data Type

Data Type Attribute	AggrDiscSeq
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=ADM on Aggregate Discount Sequence; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=ADM on Aggregate Discount Sequence; Method=Activate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

AssignGroup Database Data Type

Attributes for the AssignGroup database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the AssignGroup data type attributes, see Table 31. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 31. Attributes for the AssignGroup Database Data Type

Data Type Attribute	AssignGroup
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	Click the Release button in the UI for the changes to be effective.
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

AssignRule Database Data Type

Attributes for the AssignRule database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the AssignRule data type attributes, see Table 32. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 32. Attributes for the AssignRule Database Data Type

Data Type Attribute	AssignRule
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	Click the Release button in the UI for the changes to be effective.
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	This data type requires activation.

AttrAdjustments Database Data Type

Attributes for the AttrAdjustments database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the AttrAdjustments data type attributes, see Table 33. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 33. Attributes for the AttrAdjustments Database Data Type

Data Type Attribute	AttrAdjustments
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=ADM on Attribute Adjustments; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=ADM on Attribute Adjustments; Method=Activate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Audit Trail Admin Database Data Type

Attributes for the Audit Trail Admin database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Audit Trail Admin data type attributes, see Table 34. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 34. Attributes for the Audit Trail Admin Database Data Type

Data Type Attribute	Audit Trail Admin
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=Audit Trail Configuration Service; Method=UpdateAuditCache
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=Audit Trail Configuration Service; Method=UpdateAuditCache
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Business Rule Database Data Type

Attributes for the Business Rule database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Business Rule data type attributes, see Table 35. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 35. Attributes for the Business Rule Database Data Type

Data Type Attribute	Business Rule
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	Business service=Rule Runtime Administration; Method=Backup
Deploy Method	Business service=Rule Runtime Administration; Method=Deployment
Deploy Validation Method	
Activate Method	Business service=Rule Runtime Administration; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	Business service=Rule Runtime Administration; Method=Restore
Delete New Objects Parameter	Yes
Restore Activate Method	
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Content Fixup Database Data Type

Attributes for the Content Fixup database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Content Fixup data type attributes, see Table 36. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 36. Attributes for the Content Fixup Database Data Type

Data Type Attribute	Content Fixup
Parent Data Type	Symbolic URL
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Correspondence Template Database Data Type

Attributes for the Correspondence Template database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Correspondence Template data type attributes, see Table 37. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 37. Attributes for the Correspondence Template Database Data Type

Data Type Attribute	Correspondence Template
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	BusinessService=ADM Service; MethodName=start
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

CostList Database Data Type

Attributes for the CostList database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the CostList data type attributes, see Table 38. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 38. Attributes for the CostList Database Data Type

Data Type Attribute	CostList
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

DataMapObject Database Data Type

Attributes for the DataMapObject database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the DataMapObject data type attributes, see Table 39. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 39. Attributes for the DataMapObject Database Data Type

Data Type Attribute	DataMapObject
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Discount and EC Matrix Database Data Type

Attributes for the Discount and EC Matrix database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Discount and EC Matrix data type attributes, see Table 40. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 40. Attributes for the Discount and EC Matrix Database Data Type

Data Type Attribute	Discount and EC Matrix
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=ADM on Discount and E&C Matrix; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=ADM on Discount and E&C Matrix; Method=Activate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

EAI Data Map Database Data Type

Attributes for the EAI Data Map database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the EAI Data Map data type attributes, see Table 41. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 41. Attributes for the EAI Data Map Database Data Type

Data Type Attribute	EAI Data Map
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=EAI Data Transformation Engine; Method=Purge
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=EAI Data Transformation Engine; Method=Purge
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

EAI Dispatch Service Rule Set Database Data Type

Attributes for the EAI Dispatch Service Rule Set database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the EAI Dispatch Service Rule Set data type attributes, see Table 42. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 42. Attributes for the EAI Dispatch Service Rule Set Database Data Type

Data Type Attribute	EAI Dispatch Service Rule Set
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=EAI Dispatch Service Rule Set - ADM; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=EAI Dispatch Service Rule Set - ADM; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ExpLOV Database Data Type

Attributes for the ExpLOV database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ExpLOV data type attributes, see Table 43. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 43. Attributes for the ExpLOV Database Data Type

Data Type Attribute	ExpLOV
Parent Data Type	ExpType
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=LOV Cache Clear Service; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=LOV Cache Clear Service; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ExpType Database Data Type

Attributes for the ExpType database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ExpType data type attributes, see Table 44. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 44. Attributes for the ExpType Database Data Type

Data Type Attribute	ExpType
Parent Data Type	ExpType
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Fund Database Data Type

Attributes for the Fund database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Fund data type attributes, see Table 45. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 45. Attributes for the Fund Database Data Type

Data Type Attribute	Fund
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Host Table Database Data Type

Attributes for the Host Table database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Host Table data type attributes, see Table 46. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 46. Attributes for the Host Table Database Data Type

Data Type Attribute	Host Table
Parent Data Type	Symbolic URL
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

iHelp Database Data Type

Attributes for the iHelp database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the iHelp data type attributes, see Table 47. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 47. Attributes for the iHelp Database Data Type

Data Type Attribute	iHelp
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=Task Import/Export; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=Task Import/Export; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	Click the Activate button in the UI to activate this data type.
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

iHelp—Header Database Data Type

Attributes for the iHelp—Header database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the iHelp—Header data type attributes, see Table 48. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 48. Attributes for the iHelp—Header Database Data Type

Data Type Attribute	iHelp—Header
Parent Data Type	iHelp
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=Task Import/Export; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=Task Import/Export; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

JointWorkspace (Workspace Projects) Database Data Type

Attributes for the JointWorkspace (Workspace Projects) database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the JointWorkspace data type attributes, see Table 49. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 49. Attributes for the JointWorkspace Database Data Type

Data Type Attribute	JointWorkspace (Workspace Projects)
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	Not applicable
Deploy Method	Business service=ISS Authoring Import Export Service; Method=ADMImport
Deploy Validation Method	
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	Not applicable
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	Click the Release New Version or Release All button in the UI.
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

LOV (List of Values) Database Data Type

Attributes for the LOV (List of Values) database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the LOV data type attributes, see Table 50. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 50. Attributes for the LOV Database Data Type

Data Type Attribute	LOV (List of Values)
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=LOV Cache Clear Service; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=LOV Cache Clear Service; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	User reconnect in order to see changes
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

MessageType Database Data Type

Attributes for the MessageType database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the MessageType data type attributes, see Table 51. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 51. Attributes for the MessageType Database Data Type

Data Type Attribute	MessageType
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=ADM on Message Type; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=ADM on Message Type; Method=Activate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

PDQ (Predefined Query) Database Data Type

Attributes for the PDQ (Predefined Query) database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the PDQ data type attributes, see Table 52. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 52. Attributes for the PDQ Database Data Type

Data Type Attribute	PDQ (Predefined Query)
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Personalization—Actions Database Data Type

Attributes for the Personalization—Actions database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Personalization—Actions data type attributes, see Table 53. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 53. Attributes for the Personalization—Actions Database Data Type

Data Type Attribute	Personalization—Actions
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	For information on the behavior of this data type during deployment, see "Personalization—Actions" on page 136.

Personalization—Applets Database Data Type

Attributes for the Personalization—Applets database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Personalization—Applets data type attributes, see Table 54. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 54. Attributes for the Personalization—Applets Database Data Type

Data Type Attribute	Personalization—Applets
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=CT ADM Service; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=CT ADM Service; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Personalization—Event Defs Database Data Type

Attributes for the Personalization—Event Defs database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Personalization—Event Defs data type attributes, see Table 55. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 55. Attributes for the Personalization—Event Defs Database Data Type

Data Type Attribute	Personalization—Event Defs
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	For information on the behavior of this data type during deployment, see “Personalization—Event Defs” on page 136.

Personalization—Events Database Data Type

Attributes for the Personalization—Events database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Personalization—Events data type attributes, see Table 56. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 56. Attributes for the Personalization—Events Database Data Type

Data Type Attribute	Personalization—Events
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=CT ADM Service; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=CT ADM Service; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Personalization—Rules Database Data Type

Attributes for the Personalization—Rules database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Personalization—Rules data type attributes, see Table 57. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 57. Attributes for the Personalization—Rules Database Data Type

Data Type Attribute	Personalization—Rules
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	For information on the behavior of this data type during deployment, see "Personalization—Rules" on page 136.

Personalization—Views Database Data Type

Attributes for the Personalization—Views database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Personalization—Views data type attributes, see Table 58. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 58. Attributes for the Personalization—Views Database Data Type

Data Type Attribute	Personalization—Views
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=CT ADM Service; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=CT ADM Service; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

PriceList Database Data Type

Attributes for the PriceList database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the PriceList data type attributes, see Table 59. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 59. Attributes for the PriceList Database Data Type

Data Type Attribute	PriceList
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=ADM on Price List; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=ADM on Price List; Method=Activate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ProdFeature Database Data Type

Attributes for the ProdFeature database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ProdFeature data type attributes, see Table 60. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 60. Attributes for the ProdFeature Database Data Type

Data Type Attribute	ProdFeature
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ProdLine Database Data Type

Attributes for the ProdLine database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ProdLine data type attributes, see Table 61. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 61. Attributes for the ProdLine Database Data Type

Data Type Attribute	ProdLine
Parent Data Type	ProdFeature
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ProductCatalog Database Data Type

Attributes for the ProductCatalog database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ProductCatalog data type attributes, see Table 62. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 62. Attributes for the ProductCatalog Database Data Type

Data Type Attribute	ProductCatalog
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

ProductData Database Data Type

Attributes for the ProductData database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the ProductData data type attributes, see Table 63. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 63. Attributes for the ProductData Database Data Type

Data Type Attribute	ProductData
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	ProductData can only be updated. It cannot be inserted.

Promotion Database Data Type

Attributes for the Promotion database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Promotion data type attributes, see Table 64. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 64. Attributes for the Promotion Database Data Type

Data Type Attribute	Promotion
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Proposal Template Database Data Type

Attributes for the Proposal Template database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Proposal Template data type attributes, see Table 65. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 65. Attributes for the Proposal Template Database Data Type

Data Type Attribute	Proposal Template
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	BusinessService=ADM Service; MethodName=start
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Reports—Actuate Encyclopedia File Data Type

Attributes for the Reports—Actuate Encyclopedia file data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Reports—Actuate Encyclopedia data type attributes, see Table 66. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 66. Attributes for the Reports—Actuate Encyclopedia File Data Type

Data Type Attribute	Reports—Actuate Encyclopedia
Parent Data Type	Not applicable

Registry Parameters	TargetServerType=ReportServer;
	DestinationDirectory=reports;
	ValidFileExtensions=rox;None

Enterprise Profile Parameters	ReportServerUserName, ReportServerPassword, MaxCount, Actuate Server host, ReportFolder
Backup Method	Not applicable

Deploy Method	JavaAPI=com.siebel.adm.agent.file.javaAPI.UploadRox;
	Method=deploy

Deploy Validation Method	Not applicable
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Not applicable

Restore Method	JavaAPI=com.siebel.adm.agent.file.javaAPI.UploadRox;
	Method=restore

Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	Report Server must be running
Notes	None

Reports—Application Server File Data Type

Attributes for the Reports—Application Server file data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Reports—Application Server data type attributes, see Table 67. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 67. Attributes for the Reports—Application Server File Data Type

Data Type Attribute	Reports—Application Server
Parent Data Type	Not applicable

Registry Parameters	TargetServerType=AppServer;
	DestinationDirectory=reports/%LANG%;
	ValidFileExtensions=rox,txt

Enterprise Profile Parameters	None
Backup Method	Mbean
Deploy Method	Mbean
Deploy Validation Method	Not applicable
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Not applicable
Restore Method	Mbean
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Reports—Report Server File Data Type

Attributes for the Reports—Report Server file data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Reports—Reports Server data type attributes, see Table 68. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 68. Attributes for the Reports—Report Server File Data Type

Data Type Attribute	Reports—Report Server
Parent Data Type	Not applicable

Registry Parameters	TargetServerType=ReportServer;
	DestinationDirectory=reports/%LANG%;
	ValidFileExtensions=txt

Enterprise Profile Parameters	None
Backup Method	Mbean
Deploy Method	Mbean
Deploy Validation Method	Not applicable
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Not applicable
Restore Method	Mbean
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Responsibility Database Data Type

Attributes for the Responsibility database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Responsibility data type attributes, see Table 69. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 69. Attributes for the Responsibility Database Data Type

Data Type Attribute	Responsibility
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

SmartScript Database Data Type

Attributes for the SmartScript database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the SmartScript data type attributes, see Table 70. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 70. Attributes for the SmartScript Database Data Type

Data Type Attribute	SmartScript
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	Business service=SmartScriptADM; Method=Backup
Deploy Method	Business service=SmartScriptADM; Method=Deploy
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=SmartScriptADM; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Activates the SmartScripts in the SmartScript administration. These SmartScripts are then available in the player for users.
Restore Method	Business service=SmartScriptADM; Method=Restore
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=SmartScriptADM; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	1
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	For information on validating SmartScripts, see “SmartScript” on page 137.

SRF (Siebel Repository File) File Data Type

Attributes for the SRF (Siebel Repository File) file data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the SRF data type attributes, see Table 71. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 71. Attributes for the SRF File Data Type

Data Type Attribute	SRF (Siebel Repository File)
Parent Data Type	Not applicable

Registry Parameters	TargetServerType=AppServer;
	DestinationDirectory=objects/%LANG%;
	ValidFileExtensions=srf

Enterprise Profile Parameters	DestinationRoot
Backup Method	Mbean
Deploy Method	Mbean
Deploy Validation Method	Not applicable
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Not applicable
Restore Method	Mbean
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	The target Siebel Server must be shut down prior to deployment if it is using this SRF file.
Notes	None

StateModel Database Data Type

Attributes for the StateModel database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the StateModel data type attributes, see Table 72. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 72. Attributes for the StateModel Database Data Type

Data Type Attribute	StateModel
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Symbolic URL Database Data Type

Attributes for the Symbolic URL database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Symbolic URL data type attributes, see Table 73. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 73. Attributes for the Symbolic URL Database Data Type

Data Type Attribute	Symbolic URL
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

UIThemeMapping Database Data Type

Attributes for the UIThemeMapping database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the UIThemeMapping data type attributes, see Table 74. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 74. Attributes for the UIThemeMapping Database Data Type

Data Type Attribute	UIThemeMapping
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

UserList Database Data Type

Attributes for the UserList database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the UserList data type attributes, see Table 75. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 75. Attributes for the UserList Database Data Type

Data Type Attribute	UserList
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	This data type affects user visibility over objects of catalog and group visibility type. After import, users may see different set of products, solutions, and others.

View Database Data Type

Attributes for the View database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the View data type attributes, see Table 76. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 76. Attributes for the View Database Data Type

Data Type Attribute	View
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

VolumeDiscount Database Data Type

Attributes for the VolumeDiscount database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the VolumeDiscount data type attributes, see Table 77. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 77. Attributes for the VolumeDiscount Database Data Type

Data Type Attribute	VolumeDiscount
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=ADM on Volume Discount; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=ADM on Volume Discount; Method=Activate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Web Application Database Data Type

Attributes for the Web Application database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Web Application data type attributes, see Table 78. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 78. Attributes for the Web Application Database Data Type

Data Type Attribute	Web Application
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Webmaster File Data Type

Attributes for the Webmaster file data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the XXX data type attributes, see Table 79. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 79. Attributes for the Webmaster File Data Type

Data Type Attribute	Webmaster
Parent Data Type	Not applicable

Registry Parameters	TargetServerType=AppServer;
	DestinationDirectory=webmaster;
	ValidFileExtensions=gif,html,jpg,css,js

Enterprise Profile Parameters	DestinationRoot, WebServerHosts, ApplicationPath
Backup Method	Mbean
Deploy Method	Mbean
Deploy Validation Method	Not applicable
Activate Method	Business service=Web Public File synchronization service; Method=TriggerFileSync
Activate Validation Method	Method=ServerComponent. ADMObjMgr and ADMProc must be online.
Steps Automated by Activate or Restore Activate Methods	The business service copies the web files to the web server eApps/webmaster folder and updates the database with the new version string. The web plug-in uses this new version string to generate new version URLs.
Restore Method	Mbean
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=Web Public File synchronization service; Method=TriggerFileSync
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit.
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	The deployment service must be invoked on a single application server in an enterprise.
Notes	None

Web Service—Inbound Database Data Type

Attributes for the Web Service—Inbound database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Web Service—Inbound data type attributes, see Table 80. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 80. Attributes for the Web Service—Inbound Database Data Type

Data Type Attribute	Web Service—Inbound
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=Web Service - ADM; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=Web Service - ADM; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Web Service—Outbound Database Data Type

Attributes for the Web Service—Outbound database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Web Service—Outbound data type attributes, see Table 81. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 81. Attributes for the Web Service—Outbound Database Data Type

Data Type Attribute	Web Service—Outbound
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=Web Service - ADM; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Clear cache
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Business service=Web Service - ADM; Method=RestoreActivate
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Web Template File Data Type

Attributes for the Web Template file data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Web Template data type attributes, see Table 82. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 82. Attributes for the Web Template File Data Type

Data Type Attribute	Web Template
Parent Data Type	Not applicable

Registry Parameters	TargetServerType=AppServer;
	DestinationDirectory=webtempl;
	ValidFileExtensions=swt,swf,sws,xsl

Enterprise Profile Parameters	None
Backup Method	Mbean
Deploy Method	Mbean
Deploy Validation Method	Siebel Server must be shut down.
Activate Method	Not applicable. The new or modified template file is available after you restart the Siebel Server.
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	Not applicable
Restore Method	Mbean
Delete New Objects Parameter	Not applicable
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit. All Siebel Web Template files must be copied.
Manual Steps	Application server restart
Additional Downtime Impact	Siebel Enterprise Server restart
Deployment Prerequisites	Siebel Server shutdown
Notes	None

Workflow and Taskflow Repository Data Type

Attributes for the Workflow and Taskflow repository data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Workflow and Taskflow data type attributes, see Table 83. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 83. Attributes for the Workflow and Taskflow Repository Data Type

Data Type Attribute	Workflow and Taskflow
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	Target repository name, descriptor file name, location for backup files (UNC path), log directory, log file name, package name
Backup Method	Business service=Siebel Tools Deployment Support for ADM; Method=Backup
Deploy Method	Business service=Siebel Tools Deployment Support for ADM; Method=Deploy
Deploy Validation Method	ADMObjMgr and ADMProc must be online.
Activate Method	Business service=Siebel Tools Deployment Support for ADM; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	Business service=Siebel Tools Deployment Support for ADM; Method=Restore
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit.
Manual Steps	Activate from UI and create package (includes descriptor file/SIF pair).
Additional Downtime Impact	■ Up to 450 objects must be imported (in overwrite mode) in less than 45 minutes. ■ Up to 450 objects must be exported in under 5 minutes.

Table 83. Attributes for the Workflow and Taskflow Repository Data Type

Data Type Attribute	Workflow and Taskflow
Deployment Prerequisites	■ ADMObjMgr and ADMProc must be online. ■ You must compile on all Siebel supported platforms, since the export and import services run in either Siebel Tools or the Siebel Object Manager.
Notes	Objects are versioned. For more information, see "Workflow and Taskflow" on page 139.

Workflow Action Type Database Data Type

Attributes for the Workflow Action Type database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Workflow Action Type data type attributes, see Table 84. For more information on this data type, see Appendix A, "ADM Supported Data Types".

Table 84. Attributes for the Workflow Action Type Database Data Type

Data Type Attribute	Workflow Action Type
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit.
Manual Steps	None
Additional Downtime Impact	None

Table 84. Attributes for the Workflow Action Type Database Data Type

Data Type Attribute	Workflow Action Type
Deployment Prerequisites	None
Notes	None

Workflow Group Database Data Type

Attributes for the Workflow Group database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Workflow Group data type attributes, see Table 85. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 85. Attributes for the Workflow Group Database Data Type

Data Type Attribute	Workflow Group
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Business service=Siebel Tools Deployment Support for ADM; Method=Activate
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit.
Manual Steps	None
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Workflow Policy Database Data Type

Attributes for the Workflow Policy database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Workflow Policy data type attributes, see Table 86. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 86. Attributes for the Workflow Policy Database Data Type

Data Type Attribute	Workflow Policy
Parent Data Type	Not applicable
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit.
Manual Steps	Regenerate triggers and restart Workflow Monitor and Action agents.
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Workload Rule Database Data Type

Attributes for the Workload Rule database data type can be used toward adding new Application Deployment Manager (ADM) data types for deployment. For a list of the Workload Rule data type attributes, see Table 87. For more information on this data type, see Appendix A, “ADM Supported Data Types”.

Table 87. Attributes for the Workload Rule Database Data Type

Data Type Attribute	Workload Rule
Parent Data Type	AssignRule
Registry Parameters	None
Enterprise Profile Parameters	None
Backup Method	EAI
Deploy Method	EAI
Deploy Validation Method	Method=CheckSchemaVersion
Activate Method	Not applicable
Activate Validation Method	Not applicable
Steps Automated by Activate or Restore Activate Methods	None
Restore Method	EAI
Delete New Objects Parameter	Yes
Restore Activate Method	Not applicable
Restore Activate Validation Method	Not applicable
Max Units Per Package	No limit.
Manual Steps	Click Release button in UI to make the changes effective.
Additional Downtime Impact	None
Deployment Prerequisites	None
Notes	None

Index