

[Previous Topic](#)[Next Topic](#)

# Data preservation on cloning target instances

New York

[Subscribe](#)

...

Data preservers prevent specified target instance data from being overwritten.

## Data preservers

Sometimes it is necessary to preserve some data on an instance targeted for cloning. For example, if the target is a MID Server, you must not overwrite the MID Server [ecc\_agent] table. Preserved data is stored on the target instance before cloning begins and is restored on the target instance after cloning.

You must define data preservers on the source instance. Defining them on the target instance does not preserve the data.

Data preservers typically preserve system settings and themes, such as:

- Instance-specific authentication settings
- Bookmark [sys\_ui\_bookmark]
- Recent Selection [sys\_ui\_recent\_selection]
- User Preference [sys\_user\_preference]

Do not use data preservers to transfer large sets of data, such as user groups. If you must preserve table data, such as users, groups, and roles, consider exporting the records to a file and importing them after cloning.

## Data preservers for Multi-SSO

The system automatically creates the necessary data preservers for cloning when you activate Multiple Provider Single Sign-On integration.

## Certificate

X.509 Certificates [sys\_certificate]

None

## Core Instance Properties

System Property [sys\_properties]

- [OR] [Name] [is one of] [glide.authenticate.external, glide.authenticate.external.logout\_redirect]
- [OR] [Name] [starts with] [com.snc.integration.saml\_esig]
- [OR] [Name] [is one of] [glide.smtp.port, glide.smtp.auth, glide.smtp.encryption]
- [OR] [Name] [starts with] [glide.authenticate.multisso]
- [OR] [Name] [is] [glide.authenticate.sso.redirect.idp]

## Digest Properties

Digest Properties [digest\_properties]

None

## Identity Providers

Identity Providers [sso\_properties]

None

## SAML2 Update1 Properties

SAML2 Update1 Properties [saml2\_update1\_properties]

None

**Note:** Although you can modify these data preservers, it is good practice not to. The Digest Properties [digest\_properties], Identity Providers [sso\_properties], and SAML2 Update1 Properties [saml2\_update1\_properties] tables are required for multiple source, single sign-on (SSO) to function properly. If multiple source, single sign-on is disabled on the target instance, you can safely remove all three data preservers. Remove them at the same time, as the system terminates the clone with an error message when you attempt to clone with one or two of these tables being preserved.

## Data preservers for SAML

---

Preserving SAML SSO-related settings can prevent the target instance from using the wrong issuer and audience parameters when making authentication requests to your IdP. To preserve SAML settings, create data preservers for the following tables:

- System Property [sys\_properties]: to preserve SAML properties.
- X.509 Certificates [sys\_certificate]: to preserve SAML certificates.
- User [sys\_user]: to preserve SAML users.

You also need to preserve properties and users that are involved in SAML.

## Preservation of unpublished applications

---

You cannot use data preservers to save unpublished applications. Instead, application developers must choose how they want to preserve unpublished applications.

The cloning process does not preserve version differences for applications in development. Instead, the system clone only copies the application version installed on the source instance onto the target instance. If the target instance had a development version of the same application, the application will be editable after the clone, but it will be at whatever version was installed on the source instance. If the application was missing from the source instance, the cloning process deletes the application from the target instance.

## Create a data preserver

---

Data preservers maintain specified data on a target instance.

### Before you begin

Role required: clone\_admin or admin

### About this task

Sometimes, preserving certain data on a target instance is desirable. For example, when using a MID Server, you can avoid overwriting the MID Server [ecc\_agent] table. Preserved data is stored in a dynamically generated list on the target instance before the clone and restored on the target instance after the clone is complete. You define data preservers on the source instance.

Data preservers are primarily intended to preserve system settings and themes, such as instance-specific authentication settings. Do not use data preservers to transfer large sets of data, such as user groups. If you must preserve table data such as users, groups, and roles, consider exporting the records to a file and importing it after the clone is complete.

Consider whether to preserve the data in the following tables.

- Bookmark [sys\_ui\_bookmark]
- Recent Selection [sys\_ui\_recent\_selection]
- User Preference [sys\_user\_preference]

If you set a data preserver on a table where the source instance has more records than the target instance, the data preserved on the target instance also includes the additional records from the source instance.

For example, assume that the data preserver is already in place.

- In the source instance, the sys\_temp table contains 100 records.
- In the target instance, the sys\_temp table contains 20 records.

After the clone, the sys\_temp table in the target instance contains 100 records.

- The 20 records in the target sys\_temp table are preserved successfully (per the data preserver specification).
- The source sys\_temp table brings over the remaining 80 records to the target sys\_temp table.

To resolve this issue and to preserve only the records in the target table, create an exclude table record for the target table, in addition to setting the data preserver on the source table.

Configure preservers on the source instance. If, however, a table you want to preserve is on the target but not on the source, use the following procedure on the target and then import the clone\_data\_preserver file to the source.

## Procedure

1. On the source instance, navigate to **System Clone > Preserve Data**.
2. Click **New**.
3. Enter the table label as the **Name**, for example, User Preference for the [sys\_user\_preference] table.
4. Select the **Table** to be preserved.
5. Select the **Theme** check box if the data being preserved is a UI property.
6. Define the data to be preserved using the Condition Builder .

You can use conditions to define particular records you want to preserve during a clone. For example, to only preserve particular system properties, you can add conditions for each property name you want to preserve.

Clone Data Preserver - Application Theme Properties

Name: Application Theme Properties

Theme: ☒

Table: System Property [sys\_properties]

Conditions: Add Filter Condition Add \*OR\* Clause

Name starts with css AND OR X

or Name is glide.ui.css.name X

or Name is glide.product.description X

or Name is glide.product.description X

or Name is glide.product.image X

or Name is glide.product.icon X

Update Delete

If the clone from backup fails for some reason, the clone process fails over to the legacy clone engine. The legacy clone engine cannot preserve data from extended tables, relationships, hierarchies between tables, and dot-walked queries. You may want to reschedule a system clone or manually transfer data in such cases.

## 7. Click **Submit**.

If you want to delete the data preserver later, make sure not to modify or delete the following data preserver records:

- Core Instance Properties
- Semaphores
- Email Accounts

## Preserve SAML properties

If you want a clone target instance to keep its existing SAML integration, you must edit the **Core Instance Properties** data preserver to include the SAML properties.

### Before you begin

Role required: admin

## Procedure

1. Navigate to **System Clone > Preserve Data**.
2. Select **Core Instance Properties**.
3. Add the following **Conditions**.
  - [OR] [Name] [is one of] [glide.authenticate.external, glide.authenticate.external.logout\_redirect, glide.authenticate.failed\_requirement\_redirect]
  - [OR] [Name] [starts with] [glide.authenticate.sso.saml2]
  - [OR] [Name] [starts with] [com.snc.integration.saml\_esig]

Clone Data Preserver - Core Instance Properties

Name: Core Instance Properties

Theme: ☐

Table: System Property [sys\_properties]

Conditions:

- Add Filter Condition Add "OR" Clause
- Name is glide.sys.schedulers AND OR X
- or Name starts with glide.db X
- or Name is glide.ui.max.transaction X
- or Name starts with glide.email X
- or Name starts with glide.pop3 X
- or Name is instance\_id X
- or Name is instance\_name X
- or Name starts with glide.installation X
- or Name is one of glide.authenticate.external, glide.authenticate.external.logout\_redirect, glide.authenticate.failed\_requirement\_redirect X**
- or Name starts with glide.authenticate.sso.s X
- or Name starts with com.snc.integration.san X

**Note:** Ensure the **Theme** check box is cleared so these properties are preserved regardless of whether you preserve the instance theme.

4. Click **Update**.

## Preserve unpublished applications during a system clone

Application developers must manually save a copy of each application currently in development prior to cloning over their development instance.

### Before you begin

- Role required: admin
- Write access to the application record
- A source control repository

### About this task

The cloning process does not preserve version differences for applications in development. Instead, the system clone only copies the application version installed on the source instance onto the target instance. If the target instance had a development version of the same application, the application will be editable after the clone, but it will be at whatever version was installed on the source instance. If the application was missing from the source instance, the cloning process deletes the application from the target instance.

### Procedure

1. Use one of these actions to preserve the application on the clone target instance.

#### Version differences between instances

---

**The application version on the clone target instance is different than the source instance version.**

---

Export each application from the clone target instance. Choices include:

- (Recommended) Link each application to a source control repository.

**Note:** If the application is already linked to a source control repository, commit the latest version to it.

- Publish each application to an update set.

---

**The application is only available on the clone target instance.**

---

**The application version on the clone target instance is the same as the source instance.**

---

None. The system clone process will copy this application version onto the target instance during the clone.

---

2. Request a system clone of the source instance over the target instance.

For example, clone your production instance over your development instance.

3. After the clone process finishes, log in to the clone target instance.
4. If you saved each application to a source control repository, use one of these actions to retrieve them from the source control repository.

### **Retrieve applications from a source control repository**

---

#### **The application was previously installed on the source instance.**

---

Apply remote changes from source control repository.

---

#### **The application was never installed on the source instance.**

---

Import the application from source control repository.

---

5. If you saved each application to an update set, use one of these actions to retrieve them from the update set.

### **Retrieve applications from an update set**

---

#### **The application was previously installed on the source instance.**

---

- a. Delete the application version cloned from the source instance.
  - b. Load the update set containing the current application version.
- 

#### **The application was never installed on the source instance.**

---

Load the update set containing the current application version.

---

### **Result**

The applications previously in development are available for further development on the clone target instance.

## **Example: Preserve the Marketing Events application**

---

Suppose your company previously created version 1.0 of a custom application called Marketing Events. You have already published version 1.0 of the Marketing Events application to the application repository and installed it on your production instance.

Over time, users have submitted enhancement requests for the application, and you decide to develop version 2.0 of the Marketing Events application on a non-production instance to address these requests. As development nears completion, you want to update your non-production instance to the latest copy of production for some comprehensive testing.



Since you previously used a source control integration to develop version 1.0 of the Marketing Events application, you have already linked the Marketing Events application to a source control repository. You commit version 2.0 of the Marketing Events application to the source control repository.

You schedule a clone of the production instance over the development instance. After completion, you log in to the development instance and see that it has version 1.0 of the Marketing Events application, because that was the version installed on the source instance.

Since the application was already installed on the source instance, you apply remote changes from the source control repository to receive the latest application version. The development instance now has version 2.0 of the Marketing Events application available for further development and testing.

---

**Last updated:** July 25, 2019

**Tags:** New York, Now Platform Administration, Data Management

Send Feedback

Previous Topic

Next Topic