ORACLE®

ATG WEB COMMERCE

Version 10.0

Upgrade and Migration Guide

9.1 to 10.0

Oracle ATG
One Main Street
Cambridge, MA 02142
USA

Oracle ATG Web Commerce Upgrade and Migration Guide 9.1 to 10.0

Document Version

ATG10.0 MIGRATIONv2 12/21/10

Copyright

Copyright © 1997, 2011, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS
Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/us/corporate/accessibility/index.html.

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.

Contents

1 Introduction 1

2 New in ATG 10.0 3
- Platform 3
  - Site Administration 3
  - ATG Content Administration 3
  - GSARepository 4
  - Rest Web Services 4
- ATG Commerce 4
- ATG Commerce Reference Store 5
- ATG Search 6
- ATG Merchandising 7
- ATG Customer Intelligence 7
- ATG Campaign Optimizer 7
- ATG Commerce Service Center (CSC) 8
- ATG Outreach 8
- ATG Service 8

3 Migration Steps 11
- General Pre-Migration Tasks 11
- ATG Installation 11
- Database Schema Migration 12
  - Batch File/Shell Script Naming Conventions 12
  - Batch Files versus SQL Scripts 12
  - Before Running Migration Scripts 13
  - Running Batch Files 13
  - Running DDL Scripts 13
- Application-Specific Migration Tasks 14
- General Post-Migration Tasks 14
- Migration Testing 14

4 Oracle Database Migration 15
- Basic Platform 15
- Publishing.Base 16
- Portals 17
- B2CCommerce 17
- B2BCommerce 18
- Data Warehouse 19
- ATG Search 20

Agent 21
ATG Service 22
ATG Service Schema Migration 22
ATG Service Data Migration 24
ATG Commerce Service Center (CSC) 29
CSC Schema Migration 29
CSC Data Migration 31
ATG Service and CSC 34
ATG Service/CSC Schema Migration 34
ATG Service/CSC Data Migration 36

5 MSSQL Database Migration 43
Basic Platform 43
Publishing.Base 44
B2BCommerce 45
Data Warehouse 45
ATG Search 46
Agent 47
ATG Service 47
ATG Service Schema Migration 48
ATG Service Data Migration 49
ATG Commerce Service Center (CSC) 54
CSC Schema Migration 54
CSC Data Migration 56
ATG Service and CSC 59
ATG Service/CSC Schema Migration 59
ATG Service/CSC Data Migration 61

6 DB2 Database Migration 67
Basic Platform 67
Publishing.Base 68
B2BCommerce 68
Data Warehouse 69
ATG Search 69
Agent 71
ATG Service 71
ATG Service Schema Migration 72
ATG Service Data Migration 73
ATG Commerce Service Center (CSC) 78
CSC Schema Migration 78
CSC Data Migration 80
ATG Service and CSC 83
ATG Service/CSC Schema Migration 83
ATG Service/CSC Data Migration 85

7 Migration Notes 91
Platform Notes 91

Updating View Mappings 91
Migrating Content Targeters and Content Groups 92
ATG Commerce Reference Store (CRS) Notes 92
ATG Commerce Notes 92
Standard Catalogs / CA 93
Custom Catalogs / CA 93
Standard Catalogs / No CA 93
Custom Catalogs/ No CA 94
Deprecating the Standard Catalog 94
Standard Catalog Migration Steps 94
Updating Catalogs 95
Performing Basic Maintenance on Catalogs 96
Assigning Catalogs and Price Lists to Profiles 96
Casting beginUsable and endUsable properties as java.util.Date 96
Promotion Property Changes 96
ATG Search Notes 96
ATG Customer Intelligence Notes 98
Schema Migration Prerequisites for Customer Intelligence 98
Post-Migration Tasks for Customer Intelligence 99
ATG Outreach Notes 99
ATG Service Notes 100
Schema Migration Prerequisites for ATG Service 100
Post-Migration Tasks for ATG Service 101
ATG Commerce Service Center (CSC) Notes 103
Schema Migration Prerequisites for CSC 103
Post Migration Tasks for CSC 104
ATG Service/Commerce Service Center Notes 105
Schema Migration Prerequisites for ATG Service/CSC 105
Post Migration Tasks for ATG Service/CSC 106
ATG Campaign Optimizer Notes 108

1 Introduction

This guide describes how to upgrade from ATG 9.1 to ATG 10.0, and is written for programmers, DBAs, site administrators, and ATG partners.

Before you begin migration:

1. Review Chapter 2, New in ATG 10.0, for information on features that are new in ATG 10.0.
2. Check the Supported Environments information on the My Oracle Support Web site. Make sure you are running on a supported version of your application server, JDK, database server, and JDBC driver.
3. Review the migration steps described in Chapter 3.

Convention for File Locations

This guide uses the convention `<ATG10dir>` to represent the root directory for ATG products. By default, the Windows installation creates the root directory `C:\ATG\ATG10.0`, but its actual location can vary according to your installation.

ATG Documentation

For detailed information about ATG products, see the ATG documentation on the Oracle Technology Network Web site.

2 New in ATG 10.0

This section describes new features available in the ATG platform and applications:

- Platform
- ATG Commerce
- ATG Commerce Reference Store
- ATG Search
- ATG Merchandising
- ATG Customer Intelligence
- ATG Campaign Optimizer
- ATG Commerce Service Center (CSC)
- ATG Outreach
- ATG Service

Platform

This section describes new platform features.

Site Administration

The ATG Business Control Center provides access to the Site Administration interface, where business users can define and manage a multisite environment at a high level. Site Administration lets you perform the following tasks:

1. Create and manage registered sites and site groups.
2. Specify data to be shared within a site group.
3. Create and manage site categories, which provide data and configurable properties for sites.

For full information, see the *ATG Multisite Administration Guide*.

ATG Content Administration

Selective Cache Invalidation

You can optimize a site's performance after a switch deployment by enabling selective cache invalidation on target repositories. A repository that is thus configured invalidates its item caches selectively during deployment, rather than invalidating the contents of those caches entirely.

The SQL repository property `cacheSwitchHot`, set by default to false, is now deprecated in favor of selective cache invalidation. For more information, see the *ATG Content Administration Programming Guide*.

GSARepository

allowNullValues

`skipNullProperties` property is now deprecated. It is replaced by `allowNullValues`, which enables multi-valued properties to accept null values. You can set `allowNullValues` a property of the entire repository, or as an attribute for individual item properties.

Rest Web Services

The ATG Platform REST Web Services include the following enhancements:

- Setting a property value of type Collection or Map can now be handled as either a replace or an append.
- Support for standard JSON input.
- A number of form handler improvements.
- Property filtering enhancements, including the ability to specify a property filter on a per-request basis.
- New control parameters.

For more information, see the *ATG Web Services and Integration Framework Guide*.

Removed WAR Files

The following WAR files were removed from the WebUI module, as they contained out-of-date JavaScript and Dojo libraries:

```
dojo.war
dojo-04.war
turbogrid3.war
```

ATG Commerce

The following features are new to ATG Commerce:

Multisite Support

ATG Commerce provides full multisite support:

- All ATG Commerce repository items—including categories, products, and SKUs—can be associated with specific sites. Site association supports merchandising strategies that let you filter and promote items based on sites.
- Orders, pricelists, coupons, and search form handlers support site-awareness.
- Shopping carts can be shared across sites or restricted to a given site or group of sites.
- ATG Commerce reports let you track sales and other metrics by site.

ATG B2BCommerce is also updated to support multiple sites.

For full information about these and other ATG Commerce features, see the *ATG Commerce Programming Guide*.

Promotions Enhancements

- New UI for promotion and coupon management, providing a new navigation model and split screen and drag and drop features.
- Easy-to-use templates simplify promotion creation and management.
- Enhanced support for custom promotions, including new form-based replacement for the Discount Rule Editor screens.
- Tiered discount promotions let you define multiple levels of discounts within a single promotion.
- In a multisite environment, assign promotions to one or more registered sites.

Standard Catalog Deprecation

Previous versions of ATG Commerce included standard and custom catalogs. With this release, Commerce simplifies catalog planning and creation by providing custom catalog functionality to all users. This simplifies implementation and future expansion, and eliminates the need to run extra modules to obtain desired catalog functionality.

Existing catalogs can remain unchanged; however, the migration steps described in ATG Commerce Notes are required to ensure compatibility with current and future releases.

ATG Commerce Reference Store

The following features are new to ATG Commerce Reference Store (CRS). For full information, see the ATG Commerce Reference Store Overview.

- CRS has been recoded and reconfigured to function as a multisite implementation with three sites, ATG Store US, ATG Store Germany, and ATG Home. As a result, a number of CRS features are now site-aware.
- A variety of site-aware targeters, scenarios, and promotions have been added to CRS.
- New promotions have been added that include a coupon that supports multiple promotions and a tiered promotion.
- Multisite search functionality has been added that allows you to search the current site or multiple sites.
- CRS checkout and shipping flows have been improved.
- Text-based content for the storefronts is now contained in a repository. This allows the application to take advantage of internationalization features and it also allows merchandisers to edit the content in the ATG Business Control Center.
- Two new SKU sub-types have been added, `clothing-sku` and `furniture-sku`. These item types add properties that are appropriate for clothing and furniture items, respectively.
- A wood finish picker for products, such as furniture, that are made of wood has been added.

Note: There is no migration path from older versions of CRS to CRS 10.0. To use the 10.0 version, install and configure a new instance of CRS using the instructions in the *ATG Commerce Reference Store Installation and Configuration Guide*.

ATG Search

The following features are new to ATG Search. For full information, see the *ATG Search Administration Guide* and the *ATG Search Query Guide*.

Multisite Support

Along with other ATG applications, ATG Search supports the new multisite feature. Multisite uses for ATG Search are configured using the new Site Administration utility (see XREF).

New user interface elements allow you to target user searches to specific subsets of your content whenever you set up a new site, or to search across your entire index from any site.

Search reports now include site data, allowing you to track user activity at finer granularity.

Content Labels and Target Types

ATG Search uses a new mechanism, based on content labels and target types, for identifying the content to be searched for a given project. Independent projects have separate groupings of content for indexing. A project's content sets are grouped and identified for querying by one or more content labels. A project's target type specifies the instance of the index—for example, Production or Staging—where queries are directed.

Parallel Indexing and Index Queuing

Changes to the indexing process allow you to update multiple indexes at one time (resources permitting, and only if those indexes do not share content), and to queue indexing tasks to be performed later.

Metadata Enhancements

You can now add index-only metadata tags to unstructured content.

New Search Feedback Option

ATG Search now supports a feedback option that provides information on alternate searches using fewer words, if the original query provides insufficient results.

Diagnostics Capture

This feature, available in ATG Dynamo Server Admin, can help troubleshoot problems by capturing all relevant files required by ATG Support.

New Documentation

The ATG Search product documentation has been reorganized:

- The indexing information formerly in the *ATG Commerce Search Guide* has been moved to the *ATG Search Administration Guide*.
- A new manual, the *ATG Search Query Guide*, has been created from the querying information (including the faceted search discussions) formerly in the *ATG Commerce Search Guide* and the *ATG Search Query Reference Guide*.

The ATG Search 10.0 documentation consists of these three manuals:

ATG Search Installation and Configuration Guide
Describes how to install, configure, and run ATG Search.

ATG Search Administration Guide
Describes how to use the ATG Search Administration console to prepare and index content, make searchable content available to end users, and maintain ATG Search.

ATG Search Query Guide
Describes how to construct and issue search queries in JavaServer Pages and present the results that are returned.

ATG Merchandising

The following features are new to ATG Merchandising. For full information, see the *ATG Merchandising Administration Guide* and *ATG Merchandising Guide for Business Users*.

New UI for asset management

Provides a new navigation model and split screen and drag and drop features.

Multisite awareness for personalization assets

User segments are aware of registered sites for rule execution. Content groups and targeters are aware of registered sites for rule execution and content collection. Scenario events are aware of registered sites.

Multisite support in Search Merchandising

All aspects of Search Merchandising, including faceted search, search configuration, and search testing, support multiple registered sites. In a multisite environment you can:

- Create global facets that are available to all registered sites.
- Define and organize search configurations by site, as well as by the existing dimensions, segment and language.
- Use site as a prioritization property.

ATG Customer Intelligence

ATG Customer Intelligence provides multisite support by storing information about the sites configured in ATG Commerce. It correlates reporting information about business processes with this new dimension. Many existing ATG Customer Intelligence reports now include controls to break down information by site.

For full information, see the *ATG Customer Intelligence Installation and Configuration Guide*.

ATG Campaign Optimizer

In a multisite environment, you can preview and run tests on all registered sites or only on the site you specify.

For full information, see the *ATG Campaign Optimizer User Guide*.

ATG Commerce Service Center (CSC)

The following features are new to ATG Commerce Service Center. For full information, see the *ATG Service Installation and Configuration Guide*.

Multisite Support

ATG Commerce Service Center recognizes sites and site group configurations. Products are presented in context of current site. A site picker allows an agent to pick a site quickly.

Gift Registry

A new gift registry feature lets agents set up ATG Commerce gift registries on behalf of customers, and lets customers create a wish list.

Order Approvals

Agents can submit an approval for an order. A new approval process checks agent appeasement levels to verify approval or to escalate for approval.

Promotions Browser

Agents can view and grant existing promotions for the current order and/or customer.

Reporting

Seven new reports allow customer call center managers to track metrics.

Returns and Exchanges

Return and exchange calculations and methodology are updated and improved.

Debug Mode

Programmers can run ATG Commerce Service Center in debugging mode.

ATG Outreach

ATG Outreach now provides support for multisite campaigns. Campaigns can be applied to all the web sites managed by your ATG system or to one site only. Outreach reports have been updated to show the name of the site that generated the campaign data. Campaign actions and events are also site aware.

For full information, see the *ATG Outreach Guide for Business Users* and the *ATG Outreach Administration and Development Guide*.

ATG Service

The following features are new to ATG Service. For detailed information, see the *ATG Service Installation and Configuration Guide*.

Multisite Support

ATG Service recognizes sites and site group configurations; it uses a Self Service site type to enable ATG Self Service-specific site parameters. Existing ATG Self Service sites that are based on external user segments are now registered sites. Sites can be personalized using segments within sites.

Advanced Search Constraints

Users can select specific content from unstructured content using content sets created in ATG Search.

Owning Group Enhancements

Agents can select default owning group, or select owning groups for solutions.

New Rich Text Editor

The new editor provides additional capabilities when creating or editing a solution.

Solution Linking Validation

Links within solutions can be validated. The validation process generates a list of solutions that contain broken links.

Root Topic Selection

Agents can assign more than one root topic to an ATG Self Service site. Root topics can now also be subtopics of other topics.

3 Migration Steps

Migrating from ATG 9.1 to ATG 10.0 can be divided into the following stages:

- General pre-migration tasks
- ATG installation
- Database schema migration
- Application-specific migration tasks
- General post-migration tasks
- Migration testing

General Pre-Migration Tasks

Note: In addition to the tasks noted here, a number of ATG applications have their own pre-migration requirements. Before starting the migration, check the instructions for each installed application, under Migration Notes.

Before starting migration, complete the following tasks:

1. Back up your ATG 9.1 database.
2. Back up the `Publishing` and `PublishingAgent` directories—-for example, these CIM-generated directories under `<ATG9dir>/home/servers/`:
   - `atg_publishing_lockserver/Publishing`
   - `atg_production_lockserver/PublishingAgent`
3. If your installation includes ATG Content Administration, complete all projects.

   Note: Solution workflow projects can remain open during the migration process.
4. Shut down all ATG servers.

ATG Installation

1. Install ATG 10.0 as directed in the *ATG Installation and Configuration Guide.*
2. Copy your application modules to the ATG 10 installation. Update application code as necessary to conform to new ATG 10 functionality (see New in ATG 10.0).

Database Schema Migration

ATG provides several migration kits for upgrading the schemas of supported databases: Oracle, MSSQL, and DB2. Migration kits are available for download from My Oracle Support.

The following migration kits are available for ATG products:

- ATG platform
- ATG Search Administration
- ATG Service/ATG Commerce Service Center
- ATG Outreach
- ATG Campaign Optimizer

Each migration kit contains three database-specific directories:

```
/91to100sql/db_components/oracle
/91to100sql/db_components/db2
/91to100sql/db_components/mssql
```

Run the scripts in these directories as directed in the following chapters:

- Oracle Database Migration
- DB2 Database Migration
- MSSQL Database Migration

Batch File/Shell Script Naming Conventions

In general, each module provides batch files/shell scripts that use the following naming conventions:

Naming convention	Updates...
`*_core.{bat|sh}`	Production core schema
`*_switching.{bat|sh}`	Production switching schema
`*_publishing.{bat|sh}`	Schema for ATG Content Administration tables and versioned tables

Note: If an environment uses online (non-switching) deployment, run both `*_core` and `*_switching` scripts against the production schema.

Batch Files versus SQL Scripts

In general, the batch files/shell scripts run all required DDL scripts in the correct order. Instead of running batch files/shell scripts, you can upgrade each module by individually executing the applicable SQL scripts. Be sure to run these in the same order as they appear in the batch file or shell script.

Before Running Migration Scripts

Review the migration scripts and modify where necessary. Make sure the scripts do not overwrite custom database changes. In order to run these scripts, your database login must be the same as the one used to create the original ATG 9.1 schemas.

Before you run the migration batch files or shell scripts, include a dot (`.`) in the PATH environment variable in order to enable execution of the DDL script commands.

Running Batch Files

Use the syntax shown in the following sections to run batch files or shell scripts.

Oracle

filename.{bat|sh} *user-acct password tns-alias*

- *user-acct*: Name of the schema user account
- *password*: Password to the user account
- *tns-alias*: TNS name for the database

MSSQL

filename.{bat|sh} *user-acct password hostname db-schema*

- *user-acct*: Name of the schema user account
- *password*: Password to the user account
- *hostname*: Host name of the database server
- *db-schema*: Name of the database schema

DB2

filename.{bat|sh} *user-acct password DB2-alias*

- *user-acct*: Name of the schema user account
- *password*: Password to the user account
- *DB2-alias*: Alias for the DB2 database

Running DDL Scripts

Use the syntax shown in the following sections to run DDL scripts.

Oracle

```
sqlplus –S user-acct/password@tns-alias < ddl-pathname > logfile
```

MSSQL

```
sqlcmd -U user-acct -P password -S hostname -d db-schema
  -i ddl-pathname -o logfile
```

DB2

```
db2 –tvf ddl-pathname > logfile
```

Application-Specific Migration Tasks

After completing all tasks described in previous sections, complete migration for specific ATG applications, as documented under Migration Notes.

General Post-Migration Tasks

Note: In addition to the tasks noted here, a number of ATG applications have their own post-migration requirements. Check the instructions for each installed application, under Migration Notes.

After migration is complete, perform the following tasks:

1. If using ATG Content Administration, it is generally good practice to launch a full deployment to all workflow targets. This ensures that content on the asset management server and workflow targets is fully synchronized.

   Note: This recommendation might not apply to ATG Service installations.
2. Clear browser caches.
3. Clear the application server cache.

   For example, on JBoss, empty the contents of each server directory:

   `/work/jboss.web/localhost`
4. If your application server is JBoss version 5.0, regenerate all SOAP Web Services. See information about automatic generation of Web Services in the *ATG Web Services and Integration Framework Guide*.
5. Test the upgraded platform and applications (see Migration Testing).

Migration Testing

In order to verify the success of your migration on asset management and production servers, follow these steps:

1. Assemble EAR files for the asset management and production servers.
2. Deploy the EAR files to the application server and start the production and management servers.
3. From the ATG 10.0 BCC, verify that you can view:
   - All internal users created in ATG 9.1
   - All assets created in ATG 9.1
4. Confirm that the ATG 10.0 production server contains all assets that were accessible on the ATG 9.1 platform.
5. Create projects, where you add new assets and edit existing ones. Deploy these assets and verify that the deployment is successful.
6. Confirm that deployed file assets are in the correct location and behave as expected.

4 Oracle Database Migration

ATG 10.0 includes a number of enhancements to the database schemas used by ATG 9.1. The Oracle migration kit provides the batch files/shell scripts and DDL scripts that you need to upgrade existing Oracle schemas, in the following migration kit directory:

```
../91to100sql/db_components/oracle
```

Each section in this chapter describes the batch files/shell scripts and DDL scripts that are required to migrate a specific ATG module or group of related modules:

- Basic platform (includes DAS, DPS, and DSS)
- Publishing.Base
- Portals
- B2CCommerce
- B2BCommerce
- Data Warehouse
- ATG Search
- Agent
- ATG Service
- ATG Commerce Service Center (CSC)
- ATG Service and CSC

Each section is subdivided according to the schemas that require upgrading: core, switching, and publishing.

Note: Before running migration scripts, carefully review the Database Schema Migration section in the Migration Steps chapter.

Basic Platform

The ATG platform minimally comprises three modules whose schemas require migration: DAS, DPS, and DSS. The sections that follow document schema migration scripts for these modules.

Basic Platform Core

Batch file/shell scripts	`run_das_core.{bat|sh}`
	`run_dps_core.{bat|sh}`
	`run_dss_core.{bat|sh}`

DDL scripts	`upgrade_das_core_ddl.sql`
	`upgrade_dps_ddl.sql`
	`upgrade_dss_ddl.sql`

Basic Platform Switching

Batch file/shell script	`run_das_switching.{bat|sh}`
DDL scripts	`upgrade_das_seo_ddl.sql` `site_ddl.sql`

Basic Platform Publishing

Note: Versioned database tables must now set their `checkin_date` column to `TIMESTAMP`. If necessary, be sure to manually update your application's versioned database tables accordingly.

Batch file/shell script	`run_das_publishing.{bat|sh}`
	`run_dps_internal_users_publishing.{bat|sh}`
	`run_dps_publishing.{bat|sh}`
	`run_dss_internal_users_publishing.{bat|sh}`

DDL scripts	`versioned_site_ddl.sql`
	`upgrade_das_versioned_ddl.sql`
	`upgrade_dps_internal_users_ddl.sql`
	`upgrade_dps_versioned_ddl.sql`
	`upgrade_dss_internal_users_ddl.sql`

Publishing.Base

Prerequisite migration	Basic platform publishing
Batch file/shell script	`run_publishing_base.{bat|sh}`

DDL scripts	`upgrade_paf_ddl.sql`
	`upgrade_publishing_ddl.sql`
	`upgrade_viewmapping_ddl.sql`

Portals

Prerequisite migration	Basic platform core
Batch file/shell script	`run_portals_core.{bat|sh}`

DDL scripts	`upgrade_calendar_ddl.sql`
	`upgrade_bookmarks_ddl.sql`
	`upgrade_discussion_ddl.sql`
	`upgrade_docexch_ddl.sql`
	`upgrade_paf_ddl.sql`
	`upgrade_profile_ddl.sql`
	`upgrade_ppa_ddl.sql`

B2CCommerce

B2CCommerce Core

Prerequisite migration	Basic platform core
Batch file/shell script	`run_dcs_core.{bat|sh}`

DDL scripts	`upgrade_dcs_core_ddl.sql`
	`upgrade_abandoned_order_ddl.sql*`
	`custom_catalog_user_ddl.sql**`
	`abandoned_order_views.sql***`
	`reporting_views.sql***`
	`custom_catalog_reporting.sql***`
	`reporting_views2.sql***`
	`custom_catalog_reporting1.sql***`

* Required only if using abandoned order services
** Required if 9.1 installation uses standard catalogs
*** Use in order to recompile invalid views

B2CCommerce Switching

Prerequisite migration	Basic platform switching
Batch file/shell script	`run_dcs_switching.{bat|sh}`

DDL scripts	`commerce_site_ddl.sql`
	`upgrade_dcs_catalog_ddl.sql`
	`upgrade_custom_catalog_ddl.sql*`
	`custom_catalog_ddl.sql**`

* Required if 9.1 installation uses custom catalogs; omit if using standard catalogs
** Required if 9.1 installation uses standard catalogs

B2CCommerce Publishing

Note: Run B2CCommerce publishing migration scripts only if you use ATG Content Administration.

Prerequisite migration	Basic platform publishing
Batch file/shell scripts	`run_dcs_publishing.{bat|sh}`

DDL scripts	`versioned_commerce_site_ddl.sql`
	`upgrade_versioned_custom_catalog_ddl.sql*`
	`custom_catalog_user_ddl.sql**`
	`versioned_custom_catalog_ddl.sql**`
	`upgrade_dcs_versioned_ddl.sql`
	`upgrade_abandoned_order_ddl.sql***`

* Required if using versioned custom catalogs; omit if using standard catalogs
** Required if using versioned standard catalogs
*** Required if using abandoned_order services

B2BCommerce

B2BCommerce Core

Prerequisite migration	B2CCommerce core
Batch file/shell script	`run_b2bcommerce_core.{bat|sh}`
DDL scripts	`upgrade_b2bcommerce_ddl.sql`

B2BCommerce Switching

See B2CCommerce switching.

B2BCommerce Publishing

Note: Run B2CCommerce publishing migration scripts only if you use ATG Content Administration.

Prerequisite migration	B2CCommerce publishing
Batch file/shell script	`run_b2bcommerce_publishing.{bat|sh}`
DDL scripts	`upgrade_b2bcommerce_versioned_ddl.sql`

Data Warehouse

Note: Before you migrate the database schemas for ATG Customer Intelligence data warehouse, review the procedures in ATG Customer Intelligence Notes.

Base Data Warehouse

Batch file/shell script	`run_arf_dw_base.{bat|sh}`
DDL scripts	`upgrade_arf_dw_base_ddl.sql` `upgrade_arf_dw_base_init.sql`

ATG Commerce Reporting

Prerequisite migration	Base Data Warehouse
Batch file/shell script	`run_dcs_dw.{bat|sh}`
DDL scripts	`upgrade_dcs_dw_ddl.sql`

Search Admin Reporting

Prerequisite migration	Base Data Warehouse
Batch file/shell script	`run_arf_dw.{bat|sh}`
DDL scripts	`upgrade_arf_dw_search_ddl.sql`

ATG Service Reporting

Prerequisite migration	Base Data Warehouse Search Admin reporting
Batch file/shell script	`run_service_datawarehouse.{bat|sh}`
DDL scripts	`upgrade_service_datawarehouse_ddl.sql`

ATG Commerce Service Center Reporting

Prerequisite migration	Base Data Warehouse Search Admin reporting
Batch file/shell script	`run_dcs_dw.{bat|sh}`
DDL scripts	`upgrade_dcs_dw_ddl.sql`

ATG Search

ATG Search comprises two modules:

- DAF.Search
- SearchAdmin

After database schema migration is complete, review the procedures in ATG Search Notes.

Note: The batch files/shell scripts and DDL scripts for DAF.Search migration are in the Platform migration kit; Search Admin has its own migration kit.

ATG Search Core

Prerequisite migration	Basic platform core

Batch file/shell scripts	**Platform migration kit:**
	`run_daf_search_core.{bat|sh}`
	SearchAdmin migration kit:
	`run_search_admin.{bat|sh}`

DDL scripts	**Platform migration kit:**
	`upgrade_daf_search_routing_ddl.sql`
	`drop_search_ddl.sql*`
	`search_ddl.sql*`
	SearchAdmin migration kit:
	`upgrade_search_admin_common_ddl.sql`
	`upgrade_search_admin_customizations_ddl.sql`
	`upgrade_search_admin_data.sql`

* Drop and recreate search index tables.

ATG Search Switching

Prerequisite migration	Basic platform switching
Batch file/shell scripts	`run_daf_search_switching.{bat|sh}`
DDL scripts	`search_site_ddl.sql` `upgrade_daf_search_common_ddl.sql`

Search Publishing

Prerequisite migration	Basic platform publishing
Batch file/shell script	`run_daf_search_publishing.{bat|sh}`

DDL scripts	`versioned_search_site_ddl.sql`
	`upgrade_daf_search_common_ddl.sql`
	`upgrade_daf_search_versioned_ddl.sql`

Agent

Note: Production agent schema migration scripts are required in order to migrate the following ATG applications:

- ATG Service
- ATG Commerce Service Center (CSC)

For more information, see the applicable migration guidelines.

Agent Core

Prerequisite migration	Basic platform core
Batch file/shell script	`run_agent_production_core.{bat|sh}`
DDL scripts	`upgrade_agent_production_ddl.sql`

Agent Server

Prerequisite migration	Basic platform publishing Publishing.base
Batch file/shell script	`run_agent_agent.{bat|sh}`
DDL scripts	`upgrade_agent_agent_ddl.sql` `upgrade_agent_production_ddl.sql`

ATG Service

This section covers the following topics:

- ATG Service schema migration
- ATG Service data migration

Before starting database migration, review Schema Migration Prerequisites for ATG Service.

ATG Service Schema Migration

Service Core

Prerequisite migration	Basic platform core ATG Search Agent core

Batch file/shell script	**Platform migration kit:**
	`run_ticketing_core.{bat|sh}`
	Service10.0 migration kit:
	`run_service_production.{bat|sh}`

DDL scripts	**Platform migration kit:**
	`upgrade_ticket_ddl.sql`
	Service10.0 migration kit:
	`viewmapping_ddl.sql`
	`upgrade_unversioned_option_ddl.sql`
	`upgrade_unversioned_svc_solution_ddl.sql`
	`upgrade_service_production_ddl.sql`
	`upgrade_unversioned_offer_mgmt_ddl.sql`
	`upgrade_service_framework_production_ddl.sql`
	`upgrade_search_admin_production_ddl.sql`

Service Switching

Prerequisite migration	Basic platform switching ATG Search switching
Batch file/shell script	`run_service_switch.{bat|sh}*`

DDL scripts	`upgrade_unversioned_option_ddl.sql`
	`upgrade_unversioned_svc_solution_ddl.sql`
	`upgrade_unversioned_offer_mgmt_ddl.sql`

* Omit for an environment that uses online (non-switching) deployment.

Service Agent

Prerequisite migration	Basic platform core Basic platform publishing Publishing.base Agent server
Batch file/shell script	`run_service_agent.{bat|sh}`

DDL scripts	`upgrade_versioned_svc_solution_ddl.sql`
	`upgrade_versioned_offer_mgmt_ddl.sql`
	`upgrade_svc_publishing_ddl.sql`
	`upgrade_versioned_option_ddl.sql`

Service Data Warehouse

Prerequisite migration	Base data warehouse
Batch file/shell script	`run_service_datawarehouse.{bat|sh}`

DDL scripts	`upgrade_service_datawarehouse_ddl.sql`

Service Publishing

Prerequisite migration	Basic platform core Basic platform publishing Publishing.base Search publishing
Batch file/shell script	n/a

ATG Service Data Migration

This section explains how to migrate the data in the databases for ATG Knowledge and ATG Self Service from version 9.1 to version 10.0.

To migrate version 9.1 data to a version 10.0 database:

1. Make sure you have completed all pre-migration tasks. See General Pre-Migration Tasks and Schema Migration Prerequisites for ATG Service. Make sure you have migrated the database schema for ATG Service. See ATG Service Schema Migration.
2. Run the Service.DBMigration.91to100 Module.
3. Drop unused tables in your database schemas.
4. Import agent server data.
5. Migrate publishing data.

Run the Service.DBMigration.91to100 Module

1. Make sure you have a `PlatformLicenses.properties` file that is multisite enabled. Install this license file in your ATG 10.0 installation directory.
2. From the asset management server's `atg-ds.xml` file, copy this `<xa-datasource>` setting:

   `<jndi-name>ATGDataWarehouseDS</jndi-name>`

   to the agent server's `atg-ds.xml` file.

   Note: The `atg-ds.xml` file is in the application server's deployment directory. For example, on a JBoss application server, `atg-ds.xml` file is in the following directory:

   `jboss-root``/jboss-as/server/atg_publishing_lockserver/deploy/`
3. If you use switch deployment, configure the switching data sources for the `Service.DBMigration` module. Edit the following files in:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/`

   - `JTDataSource_switchA.properties`:

     ```
     $class=atg.nucleus.JNDIReference
     JNDIName=java:/ATGSwitchingDS_A
     ```

   - `JTDataSource_switchB.properties`:

```
$class=atg.nucleus.JNDIReference
JNDIName=java:/ATGSwitchingDS_B
```

Note: Replace *ATGSwitchingDS_A* and *ATGSwitchingDS_B* with the JNDI names for the switching data sources as they are configured in the `atg-ds.xml` file for your application server.

4. In a command-line terminal, change to this directory:

   `<ATG10dir>/home/bin`

5. Assemble the `Service.DBMigration.91to100` module with the `<ATG10dir>/home/bin/runAssembler` script as follows:

   ```
   runAssembler –server atg_svcagent_lockserver –
   standalone atg_svcagent_lockserver.ear -
   m Service.DBMigration.91to100
   ```

6. Start the module.
7. Run CIM and deploy the agent EAR again.

Drop Unused Tables

Note: If you intend to run a full deployment after migration, skip steps 1-6.

1. Grant permission to create a private synonym on the agent data source schema to the production core schema user. The following command grants permission to create a private synonym to this user:

   `GRANT CREATE SYNONYM TO production-schema-user`

2. Grant SELECT permission on the `svc_soln` and `avm_devline` tables in the agent data source schema to the user of the production core schema.

   The following commands grant SELECT permission to the production core schema user. Replace *production-schenma-user* with the name of the production core user.

   ```
   GRANT SELECT ON svc_soln TO production-schema-user;
   GRANT SELECT ON avm_devline TO production-schema-user;
   ```

3. If you use switch deployment, repeat step 2 for each switching data source schema user.
4. In the Service migration kit, edit the following script:

   `../oracle/complete_solution_data_migration_ddl.sql`

   - Replace <svcagent_schema> with the name of the agent data source schema user.
   - Replace <production_schema> with the name of the production core schema user. If you use switch deployment, replace <production_schema> with the name of one schema user.

5. Run `complete_solution_data_migration_ddl.sql`
6. If you use switch deployment, repeat steps 4 and 5 for each production schema user.
7. If you use switch deployment, run this migration script on each switching schema; otherwise, run it on the production schema:

   `run_service_production_complete_solution_migration.{bat|sh}`

8. Run the following script on the production core schema:

   `run_post_drop_production.{bat|sh}`

9. Run the following script on the production switching schema:

   `run_post_drop_switch.{bat|sh}`

10. Run the following script for your agent data source schema. If you installed ATG Knowledge and ATG Self Service with the Configuration and Installation Manager (CIM) utility, comment out the call to the script `drop_useroption_ddl.sql`.

    `run_post_drop_agent.{bat|sh}`

11. Edit the script `run_arf_dw.{bat|sh}` from the platform migration kit. Uncomment the call to the `drop_arf_search_site.sql` script. Comment out all other SQL script calls in the file.

12. Run the following script for your data warehouse schema:

    `run_arf_dw.{bat|sh}`

Import Agent Server Data

To import Agent server data, follow these steps:

1. Start the Configuration and Installation Manager (CIM) utility that you used to install ATG 10.0. Deploy `atg_svcagent_lockserver.ear`.

2. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/`

3. Configure the following properties files with connection information as noted:

   - `FakeXADataSource_agent.properties`

     Set connection information for the agent schema.

   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

   For information about configuring `FakeXADataSource` files, see the *ATG Service Installation and Configuration Guide*.

4. Update the view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   ```

5. Start the agent server. After startup, let the server run for five minutes, then shut it down.

6. Copy:

   `<ATG10dir>/home/servers/atg_svcagent_lockserver`

   to:

   `<ATG10dir>/home/servers/svcagent_import91to100`

7. Change to this directory:

   `<ATG10dir>/home/servers/svcagent_import91to100/localconfig`

8. In this directory, configure the agent server as indicated in the following table.

   Note: Create directories and properties files as needed.

Path	Configuration
/atg/dynamo/service/jdbc	FakeXADataSource_agent.properties: Points to the agent schema FakeXADataSource_production.properties: Points to the production core schema. FakeXADataSource.properties: Points to the agent schema
/atg/dynamo/service/jdbc	If using switch deployment: SwitchingDataSourceA.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchA SwitchingDataSourceB.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchB FakeXADataSource_switchA and FakeXADataSource_switchB point to the switching schemas.
/atg/reporting/datawarehouse	FakeXADataSource.properties: Points to the data warehouse schema
/atg/reporting/datawarehouse/loaders	FakeXADataSource.properties: Points to the production core schema
/atg/search/routing	RoutingSearchService.properties: cacheService=
/atg/epub/deployment	DeploymentFulfiller.properties: enable=false

9. Delete the following directories:

   <ATG10dir>/home/servers/svcagent_import91to100/Publishing
   <ATG10dir>/home/servers/svcagent_import91to100/PublishingAgent

10. Run the following script in the Service migration kit:

    Service10.0/migration/91to100sql/scripts/run-agent-import.{bat|sh}

    Pass the name of the server directory you created earlier in this procedure as a parameter when you invoke the script. For example:

    run-agent-import.sh svcagent_import91to100

11. Repeat the previous step twice. On the third iteration, the script should execute error-free.

Migrate Publishing Data

To migrate publishing data:

1. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/`

2. Configure the following properties files with connection information as noted:

   - `FakeXADataSource.properties`

     Set connection information for the publishing schema.

   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

3. Update view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   <ATG10dir>/SiteAdmin/Versioned/install/importSiteAdmin.{bat|sh}
   <ATG10dir>/SiteAdmin/Search/install/importSiteAdminSearch.{bat|sh}
   ```

4. Invoke the startSQLRepository utility as follows:

   UNIX:

   ```
   <ATG10dir>/home/bin/startSQLRepository -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>/Search10.0/SearchAdmin/common/data/initial/search
   admin-profile.xml
   ```

   Windows:

   ```
   <ATG10dir>\home\bin\startSQLRepository.bat -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>\Search10.0\SearchAdmin\common\data\initial\search
   admin-profile.xml
   ```

5. Start the asset management server. Let the server run for several minutes after startup is complete, then shut it down.

6. Copy this directory:

   `<ATG9.1dir>/home/servers/atg_publishing_lockserver`

   to:

   `<ATG10dir>/home/servers/svcpub_import91to100`

7. Create this directory:

   ```
   <ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/dynamo
   /service/jdbc
   ```

8. In the new directory, configure the following properties files, as noted:

   - `FakeXADataSource.properties`

     Set connection information for the publishing schema.

   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

   - `FakeXADataSource_agent.properties`

     Set connection information for the agent data source schema.

9. Change to this directory:

   `<ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/`

10. Edit the following properties files, as noted:
    - `/reporting/datawarehouse/FakeXADataSource.properties`

      Set connection information for the reporting data warehouse schema.
    - `/reporting/datawarehouse/loaders/FakeXADataSource.properties`

      Set connection information for the production core schema.
    - `/search/routing/RoutingSearchService.properties`

      `cacheService=`
    - `/epub/deployment/DeploymentFulfiller.properties`

      `enable=false`
11. Delete this directory:

    `<ATG10dir>/home/servers/svcpub_import91to100/Publishing`
12. Run the following script in the Service10.0 migration kit in order to import the Self Service sites:

    `../migration/91to100sql/scripts/run-management-import.{bat|sh}`

    This script uses the `svc_multisite_91to100.xml` file that was generated earlier by running the `Service.DBMigration.91to100` module.

    For example:

    `run-management-import.sh svcpub_import91to100`

ATG Commerce Service Center (CSC)

This section describes database migration procedures for customers who have ATG Commerce Service Center installed. The following topics are covered:

- CSC schema migration
- CSC data migration

Before starting database migration, review Schema Migration Prerequisites for CSC.

CSC Schema Migration

Note: Unlike CSC 9.1, CSC 10.0 does not require installation of ATG Service. The CSC 10.0 schema migration migrates the ATG Knowledge and ATG Service Framework tables, then drops the ATG Knowledge-only tables as described in CSC Data Migration.

CSC Core

Prerequisite migration	Basic platform core ATG Search Agent core B2CCommerce core

Migration scripts	**Service10.0 migration kit:**
	`run_service_production.{bat|sh}*`
	Platform migration kit:
	`run_compile_abandoned_orders_views_core.{bat|sh}`
	`run_click_to_connect.{bat|sh}**`
	`run_ticketing_core.{bat|sh}`
	CSC10.0 migration kit:
	`run_dcs_csr_production.{bat|sh}`
	ATG10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql***`

* Comment out execution of the DDL script `upgrade_search_admin_production_ddl.sql`.
** Required only if running ATG ClicktoConnect.
*** Required only if CRS Search add-on was not previously installed. Omit if using switch deployment.

CSC Switching

Prerequisite migration	Service switching B2CCommerce switching

Migration scripts	**CSC10.0 migration kit:**
	`run_dcs_csr_switch.{bat|sh}`
	ATG10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

CSC Agent

Prerequisite migration	Basic platform core Basic platform publishing Publishing.base Agent server

Migration scripts	**Service10.0 migration kit**:
	`run_service_agent.{bat|sh}`
	CSC10.0 migration kit:
	`run_dcs_csr_agent.{bat|sh}`

CSC Data Warehouse

Required only if running ATG Reporting.

Prerequisite migration	Base Data Warehouse ATG Commerce reporting Search Admin reporting
Batch file/shell script	n/a

CSC Publishing

Prerequisite migration	Basic platform Publishing.base Search publishing B2CCommerce publishing

Migration scripts	**CSC10.0 migration kit:**
	`run_dcs_csr_management.{bat|sh}`
	ATG 10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

CSC Data Migration

This section explains how to migrate CSC data from version 9.1 to version 10.0. Before you begin migrating CSC data, be sure you:

- Complete all pre-migration tasks. See General Pre-Migration Tasks and Schema Migration Prerequisites for CSC.
- Migrate the database schema for ATG Service.
- Make sure you have a PlatformLicenses.properties file that is multisite enabled. Install this license file in your ATG 10.0 installation directory.

CSC data migration comprises the following tasks:

1. Drop unused tables.
2. Update view mappings.
3. Start the agent server.
4. Configure the agent server.
5. Import agent server data.
6. Import Agent publishing data.

Drop Unused Tables

To drop unused tables from your migrated database, run the following DDL scripts:

- `run_post_drop_production.{bat|sh}` – Run on the production core schema.
- `run_post_drop_switch.{bat|sh}` – Run on the production switching schema.
- `run_post_drop_agent.{bat|sh}` – Run on the agent schema.

Update View Mappings

1. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/`
2. Configure the following properties files with connection information as noted:
   - `FakeXADataSource_agent.properties`

     Set connection information for the agent schema.
   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.
3. Update the view mappings in the ATG Content Administration schema by running the ATG 10.0 import scripts that create view mappings:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetui.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   ```

Start the Agent Server

Start the new 10.0 Agent server, in order to initialize new publishing workflows. After server startup is complete, allow it to run for approximately five minutes, then shut the server down.

Configure the Agent Server

Change to this directory:

`<ATG10dir>/home/servers/svcagent_import91to100/localconfig`

In this directory, configure the agent server as indicated in the following table.

Note: Create directories and properties files as needed.

Path	Configuration
`/atg/dynamo/service/jdbc`	`FakeXADataSource_agent.properties`: Points to the agent schema `FakeXADataSource_production.properties`: Points to the production core schema. `FakeXADataSource.properties`: Points to the agent schema

Path	Configuration
/atg/dynamo/service/jdbc	If using switch deployment: SwitchingDataSourceA.properties: Points to FakeXADataSource_switchA SwitchingDataSourceB.properties: Points to FakeXADataSource_switchB FakeXADataSource_switchA and FakeXADataSource_switchB point to the switching schemas.
/atg/reporting/datawarehouse	FakeXADataSource.properties: Points to the data warehouse schema
/atg/reporting/datawarehouse/loaders	FakeXADataSource.properties: Points to the production core schema
/atg/search/routing	RoutingSearchService.properties: cacheService=
/atg/commerce/catalog	ProductCatalog.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/commerce/pricing/priceLists	PriceLists.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/assetregistry	AssetRegistry.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_agent
/atg/search/repository	RefinementRepository.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production

Import Agent Server Data

From the DYNAMO_HOME/CSC10.0/migration/91to100sql/scripts directory, run the following script:

```
run-csc-only-agent-import.{bat|sh} svcagent_import91to100
```

Import Agent Publishing Data

1. Change to the following directory:

   <ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/

2. Configure two FakeXADataSource properties files with the appropriate connection information:

   - FakeXADataSource.properties

Set connection information for the publishing schema.

- `FakeXADataSource_production.properties`

  Set connection information for the production schema.

3. Update the view mappings in the ATG Content Administration schema by running the following ATG 10.0 import scripts:

```
<ATG10dir>/BIZUI/install/importBizui.{bat|sh}
<ATG10dir>/BCC/install/importBCC.{bat|sh}
<ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
<ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
<ATG10dir>/SiteAdmin/Versioned/install/importSiteAdmin.{bat|sh}
<ATG10dir>/SiteAdmin/Search/install/importSiteAdminSearch.{bat|sh}
<ATG10dir>/DCS-UI/install/importMerchandising.{bat|sh}
<ATG10dir>/DCS-UI/Versioned/install/importDCSUIVersioned.{bat|sh}
```

4. Run the `startSQLRepository` command:

```
<ATG10dir>/home/bin/startSQLRepository.sh -m DPS.InternalUsers
–repository /atg/userprofiling/InternalProfileRepository
-
import <ATG10dir>/Search10.0/SearchAdmin/common/data/initial/search
admin-profile.xml
```

ATG Service and CSC

This section describes database migration procedures for customers who have both ATG Service and ATG Commerce Service Center installed. The following topics are covered:

- ATG Service/CSC schema migration
- ATG Service/CSC data migration

Before starting database migration, review Schema Migration Prerequisites for ATG Service/CSC.

ATG Service/CSC Schema Migration

Service/CSC Core

Prerequisite migration	Basic platform core ATG Search core B2CCommerce core Agent core

Migration scripts	**Platform migration kit:**
	`run_ticketing_core.{bat|sh}`
	`run_compile_abandoned_orders_views_core.{bat|sh}`
	`run_click_to_connect.{bat|sh}`*
	Service10.0migration kit:
	`run_service_production.{bat|sh}`
	CSC10.0migration kit:
	`run_dcs_csr_production.{bat|sh}`
	ATG10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql`**

* Required only if running ATG ClicktoConnect
** Required only if CRS Search add-on was not previously installed. Omit if using switch deployment.

Service/CSC Switching

Prerequisite migration	Basic Platform switching ATG Search switching B2CCommerce switching

Migration script	**Service10.0migration kit:**
	`run_service_switch.{bat|sh}`
	CSC10.0migration kit:
	`run_dcs_csr_switch.{bat|sh}`
	ATG10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql`*

* Required only if CRS Search add-on was not previously installed.

Service/CSC Agent

Prerequisite migration	Platform core Basic Platform Publishing Publishing.Base Agent server

Migration scripts	**Service10.0migration kit:**
	`run_service_agent.{bat|sh}`
	CSC10.0migration kit:
	`run_dcs_csr_agent.{bat|sh}`

Service/CSC Data Warehouse

Prerequisite migration	Base Data Warehouse ATG Commerce reporting Search Admin reporting
Migration scripts	**Service10.0migration kit:** `run_service_datawarehouse.{bat|sh}`

Service/CSC Publishing

Prerequisite migration	Basic platform Publishing.Base Search publishing B2CCommerce publishing

Migration scripts	**CSC 10 migration kit:**
	`run_dcs_csr_management.{bat|sh}`
	ATG 10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

ATG Service/CSC Data Migration

This section explains how to migrate the data in the databases for ATG Service and ATG Commerce Service Center (CSC) from version 9.1 to version 10.0.

To migrate version 9.1 data to a version 10.0 database, perform the following tasks:

1. Run the Service.DBMigration.91to100 module.
2. Drop unused tables.
3. Import agent server data.
4. Import publishing data.

Run the Service.DBMigration.91to100 Module

1. Make sure you have a `PlatformLicenses.properties` file that is multisite enabled. Install this license file in your ATG 10.0 installation directory.
2. From the asset management server's `atg-ds.xml` file, copy this <xa-datasource> setting:

   `<jndi-name>ATGDataWarehouseDS</jndi-name>`

   to the agent server's `atg-ds.xml` file.

   Note: The `atg-ds.xml` file is in the application server's deployment directory. For example, on a JBoss application server, `atg-ds.xml` file is in the following directory:

jboss-root/jboss-as/server/atg_publishing_lockserver/deploy/

3. If you use switch deployment, configure the switching data sources for the Service.DBMigration module in the following files:
   - <ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/JTDataSource_switchA.properties:

     ```
     $class=atg.nucleus.JNDIReference
     JNDIName=java:/ATGSwitchingDS_A
     ```

   - <ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/JTDataSource_switchB.properties:

     ```
     $class=atg.nucleus.JNDIReference
     JNDIName=java:/ATGSwitchingDS_B
     ```

   Note: Replace ATGSwitchingDS_A and ATGSwitchingDS_B with the JNDI names for the switching data sources as they are configured in your application server's atg-ds.xml file.
4. In a command-line terminal, change to this directory:

   <ATG10dir>/home/bin
5. Assemble the Service.DBMigration.91to100 module with the <ATG10dir>/home/bin/runAssembler script as follows:

   ```
   runAssembler –server atg_svcagent_lockserver –
   standalone atg_svcagent_lockserver.ear -
   m Service.DBMigration.91to100
   ```
6. Start the module.

   Note: If you see an Abstract Deployment Context error, you might have two ATG EAR files in the agent server's deployment directory. In this case, delete the extra EAR file.
7. When startup is complete (on display of the Nucleus Running message), stop the application server.

Drop Unused Tables

To drop unused tables from your migrated database:

1. Grant permission to create a private synonym on the agent data source schema to the production core schema user. The following command grants permission to create a private synonym to this user:

   ```
   GRANT CREATE SYNONYM TO production-schema-user
   ```
2. Grant SELECT permission on the svc_soln and avm_devline tables in the agent data source schema to the user of the production core schema.

   The following commands grant SELECT permission to the production core schema user. Replace *production-user* with the name of the production core user.

   ```
   GRANT SELECT ON svc_soln TO production-schema-user;
   GRANT SELECT ON avm_devline TO production-schema-user;
   ```
3. If you use switch deployment, repeat step 2 for each switching data source schema user.
4. In the Service migration kit, edit the following script:

   ../oracle/complete_solution_data_migration_ddl.sql

- Replace <svcagent_schema> with the name of the agent data source schema user.
- Replace <production_schema> with the name of the production core schema user. If you use switch deployment, replace <production_schema> with the name of one schema user.

5. Run `complete_solution_data_migration_ddl.sql`
6. If you use switch deployment, repeat steps 4 and 5 for each production schema user.
7. **Note:** If you intend to run a full deployment after migration, skip this step.

   If you use switch deployment, run this migration script on each switching schema; otherwise, run it on the production schema:

   `run_service_production_complete_solution_migration.{bat|sh}`
8. Run the following script on the production core schema:

   `run_post_drop_production.{bat|sh}`
9. Run the following script on the production switching schema:

   `run_post_drop_switch.{bat|sh}`
10. Run the following script on your agent data source schema.

    `run_post_drop_agent.{bat|sh}`

    If you installed ATG Knowledge and ATG Self Service with the Configuration and Installation Manager (CIM) utility, comment out the call to the script `drop_useroption_ddl.sql`.
11. Edit the script `run_arf_dw.{bat|sh}` from the platform migration kit. Uncomment the call to the `drop_arf_search_site.sql` script. Comment out all other SQL script calls in the file.
12. Run the following script on your data warehouse schema.

    `run_arf_dw.{bat|sh}`

Import Agent Server Data

To import Agent server data, follow these steps:

1. Start the Configuration and Installation Manager (CIM) utility that you used to install ATG 10.0. Deploy `atg_svcagent_lockserver.ear`.
2. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc`
3. Configure two FakeXADataSource properties files with the appropriate connection information:
   - `FakeXADataSource.properties`

     Set connection information for the agent schema.
   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

   For information about configuring `FakeXADataSource` files, see the *ATG Service Installation and Configuration Guide*.
4. Update the view mappings in the publishing schema by running the following scripts:

   `<ATG10dir>/BIZUI/install/importBizui.{bat|sh}`
   `<ATG10dir>/BCC/install/importBCC.{bat|sh}`

```
<ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
<ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
```

5. Start the agent server. After startup is complete, let the server run for five minutes so it can initialize new publishing workflows, then shut it down.
6. Copy:

   `<ATG10dir>/home/servers/atg_svcagent_lockserver`

   to this new directory:

   `<ATG10dir>/home/servers/svcagent_import91to100`
7. Change to this directory:

   `<ATG10dir>/home/servers/svcagent_import91to100/localconfig`
8. In this directory, configure the agent server as indicated in the following table.

   Note: Create directories and properties files as needed.

Path	Configuration
`/atg/dynamo/service/jdbc`	`FakeXADataSource_agent.properties`: Points to the agent schema `FakeXADataSource_production.properties`: Points to the production schema. `FakeXADataSource.properties`: Points to the agent schema

`/atg/dynamo/service/jdbc`	If using switch deployment:
	`SwitchingDataSourceA.properties`:
	`$class=atg.service.jdbc.MonitoredDataSource`
	`dataSource=\`
	`/atg/dynamo/service/jdbc/FakeXADataSource_switchA`
	`SwitchingDataSourceB.properties`:
	`$class=atg.service.jdbc.MonitoredDataSource`
	`dataSource=\`
	`/atg/dynamo/service/jdbc/FakeXADataSource_switchB`
	`FakeXADataSource_switchA` and `FakeXADataSource_switchB` point to the switching schemas.

`/atg/reporting/datawarehouse`	`FakeXADataSource.properties`: Points to the data warehouse schema
`/atg/reporting/datawarehouse/loaders`	`FakeXADataSource.properties`: Points to the production schema
`/atg/search/routing`	`RoutingSearchService.properties`: `cacheService=`
`/atg/epub/deployment`	`DeploymentFulfiller.properties`: `enable=false`

Path	Configuration
/atg/commerce/catalog	ProductCatalog.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/commerce/pricing/priceLists	PriceLists.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/assetregistry	AssetRegistry.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_agent
/atg/search/repository	RefinementRepository.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production

9. From the Service migration kit's /scripts directory, run this script:

   ```
   run-agent-import.{bat|sh} svcagent_import91to100
   ```

10. Repeat the previous step twice. On the third iteration, the script should execute error-free.
11. From the CSC migration kit's /scripts directory, run this script:

    ```
    run-csc-knowledge-agent-import.{bat|sh} svcagent_import91to100
    ```

Import Publishing Data

1. Change to the following directory:

   ```
   <ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/
   ```

   Configure the following properties files with connection information as noted:

   - FakeXADataSource.properties

     Set connection information for the publishing schema.
   - FakeXADataSource.properties

     Set connection information for the publishing schema.
   - FakeXADataSource_production.properties

     Set connection information for the production core schema.

2. Update view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   <ATG10dir>/SiteAdmin/Versioned/install/importSiteAdmin.{bat|sh}
   <ATG10dir>/SiteAdmin/Search/install/importSiteAdminSearch.{bat|sh}
   <ATG10dir>/DCS-UI/install/importMerchandising.{bat|sh}
   <ATG10dir>/DCS-UI/Versioned/install/importDCSUIVersioned.{bat|sh}
   ```

3. Invoke the startSQLRepository utility as follows:

   UNIX:

```
<ATG10dir>/home/bin/startSQLRepository -m DPS.InternalUsers -
repository /atg/userprofiling/InternalProfileRepository -
import <ATG10dir>/Search10.0/SearchAdmin/common/data/initial/search
admin-profile.xml
```

Windows:

```
<ATG10dir>\home\bin\startSQLRepository.bat -m DPS.InternalUsers -
repository /atg/userprofiling/InternalProfileRepository -
import <ATG10dir>\Search10.0\SearchAdmin\common\data\initial\search
admin-profile.xml
```

4. Start the asset management server. Let the server run for several minutes after startup is complete so it can initialize new publishing workflows, then shut it down.
5. Copy this directory:

   `<ATG10dir>/home/servers/atg_publishing_lockserver`

   to:

   `<ATG10dir>/home/servers/svcpub_import91to100`
6. Create this directory:

   `<ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/dynamo/service/jdbc`
7. In the new directory, configure the following properties files, as noted:
   - `FakeXADataSource.properties`

     Set connection information for the publishing schema.
   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.
   - `FakeXADataSource_agent.properties`

     Set connection information for the agent data source schema.
8. Change to this directory:

   `<ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/`
9. Edit the following properties files, as noted:
   - `/reporting/datawarehouse/FakeXADataSource.properties`

     Set connection information for the reporting data warehouse schema.
   - `/reporting/datawarehouse/loaders/FakeXADataSource.properties`

     Set connection information for the production core schema.
   - `/search/routing/RoutingSearchService.properties`

     `cacheService=`
   - `/epub/deployment/DeploymentFulfiller.properties`

     `enable=false`
10. Run the following script in the Service10.0migration kit, to import the ATG Self Service sites:

    `../migration/91to100sql/scripts/run-management-import.{bat|sh}`

    Pass the name of the server directory you created in step 6 as a parameter when you invoke the script. For example:

    `run-management-import.sh svcpub_import91to100`

5 MSSQL Database Migration

ATG 10.0 includes a number of enhancements to the database schemas used by ATG 9.1. The MSSQL migration kit provides the batch files/shell scripts and DDL scripts that you need to upgrade existing MSSQL schemas, in the following migration kit directory:

```
/91to100sql/db_components/mssql
```

Each section in this chapter describes the batch files/shell scripts and DDL scripts that are required to migrate a specific ATG module or group of related modules:

- Basic platform (includes DAS, DPS, and DSS)
- Publishing.Base
- B2CCommerce
- B2BCommerce
- Data Warehouse
- ATG Search
- Agent
- ATG Service
- ATG Commerce Service Center (CSC)
- ATG Service and CSC

Each section is subdivided according to the schemas that require upgrading: core, switching, and publishing.

Note: Before running migration scripts, carefully review the Database Schema Migration section in the Migration Steps chapter.

Basic Platform

The ATG platform minimally comprises three modules whose schemas require migration: DAS, DPS, and DSS. The sections that follow document schema migration scripts for these modules.

- Basic platform core
- Basic platform switching
- Basic platform publishing

Basic Platform Core

Batch file/shell scripts	`run_das_core.{bat|sh}`
	`run_dps_core.{bat|sh}`
	`run_dss_core.{bat|sh}`

DDL scripts	`upgrade_das_core_ddl.sql`
	`upgrade_dps_ddl.sql`
	`upgrade_dss_ddl.sql`

Basic Platform Switching

Batch file/shell script	`run_das_switching.{bat|sh}`
DDL scripts	`upgrade_das_seo_ddl.sql` `site_ddl.sql`

Basic Platform Publishing

Note: Versioned database tables must now set their `checkin_date` column to `TIMESTAMP`. If necessary, be sure to manually update your application's versioned database tables accordingly.

Batch file/shell script	`run_das_publishing.{bat|sh}`
	`run_dps_internal_users_publishing.{bat|sh}`
	`run_dss_internal_users_publishing.{bat|sh}`

DDL scripts	`versioned_site_ddl.sql`
	`upgrade_das_versioned_ddl.sql`
	`upgrade_dps_internal_users_ddl.sql`
	`upgrade_dss_internal_users_ddl.sql`

Publishing.Base

Prerequisite migration	Basic platform publishing
Batch file/shell script	`run_publishing_base.{bat|sh}`
DDL scripts	`upgrade_publishing_ddl.sql` `upgrade_viewmapping_ddl.sql`

B2BCommerce

The MSSQL migration kit has no migration scripts that are specific to B2BCommerce. B2BCommerce customers only need to run the B2CCommerce migration scripts.

Data Warehouse

Note: Before you migrate the database schemas for ATG Customer Intelligence data warehouse, review the procedures in ATG Customer Intelligence Notes.

Base Data Warehouse

Batch file/shell script	`run_arf_dw_base.{bat|sh}`
DDL scripts	`upgrade_arf_dw_base_ddl.sql`

ATG Commerce Reporting

Prerequisite migration	Basic data warehouse
Batch file/shell script	`run_dcs_dw.{bat|sh}`
DDL scripts	`upgrade_dcs_dw_ddl.sql`

Search Admin Reporting

Prerequisite migration	Basic data warehouse
Batch file/shell script	`run_arf_dw.{bat|sh}`
DDL scripts	`upgrade_arf_dw_search_ddl.sql`

ATG Commerce Service Center Reporting

Prerequisite migration	Basic data warehouse Search Admin reporting
Batch file/shell script	`run_dcs_dw.{bat|sh}`
DDL scripts	`upgrade_dcs_dw_ddl.sql`

ATG Search

ATG Search comprises two modules:

- DAF.Search
- Search Admin

After database schema migration is complete, review the procedures in ATG Search Notes.

Note: The batch files/shell scripts and DDL scripts for DAF.Search migration are in the Platform migration kit; Search Admin has its own migration kit.

ATG Search Core

Prerequisite migration	Basic platform core

Batch file/shell scripts	**Platform migration kit:**
	`run_daf_search_core.{bat|sh}`
	SearchAdmin migration kit:
	`run_search_admin.{bat|sh}`

DDL scripts	**Platform migration kit:**
	`upgrade_daf_search_routing_ddl.sql`
	`drop_search_ddl.sql*`
	`search_ddl.sql*`
	SearchAdmin migration kit:
	`upgrade_search_admin_common_ddl.sql`
	`upgrade_search_admin_customizations_ddl.sql`
	`upgrade_search_admin_data.sql`

* Drop and recreate search index tables.

ATG Search Switching

Prerequisite migration	Basic platform switching
Batch file/shell scripts	`run_daf_search_switching.{bat|sh}`
DDL scripts	`search_site_ddl.sql` `upgrade_daf_search_common_ddl.sql`

Search Publishing

Prerequisite migration	Basic platform publishing
Batch file/shell script	`run_daf_search_publishing.{bat|sh}`

DDL scripts	`versioned_search_site_ddl.sql` `upgrade_daf_search_common_ddl.sql`

Agent

Note: Production agent schema migration scripts are required in order to migrate the following ATG applications:

- ATG Service
- ATG Commerce Service Center (CSC)

For more information, see the applicable migration guidelines.

Agent Core

Prerequisite migration	Basic platform core
Batch file/shell script	`run_agent_production_core.{bat|sh}`
DDL scripts	`upgrade_agent_production_ddl.sql`

Agent Server

Prerequisite migration	Basic platform publishing Publishing.base
Batch file/shell script	`run_agent_agent.{bat|sh}`
DDL scripts	`upgrade_agent_agent_ddl.sql` `upgrade_agent_production_ddl.sql`

ATG Service

This section covers the following topics:

- ATG Service schema migration
- ATG Service data migration

Before starting database migration, review Schema Migration Prerequisites for ATG Service.

ATG Service Schema Migration

Service Core

Prerequisite migration	Basic platform core ATG Search Agent core
Batch file/shell script	`run_service_production.{bat|sh}`

DDL scripts	`viewmapping_ddl.sql`
	`upgrade_unversioned_option_ddl.sql`
	`upgrade_unversioned_svc_solution_ddl.sql`
	`upgrade_service_production_ddl.sql`
	`upgrade_search_admin_production_ddl.sql`

Service Switching

Prerequisite migration	Basic platform switching ATG Search switching
Batch file/shell script	`run_service_switch.{bat|sh}*`
DDL scripts	`upgrade_unversioned_option_ddl.sql` `upgrade_unversioned_svc_solution_ddl.sql`

* Omit for an environment that uses online (non-switching) deployment.

Service Agent

Prerequisite migration	Basic platform switching Basic platform publishing Publishing.base Agent server
Batch file/shell script	`run_service_agent.{bat|sh}`
DDL scripts	`upgrade_versioned_option_ddl.sql` `upgrade_service_agent_ddl.sql`

Service Data Warehouse

Prerequisite migration	Basic data warehouse

Batch file/shell script	`run_service_datawarehouse.{bat|sh}`
DDL scripts	`upgrade_service_datawarehouse_ddl.sql`

Service Publishing

Prerequisite migration	Basic platform core Basic platform publishing Publishing. Base Search publishing
Batch file/shell script	n/a

ATG Service Data Migration

This section explains how to migrate the data in the databases for ATG Knowledge and ATG Self Service from version 9.1 to version 10.0.

To migrate version 9.1 data to a version 10.0 database:

1. Make sure you have completed all pre-migration tasks. See General Pre-Migration Tasks and Schema Migration Prerequisites for ATG Service. Make sure you have migrated the database schema for ATG Service. See ATG Service Schema Migration.
2. Run the Service.DBMigration.91to100 module.
3. Drop unused tables in your database schemas.
4. Import agent server data.
5. Import publishing data.

Run the Service.DBMigration.91to100 Module

1. Make sure you have a `PlatformLicenses.properties` file that is multisite enabled. Install this license file in your ATG 10.0 installation directory.
2. Copy the `xa-datasource` element with the `jndi-name` `ATGDataWarehouseDS` from the `atg-ds.xml` file for your asset management server to the `atg-ds.xml` file for your agent server. The `atg-ds.xml` file is in the deployment directory of your application server.

   For example, the `atg-ds.xml` file is in the following directory of a JBoss application server installation: `jboss-eap-5.0/jboss-as/server/atg_publishing_lockserver/deploy/atg-ds.xml`.
3. If you use switch deployment, configure the switching data sources for the `Service.DBMigration` module. Edit the following files in:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/`

   - `JTDataSource_switchA.properties`:

     ```
     $class=atg.nucleus.JNDIReference
     JNDIName=java:/ATGSwitchingDS_A
     ```

- JTDataSource_switchB.properties:

  ```
  $class=atg.nucleus.JNDIReference
  JNDIName=java:/ATGSwitchingDS_B
  ```

Note: Replace *ATGSwitchingDS_A* and *ATGSwitchingDS_B* with the JNDI names for the switching data sources as they are configured in the `atg-ds.xml` file for your application server.

4. Assemble the `Service.DBMigration.91to100` module with the `<ATG10dir>/home/bin/runAssembler` script as follows:

   ```
   runAssembler –server atg_svcagent_lockserver –
   standalone atg_svcagent_lockserver.ear -
   m Service.DBMigration.91to100
   ```

5. Start the module.

   Note: If you see an Abstract Deployment Context error, you might have two ATG EAR files in the agent server's deployment directory. In this case, delete the extra EAR file.

6. When startup is complete (on display of the Nucleus Running message), stop the application server.

Drop Unused Tables

To drop unused tables from your migrated database:

1. Grant permission to create a private synonym on the agent data source schema to the production core schema user.

   The following commands grant select privilege to the production core schema user. Replace *<svcagent_db>* with the name of the agent data source schema. Replace *<production_user>* with the name of the production core schema user.

   ```
   USE <svcagent_db>
   CREATE USER <production_user>
   GRANT SELECT TO <production_user>
   ```

2. Run the following script on the production core schema.

   `run_post_drop_production.{bat|sh}`

3. Run the following script on the production switching schema:

   `run_post_drop_switch.{bat|sh}`

4. Run the following script for your agent data source schema. If you installed ATG Knowledge and ATG Self Service with the Configuration and Installation Manager (CIM) utility, comment out the call to the script `drop_useroption_ddl.sql`.

   `run_post_drop_agent.{bat|sh}`

   Include parameters to supply the user account, password, and connection information that the script requires.

5. Edit the script `run_arf_dw.{bat|sh}` from the platform migration kit. Uncomment the call to the `drop_arf_search_site.sql` script. Comment out all other SQL script calls in the file.

6. Run the following script for your data warehouse schema:

   `run_arf_dw.{bat|sh}`

Import Agent Server Data

To import Agent server data, follow these steps:

1. Start the Configuration and Installation Manager (CIM) utility that you used to install ATG 10.0. Deploy `atg_svcagent_lockserver.ear`.
2. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc`
3. Configure two FakeXADataSource properties files with the appropriate connection information:
   - `FakeXADataSource.properties`
   - `FakeXADataSource_production.properties`

   For information about configuring FakeXADataSource files, see the *ATG Service Installation and Configuration Guide*.
4. Update the view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   ```
5. Start the agent server. After startup, let the server run for five minutes, then shut it down.
6. Copy:

   `<ATG10dir>/home/servers/atg_svcagent_lockserver`

   to:

   `<ATG10dir>/home/servers/svcagent_import91to100`
7. Change to this directory:

   `<ATG10dir>/home/servers/svcagent_import91to100/localconfig`
8. In this directory, configure the agent server as indicated in the following table.

   Note: Create directories and properties files as needed.

Path	Configuration
`/atg/dynamo/service/jdbc`	`FakeXADataSource_agent.properties`: Points to the agent schema `FakeXADataSource_production.properties`: Points to the production core schema. `FakeXADataSource.properties`: Points to the agent schema

Path	Configuration
/atg/dynamo/service/jdbc	If using switch deployment: SwitchingDataSourceA.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchA SwitchingDataSourceB.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchB FakeXADataSource_switchA and FakeXADataSource_switchB point to the switching schemas.
/atg/reporting/datawarehouse	FakeXADataSource.properties: Points to the data warehouse schema
/atg/reporting/datawarehouse/loaders	FakeXADataSource.properties: Points to the production core schema
/atg/search/routing	RoutingSearchService.properties: cacheService=
/atg/epub/deployment	DeploymentFulfiller.properties: enable=false

9. Delete the following directories:

   ```
   <ATG10dir>/home/servers/svcagent_import91to100/Publishing
   <ATG10dir>/home/servers/svcagent_import91to100/PublishingAgent
   ```

10. Run the following script in the Service migration kit:

    ```
    Service10.0/migration/91to100sql/scripts/run-agent-import.{bat|sh}
    ```

    Pass the name of the server directory you created earlier in this procedure as a parameter when you invoke the script. For example:

    ```
    run-agent-import.sh svcagent_import91to100
    ```

11. Repeat the previous step twice. On the third iteration, the script should execute error-free..

Import Publishing Data

To import publishing data:

1. Change to the following directory:

   ```
   <ATG10dir>/home/localconfig/atg/dynamo/service/jdbc
   ```

2. Configure two FakeXADataSource properties files with the appropriate connection information:

   - FakeXADataSource.properties

     Set connection information for the publishing schema.

- `FakeXADataSource.properties`

  Set connection information for the publishing schema.

- `FakeXADataSource_production.properties`

  Set connection information for the production core schema.

3. Update view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   <ATG10dir>/SiteAdmin/Versioned/install/importSiteAdmin.{bat|sh}
   <ATG10dir>/SiteAdmin/Search/install/importSiteAdminSearch.{bat|sh}
   ```

4. Invoke the startSQLRepository utility using as follows:

   UNIX:

   ```
   <ATG10dir>/home/bin/startSQLRepository -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>/Search10.0/SearchAdmin/common/data/initial/search
   admin-profile.xml
   ```

   Windows:

   ```
   <ATG10dir>\home\bin\startSQLRepository.bat -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>\Search10.0\SearchAdmin\common\data\initial\search
   admin-profile.xml
   ```

5. Start the asset management server. Let the server run for several minutes after startup is complete, then shut it down.

6. Copy this directory:

   ```
   <ATG10dir>/home/servers/atg_publishing_lockserver
   ```

   to:

   ```
   <ATG10dir>/home/servers/svcpub_import91to100
   ```

7. Create this directory:

   ```
   <ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/dynamo
   /service/jdbc
   ```

8. In the new directory, configure the following properties files, as noted:

   - `FakeXADataSource.properties`

     Set connection information for the publishing schema.

   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

   - `FakeXADataSource_agent.properties`

     Set connection information for the agent data source schema.

9. Change to this directory:

   ```
   <ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/
   ```

10. Edit the following properties files, as noted:

    - `/reporting/datawarehouse/FakeXADataSource.properties`

      Set connection information for the reporting data warehouse schema.

- `/reporting/datawarehouse/loaders/FakeXADataSource.properties`

  Set connection information for the production core schema.
- `/search/routing/RoutingSearchService.properties`

  `cacheService=`
- `/epub/deployment/DeploymentFulfiller.properties`

  `enable=false`

11. Delete this directory:

    `<ATG10dir>/home/servers/svcpub_import91to100/Publishing`
12. Run the following script in the Service10.0migration kit in order to import the Self Service sites:

    `../migration/91to100sql/scripts/run-management-import.{bat|sh}`

    This script uses the `svc_multisite_91to100.xml` file that was generated earlier by running the `Service.DBMigration.91to100` module.

    For example:

    `run-management-import.sh svcpub_import91to100`

ATG Commerce Service Center (CSC)

This section describes database migration procedures for customers who have ATG Commerce Service Center installed. The following topics are covered:

- CSC schema migration
- CSC data migration

Before starting database migration, review Schema Migration Prerequisites for CSC.

CSC Schema Migration

Note: Unlike CSC 9.1, CSC 10.0 does not require installation of ATG Service. The CSC 10.0 schema migration migrates the ATG Knowledge and ATG Service Framework tables, then drops the ATG Knowledge-only tables as described in CSC Data Migration.

CSC Core

Prerequisite migration	Basic platform core ATG Search Agent core B2CCommerce core

Migration scripts	**Service migration kit:**
	`run_service_production.{bat|sh}*`
	Platform migration kit:
	`run_click_to_connect_core.{bat|sh}**`
	CSC 10 migration kit:
	`run_dcs_csr_production.{bat|sh}`
	ATG10 installation:
	`<ATG10dir>/DAF/Search/common/sql/db_components/mssql/refinement_ddl.sql***`

* Comment out execution of the DDL script `upgrade_search_admin_production_ddl.sql`
** Required only if running ATG ClicktoConnect
*** Required only if CRS Search add-on was not previously installed. Omit if using switch deployment

CSC Switching

Prerequisite migration	Service switching B2CCommerce switching

Batch file/shell script	**CSC migration kit:**
	`run_dcs_csr_switch.{bat|sh}`
	ATG10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

CSC Agent

Prerequisite migration	Basic platform core Basic platform publishing Publishing.base Agent server

Batch file/shell script	**Service migration kit**:
	`run_service_agent.{bat|sh}`
	CSC migration kit:
	`run_dcs_csr_agent.{bat|sh}`

CSC Data Warehouse

Required only if running ATG Reporting.

Prerequisite migration	Base Data Warehouse ATG Commerce reporting Search Admin reporting
Batch file/shell script	n/a

CSC Publishing

Prerequisite migration	Basic platform Publishing.base Search publishing B2CCommerce publishing

Migration scripts	**CSC migration kit**:
	`run_dcs_csr_management.{bat|sh}`
	ATG 10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

CSC Data Migration

This section explains how to migrate CSC data from version 9.1 to version 10.0. Before you begin migrating CSC data, be sure you:

- Complete all pre-migration tasks. See General Pre-Migration Tasks and Schema Migration Prerequisites for CSC.
- Migrate the database schema for ATG Service.

CSC data migration comprises the following tasks:

1. Drop unused tables.
2. Update view mappings.
3. Start the agent server.
4. Configure the agent server.
5. Import agent server data.
6. Import Agent publishing data.

Drop Unused Tables

To drop unused tables from your migrated database, run the following DDL scripts:

- `run_post_drop_production.{bat|sh}` – Run on the production core schema.
- `run_post_drop_switch.{bat|sh}` – Run on the production switching schema.
- `run_post_drop_agent.{bat|sh}` – Run on the agent schema.

Update View Mappings

1. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/`

2. Configure the following properties files with connection information as noted:
   - `FakeXADataSource_agent.properties`

     Set connection information for the agent schema.
   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

3. Update the view mappings in the ATG Content Administration schema by running the ATG 10.0 import scripts that create view mappings:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetui.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   ```

Start the Agent Server

Start the new 10.0 Agent server, in order to initialize new publishing workflows. After server startup is complete, allow it to run for approximately five minutes, then shut the server down.

Configure the Agent Server

Change to this directory:

`<ATG10dir>/home/servers/svcagent_import91to100/localconfig`

In this directory, configure the agent server as indicated in the following table.

Note: Create directories and properties files as needed.

Path	Configuration
`/atg/dynamo/service/jdbc`	`FakeXADataSource_agent.properties`: Points to the agent schema `FakeXADataSource_production.properties`: Points to the production core schema. `FakeXADataSource.properties`: Points to the agent schema
`/atg/dynamo/service/jdbc`	If using switch deployment: `SwitchingDataSourceA.properties`: Points to `FakeXADataSource_switchA` `SwitchingDataSourceB.properties`: Points to `FakeXADataSource_switchB` `FakeXADataSource_switchA` and `FakeXADataSource_switchB` point to the switching schemas.

Path	Configuration
/atg/reporting/datawarehouse	FakeXADataSource.properties: Points to the data warehouse schema
/atg/reporting/datawarehouse/loaders	FakeXADataSource.properties: Points to the production core schema
/atg/search/routing	RoutingSearchService.properties: cacheService=
/atg/commerce/catalog	ProductCatalog.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/commerce/pricing/priceLists	PriceLists.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/assetregistry	AssetRegistry.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_agent
/atg/search/repository	RefinementRepository.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production

Import Agent Server Data

From the DYNAMO_HOME/CSC10.0/migration/91to100sql/scripts directory, run the following script:

```
run-csc-only-agent-import.{bat|sh} svcagent_import91to100
```

Import Agent Publishing Data

1. Change to the following directory:

   <ATG10dir>/home/localconfig/atg/dynamo/service/jdbc

2. Configure two FakeXADataSource properties files with the appropriate connection information:

   - FakeXADataSource.properties

     Set connection information for the publishing schema.

   - FakeXADataSource_production.properties

     Set connection information for the production core schema .

3. Update the view mappings in the ATG Content Administration schema by running the following ATG 10.0 import scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   ```

```
<ATG10dir>/SiteAdmin/Versioned/install/importSiteAdmin.{bat|sh}
<ATG10dir>/SiteAdmin/Search/install/importSiteAdminSearch.{bat|sh}
<ATG10dir>/DCS-UI/install/importMerchandising.{bat|sh}
<ATG10dir>/DCS-UI/Versioned/install/importDCSUIVersioned.{bat|sh}
```

4. Run the `startSQLRepository` command:

```
<ATG10dir>/home/bin/startSQLRepository.sh -m DPS.InternalUsers –
repository
/atg/userprofiling/InternalProfileRepository -import
<ATG10dir>/Search10.0/SearchAdmin/common/data/initial/searchadmin-
profile.xml
```

ATG Service and CSC

This section describes database migration procedures for customers who have both ATG Service and ATG Commerce Service Center installed. The following topics are covered:

- ATG Service/CSC schema migration
- ATG Service/CSC data migration

Before starting database migration, review Schema Migration Prerequisites for ATG Service/CSC.

ATG Service/CSC Schema Migration

Service/CSC Core

Prerequisite migration	Basic platform core ATG Search core B2CCommerce core Agent core

Migration scripts	**Platform migration kit:**
	`run_click_to_connect.{bat|sh}`*
	Service10.0migration kit:
	`run_service_production.{bat|sh}`
	CSC10.0migration kit:
	`run_dcs_csr_production.{bat|sh}`
	ATG10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql`**

* Required only if running ATG ClicktoConnect
** Required only if CRS Search add-on was not previously installed. Omit if using switch deployment.

Service/CSC Switching

Prerequisite migration	Basic Platform switching ATG Search switching B2CCommerce switching

Migration script	**Service10.0migration kit:**
	`run_service_switch.{bat|sh}`
	CSC10.0migration kit:
	`run_dcs_csr_switch.{bat|sh}`
	ATG 10.0 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

Service/CSC Agent

Prerequisite migration	Platform core Basic Platform Publishing Publishing.Base Agent server

Migration scripts	**Service10.0 migration kit:**
	`run_service_agent.{bat|sh}`
	CSC10.0 migration kit:
	`run_dcs_csr_agent.{bat|sh}`

Service/CSC Data Warehouse

Prerequisite migration	Base Data Warehouse ATG Commerce reporting Search Admin reporting
Migration scripts	**Service10.0 migration kit:** `run_service_datawarehouse.{bat|sh}`

Service/CSC Publishing

Prerequisite migration	Basic platform Publishing.Base Search publishing B2CCommerce publishing

Migration scripts	**CSC 10 migration kit:**
	`run_dcs_csr_management.{bat|sh}`
	ATG 10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

ATG Service/CSC Data Migration

This section explains how to migrate the data in the databases for ATG Service and ATG Commerce Service Center (CSC) from version 9.1 to version 10.0.

To migrate version 9.1 data to a version 10.0 database, perform the following tasks:

1. Run the Service.DBMigration.91to100 module.
2. Drop unused tables.
3. Import agent server data.
4. Migrate publishing data.

Run the Service.DBMigration.91to100 Module

1. From the asset management server's `atg-ds.xml` file, copy this <xa-datasource> setting:

   `<jndi-name>ATGDataWarehouseDS</jndi-name>`

   to the agent server's `atg-ds.xml` file.

   Note: The `atg-ds.xml` file is in the application server's deployment directory. For example, on a JBoss application server, `atg-ds.xml` file is in the following directory:

   `jboss-root``/jboss-as/server/atg_publishing_lockserver/deploy/`

2. If you use switch deployment, configure the switching data sources for the `Service.DBMigration` module in the following files:

   - `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/JTDataSource_switchA.properties`:

     ```
     $class=atg.nucleus.JNDIReference
     JNDIName=java:/ATGSwitchingDS_A
     ```

   - `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/JTDataSource_switchB.properties`:

     ```
     $class=atg.nucleus.JNDIReference
     JNDIName=java:/ATGSwitchingDS_B
     ```

Note: Replace `ATGSwitchingDS_A` and `ATGSwitchingDS_B` with the JNDI names for the switching data sources as they are configured in your application server's `atg-ds.xml` file.

3. In a command-line terminal, change to this directory:

   `<ATG10dir>/home/bin`

4. Assemble the `Service.DBMigration.91to100` module with the `<ATG10dir>/home/bin/runAssembler` script as follows:

   ```
   runAssembler –server atg_svcagent_lockserver –
   standalone atg_svcagent_lockserver.ear -
   m Service.DBMigration.91to100
   ```

5. Start the module.

   Note: If you see an Abstract Deployment Context error, you might have two ATG EAR files in the agent server's deployment directory. In this case, delete the extra EAR file.

6. When startup is complete (on display of the Nucleus Running message), stop the application server.

Drop Unused Tables

Note: If you intend to run a full deployment after migration, skip steps 1-4:

1. Grant the database user access to tables on both the agent and switching/production schemas. Grant select privilege of the agent database to production users:

   ```
   USE svcagent_db
   CREAT USER production_user
   GRANT SELECT to production_user
   ```

2. In the Service migration kit, edit the following script:

   `../mssql/complete_solution_data_migration_ddl.sql`

   Replace `<svcagent_schema>` with the name of the agent data source schema user.

3. If you use switch deployment, repeat step 2 for each switching data source schema user.

4. Run `complete_solution_data_migration_ddl.sql`

5. If you use switch deployment, run this migration script on each switching schema; otherwise, run it on the production schema:

   `run_service_production_complete_solution_migration.{bat|sh}`

6. Run the following script on the production core schema:

   `run_post_drop_production.{bat|sh}`

7. Run the following script on the production switching schema:

   `run_post_drop_switch.{bat|sh}`

8. Run the following script on your agent data source schema. If you installed ATG Knowledge and ATG Self Service with the Configuration and Installation Manager (CIM) utility, comment out the call to the script `drop_useroption_ddl.sql`.

   `run_post_drop_agent.{bat|sh}`

9. Edit the script `run_arf_dw.{bat|sh}` from the platform migration kit. Uncomment the call to the `drop_arf_search_site.sql` script. Comment out all other SQL script calls in the file.

10. Run the following script on your data warehouse schema.

```
run_arf_dw.{bat|sh}
```

Import Agent Server Data

To import Agent server data, follow these steps:

1. Start the Configuration and Installation Manager (CIM) utility that you used to install ATG 10.0. Deploy `atg_svcagent_lockserver.ear`.
2. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc`
3. Configure two FakeXADataSource properties files with the appropriate connection information:
   - `FakeXADataSource.properties`

     Set connection information for the agent schema.
   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

   For information about configuring `FakeXADataSource` files, see the *ATG Service Installation and Configuration Guide*.
4. Update the view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   ```
5. Start the agent server. After startup is complete, let the server run for five minutes so it can initialize new publishing workflows, then shut it down.
6. Copy:

   `<ATG10dir>/home/servers/atg_svcagent_lockserver`

   to this new directory:

   `<ATG10dir>/home/servers/svcagent_import91to100`
7. In this directory, configure the agent server as indicated in the following table.

   Note: Create directories and properties files as needed.

Path	Configuration
`/atg/dynamo/service/jdbc`	`FakeXADataSource_agent.properties`: Points to the agent schema `FakeXADataSource_production.properties`: Points to the production schema. `FakeXADataSource.properties`: Points to the agent schema

Path	Configuration
/atg/dynamo/service/jdbc	If using switch deployment: SwitchingDataSourceA.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchA SwitchingDataSourceB.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchB FakeXADataSource_switchA and FakeXADataSource_switchB point to the switching schemas.
/atg/reporting/datawarehouse	FakeXADataSource.properties: Points to the data warehouse schema
/atg/reporting/datawarehouse/loaders	FakeXADataSource.properties: Points to the production schema
/atg/search/routing	RoutingSearchService.properties: cacheService=
/atg/epub/deployment	DeploymentFulfiller.properties: enable=false
/atg/commerce/catalog	ProductCatalog.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/commerce/pricing/priceLists	PriceLists.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/assetregistry	AssetRegistry.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_agent
/atg/search/repository	RefinementRepository.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production

8. From the Service migration kit's /scripts directory, run this script:

   ```
   run-agent-import.{bat|sh} svcagent_import91to100
   ```

9. Repeat the previous step twice. On the third iteration, the script should execute error-free.

10. From the CSC migration kit's /scripts directory, run this script:

    ```
    run-csc-knowledge-agent-import.{bat|sh} svcagent_import91to100
    ```

Migrate Publishing Data

1. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/`

2. Configure two FakeXADataSource properties files with the appropriate connection information:

   - `FakeXADataSource.properties`

     Set connection information for the publishing schema.

   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

3. Update view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   <ATG10dir>/SiteAdmin/Versioned/install/importSiteAdmin.{bat|sh}
   <ATG10dir>/SiteAdmin/Search/install/importSiteAdminSearch.{bat|sh}
   <ATG10dir>/DCS-UI/install/importMerchandising.{bat|sh}
   <ATG10dir>/DCS-UI/Versioned/install/importDCSUIVersioned.{bat|sh}
   ```

4. Invoke the startSQLRepository utility as follows:

   UNIX:

   ```
   <ATG10dir>/home/bin/startSQLRepository -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>/Search10.0/SearchAdmin/common/data/initial/search
   admin-profile.xml
   ```

   Windows:

   ```
   <ATG10dir>\home\bin\startSQLRepository.bat -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>\Search10.0\SearchAdmin\common\data\initial\search
   admin-profile.xml
   ```

5. Start the asset management server. Let the server run for several minutes after startup is complete so it can initialize new publishing workflows, then shut it down.

6. Copy this directory:

   `<ATG10dir>/home/servers/atg_publishing_lockserver`

   to:

   `<ATG10dir>/home/servers/svcpub_import91to100`

7. Create this directory:

   ```
   <ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/dynamo
   /service/jdbc
   ```

8. In the new directory, configure the following properties files, as noted:

   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

   - `FakeXADataSource_agent.properties`

     Set connection information for the agent data source schema.

9. Change to this directory:

   `<ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/`

10. Edit the following properties files, as noted:

    - `/reporting/datawarehouse/FakeXADataSource.properties`

      Set connection information for the reporting data warehouse schema.

    - `/reporting/datawarehouse/loaders/FakeXADataSource.properties`

      Set connection information for the production core schema.

    - `/search/routing/RoutingSearchService.properties`

      `cacheService=`

    - `/epub/deployment/DeploymentFulfiller.properties`

      `enable=false`

11. Run the following script in the Service10.0 migration kit, to import the ATG Self Service sites:

    `../migration/91to100sql/scripts/run-management-import.{bat|sh}`

    Pass the name of the server directory you created in step 6 as a parameter when you invoke the script. For example:

    `run-management-import.sh svcpub_import91to100`

6 DB2 Database Migration

ATG 10.0 includes a number of enhancements to the database schemas used by ATG 9.1. The DB2 migration kit provides the batch files/shell scripts and DDL scripts that you need to upgrade existing DB2 schemas, in the following migration kit directory:

```
/91to100sql/db_components/db2
```

Each section in this chapter describes the batch files/shell scripts and DDL scripts that are required to migrate a specific ATG module or group of related modules:

- Basic platform (includes DAS, DPS, and DSS)
- Publishing.Base
- B2CCommerce
- B2BCommerce
- Data Warehouse
- ATG Search
- Agent
- ATG Service
- ATG Commerce Service Center (CSC)
- ATG Service and CSC

Each section is subdivided according to the schemas that require upgrading: core, switching, and publishing.

Note: Before running migration scripts, carefully review the Database Schema Migration section in the Migration Steps chapter.

Basic Platform

The ATG platform minimally comprises three modules whose schemas require migration: DAS, DPS, and DSS. The sections that follow document schema migration scripts for these modules.

Basic Platform Core

Batch file/shell scripts	`run_das_core.{bat|sh}`
	`run_dps_core.{bat|sh}`
	`run_dss_core.{bat|sh}`

DDL scripts	`upgrade_das_core_ddl.sql`
	`upgrade_dps_ddl.sql`
	`upgrade_dss_ddl.sql`

Basic Platform Switching

Batch file/shell script	`run_das_switching.{bat|sh}`
DDL scripts	`upgrade_das_seo_ddl.sql` `site_ddl.sql`

Basic Platform Publishing

Note: Versioned database tables must now set their `checkin_date` column to `TIMESTAMP`. If necessary, be sure to manually update your application's versioned database tables accordingly.

Batch file/shell script	`run_das_publishing.{bat|sh}`
	`run_dps_internal_users_publishing.{bat|sh}`
	`run_dss_internal_users_publishing.{bat|sh}`

DDL scripts	`versioned_site_ddl.sql`
	`upgrade_das_versioned_ddl.sql`
	`upgrade_dps_internal_users_ddl.sql`
	`upgrade_dss_internal_users_ddl.sql`

Publishing.Base

Prerequisite migration	Basic platform publishing
Batch file/shell script	`run_publishing_base.{bat|sh}`
DDL scripts	`upgrade_publishing_ddl.sql` `upgrade_viewmapping_ddl.sql`

B2BCommerce

The DB2 migration kit has no migration scripts that are specific to B2BCommerce. B2BCommerce customers only need to run the B2CCommerce migration scripts.

Data Warehouse

Note: Before you migrate the database schemas for ATG Customer Intelligence data warehouse, review the procedures in ATG Customer Intelligence Notes.

Base Data Warehouse

Batch file/shell script	`run_arf_dw_base.{bat|sh}`
DDL scripts	`upgrade_arf_dw_base_ddl.sql` `upgrade_arf_dw_base_init.sql`

ATG Commerce Reporting

Prerequisite migration	Base Data Warehouse
Batch file/shell script	`run_dcs_dw.{bat|sh}`
DDL scripts	`upgrade_dcs_dw_ddl.sql`

Search Admin Reporting

Prerequisite migration	Base Data Warehouse
Batch file/shell script	`run_arf_dw.{bat|sh}`
DDL scripts	`upgrade_arf_dw_search_ddl.sql`

ATG Commerce Service Center Reporting

Prerequisite migration	Base Data Warehouse Search Admin reporting
Batch file/shell script	`run_dcs_dw.{bat|sh}`
DDL scripts	`upgrade_dcs_dw_ddl.sql`

ATG Search

ATG Search comprises two modules:

- DAF.Search
- Search Admin

After database schema migration is complete, review the procedures in ATG Search Notes.

Note: The batch files/shell scripts and DDL scripts for DAF.Search migration are in the Platform migration kit; Search Admin has its own migration kit.

ATG Search Core

Prerequisite migration	Basic platform core

Batch file/shell scripts	**Platform migration kit:**
	`run_daf_search_core.{bat|sh}`
	SearchAdmin migration kit:
	`run_search_admin.{bat|sh}`

DDL scripts	**Platform migration kit:**
	`upgrade_daf_search_routing_ddl.sql`
	`drop_search_ddl.sql*`
	`search_ddl.sql*`
	SearchAdmin migration kit:
	`upgrade_search_admin_common_ddl.sql`
	`upgrade_search_admin_customizations_ddl.sql`
	`upgrade_search_admin_data.sql`

* Drop and recreate search index tables.

ATG Search Switching

Prerequisite migration	Basic platform switching
Batch file/shell scripts	`run_daf_search_switching.{bat|sh}`
DDL scripts	`search_site_ddl.sql` `upgrade_daf_search_common_ddl.sql`

Search Publishing

Prerequisite migration	Basic platform publishing
Batch file/shell script	`run_daf_search_publishing.{bat|sh}`
DDL scripts	`versioned_search_site_ddl.sql` `upgrade_daf_search_common_ddl.sql`

Agent

Note: Production agent schema migration scripts are required in order to migrate the following ATG applications:

- ATG Service
- ATG Commerce Service Center (CSC)

For more information, see migration guidelines for these products.

Agent Core

Prerequisite migration	Basic platform core
Batch file/shell script	`run_agent_production_core.{bat|sh}`
DDL scripts	`upgrade_agent_production_ddl.sql`

Agent Server

Prerequisite migration	Basic platform publishing Publishing.base
Batch file/shell script	`run_agent_agent.{bat|sh}`
DDL scripts	`upgrade_agent_agent_ddl.sql` `upgrade_agent_production_ddl.sql`

ATG Service

This section covers the following topics:

- ATG Service schema migration
- ATG Service data migration

Before starting database migration, review Schema Migration Prerequisites for ATG Service.

ATG Service Schema Migration

Service Core

Prerequisite migration	Basic platform core ATG Search Agent core
Batch file/shell script	`run_service_production.{bat|sh}`

DDL scripts	`viewmapping_ddl.sql`
	`upgrade_unversioned_option_ddl.sql`
	`upgrade_unversioned_svc_solution_ddl.sql`
	`upgrade_service_production_ddl.sql`
	`upgrade_search_admin_production_ddl.sql`

* Omit for CSC-only migration

Service Switching

Prerequisite migration	Basic platform switching ATG Search switching
Batch file/shell script	`run_service_switch.{bat|sh}*`
DDL scripts	`upgrade_unversioned_option_ddl.sql` `upgrade_unversioned_svc_solution_ddl.sql`

* Omit for an environment that uses online (non-switching) deployment.

Service Agent

Prerequisite migration	Basic platform switching Basic platform publishing Publishing.base Agent server
Batch file/shell script	`run_service_agent.{bat|sh}`

DDL scripts	`upgrade_versioned_svc_solution_ddl.sql`
	`upgrade_svc_publishing_ddl.sql`
	`upgrade_versioned_option_ddl.sql`

Service Data Warehouse

Prerequisite migration	Basic data warehouse
Batch file/shell script	`run_service_datawarehouse.{bat|sh}`
DDL scripts	`upgrade_service_datawarehouse_ddl.sql`

Service Publishing

Prerequisite migration	Basic platform core Basic platform publishing Publishing. Base Search publishing
Batch file/shell script	n/a

ATG Service Data Migration

This section explains how to migrate the data in the databases for ATG Knowledge and ATG Self Service from version 9.1 to version 10.0.

To migrate version 9.1 data to a version 10.0 database:

1. Make sure you have completed all pre-migration tasks. See General Pre-Migration Tasks and Schema Migration Prerequisites for ATG Service. Make sure you have migrated the database schema for ATG Service. See ATG Service Schema Migration.
2. Run the Service.DBMigration.91to100 module.
3. Drop unused tables in your database schemas.
4. Import agent server data.
5. Migrate publishing data.

Run the Service.DBMigration.91to100 Module

1. Make sure you have a `PlatformLicenses.properties` file that is multisite enabled. Install this license file in your ATG 10.0 installation directory.
2. Copy the `xa-datasource` element with the `jndi-name` `ATGDataWarehouseDS` from the `atg-ds.xml` file for your asset management server to the `atg-ds.xml` file for your agent server. The `atg-ds.xml` file is in the deployment directory of your application server.

   For example, the `atg-ds.xml` file is in the following directory of a JBoss application server installation: `jboss-eap-5.0/jboss-as/server/atg_publishing_lockserver/deploy/atg-ds.xml`.
3. If you use switch deployment, configure the switching data sources for the `Service.DBMigration` module. Edit the following files in:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/`

- JTDataSource_switchA.properties:

  ```
  $class=atg.nucleus.JNDIReference
  JNDIName=java:/ATGSwitchingDS_A
  ```

- JTDataSource_switchB.properties:

  ```
  $class=atg.nucleus.JNDIReference
  JNDIName=java:/ATGSwitchingDS_B
  ```

Note: Replace *ATGSwitchingDS_A* and *ATGSwitchingDS_B* with the JNDI names for the switching data sources as they are configured in the `atg-ds.xml` file for your application server.

4. Assemble the `Service.DBMigration.91to100` module with the `<ATG10dir>/home/bin/runAssembler` script as follows:

   ```
   runAssembler –server atg_svcagent_lockserver –
   standalone atg_svcagent_lockserver.ear -
   m Service.DBMigration.91to100
   ```

5. Start the module.
6. When startup is complete (on display of the Nucleus Running message), stop the application server.

Drop Unused Tables

Note: If you intend to run a full deployment after migration, skip steps 1-4:

1. Grant permission to create a private synonym on the agent data source schema to the production core schema user.

   The following commands grant select privilege to the production core schema user. Replace *<production_user>* with the name of the production core schema user.

   ```
   GRANT CREATE SYNONYM TO <production_user>
   ```

2. Grant SELECT permission on the `svc_soln` and `avm_devline` tables in the agent data source schema to the user of the production core schema.

   The following commands grant SELECT permission to the production core schema user. Replace *<production_user>* with the name of the production core user.

   ```
   GRANT SELECT ON svc_soln TO USER <production_schema>;
   GRANT SELECT ON avm_devline TO USER <production_schema>;
   COMMIT;
   ```

3. In the Service migration kit, run this script:

   ```
   ../db2/complete_solution_data_migration_ddl.sql
   ```

4. If you use switch deployment, repeat steps 2 and 3 for each production schema user.
5. If you use switch deployment, run this migration script on each switching schema; otherwise, run it on the production schema:

   ```
   run_service_production_complete_solution_migration.{bat|sh}
   ```

6. Run the following script on the production core schema:

   ```
   run_post_drop_production.{bat|sh}
   ```

7. Run the following script on the production switching schema:

   ```
   run_post_drop_switch.{bat|sh}
   ```

8. Run the following script for your agent data source schema. If you installed ATG Knowledge and ATG Self Service with the Configuration and Installation Manager (CIM) utility, comment out the call to the script `drop_useroption_ddl.sql`.

   `run_post_drop_agent.{bat|sh}`

9. Edit the script `run_arf_dw.{bat|sh}` from the platform migration kit. Uncomment the call to the `drop_arf_search_site.sql` script. Comment out all other SQL script calls in the file.

10. Run the following script for your data warehouse schema:

    `run_arf_dw.{bat|sh}`

Import Agent Server Data

To import Agent server data, follow these steps:

1. Start the Configuration and Installation Manager (CIM) utility that you used to install ATG 10.0. Deploy `atg_svcagent_lockserver.ear`.

2. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc`

3. Configure the following properties files with connection information as noted:

   - `FakeXADataSource_agent.properties`

     Set connection information for the agent schema.

   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

   For information about configuring `FakeXADataSource` files, see the *ATG Service Installation and Configuration Guide*.

4. Update the view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   ```

5. Start the agent server. After startup, let the server run for five minutes, then shut it down.

6. Copy:

   `<ATG10dir>/home/servers/atg_svcagent_lockserver`

   to:

   `<ATG10dir>/home/servers/svcagent_import91to100`

7. Change to this directory:

   `<ATG10dir>/home/servers/svcagent_import91to100/localconfig`

8. In this directory, configure the agent server as indicated in the following table.

   Note: Create directories and properties files as needed.

Path	Configuration
/atg/dynamo/service/jdbc	FakeXADataSource_agent.properties: Points to the agent schema FakeXADataSource_production.properties: Points to the production core schema. FakeXADataSource.properties: Points to the agent schema
/atg/dynamo/service/jdbc	If using switch deployment: SwitchingDataSourceA.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchA SwitchingDataSourceB.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchB FakeXADataSource_switchA and FakeXADataSource_switchB point to the switching schemas.
/atg/reporting/datawarehouse	FakeXADataSource.properties: Points to the data warehouse schema
/atg/reporting/datawarehouse/loaders	FakeXADataSource.properties: Points to the production core schema
/atg/search/routing	RoutingSearchService.properties: cacheService=
/atg/epub/deployment	DeploymentFulfiller.properties: enable=false

9. Delete the following directories:

   ```
   <ATG10dir>/home/servers/svcagent_import91to100/Publishing
   <ATG10dir>/home/servers/svcagent_import91to100/PublishingAgent
   ```

10. Run the following script in the Service migration kit:

    ```
    Service10.0/migration/91to100sql/scripts/run-agent-import.{bat|sh}
    ```

    Pass the name of the server directory you created earlier in this procedure as a parameter when you invoke the script. For example:

    ```
    run-agent-import.sh svcagent_import91to100
    ```

Migrate Publishing Data

To migrate publishing data:

1. Change to the following directory:

```
<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/
```

2. Configure two FakeXADataSource properties files with the appropriate connection information:
   - `FakeXADataSource.properties`

     Set connection information for the publishing schema.
   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.
3. Update view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   <ATG10dir>/SiteAdmin/Versioned/install/importSiteAdmin.{bat|sh}
   <ATG10dir>/SiteAdmin/Search/install/importSiteAdminSearch.{bat|sh}
   ```
4. Invoke the startSQLRepository utility as follows:

   UNIX:

   ```
   <ATG10dir>/home/bin/startSQLRepository -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>/Search10.0/SearchAdmin/common/data/initial/search
   admin-profile.xml
   ```

   Windows:

   ```
   <ATG10dir>\home\bin\startSQLRepository.bat -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>\Search10.0\SearchAdmin\common\data\initial\search
   admin-profile.xml
   ```
5. Start the asset management server. Let the server run for several minutes after startup is complete, then shut it down.
6. Copy this directory:

   ```
   <ATG10dir>/home/servers/atg_publishing_lockserver
   ```

   to:

   ```
   <ATG10dir>/home/servers/svcpub_import91to100
   ```
7. Create this directory:

   ```
   <ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/dynamo
   /service/jdbc
   ```
8. In the new directory, configure the following properties files, as noted:
   - `FakeXADataSource.properties`

     Set connection information for the publishing schema.
   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.
   - `FakeXADataSource_agent.properties`

     Set connection information for the agent data source schema.
9. Change to this directory:

   ```
   <ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/
   ```

10. Edit the following properties files, as noted:
    - `/reporting/datawarehouse/FakeXADataSource.properties`

      Set connection information for the reporting data warehouse schema.
    - `/reporting/datawarehouse/loaders/FakeXADataSource.properties`

      Set connection information for the production core schema.
    - `/search/routing/RoutingSearchService.properties`

      `cacheService=`
    - `/epub/deployment/DeploymentFulfiller.properties`

      `enable=false`
11. Delete this directory:

    `<ATG10dir>/home/servers/svcpub_import91to100/Publishing`
12. Run the following script in the Service10.0 migration kit in order to import the Self Service sites:

    `../migration/91to100sql/scripts/run-management-import.{bat|sh}`

    This script uses the `svc_multisite_91to100.xml` file that was generated earlier by running the `Service.DBMigration.91to100` module.

    For example:

    `run-management-import.sh svcpub_import91to100`

ATG Commerce Service Center (CSC)

This section describes database migration procedures for customers who have ATG Commerce Service Center installed. The following topics are covered:

- CSC schema migration
- CSC data migration

Before starting database migration, review Schema Migration Prerequisites for CSC.

CSC Schema Migration

Note: Unlike CSC 9.1, CSC 10.0 does not require installation of ATG Service. The CSC 10.0 schema migration migrates the ATG Knowledge and ATG Service Framework tables, then drops the ATG Knowledge-only tables as described in CSC Data Migration.

CSC Core

Prerequisite migration	Service core B2CCommerce core

Migration scripts	**Service migration kit:**
	`run_service_production.{bat|sh}*`
	Platform migration kit:
	`run_click_to_connect_core.{bat|sh}**`
	CSC 10.0 migration kit:
	`run_dcs_csr_production.{bat|sh}`
	ATG10 installation:
	`<ATG10dir>/DAF/Search/common/sql/db_components/db2/refinement_ddl.sql***`

* Comment out execution of the DDL script `upgrade_search_admin_production_ddl.sql`
** Required only if running ATG ClicktoConnect
*** Required only if CRS Search add-on was not previously installed. Omit if using switch deployment

CSC Switching

Prerequisite migration	Service switching B2CCommerce switching

Batch file/shell script	**CSC migration kit:**
	`run_dcs_csr_switch.{bat|sh}`
	ATG10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed

CSC Agent

Prerequisite migration	Basic platform core Basic platform publishing Publishing.base Agent server

Batch file/shell script	**Service migration kit**:
	`run_service_agent.{bat|sh}`
	CSC migration kit:
	`run_dcs_csr_agent.{bat|sh}`

CSC Data Warehouse

Prerequisite migration	Base data warehouse ATG Commerce reporting Search Admin reporting
Batch file/shell script	n/a

CSC Publishing

Prerequisite migration	Basic platform Publishing.base Search publishing B2CCommerce publishing

Migration scripts	**CSC migration kit**:
	`run_dcs_csr_management.{bat|sh}`
	ATG 10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

CSC Data Migration

This section explains how to migrate CSC data from version 9.1 to version 10.0. Before you begin migrating CSC data, be sure you:

- Complete all pre-migration tasks. See General Pre-Migration Tasks and Schema Migration Prerequisites for CSC.
- Migrate the database schema for ATG Service.

CSC data migration comprises the following tasks:

1. Drop unused tables.
2. Update view mappings.
3. Start the agent server.
4. Configure the agent server.
5. Import agent server data.
6. Import Agent publishing data.

Drop Unused Tables

To drop unused tables from your migrated database, run the following DDL scripts:

- `run_post_drop_production.{bat|sh}` – Run on the production core schema.
- `run_post_drop_switch.{bat|sh}` – Run on the production switching schema.

- `run_post_drop_agent.{bat|sh}` – Run on the agent schema.

Update View Mappings

1. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/`

2. Configure the following properties files with connection information as noted:
   - `FakeXADataSource_agent.properties`

     Set connection information for the agent schema.
   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.
3. Update the view mappings in the ATG Content Administration schema by running the ATG 10.0 import scripts that create view mappings:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetui.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   ```

Start the Agent Server

Start the new 10.0 Agent server, in order to initialize new publishing workflows. After server startup is complete, allow it to run for approximately five minutes, then shut the server down.

Configure the Agent Server

Change to this directory:

`<ATG10dir>/home/servers/svcagent_import91to100/localconfig`

In this directory, configure the agent server as indicated in the following table.

Note: Create directories and properties files as needed.

Path	Configuration
`/atg/dynamo/service/jdbc`	`FakeXADataSource_agent.properties`: Points to the agent schema `FakeXADataSource_production.properties`: Points to the production core schema. `FakeXADataSource.properties`: Points to the agent schema
`/atg/dynamo/service/jdbc`	If using switch deployment: `SwitchingDataSourceA.properties`: Points to `FakeXADataSource_switchA` `SwitchingDataSourceB.properties`: Points to `FakeXADataSource_switchB` `FakeXADataSource_switchA` and `FakeXADataSource_switchB` point to the switching schemas.

Path	Configuration
/atg/reporting/datawarehouse	FakeXADataSource.properties: Points to the data warehouse schema
/atg/reporting/datawarehouse/loaders	FakeXADataSource.properties: Points to the production core schema
/atg/search/routing	RoutingSearchService.properties: cacheService=
/atg/commerce/catalog	ProductCatalog.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/commerce/pricing/priceLists	PriceLists.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/assetregistry	AssetRegistry.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_agent
/atg/search/repository	RefinementRepository.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production

Import Agent Server Data

From the DYNAMO_HOME/CSC10.0/migration/91to100sql/scripts directory, run the following script:

```
run-csc-only-agent-import.{bat|sh} svcagent_import91to100
```

Import Agent Publishing Data

1. Change to the following directory:

   <ATG10dir>/home/localconfig/atg/dynamo/service/jdbc

2. Configure two FakeXADataSource properties files with the appropriate connection information:

   - FakeXADataSource.properties

     Set connection information for the publishing schema.

   - FakeXADataSource_production.properties

     Set connection information for the production core schema .

   For information about configuring FakeXADataSource files, see the *ATG Service Installation and Configuration Guide*.

   <atg10dir>/home/localconfig/atg/dynamo/service/jdbc/

3. Update the view mappings in the ATG Content Administration schema by running the following ATG 10.0 import scripts:

```
<ATG10dir>/BIZUI/install/importBizui.{bat|sh}
<ATG10dir>/BCC/install/importBCC.{bat|sh}
<ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
<ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
<ATG10dir>/SiteAdmin/Versioned/install/importSiteAdmin.{bat|sh}
<ATG10dir>/SiteAdmin/Search/install/importSiteAdminSearch.{bat|sh}
<ATG10dir>/DCS-UI/install/importMerchandising.{bat|sh}
<ATG10dir>/DCS-UI/Versioned/install/importDCSUIVersioned.{bat|sh}
```

4. Run the `startSQLRepository` command:

```
<ATG10dir>/home/bin/startSQLRepository.sh -m DPS.InternalUsers
–repository /atg/userprofiling/InternalProfileRepository
-
import <ATG10dir>/Search10.0/SearchAdmin/common/data/initial/search
admin-profile.xml
```

ATG Service and CSC

This section describes database migration procedures for customers who have both ATG Service and ATG Commerce Service Center installed. The following topics are covered:

- ATG Service/CSC schema migration
- ATG Service/CSC data migration

Before starting database migration, review Schema Migration Prerequisites for ATG Service/CSC.

ATG Service/CSC Schema Migration

Service/CSC Core

Prerequisite migration	Basic platform core ATG Search core B2CCommerce core Agent core

Migration scripts	**Platform migration kit:**
	`run_ticketing_core.{bat|sh}`
	`run_click_to_connect.{bat|sh}*`
	Service10.0 migration kit:
	`run_service_production.{bat|sh}`
	CSC 10.0 migration kit:
	`run_dcs_csr_production.{bat|sh}`
	ATG 10.0 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql**`

* Required only if running ATG ClicktoConnect
** Required only if CRS Search add-on was not previously installed. Omit if using switch deployment.

Service/CSC Switching

Prerequisite migration	Basic Platform switching ATG Search switching B2CCommerce switching

Migration script	**Service10.0 migration kit:**
	`run_service_switch.{bat|sh}`
	CSC10.0 migration kit:
	`run_dcs_csr_switch.{bat|sh}`
	ATG 10.0 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

Service/CSC Agent

Prerequisite migration	Platform core Basic Platform Publishing Publishing.Base Agent server

Migration scripts	**Service10.0 migration kit:**
	`run_service_agent.{bat|sh}`
	CSC10.0 migration kit:
	`run_dcs_csr_agent.{bat|sh}`

Service/CSC Data Warehouse

Prerequisite migration	Base Data Warehouse ATG Commerce reporting Search Admin reporting
Migration scripts	**Service10.0 migration kit:** `run_service_datawarehouse.{bat|sh}`

Service/CSC Publishing

Prerequisite migration	Basic platform Publishing.Base Search publishing B2CCommerce publishing

Migration scripts	**CSC 10 migration kit:**
	`run_dcs_csr_management.{bat|sh}`
	ATG 10 installation:
	`../DAF/Search/common/sql/db_components/oracle/refinement_ddl.sql*`

* Required only if CRS Search add-on was not previously installed.

ATG Service/CSC Data Migration

This section explains how to migrate the data in the databases for ATG Service and ATG Commerce Service Center (CSC) from version 9.1 to version 10.0.

To migrate version 9.1 data to a version 10.0 database, perform the following tasks:

1. Run the Service.DBMigration.91to100 module.
2. Drop unused tables.
3. Import agent server data.
4. Import publishing data.

Run the Service.DBMigration.91to100 Module

1. From the asset management server's `atg-ds.xml` file, copy this <xa-datasource> setting:

   `<jndi-name>ATGDataWarehouseDS</jndi-name>`

   to the agent server's `atg-ds.xml` file.

   Note: The `atg-ds.xml` file is in the application server's deployment directory. For example, on a JBoss application server, `atg-ds.xml` file is in the following directory:

   `jboss-root``/jboss-as/server/atg_publishing_lockserver/deploy/`

2. If you use switch deployment, configure the switching data sources for the `Service.DBMigration` module in the following files:

   - `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/JTDataSource_switchA.properties`:

     ```
     $class=atg.nucleus.JNDIReference
     JNDIName=java:/ATGSwitchingDS_A
     ```

   - `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc/JTDataSource_switchB.properties`:

     ```
     $class=atg.nucleus.JNDIReference
     JNDIName=java:/ATGSwitchingDS_B
     ```

Note: Replace `ATGSwitchingDS_A` and `ATGSwitchingDS_B` with the JNDI names for the switching data sources as they are configured in your application server's `atg-ds.xml` file.

3. In a command-line terminal, change to this directory:

   `<ATG10dir>/home/bin`

4. Assemble the `Service.DBMigration.91to100` module with the `<ATG10dir>/home/bin/runAssembler` script as follows:

   ```
   runAssembler –server atg_svcagent_lockserver –
   standalone atg_svcagent_lockserver.ear -
   m Service.DBMigration.91to100
   ```

5. Start the module.

   Note: If you see an Abstract Deployment Context error, you might have two ATG EAR files in the agent server's deployment directory. In this case, delete the extra EAR file.

6. When startup is complete (on display of the Nucleus Running message), stop the application server.

Drop Unused Tables

To drop unused tables from your migrated database:

1. Grant the production user the ability to create a private synonym by using the `GRANT CREATE SYNONYM TO` *`production_schema`* command.
2. The `svcagent` schema must grant SELECT access to the `svc_soln` and `avm_devline` tables:

   ```
   GRANT SELECT ON svc_soln TO USER non-versioned_production_schema;
   GRANT SELECT ON avm_devline TO USER non-
   versioned_production_schema;
   COMMIT;
   ```

3. If you use switch deployment, repeat step 2 for each switching data source schema user.
4. In the Service migration kit, edit the following script:

   `../db2/complete_solution_data_migration_ddl.sql`

   - Replace `<svcagent_schema>` with the name of the agent data source schema user.
   - Replace `<production_schema>` with the name of the production core schema user. If you use switch deployment, replace `<production_schema>` with the name of one schema user.

5. Run `complete_solution_data_migration_ddl.sql`
6. If you use switch deployment, repeat steps 4 and 5 for each production schema user.
7. **Note:** If you intend to run a full deployment after migration, skip this step.

   If you use switch deployment, run this migration script on each switching schema; otherwise, run it on the production schema:

   `run_service_production_complete_solution_migration.{bat|sh}`

8. Run the following script on the production core schema:

   `run_post_drop_production.{bat|sh}`

9. Run the following script on the production switching schema:

```
run_post_drop_switch.{bat|sh}
```

10. Run the following script on your agent data source schema. If you installed ATG Knowledge and ATG Self Service with the Configuration and Installation Manager (CIM) utility, comment out the call to the script `drop_useroption_ddl.sql`.

    ```
    run_post_drop_agent.{bat|sh}
    ```

11. Edit the script `run_arf_dw.{bat|sh}` from the platform migration kit. Uncomment the call to the `drop_arf_search_site.sql` script. Comment out all other SQL script calls in the file.
12. Run the following script on your data warehouse schema.

    ```
    run_arf_dw.{bat|sh}
    ```

Import Agent Server Data

To import Agent server data, follow these steps:

1. Start the Configuration and Installation Manager (CIM) utility that you used to install ATG 10.0. Deploy `atg_svcagent_lockserver.ear`.
2. Change to the following directory:

   ```
   <ATG10dir>/home/localconfig/atg/dynamo/service/jdbc
   ```

3. Configure two FakeXADataSource properties files with the appropriate connection information:
   - `FakeXADataSource.properties`

     Set connection information for the agent schema.
   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

   For information about configuring `FakeXADataSource` files, see the *ATG Service Installation and Configuration Guide*.
4. Update the view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   ```

5. Start the agent server. After startup is complete, let the server run for five minutes so it can initialize new publishing workflows, then shut it down.
6. Copy:

   ```
   <ATG10dir>/home/servers/atg_svcagent_lockserver
   ```

   to this new directory:

   ```
   <ATG10dir>/home/servers/svcagent_import91to100
   ```

7. In this directory, configure the agent server as indicated in the following table.

   Note: Create directories and properties files as needed.

Path	Configuration
/atg/dynamo/service/jdbc	FakeXADataSource_agent.properties: Points to the agent schema FakeXADataSource_production.properties: Points to the production schema. FakeXADataSource.properties: Points to the agent schema
/atg/dynamo/service/jdbc	If using switch deployment: SwitchingDataSourceA.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchA SwitchingDataSourceB.properties: $class=atg.service.jdbc.MonitoredDataSource dataSource=\ /atg/dynamo/service/jdbc/FakeXADataSource_switchB FakeXADataSource_switchA and FakeXADataSource_switchB point to the switching schemas.
/atg/reporting/datawarehouse	FakeXADataSource.properties: Points to the data warehouse schema
/atg/reporting/datawarehouse/loaders	FakeXADataSource.properties: Points to the production schema
/atg/search/routing	RoutingSearchService.properties: cacheService=
/atg/epub/deployment	DeploymentFulfiller.properties: enable=false
/atg/commerce/catalog	ProductCatalog.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/commerce/pricing/priceLists	PriceLists.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production
/atg/assetregistry	AssetRegistry.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_agent
/atg/search/repository	RefinementRepository.properties: dataSource=\ /atg/dynamo/service/jdbc/JTDataSource_production

8. From the Service migration kit's `/scripts` directory, run this script:

   `run-agent-import.{bat|sh} svcagent_import91to100`

9. Repeat the previous step twice. On the third iteration, the script should execute error-free.

10. From the CSC migration kit's `/scripts` directory, run this script:

    `run-csc-knowledge-agent-import.{bat|sh} svcagent_import91to100`

Import Publishing Data

1. Change to the following directory:

   `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc`

2. Configure two FakeXADataSource properties files with the appropriate connection information:

   - `FakeXADataSource.properties`

     Set connection information for the publishing schema.

   - `FakeXADataSource_production.properties`

     Set connection information for the production core schema.

3. Update view mappings in the publishing schema by running the following scripts:

   ```
   <ATG10dir>/BIZUI/install/importBizui.{bat|sh}
   <ATG10dir>/BCC/install/importBCC.{bat|sh}
   <ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}
   <ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}
   <ATG10dir>/SiteAdmin/Versioned/install/importSiteAdmin.{bat|sh}
   <ATG10dir>/SiteAdmin/Search/install/importSiteAdminSearch.{bat|sh}
   <ATG10dir>/DCS-UI/install/importMerchandising.{bat|sh}
   <ATG10dir>/DCS-UI/Versioned/install/importDCSUIVersioned.{bat|sh}
   ```

4. Invoke the startSQLRepository utility as follows:

   UNIX:

   ```
   <ATG10dir>/home/bin/startSQLRepository -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>/Search10.0/SearchAdmin/common/data/initial/search
   admin-profile.xml
   ```

   Windows:

   ```
   <ATG10dir>\home\bin\startSQLRepository.bat -m DPS.InternalUsers -
   repository /atg/userprofiling/InternalProfileRepository -
   import <ATG10dir>\Search10.0\SearchAdmin\common\data\initial\search
   admin-profile.xml
   ```

5. Start the asset management server. Let the server run for several minutes after startup is complete so it can initialize new publishing workflows, then shut it down.

6. Copy this directory:

   `<ATG10dir>/home/servers/atg_publishing_lockserver`

   to:

   `<ATG10dir>/home/servers/svcpub_import91to100`

7. Create this directory:

<ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/dynamo/service/jdbc

8. In the new directory, configure the following properties files, as noted:
   - FakeXADataSource_production.properties

     Set connection information for the production core schema.
   - FakeXADataSource_agent.properties

     Set connection information for the agent data source schema.
9. Change to this directory:

   <ATG10dir>/home/servers/svcpub_import91to100/localconfig/atg/
10. Edit the following properties files, as noted:
    - /reporting/datawarehouse/FakeXADataSource.properties

      Set connection information for the reporting data warehouse schema.
    - /reporting/datawarehouse/loaders/FakeXADataSource.properties

      Set connection information for the production core schema.
    - /search/routing/RoutingSearchService.properties

      cacheService=
    - /epub/deployment/DeploymentFulfiller.properties

      enable=false
11. Run the following script in the Service10.0 migration kit, to import the ATG Self Service sites:

    ../migration/91to100sql/scripts/run-management-import.{bat|sh}

    Pass the name of the server directory you created in step 6 as a parameter when you invoke the script. For example:

    run-management-import.sh svcpub_import91to100

7 Migration Notes

All sections in this chapter focus on an ATG feature or application that has its own migration-related requirements, beyond the general requirements and tasks that are outlined earlier in this manual under Migration Steps.

This chapter contains the following sections:

- Platform Notes
- ATG Commerce Reference Store (CRS) Notes
- ATG Commerce Notes
- ATG Search Notes
- ATG Customer Intelligence Notes
- ATG Outreach Notes
- ATG Service Notes
- ATG Commerce Service Center (CSC) Notes
- ATG Service/Commerce Service Center Notes
- ATG Campaign Optimizer Notes

Platform Notes

This section covers the following platform migration tasks:

- Updating view mappings
- Migrating content targeters and content groups

Updating View Mappings

After you install ATG 10.0 and upgrade database schemas on production and asset-management servers, you must update view mapping data on the asset management servers in the following steps:

1. Configure data source properties.
2. Run import scripts.

Configure Data Source Properties

Before you import the new ATG 10.0 view mappings, edit the following files in `<ATG10dir>/home/localconfig/atg/dynamo/service/jdbc`

- `FakeXADataSource.properties`
- `FakeXADataSource_production.properties`

Run Import Scripts

Run the following import scripts to import new view mappings:

1. `<ATG10dir>/BIZUI/install/importBIZUI.{bat|sh}`
2. `<ATG10dir>/AssetUI/install/importAssetUI.{bat|sh}`
3. `<ATG10dir>/DPS-UI/install/importDPSUI.{bat|sh}`

   Note: If execution of this script generates errors, run it again.
4. Optional: `<ATG10dir>/DPS-UI/install/importDPSUIExamples.{bat|sh}`
5. If you have ATG Search installed:
   - Set the environment variable `SEARCH_VERSION` to 10.0
   - Run this script:

     `<ATG10dir>/Search10.0/SearchAdmin/install/create-search-admin.{bat|sh}`

For general information on view mapping, see the *Customizing Asset Display* chapter of the *ATG Content Administration Programming Guide*.

Migrating Content Targeters and Content Groups

Content targeters and content groups that you created in previous versions of ATG require migration to work correctly in ATG 10.0.

1. Edit each existing targeter and content group file before importing it into ATG 10.0. Add a `siteRulesets` property to each one and set it to the following value:

   `siteRulesets^=/Constants.null`
2. After you initialize your sites by performing a full ATG Content Administration deployment, open each targeter and content group in the ATG Business Control Center and save it.
3. Deploy and complete the project. Do not make any further modifications to these assets until this step is complete.

ATG Commerce Reference Store (CRS) Notes

ATG Commerce Reference Store was rewritten for the 10.0 platform. To use the 10.0 version, install and configure a new CRS instance as described in the *ATG Commerce Reference Store Installation and Configuration Guide*.

If an application is based on an earlier version of CRS, follow the standard migration paths described in this document.

ATG Commerce Notes

The steps required to migrate ATG Commerce depend on whether you use standard or custom catalogs, and whether you use ATG Content Administration (CA). Accordingly, this section is divided into four sections that address the following combinations:

- Standard catalogs / CA
- Custom catalogs / CA
- Standard catalogs / no CA
- Custom catalogs / no CA

This section also covers the following topics:

- Deprecating the standard catalog
- Standard catalog migration steps
- Updating catalogs
- Performing basic maintenance on catalogs
- Assigning catalogs and price lists to profiles
- Casting beginUsable and endUsable properties as java.util.Date

Prerequisites

Before you migrate ATG Commerce, complete the procedures described earlier under General Pre-Migration Tasks.

Standard Catalogs / CA

1. Run one of the following sets of schema migration scripts:
   - B2CCommerce core
     B2CCommerce publishing
   - B2BCommerce core
     B2BCommerce publishing
2. On the asset management server, migrate the standard catalog (see Deprecating the Standard Catalog).
3. Perform all general post-migration tasks as described earlier in this guide, including full deployment from CA.

Custom Catalogs / CA

1. Run one of the following sets of schema migration scripts:
   - B2CCommerce core
     B2CCommerce publishing
   - B2BCommerce core
     B2BCommerce publishing
2. On the asset management server, run the Catalog Update Service as described in Updating Catalogs.
3. Perform all general post-migration tasks as described earlier in this guide, including full deployment from CA.

Standard Catalogs / No CA

1. Run the migration scripts for either B2Ccommerce core or B2Bcommerce core.
2. On the production server, migrate the standard catalog (see Deprecating the Standard Catalog).

3. On the production server, run the Basic Maintenance service as described in Performing Basic Maintenance on Catalogs.

Custom Catalogs/ No CA

1. Run the migration scripts for either B2CCommerce core or B2BCommerce core .
2. On the production server, run the Catalog Update Service as described in Updating Catalogs.
3. On the production server, run the Basic Maintenance service as described in Performing Basic Maintenance on Catalogs.

Deprecating the Standard Catalog

Standard catalogs are deprecated in ATG 10.0. You should be aware of the following changes to ATG Commerce catalogs and modify your applications where necessary:

- The B2BCommerce module no longer depends on `DCS.CustomCatalogs`.
- In previous versions, the `ProductCatalog` repository name was `ProductCatalog` in module DCS, and `CustomProductCatalog` in module `DCS.CustomCatalog`. New customers who run DCS will get `ProductCatalog`. In order to preserve the old repository name, continue to run `DCS.CustomCatalog` by adding `DCS.CustomCatalogs` to startup commands and manifests, inasmuch as B2BCommerce no longer uses this module. The repository name change might affect slot behavior. You can modify slots to reference `repositoryPath` instead of `repositoryName`.
- The `DCS.CustomCatalogs` module is still available. Although now largely empty, the module serves the following objectives:
  - Configures the `ProductCatalog.repositoryName` property to `CustomProductCatalog` and overrides the DCS value of `ProductCatalog`.
  - Configures the ACC's `admins.xml` file to match the repository name.
  - Registers a `CustomCatalogs` module with the ModuleManager. A `DCS.CustomCatalogs.Versioned` module also remains that configures the `repositoryName` for `ProductCatalog_production` and `ProductCatalog_staging`

  By retaining this module, you can continue to reference the name `CustomProductCatalog` in your application. This module also lets you retain references to `DCS.CustomCatalog` on command lines and manifest files.
- `product.parentCatalog` is not writable in custom catalogs. Consequently, you might need to change bulk import data and programs.
- The DCS module now adds the `user.catalog` property in `userProfile.xml` before B2CCommerce is included through XML file combination. As a result, the catalog appears before `creditCards` and `secondaryAddresses`, which may cause XML file combination to yield startup errors.

Standard Catalog Migration Steps

If your application used standard catalogs, you must migrate it in the following steps:

1. Assemble the application with the Catalog Migration Service by including the module `DCS.CustomCatalogMigration`.
2. Run the catalog migration service.
3. If running Content Administration, deploy the migrated catalog.
4. If necessary, update your migrated application's `admins.xml` file in order to ensure that all catalog-related menus and items are visible in the ACC. Check that items that were deleted or renamed in the Core Commerce `admins.xml` file have corresponding updates in the application's `admins.xml` files.

Run the Catalog Migration Service

After you assemble the Catalog Migration Service in the application, launch the application and access the Catalog Migration Service as follows:

1. Access ATG Dynamo Administration on your production server.
2. Choose Commerce Administration.
3. From the Commerce Administration page, choose Standard to Custom Catalog Migration and perform the following tasks:
   - Enter the Nucleus path to your catalog repository component in the Catalog Repository To Migrate: field.
   - Enter a display name for the new custom catalog.
   - Click Migrate. Status updates display in the browser as the migration takes place.
4. After catalog migration is complete, exit ATG Dynamo Administration and stop the production server.

Deploy the migrated catalog

If running Content Administration, you deploy the migrated catalog as follows:

1. Reassemble the application,. This time, omit the module `DCS.CustomCatalogMigration`.
2. Start the asset management server and log into the BCC.
3. Under Content Administration, click Browse Projects and access the Catalog Migration project.
4. Advance this project to deployment.

Updating Catalogs

If you are migrating a Commerce application that already uses custom catalogs, you run the Catalog Update service. This service ensures that relationships are properly defined between a catalog and its categories, products, and SKUs. You run this service as follows:

1. Log into the ATG Dynamo Server Admin and navigate to the Commerce Administration page.
2. Under Catalog Maintenance, choose Catalog Update.
3. Click Start Process.
4. Click the Refresh Status link to update the status information and see when the process has completed.

Performing Basic Maintenance on Catalogs

The Basic Maintenance service executes standard batch maintenance services against the DCS catalog. You run this service as follows:

1. Log into the ATG Dynamo Server Admin and navigate to the Commerce Administration page.
2. Under Catalog Maintenance, choose Basic Maintenance.
3. Click Start Process.
4. If desired, you can view information, errors and warnings by clicking View Status on the Commerce Administration page.

Assigning Catalogs and Price Lists to Profiles

If you assign catalogs or price lists to user profiles by means of a custom pipeline servlet or a scenario, set the `useProfile` property in the `/atg/commerce/util/ContextValueRetriever` component to `true`. Otherwise the profile assignments may be overridden by default values set in the `CatalogTools` and `PriceListManager` components or by multisite configuration settings. Refer to *Assigning a Catalog to a User* in the *ATG Commerce Programming Guide* for more information.

Casting beginUsable and endUsable properties as java.util.Date

In ATG 10, `beginUsable` and `endUsable` properties of promotion repository items changed from `date` to `timestamp`. Application code that casts either property to `java.sql.Date` must be changed to `java.util.Date`.

Promotion Property Changes

Two changes to the promotions repository items may affect code you have written that refers to those items.

First, the repository definition of the `uses` property of the `promotion` repository item has changed from enumerated to an integer, and no longer handles the following string:

```
![CDATA[Infinite]]
```

If you need to import promotions with infinite uses, set the use property to `-1`.

Second, the name of the `promotion` property of the `promotion` claimable item descriptor has changed to `promotions`. The `promotions` property is a collection rather than a single promotion, and must be treated accordingly in your code.

ATG Search Notes

This section discusses migration tasks for ATG Search.

Deleting Old Indexes

ATG recommends that before you delete any old index files from the `deployshare` directory of your Search hosts before indexing with ATG Search 10.0.

Reindexing Search Content

After database schema migration is complete, you must reindex all of your search content. If an index includes items from the ATG Commerce catalog repository, complete the ATG Commerce migration process, including running the Catalog Maintenance Service, before reindexing.

Identifying Search Environments

ATG Search 10.0 uses a new mechanism, based on content labels and target types, for identifying the content to be searched for a given project. This mechanism is implemented by the `/atg/search/routing/DynamicTargetGenerator` component, which is enabled by default. If you want to instead continue to use the search environment name for search targeting, you can disable `DynamicTargetGenerator` by setting its `enabled` property to `false`. Note, however, that if `DynamicTargetGenerator` is disabled, you cannot use the ATG Merchandising Search Testing feature.

Querying Properties of a Product's Parent Categories

In ATG 9.1 and before, the default XML definition file for the `/atg/commerce/search/ProductCatalogOutputConfig` component included properties of the product's `parentCategory` item, such as `parentCategory.displayName`. The XHTML document for the product included the values for a single parent category (the product's default parent category).

In ATG 10.0, the definition file has been modified to instead include properties of the product's `parentCategories` item, such as `parentCategories.displayName`, so that the XHTML document includes values for all of the product's parent categories, not just the default one:

```
<item property-name="parentCategories" parent-property="childProducts"
    is-multi="true">
  <meta-properties>
    <property name="$repository.repositoryName" suppress="true"/>
    <property name="$itemDescriptor.itemDescriptorName" suppress="true"/>
    <property name="$repositoryId" type="string"
      output-name="parentCategory.$repositoryId"/>
  </meta-properties>
  <text-properties>
    <property name="displayName" output-name="parentCategory.displayName"/>
    <property name="keywords" filter="concat"
      output-name="parentCategory.keywords"/>
  </text-properties>
</item>
```

Notice that the `output-name` attribute is used to change the names of the indexed properties to use `parentCategory` rather than `parentCategories`. This was done so existing JSP code that displays properties of the `parentCategory` item continues to work.

Note, however, that if a product has multiple parent categories, a query will by default return only the properties of one of the parent categories. So, for example, `parentCategories.displayName` will return a single value, even though the XHTML document includes `displayName` values for multiple parent categories. Unlike in ATG 9.1, the category whose values are returned may not actually be the default category.

Typically, you want to return values of properties of all categories in `parentCategories`, not just one. To do this, add these properties to the `listableProperties` property of the `QueryRequest` component:

```
listableProperties+=parentCategory.$repositoryId,\
    parentCategory.displayName,parentCategory.keywords
```

The values of these properties then be returned as List values. Note that you may also need to change your JSP code to handle List values.

ATG Customer Intelligence Notes

This section covers the following topics:

- Schema Migration Prerequisites for Customer Intelligence
- Post-Migration Tasks for Customer Intelligence

Schema Migration Prerequisites for Customer Intelligence

Perform the following steps before migrating the database schemas for ATG Customer Intelligence data warehouse:

Perform the following tasks before migrating the database schemas for ATG Customer Intelligence data warehouse:

1. Complete all general pre-migration tasks as described earlier in this guide.
2. Make sure your production servers have logged all reporting data and that the log files have been rotated.

   Shutting these servers down gracefully completes all logging and will rotate all log files.
3. Make sure the data warehouse loader server has loaded all logged data into the data warehouse. Load logged data manually by invoking the `loadAllAvailable` method of each loader component (see Data Warehouse Loader Components).

Data Warehouse Loader Components

The following table lists the data warehouse loader components for each product. Invoke the `loadAllAvailable` method of each component to ensure all logged data is loaded into the ATG Customer Intelligence data warehouse.

Product	Loader components
Platform	`SiteVisitLoader`
ATG Commerce	`OrderSubmitLoader`

ATG Search	`SearchQueryLoader`
	`ViewContentLoader`
	`UseSearchEnvironmentLoader`

ATG Commerce Service Center	`CallEventLoader`
	`OrderSubmitLoader`
	`SubmitReturnLoader`

Product	Loader components

ATG Outreach	`CampaignAuditTrailLoader`
	`CampaignEnteredLoader`
	`CampaignLoader`
	`EmailClickLoader`
	`EmailOpenLoader`
	`InboundEmailLoader`
	`OptOutLoader`
	`OutboundEmailLoader`

Invoking the loadAllAvailable Method

To invoke the `loadAllAvailable` method of a data warehouse loader component:

1. Navigate to the loader component in the ATG Dynamo Administration application. The loader components are available at the following URL:

   `http://`*`server-name`*`:`*`port`*`/dyn/admin/nucleus/atg/reporting/datawarehouse/loaders/`
2. Follow the link to the `globalLockRequired` property and set it to `false`.
3. Follow the link to the `loadAllAvailable` method and choose to invoke it.

Post-Migration Tasks for Customer Intelligence

After you migrate the database schemas for ATG Customer Intelligence data warehouse, perform the following steps:

1. Move your version 9.1 installation of ATG Customer Intelligence aside. Retain all files in case they are needed for reference.
2. Install ATG Customer Intelligence version 10.0. See instructions in the *ATG Customer Intelligence Installation and Configuration Guide*.
3. Reproduce any changes that your organization made to the metadata model.

Note: The metadata model for version 10.0 of ATG Customer Intelligence has changed significantly. The migration tools for ATG 10.0 do not automatically merge your changes with the new metadata model. You might be able to use the synchronize projects feature of the IBM Cognos 8 Framework Manager application to replay some customizations to the metadata model. This application is a component of ATG Customer Intelligence. See information about synchronizing projects in the IBM Cognos 8 Framework Manager User Guide.

ATG Outreach Notes

To migrate an ATG Outreach 9.1 installation for use with ATG Outreach 10.0, complete the following steps:

1. Review and follow the procedures described in Migration Steps. Make sure you stop all running campaigns, as described in the *ATG Outreach Guide for Business Users*.

2. Complete the procedures for updating the database as described in the appropriate Database Migration section of this guide. Make sure you complete the steps for the ATG Outreach tables as well as the platform tables.
3. Install and configure ATG Outreach 10.0 as described in the *ATG Outreach Administration and Development Guide*.
4. Move any custom pages (for example, custom opt-in or opt-out forms) deployed in the Outreach Web application under `/ACO9.1/ACO/communication/common/j2ee-apps/Outreach.ear/Outreach.war` to `/ACO10.0/ACO/communication/common/j2ee-apps/Outreach.ear/Outreach.war`.

ATG Service Notes

In addition to database schema migration and other general tasks described elsewhere in this guide, you must perform a number of tasks for ATG Service migration. These are covered in the following sections:

- Schema Migration Prerequisites for ATG Service
- Post-Migration Tasks for ATG Service

Schema Migration Prerequisites for ATG Service

Perform the following tasks before migrating the databases for ATG Service from version 9.1 to 10.0:

1. Complete all general pre-migration tasks as described earlier in this guide.
2. Move service-related directories to the new installation from your ATG 9.1 installation.

Note: Migrating to version 10.0 converts your segment-based ATG Self Service sites into sites based on the new multisite functionality.

Move Service-Related Directories to the New Installation

Copy the directories listed in the following table from your ATG 9.1 installation into your ATG 10.0 installation.

Source Directory in <ATG9dir>	Destination Directory in <ATG10dir>
`/home/doc`	`/home/doc`
`/home/Publishing/versionFileStore`	`/home/Publishing/versionFileStore`
`/home/localconfig/atg/registry/RepositoryGroups`	`/home/localconfig/atg/registry/RepositoryGroups`
`/home/servers/atg_production_lockserver/PublishingAgent`	`/home/servers/atg_production_lockserver/PublishingAgent`
`/home/servers/atg_publishing_lockserver/Publishing/versionFileStore`	`/home/servers/atg_publishing_lockserver/Publishing/versionFileStore`
`/home/servers/atg_svcagent_lockserver/PublishingAgent`	`/home/servers/atg_svcagent_lockserver/PublishingAgent`

/home/servers/atg_svcagent_lockserver/Publishing/versionFileStore	/home/servers/atg_svcagent_lockserver/Publishing/versionFileStore
/home/servers/atg_dw_loader_lockserver/PublishingAgent	/home/servers/atg_dw_loader_lockserver/PublishingAgent

If you use the PublishingWebAgent, also copy <ATGWeb9dir>/home/PublishingWebAgent to <ATGWeb10dir>/home/PublishingWebAgent.

Post-Migration Tasks for ATG Service

After you finish migrating the database schema and data for ATG Service from version 9.1 to 10.0, you must perform the following tasks:

1. Start ATG service Servers.
2. Deploy from the Agent Server.
3. Deploy from the Asset Management Server.
4. Create the ATG Service Search Environment.
5. Copy Custom ATG Service Workflows.
6. If required, map brand-based sites to migrated sites.

Start ATG Service Servers

Start the following servers:

- production
- agent
- asset management
- data warehouse loader

Deploy from the Agent Server

1. From the BCC, open ATG Content Administration and choose Browse Projects.
2. Advance the following projects to deployment:
   - 91to100ServiceFramework
   - 91to100Option

Deploy from the Asset Management Server

1. from the BCC, open ATG Content Administration and choose Admin Console.
2. Edit the production site configuration by adding the following repository mapping:

   SiteRepository > SiteRepository_production
3. Perform a full deployment to the production site.
4. From the BCC, open ATG Content Administration and choose Browse Projects.
5. Advance the 91to100SelfServiceSites project to deployment.
6. If your existing ATG Self Service installation uses brand-based sites, you must map brand-based sites to migrated sites.

Create the ATG Service Search Environment

After completing the database migration tasks, create an ATG Search Live Indexing search environment. See information about configuring search environments in the *ATG Service Installation and Configuration Guide*.

1. Open the ATG Dynamo Server Admin at `http://hostname:port/dyn/admin/nucleus/atg/search/routing/LiveIndexingService/`
2. Click the Create New Environment link. A button appears for creating profile search environments.
3. Click the button to create the environments and enter the details of your search engine. You can either select the check box for your existing machine or enter the address of another machine.
4. Click the Environments link to display the current default environments. Two environments, a live indexing and a bulk indexing, are created for the profile process.

   Note: If you choose different environment names than the default `ATGProfile`, you must edit the `/atg/userprofiling/search/ProfileSearchConfiguration` component to reflect the search environment names and corresponding logical partition names.
5. Use the Environment page to add, delete, or administer your environments.
6. Open the component `/atg/userprofiling/search/ProfileOutputConfig` and execute the `bulkLoad` method.

Copy Custom ATG Service Workflows

Copy custom workflows from the ATG 9.1 server installation, in:

`localconfig/atg/registry/data/epubworkflows`

to the corresponding directory in the ATG 10.0 installation.

Map Brand-Based Sites to Migrated Sites

If your existing ATG Self Service installation uses brand-based sites, map the brand parameter for each site to the corresponding new site in version 10.0:

1. Examine the `/atg/multisite/SiteRepository` repository and find the site names that have been created based on your existing brands. The site name is generated from the first 40 characters of the brand name, minus spaces. See information about repositories in the *ATG Repository Guide*.
2. Configure the `/atg/svc/multisite/BrandRuleFiler` component of your production server. Edit or create the file `<ATG10dir>/home/server/servername/localconfig/atg/svc/multisite/BrandRuleFilter.properties`.

   Add the configuration property `brandToSiteIdMap`. Include a comma-separated list of site identifier and brand pairs as its value.
3. Configure the `/atg/svc/email/EmailLandingPage` component of your agent server. Edit or create the file `<ATG10dir>/home/server/servername/localconfig/atg/svc/email/EmailLandingPage.properties`.

   Add the configuration property `siteIdToBrandMap`. Include a comma-separated list of site identifier and brand pairs as its value.

The following example shows the /atg/svc/multisite/BrandRuleFiler configuration for two brands that are mapped to new site identifiers in a version 10.0 installation.

```
# localconfig/atg/svc/multisite/BrandRuleFilter.properties:
brandToSiteIdMap=MyNewSiteID=MyBrandX,MyOtherNewSiteID=MyBrandY
```

The following example shows the /atg/svc/email/EmailLandingPage configuration for two brands that are mapped to new site identifiers in a version 10.0 installation.

```
# localconfig/atg/svc/email/EmailLandingPage.properties:
siteIdToBrandMap=MyNewSiteID=MyBrandX,MyOtherNewSiteID=MyBrandY
```

ATG Commerce Service Center (CSC) Notes

In addition to database schema migration and other general tasks described elsewhere in this guide, you must perform a number of tasks for CSC migration. These are covered in the following sections:

- Schema Migration Prerequisites for CSC
- Post-Migration Tasks for CSC

Schema Migration Prerequisites for CSC

Perform the following tasks before migrating the databases for ATG Service from version 9.1 to 10.0:

1. Complete all general pre-migration tasks as described earlier in this guide.
2. Make sure versions 10.0 of CSC and ATG Knowledge are installed.
3. Move CSC-related directories to the new installation from your ATG 9.1 installation.

Move CSC-Related Directories to the New Installation

Copy the directories listed in the following table from your ATG 9.1 installation into your ATG 10.0 installation.

Source Directory	Destination Directory
<ATGWeb9dir>/home/PublishingWebAgent	<ATGWeb10dir>/home/PublishingWebAgent
<ATG9dir>/home/Publishing/versionFileStore	<ATG10dir>/home/Publishing/versionFileStore
<ATG9dir>/home/servers/atg_production_lockserver/PublishingAgent	<ATG10dir>/home/servers/atg_production_lockserver/PublishingAgent
<ATG9dir>/home/servers/atg_svcagent_lockserver/PublishingAgent	<ATG10dir>/home/servers/atg_svcagent_lockserver/PublishingAgent

Post Migration Tasks for CSC

After you have migrated your database schemas and data perform the following post-migration procedures.

1. Deploy to CSC non-versioned repositories.
2. Create CSC search environment.
3. Copy custom CSC workflows.

Deploy to CSC Non-Versioned Repositories

Note: The following procedure is required only if a publishing site is already set up on the Agent server. If no publishing site is set up on the Agent server, you can skip this procedure.

1. Start the Agent server and log onto the BCC Project Browser. The Project Browser should display the following projects:

   ```
   91to100ServiceFramework
   91to100Option
   csc91to100Option
   csc91to100ServiceFramework
   cscOnly91to100ServiceFramework
   ```

2. Check in the following projects:

   ```
   91to100ServiceFramework
   91to100Option
   csc91to100Option
   ```

3. Open the following project:

   ```
   css91to100ServiceFramework
   ```

4. Switch to the assets tab, and resolve a merge conflict on the `cmcHelpfulPanels` Panel Stack Definition:
   - Select Merge with Latest.
   - Select Use M y Version.
   - Check in the project.
5. Open the following project:

   ```
   cscOnly91to100ServiceFramework
   ```

6. Switch to the assets tab and resolve a merge conflict on the `WsRespondPanelStack` Panel Stack Definition:
   - Select Merge with Latest.
   - Select Use My Version. This removes the panel `communicateLinkedSolutionsPanel` from the `WsRespondPanelStack.panelIds`.
   - Check in the project.
7. Complete the workflow for each project by performing the following actions:
   - Approve and Deploy to Production
   - Verify Production Deployment

   Note: If you u switching data sources, perform a full deployment.

Create CSC Search Environment

After completing the database migration tasks (see the pertinent section for your database), you must create an ATG Search Live Indexing search environment to perform Profile and Order searches. CSC uses an embedded search method for customer profile and order searches that are included in the `DPS.Search.Index` and `DCS.Search.Order.Index` modules on each customer-facing and management server. For additional information, refer to the *ATG Commerce Service Center Installation and Programming Guide.*

1. Open the ATG Dynamo Server Admin on the agent server.
2. Open the Nucleus component `/atg/search/routing/LiveIndexingService`.
3. Create the ATGProfile search environment by entering the details of your search engine.
4. Create the ATGOrder search environment by providing the search engine information.
5. Open the component `/atg/userprofiling/search/ProfileOutputConfig` and execute the `bulkLoad` method.
6. Open the component `/atg/commerce/search/OrderOutputConfig` and execute the `bulkLoad` method.

Copy Custom CSC Workflows

Copy custom workflows from the ATG 9.1 server installation, in:

`localconfig/atg/registry/data/epubworkflows`

to the corresponding directory in the ATG 10.0 installation.

ATG Service/Commerce Service Center Notes

In addition to database schema migration and other general tasks described elsewhere in this guide, you must perform a number of tasks for ATG Service/CSC migration.

Schema Migration Prerequisites for ATG Service/CSC

Perform the following tasks before migrating the databases for ATG Service from version 9.1 to 10.0:

1. Complete all general pre-migration tasks as described earlier in this guide.
2. Move directories to the new installation from your ATG 9.1 installation.

Move Directories to the New Installation

Copy the directories listed in the following table from your ATG 9.1 installation into your ATG 10.0 installation.

Source directory	Destination directory
`<ATGWeb9dir>/home/PublishingWebAgent`	`<ATGWeb10dir>/home/PublishingWebAgent`
`<ATG9dir>/home/Publishing/versionFileStore`	`<ATG10dir>/home/Publishing/versionFileStore`

<ATG9dir>/home/servers/atg_production_lockserver/PublishingAgent	<ATG10dir>/home/servers/atg_production_lockserver/PublishingAgent
<ATG9dir>/home/servers/atg_svcagent_lockserver/PublishingAgent	<ATG10dir>/home/servers/atg_svcagent_lockserver/PublishingAgent

Post Migration Tasks for ATG Service/CSC

After you have migrated your database schemas and data perform the following post-migration procedures.

1. Deploy to ATG Service/CSC Non-Versioned Repositories.
2. Create the ATG Service/CSC Search Environment.
3. Copy Custom ATG Service/CSC Workflows.

Deploy to ATG Service/CSC Non-Versioned Repositories

The startSQLRepository scripts that you ran during data migration imported changes into versioned repositories. You must deploy these changes to the non-versioned repositories so they are available in Service Center.

1. Start the agent, asset management, and production servers.
2. On the asset management server, log onto the BCC. Open ATG Content Administration and choose Admin Console.
3. Edit the production site configuration by adding the following repository mapping:

   `SiteRepository` > `SiteRepository_production`
4. On the asset management server, log onto the BCC Project Browser. The Project Browser should display the following projects:

   ```
   91to100ServiceFramework
   91to100Option
   csc91to100ServiceFramework
   csc91to100Option
   ```
5. Check in the following projects:

   ```
   91to100ServiceFramework
   91to100Option
   csc91to100Option
   ```
6. Open the following project:

   `css91to100ServiceFramework`
7. Switch to the assets tab, and resolve a merge conflict on the `cmcHelpfulPanels` Panel Stack Definition:
   - Select the Merge with Latest option
   - Select the Use M y Version option.
   - Check in the project.
8. Open the following project:

   `cscOnly91to100ServiceFramework`

9. Switch to the assets tab and resolve a merge conflict on the `WsRespondPanelStack` Panel Stack Definition:
   - Select the Merge with Latest option.
   - Select the Use My Version option. This removes the `communicateLinkedSolutionsPanel` panel from the `WsRespondPanelStack.panelIds`.
   - Check in the project.
10. Complete the workflow for each project by performing the following actions:
    - Approve and Deploy to Production
    - Verify Production Deployment

    Note: If you u switching data sources, perform a full deployment.

Create the ATG Service/CSC Search Environment

After completing the database migration tasks, create an ATG Search Live Indexing search environment. See information about configuring search environments in the *ATG Service Installation and Configuration Guide*.

1. Open the ATG Dynamo Server Admin at `http://`*`hostname`*`:`*`port`*`/dyn/admin/nucleus/atg/search/routing/LiveIndexingService/`
2. Click the Create New Environment link. A button appears for creating profile search environments.
3. Click the button to create the environments and enter the details of your search engine. You can either select the check box for your existing machine or enter the address of another machine.
4. Click the Environments link to display the current default environments. Two environments, a live indexing and a bulk indexing, are created for the profile process.

   Note: If you choose different environment names than the default `ATGProfile`, you must edit the `/atg/userprofiling/search/ProfileSearchConfiguration` component to reflect the search environment names and corresponding logical partition names.
5. Use the Environment page to add, delete, or administer your environments.
6. Open the component `/atg/userprofiling/search/ProfileOutputConfig` and execute the `bulkLoad` method.
7. Open the component `/atg/commerce/search/OrderOutputConfig` and execute the `bulkLoad` method.

Copy Custom ATG Service/CSC Workflows

Copy custom workflows from the ATG 9.1 server installation, in:

`localconfig/atg/registry/data/epubworkflows`

to the corresponding directory in your ATG 10.0 installation.

ATG Campaign Optimizer Notes

To migrate an ATG Campaign Optimizer 9.1 installation for use with ATG 10.0, complete the following steps:

1. Make sure you stop all running tests, as described in the *ATG Campaign Optimizer User Guide*.
2. Review and follow the general upgrade procedures described in Migration Steps.
3. Complete the procedures for updating the database as described in the Database Schema Migration section of this guide. Make sure you complete the steps for the ATG Campaign Optimizer tables as well as the platform tables.
4. Install and configure ATG Campaign Optimizer 10.0 as described in the *ATG Campaign Optimizer User Guide*.