HOPEX ADMINISTRATION

HOPEX V2

Information in this document is subject to change and does not represent a commitment on the part of MEGA International.

No part of this document is to be reproduced, transmitted, stored in a retrieval system, or translated into any language in any form by any means, without the prior written permission of MEGA International.

© MEGA International, Paris, 1996 - 2016

All rights reserved.

HOPEX is a registered trademarks of MEGA International.

Windows is a registered trademark of Microsoft Corporation.

The other trademarks mentioned in this document belong to their respective owners.

.

Administrator Guide

ABOUT HOPEX ADMINISTRATION

HOPEX products can be used on a stand-alone workstation or in a configuration including dozens of users.

This guide is for the person responsible for repository and user administration. When there are only a few users, administration is usually done by one of the users. In such cases, it mainly consists of carrying out regular backups and reorganizing repositories when required.

Chapters ("Managing Users", page 9 and "Managing Repositories", page 73) cover most administration requirements of a structure with only a few users. In a structure with many users, the administrator must respond to more specific requirements, which are detailed in the following chapters.

Most of the functions described here can be used by the repository and user administrator, whatever the products enabled through his/her security key. However, certain functions are only available with specific technical modules (**HOPEX Power Studio**, **HOPEX Power Supervisor**, or **HOPEX Collaboration Manager**). These are indicated by a note in the text.

HOPEX administration is managed from the **HOPEX Administration** application (Windows Front-End) "Administration.exe" or from the **HOPEX Administration** desktop (Web Front-End). Some actions can also be performed from the main **HOPEX** application (Windows Front-End) "Hopex.exe" or from certain **HOPEX** desktops (Web Front-End).

The **HOPEX** administration applications (Windows Front-End and Web Front-End) are designed for administrators **HOPEX**: they are used to manage environments, repositories, users, etc.

A **HOPEX** installation can contain a large number of environments, repositories and users. To facilitate their management, **HOPEX Administration** provides all the key concepts and tools required for their administration in a unified hierarchical structure.

PRESENTATION OF THIS GUIDE

This guide concerns **HOPEX** administration from the **HOPEX Administration** application (Windows Front-End).

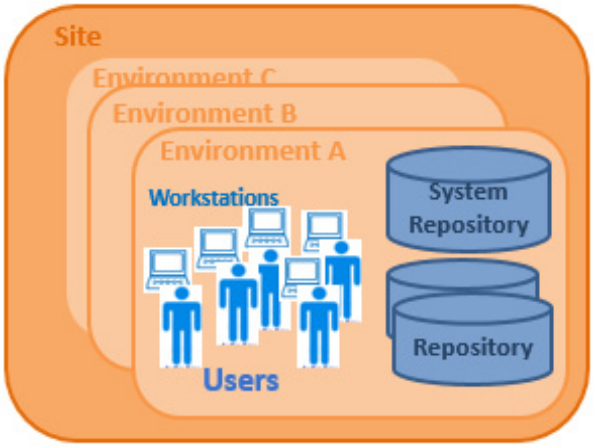
> ☛ *To perform **HOPEX** administration tasks from the **Administration HOPEX** desktop (Web Front-End), see the HOPEX Administration - Supervisor Web guide.*

It includes the following chapters:

- "Access HOPEX Administration", page 5: how to access **HOPEX Administration**.
- "Managing Users", page 9: to create and define users, and user profiles.
- "Managing Repositories", page 73: to create, save, restore, check, reorganize, copy and move repositories.
- "Managing workspaces", page 133: principle of private workspaces, dispatch and refresh private workspaces, and lock management.
- "Managing Environments", page 167: to create, back up, restore, check, copy and move an environment.
- "Managing Events", page 185: to supervise events with the **HOPEX Server Supervisor** tool.
- "Managing objects", page 205: Advanced administration functions available with:
  - the **HOPEX Power Studio** technical module to extract objects
  - the **HOPEX Power Supervisor** technical module to manage user interface (UI) access, query isolated objects, compare objects in two repositories.
- "Managing Data Writing Access", page 249: to set up management of organized projects in the form of data writing access, from the **HOPEX Power Supervisor** technical module.
- "Managing Data Reading Access", page 275: to install a confidentiality strategy using a reading access diagram and access areas.
- "Command File Syntax", page 307: description of the syntax used in command files.
- "Frequently Asked Questions", page 337: answers to frequently asked questions and some administration tips.
- "Glossary", page 343: definition of the main terms used in this guide.

HOPEX Structure

Some basic knowledge is required to understand the architecture and operation of **HOPEX**.



HOPEX is organized in four tiers:

- ***site***
  A site groups together everything that is shared by all **HOPEX** users on the same local network: the programs, standard configuration files, online help files, standard shapes, workstation installation programs, and version upgrade programs.
- ***environment***
  An environment groups a set of users, the repositories on which they can work, and the system repository. It is where user private workspaces, users, system data, etc. are managed.
- ***workstation***
  A workstation is defined for each computer connected to the environment. A workstation contains programs and a configuration file that allow you to use **HOPEX** on that machine.
- ***user***
  A user is a person (or person group) with a login. A user:
  - has a specific workspace in each repository.
  - can connect to a repository from all workstations connected to the environment in which this repository is referenced.
  - has a specific configuration and is authorized to access specific product functions and repositories in the environment.

ACCESS HOPEX ADMINISTRATION

The **Administration** application (Windows Front-End) is the **HOPEX** administration application accessible from the Windows desktop. This application contains the tools required to manage users, repositories, environments and private workspaces. It is used to manage users (individuals, business roles and profiles, access to GUIs, writing access as well confidentiality using reading access management, servers, LDAP servers) and repositories (workspaces, locks, repository snapshots, Scheduler).

The **Administration** desktop (Web Front-End) is the **HOPEX** administration application available via an internet browser. This application is used to manage users (persons, person groups, business roles, profiles, LDAP servers), repositories (workspaces, locks, repository, repository snapshots) and UI accesses. This application also provides access to tools (Excel import/export, Import/Export of command files, Scheduler, Exchange Rate) and is used to manage person skills.

☛ *The **Administration** desktop (Web Front-End) is described in the Web HOPEX Administration - Supervisor Guide.*

The points covered here are:

✓ "Connecting to HOPEX Administration", page 6
✓ "Connecting to an Environment", page 7

CONNECTING TO HOPEX ADMINISTRATION

The **Administration** application is the **HOPEX** *administration* application accessible from the Windows desktop.

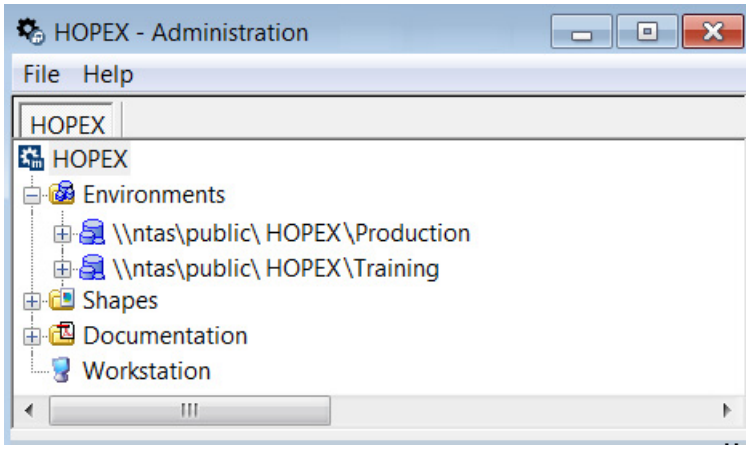
To access the **HOPEX Administration** application **(Windows Front-End)**:

❯ Double-click the "Administration.exe" file .

☛ *The administration icon is created during setup of an administrator workstation.*

The **HOPEX Administration** main window opens.

☛ *Environments preceded by a red icon are not accessible.*



HOPEX Administration is presented in the form of a navigation tree:

- containing the elements you can manage:
  - *site*
  - *environments*
  - *repositories*
- enabling access to:
  - the list of shapes used in **HOPEX**
  - technical characteristics of the **HOPEX** installation on your workstation.

CONNECTING TO AN ENVIRONMENT

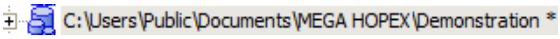
To connect to an environment:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. Expand the **Environments** folder.
   The list of referenced environments appears.

   ☛ *The asterisk that may appear after the environment name means that you must compile the metamodel and/or the technical data, see "Compiling an Environment", page 177.*

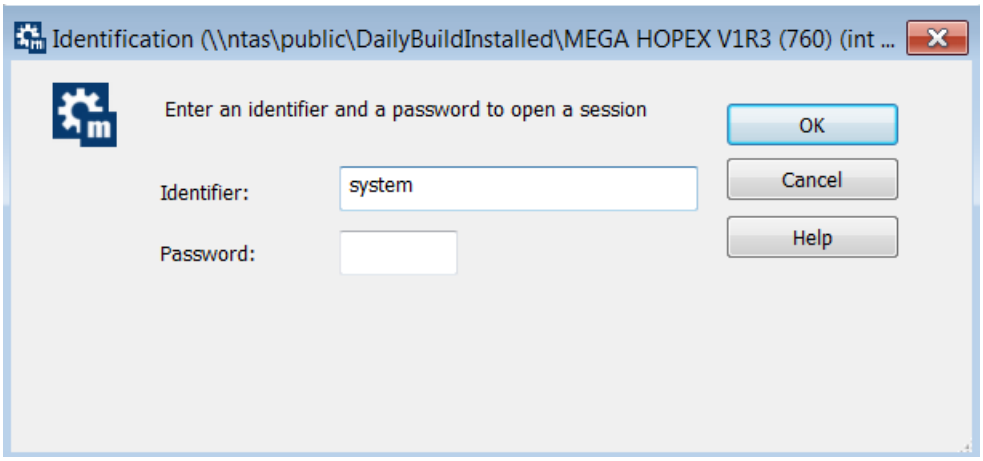


3. Right-click the environment you want to connect to and select **Open**.
   The environment connection dialog box appears.
4. In the **Identifier** field, enter the identifier of an *Administrator*.

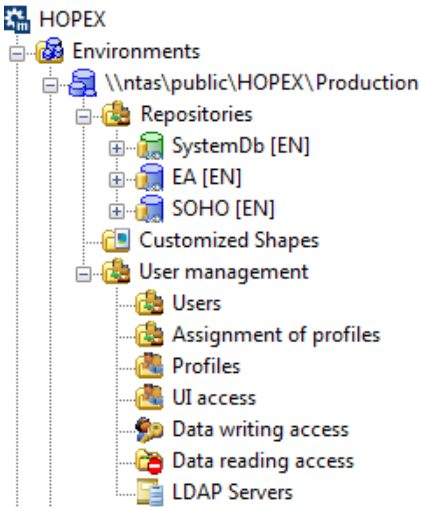
   ☛ *Administrator and Mega users own administration rights. The Administrator user identifier is System. The Mega user identifier is Mega.*

5. (optional) Type the user **Password**.

   ☛ *By default, Administrator and Mega users do not have a password.*



6. Click **OK**.
   The content of the environment folder is available.



This environment folder contains the folders:

- **Repositories** containing repositories referenced in the environment
- **Customized Shapes** containing **HOPEX** shapes customized by the user. They are stored in the Mega_Usr folder of the environment.
- **User Management** to manage:

the *users*

- the *Assignment of profiles* of each user
- the *profiles*
- the *UI Access*
- the *Data Writing Access areas*
- the *Data Reading Access areas*
- the *LDAP Servers*.

MANAGING USERS

The **Administration** application is equipped with tools required for user management.

This chapter explains how to create and manage *users* and how to define and modify their characteristics.

☛ *Managing person groups is a functionality available in **HOPEX** (Web Front-End). For more information on managing this functionality, see the Web HOPEX Administration - Supervisor guide.*

The following points are covered here:

- ✓ "Actions to be Performed to Define a User", page 10
- ✓ "Introduction to Profile Management", page 13
- ✓ "Introduction to User Management", page 26
- ✓ "Managing Profiles", page 33 (available with **HOPEX Power Supervisor**)
- ✓ "Access to User Management", page 44
- ✓ "Managing Users", page 52
- ✓ "Managing user options", page 60
- ✓ "Authentication in HOPEX", page 63

Actions to be Performed to Define a User

To define a *user*, some actions are compulsory, while others are only necessary depending on **HOPEX** options selected, and others are optional.

A user is a person with a login.

See:

- "Before defining a user: profile concepts", page 10
- "Compulsory Actions to be Performed to Define a User", page 11
- "Optional Actions to be Performed to Define a User", page 11
- "Other Actions to Set or Manage a User", page 12

Before defining a user: profile concepts

To connect to **HOPEX** a user select the profile with which he/she wants to work.

This profile defines:

- the products accessible

  If a user already has restricted access rights to products (see "Viewing Login Characteristics", page 51), the products accessible to this user are at the intersection of values of the Command Line attribute of the user login and profile.

- the applications and the desktops which the user is going to access.

  See "Connection schema: Person - Profile - Application", page 14.

  See "Connection schema: Person - Profile - Application - Desktops", page 15.

- the user's access rights to UIs (permissions)

Assigning a profile to a person defines:

See "Assigning a Profile to a Person", page 41.

- the repository concerned by the assignment
- the person's access rights to repositories with this profile assignment
- (optional, with read-only access to the repository) connection repository snapshot
- (optional) the validity period of the assignment

Compulsory Actions to be Performed to Define a User

To create a user who can connect to **HOPEX** you must:

- define the name of the person

  ☛ *See "Creating Users", page 52.*

- define the login of the user

  A person must have a login to be able to connect to HOPEX.

  ☛ *See "Defining the Login of a Person", page 56.*

  ☛ *The login of the user is created automatically on creation of the person, see "Creating Users", page 52 (If necessary, see "Creating the Login of a Person", page 55).*

  ☛ *The login status must be active so the person can connect, see "Inactive user (Status)", page 31.*

- (recommended) define the e-mail address of the person

  ☛ *See "Defining a Person", page 53.*

  ☛ *The e-mail address is necessary, for example, for distributing documents, receiving notifications and questionnaires, or when a Web user lost his/her password.*

- assign a profile to a person

  The user must have at least one profile assigned to be able to connect to HOPEX.

  ☛ *See "Assigning a Profile to a Person", page 41.*

Optional Actions to be Performed to Define a User

According to the selected options you must:

- (recommended) define the e-mail address of the person

  ☛ *The e-mail address is necessary, for example, for distributing documents, receiving notifications and questionnaires, or when a Web user lost his/her password.*

- (where writing access management is activated) define the writing access area of the user

  ☛ *See "Defining a Person", page 53.*

- (where reading access management is activated) define the reading access area of the user

  ☛ *See "Defining a Person", page 53.*

Other Actions to Set or Manage a User

You can:

- define the telephone number and initials of the person

  ☛ *See "Defining a Person", page 53.*

- restrict user access to certain products

  ☛ *The products accessible to this user are at the intersection of the values of the **Command Line** attribute of the user login and profile.*

  ☛ *See "Defining the Login of a Person", page 56.*

  ☛ *See "Configuring a Profile", page 37.*

- modify user authentication mode

  ☛ *See "Defining the Login of a Person", page 56.*

- make the user inactive.

  ☛ *See "Defining the Login of a Person", page 56.*

  ☛ *See "Preventing User Connection", page 57.*

INTRODUCTION TO PROFILE MANAGEMENT

Managing users involves managing profiles. A user connects to **HOPEX** with a specific profile that determines the **HOPEX** application to which the user connects and the desktops with which it is associated.

See:

- "Description of a profile", page 13
- "Connection schema: Person - Profile - Application", page 14
- "Connection schema: Person - Profile - Application - Desktops", page 15
- "Connection schema: Person - Profile - Application - Specific Desktops", page 17
- "The Administration profiles provided", page 19
- "Profile Properties", page 22

Description of a profile

The description of a profile includes:

- the definition of the profile
- the definition of the profile assignment to a person

☛ *See "Profile Properties", page 22.*

Definition of the profile

A profile defines the function of a person or person group in the enterprise

```
Example: Application Portfolio Manager, Enterprise
Architect).
```

The profile defines:

- the products accessible

☛ *See "Products accessible on the license (Command Line)", page 22.*

If a user already has restricted access rights to products (see "Viewing Login Characteristics", page 51), the products accessible to this user are at the intersection of values of the Command Line attribute of the user login and profile.

- the applications and the desktops which the user is going to access.

☛ *See "Connection schema: Person - Profile - Application", page 14.*

☛ *See "Connection schema: Person - Profile - Application - Desktops", page 15.*

☛ *See "Defining applications accessible to profile users", page 38 and "Defining application desktops accessible to profile users", page 39.*

- the user's access rights to UIs (permissions)

☛ *See "Managing UI Access", page 235.*

Profile assignment

You must assign each person at least one profile so that this person can connect to **HOPEX**.

☛ *By default, no profiles is assigned to a person.*

Assigning a profile to a person defines:

☛ *See "Assigning a Profile to a Person", page 41.*

- the repository concerned by the assignment
- the person's access rights to repositories with this profile assignment
- (optional, with read-only access to the repository) connection repository snapshot
- (optional) the validity period of the assignment

Connection schema: Person - Profile - Application

To connect to **HOPEX**, a person must have:

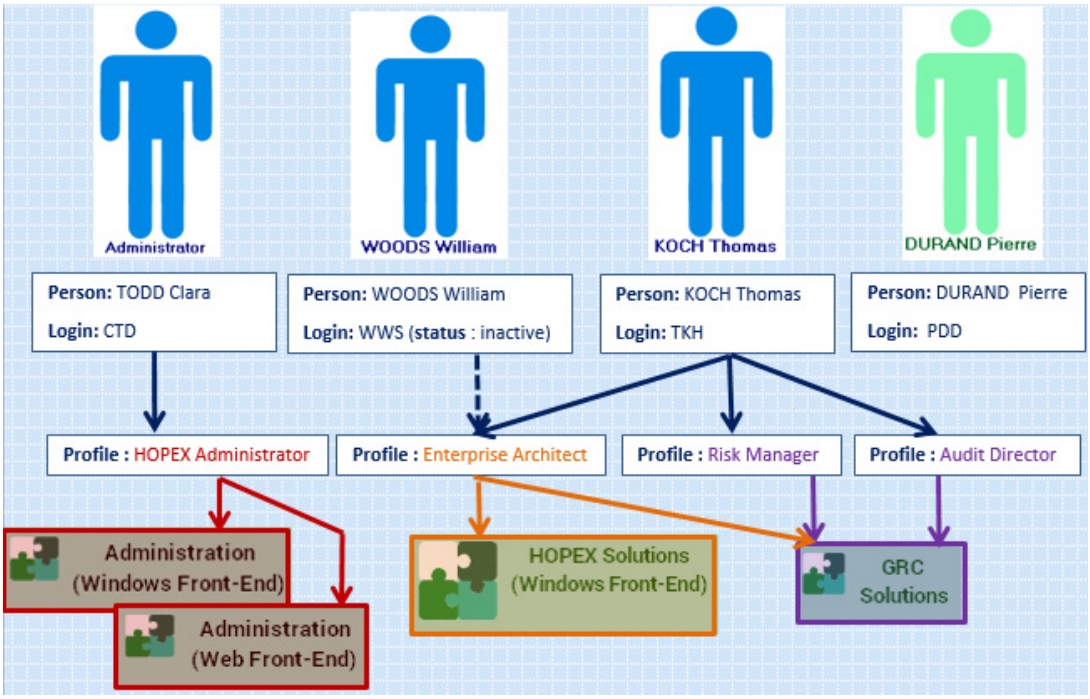
- a login

  ☛ *See "Creating Users", page 52.*

  ☛ *The login status must be active so the person can connect, see "Inactive user (Status)", page 31.*

- at least one profile.

  ☛ *See "Assigning a Profile to a Person", page 41.*



In the above example:

- Clara TODD has a login and the **HOPEX Administrator** profile assigned: she can connect to **Administration** applications (Windows Front-Endand Web Front-End).
- William WOODS has the **Enterprise Architect** profile assigned but the status of his login is inactive:
  he cannot connect to **HOPEX Solutions (Windows Front-End)** or **GRC Solutions** applications.
- Thomas KOCH has a login and the **Enterprise Architect**, **Risk Manager** and **Audit Director** profiles assigned:
  he can connect to **HOPEX Solutions (Windows Front-End)** and **GRC Solutions** applications.
- Pierre DURAND has a login but does not have an assigned profile:
  he cannot connect to **HOPEX**.

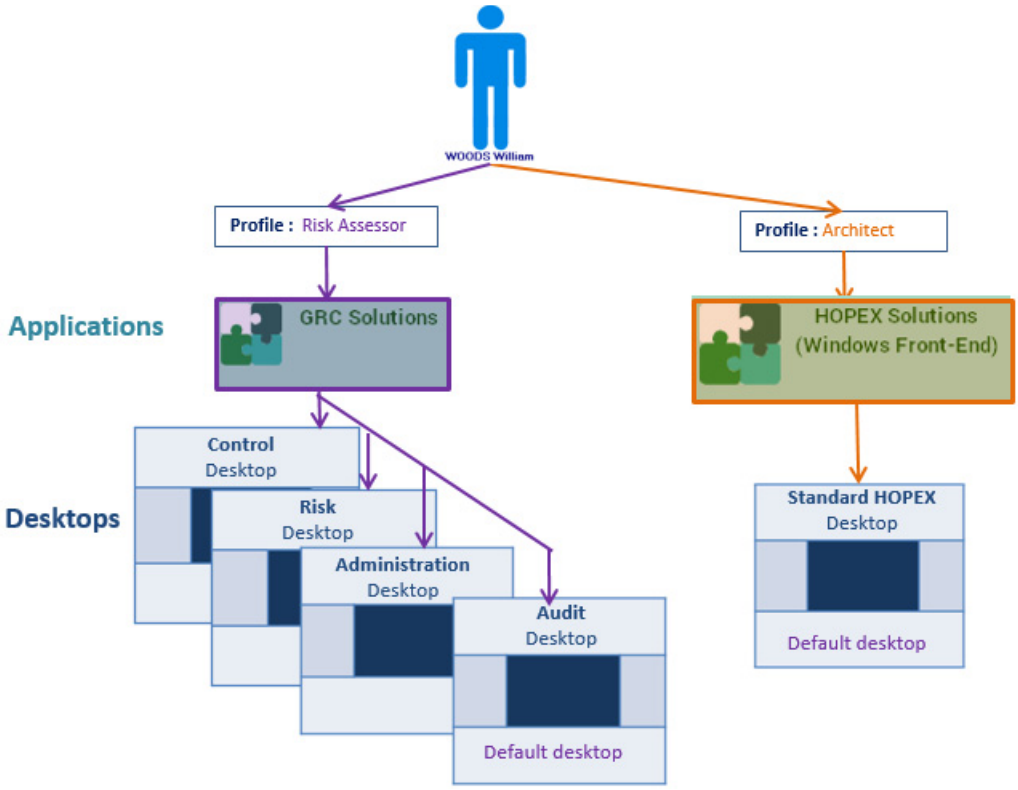
Connection schema: Person - Profile - Application - Desktops

So that a user of a profile can connect to an application, you must connect this application to the profile concerned.

All desktops connected to the application are then accessible.

☛ *To modify a profile supplied by **MEGA**, **MEGA** recommends you create a new profile; see "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.*

☛ *To enable access to only certain desktops of the application, see "Connection schema: Person - Profile - Application - Specific Desktops", page 17.*



Example:

The **GRC Solutions** application is linked to the **Risk Assessor** profile and the **HOPEX Solutions (Windows Front-End)** application is linked to the **Architect** profile.

No desktops of the **GRC Solutions** or **HOPEX Solutions (Windows Front-End)** application are directly linked to the **Risk Assessor** and **Architect** profiles.

The user William WOODS, who has the **Risk Assessor** and **Architect** profiles, has access to all the **GRC Solutions** and **HOPEX Solutions (Windows Front-End)** applications.

Connection schema: Person - Profile - Application - Specific Desktops

A user can connect to applications via customized desktops according to actions to be performed.

If an application contains several desktops, you can specifically define application desktops that are accessible to the concerned profile. To do this, you must connect to the profile:

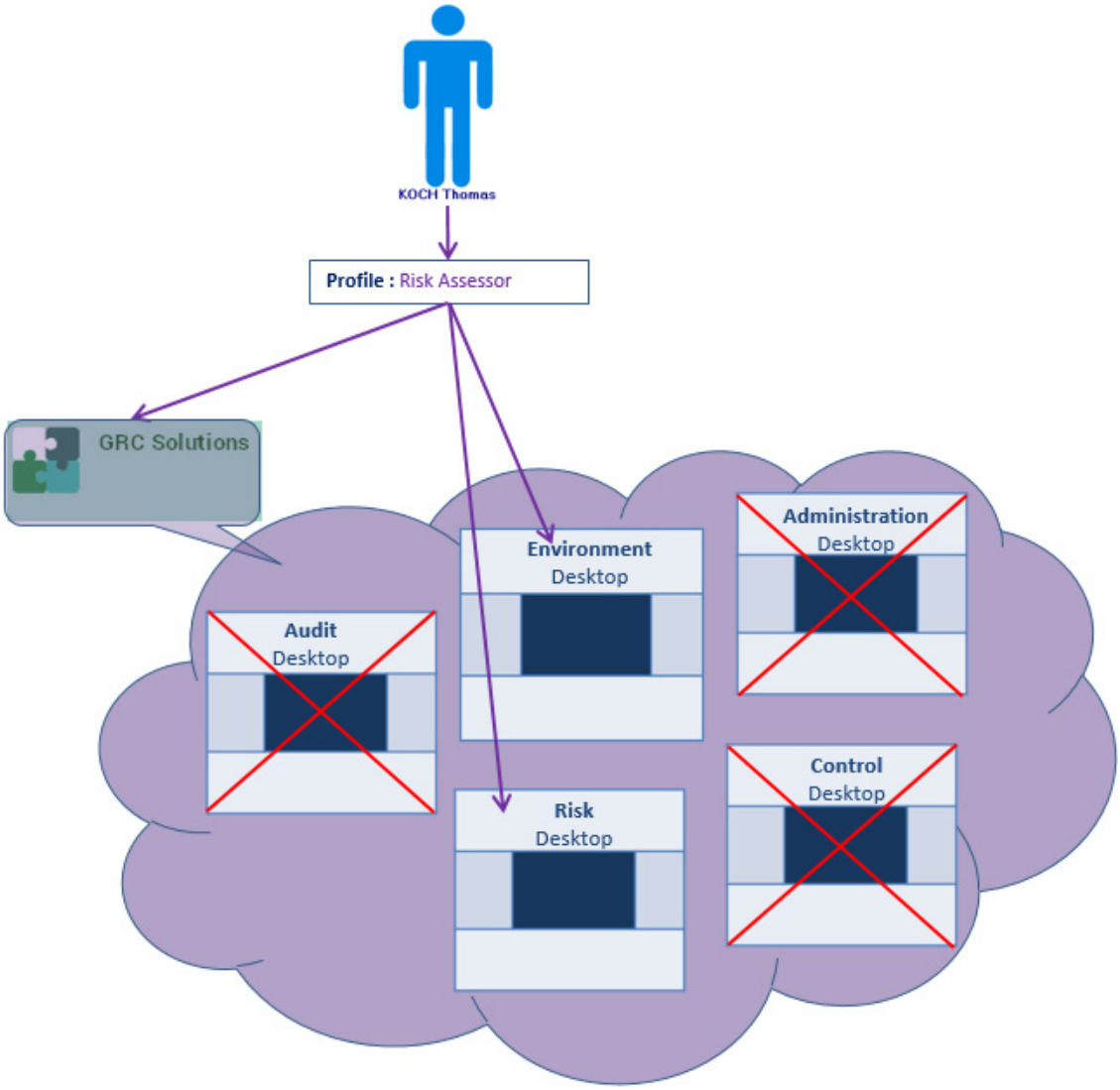
> ☛ *To modify a profile delivered by **MEGA**, you must have rights to modify **HOPEX** data. Alternatively, you can create a new profile, see "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.*

- the application containing the desktops.

  > ☛ *See "Defining applications accessible to profile users", page 38.*

- the desktops to which you want users of the profile to connect.
  The application desktops that are not connected to the profile are not accessible to users of the profile.

  > ☛ *To enable access to only certain desktops of the application, see "Defining application desktops accessible to profile users", page 39.*



Example:

The **Risk Assessor** profile is connected:

- to the **GRC Solutions** application which contains in particular the "Audit", "Environment" and "Administration" desktops.

- to the "Risk" and "Environment" desktops of the "GRC Solutions" application.

User Thomas KOCH with the **Risk Assessor** profile can connect only to the "Risk" and "Environment" desktops of the **GRC Solutions** application. The "Audit", "Control" and "Administration" desktops are not authorized.

The Administration profiles provided

Administration profiles are supplied at installation with defined rights and access to applications:

When several users with an Administration profile connect to ***HOPEX Administration*** at the same time, certain actions, such as user management, are exclusive.

These profiles are dedicated to:

- global Administration, with exclusive access to **Administration** applications (Windows Front-End and Web Front-End):
  - **HOPEX Administrator**

    ☛ *See "HOPEX Administrator profile", page 19.*
  - **HOPEX Administrator - Production**

    ☛ *See "HOPEX Administrator - Production profile", page 21.*
- Administration (Web Front-End), with exclusive access to the **Web Administration** desktop:
  User Management Web Administrator

  ☛ *See "User Management Web Administrator profile", page 21.*
- functional Administration (Web Front-End), with access to the **Web Administration** desktop and to Solution-specific desktops:
  <Solution name> functional Administrator

  Example: **ITPM functional Administrator** gives access to Environment, ITPM, and Administration desktops.

  ☛ *See "Functional Administrator profile of a Solution", page 21.*

If needed you can modify the rights and access to applications defined on these profiles.

☛ *See "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35 and "Configuring a Profile", page 37.*

HOPEX Administrator profile

☛ *When several users with an Administration profile connect to **HOPEX Administration** at the same time, certain actions are exclusive (e.g.: user management).*

In the **Administration** application (Windows Front-End), the **HOPEX Administrator** profile allows, in particular, to manage :

> ☛ *For information on the HOPEX Administrator profile in the Web Administration desktop, see HOPEX Administration - Supervisor guide.*

- environments

  ☛ *See "Managing Environments", page 167.*

- repositories

  ☛ *See "Managing Repositories", page 73.*

- Profiles

  ☛ *See "Managing Profiles", page 33.*

- users (Persons and Logins)

  ☛ *See "Managing Users", page 52.*

- workspaces

  ☛ *See "Managing workspaces", page 133.*

- objects

  ☛ *See "Managing objects", page 205.*

- Permissions

  ☛ *See "Managing UI Access (Permissions)", page 233.*

- LDAP servers

  ☛ *See "LDAP Authentication", page 66.*

It also allows to perform tasks linked to:

- file import/export:
  - Excel

    ☛ *See the **HOPEX Common Features** guide, "Exchanging Data With Excel" chapter.*

  - XMG/MGL/MGR

    ☛ *See "Importing command files", page 103.*

    ☛ *See "Exporting Objects", page 206.*

  - Visio

    ☛ *See **HOPEX Common Features - Visio Import**.*

- Snapshot use

  ☛ *To create a repository snapshot, see the **HOPEX Collaboration Manager - Repository Snapshots** guide.*

- Scheduler use

  ☛ *See **HOPEX Power Studio - Scheduler** guide.*

(Web specific) It also allows to manage:

- *User groups* (Person groups and Logins)

  🕮 *(Web Front-End specific) A person group groups persons in a group. These persons share the same connection characteristics.*

  ☛ *For information on user groups, see the HOPEX Administration - Supervisor Web guide.*

- Business Roles

  ☛ *For more information on business roles, see the **HOPEX Administration - Supervisor Web** guide.*

HOPEX Administrator - Production profile

The **HOPEX Administrator - Production** profile is the equivalent of the **HOPEX Administrator** profile without permission management rights.

User Management Web Administrator profile

The **User Management Web Administrator** profile allows, in particular, to manage:

> ☛ *For information on the User Management Web Administrator, see the **HOPEX Administration - Supervisor Web** guide.*

- users (Persons and Logins)
- *User groups* (Person groups and Logins)

  > 🕮 *(Web Front-End specific) A person group groups persons in a group. These persons share the same connection characteristics.*
  >
  > ☛ *For information on user groups, see the **HOPEX Administration - Supervisor Web** guide.*

- Business Roles

  > ☛ *For more information on business role management, see the **HOPEX Administration - Supervisor Web** guide.*

It also gives access to management of:

- LDAP servers
- Locks.

Functional Administrator profile of a Solution

Each **<Solution name> functional Administrator** gives access to the Administration desktop and to Solution-specific desktops.

> ☛ *For information on the functional Administrator of a Solution, see the HOPEX Administration - Supervisor Web guide.*

From an administration point of view, the **<Solution name> Functional Administrator** profile allows, in particular, to manage:

- Profiles
- users (Persons and Logins)
- *User groups* (Person groups and Logins)

  > ☛ *For information on user group management, see the HOPEX Administration - Supervisor Web guide.*

- Business Roles

  > ☛ *For more information on business role management, see the **HOPEX Administration - Supervisor Web** guide.*

It also gives access to:

- the Scheduler use

  > ☛ *See **HOPEX Power Studio - Scheduler** guide.*

- LDAP server management.

Profile Properties

A profile enables definition of the same connection parameters and rights to a set of users.

☛ *See "Introduction to Profile Management", page 13.*

☛ *To assign a profile to a person, see "Assigning a Profile to a Person", page 41.*

☛ *To manage profiles, see "Managing Profiles", page 33.*

Name

The **Name** of a profile can comprise letters, figures and/or special characters.

Products accessible on the license (Command Line)

The **Command Line** field enables definition of products that can be accessed by users with the current profile.

Format of the command is:

/RW'<accessible Product A code>;<accessible Product B code>;<...>'

For example: You have licenses for products **HOPEX Business Process Analysis, HOPEX IT Portfolio Management and other HOPEX products.** To authorize only **HOPEX Business Process Analysis** and **HOPEX IT Portfolio Management** modules to users that have this profile, enter:

/RW'HBPA;APM'

☛ *To found out the product code, see "Access your list of available products", page 341 or on line documentation: **Concepts > Products**.*

If a user already has access rights restricted by the Command Line attribute on his/her Login (see "Viewing Login Characteristics", page 51), the products accessible to this user

are at the intersection of values of the Command Line attribute of the user login and profile.

		Profile 1	Profile 2
	Command line	**RW:/'APM'**	**none**
User A	RW:/'APM;HBPA'	user A has access to **HOPEX IT Portfolio Management**	user A has access to: **HOPEX IT Portfolio Management** and **HOPEX Business Process Analysis**
User B	RW:/'HBPA'	user B cannot access any product	user B has access to **HOPEX Business Process Analysis**
User C	none	user C has access to **HOPEX IT Portfolio Management**	user C can access all products for which he/she has the license (**HOPEX IT Portfolio Management** and **HOPEX Business Process Analysis**)

*Restrictions on products for users and profiles that have licenses for **HOPEX IT Portfolio Management** and **HOPEX Business Process Analysis.***

Assignable

The **Assignable** attribute defines if the profile is assignable to a Login or not. Certain profiles are created to aggregate other profiles.

☺ *This attribute enables filtering of profiles and improves visibility of profiles to be assigned.*

☛ *The default value is "No".*

Administrator profile

Only the user whose current profile has the **Administrator Profile** attribute with value "Yes" can:

- grant administrator profile to another user.
- declare a profile as administrator.
  That is, specify value "Yes" for the **Administrator Profile** attribute of any profile.

The default value of **Administrator Profile** is "No".

Profile display

A profile is provided by default at connection when it is not included in another profile.

The **Profile Display** attribute defines when the profile is provided at connection:

- "always": the profile is provided at connection even if it is included in the definition of another profile,

  ☛ *See "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.*

- "If not included in another profile" (default value): the profile is provided at connection only if it is not included in another profile.

Profile status

The **Profile Status** attribute is used to define the profile as inactive if necessary.

_GUIName

The **_GUIName** attribute enables definition of the profile name display in the interface.

MetaPicture

The **MetaPicture** attribute enables customization of the icon representing the current profile.

Working Environment Template

The working environment template defines the appearance of the desktop.

Profile used

The **Profile used** tab contains the profiles used to define the current profile by aggregation of these profiles.

☛ *See "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.*

Persons

The **Persons** tab lists all the persons who are connected to the current profile.

Available applications

The **Available Applications** tab is used to define the applications to which the current profile gives access.

☛ *See "Defining applications accessible to profile users", page 38.*

Available desktops

The **Available Desktops** tab is used to define a **Working Environment Template**, which gives access to a specific desktop.

In cases where a **Working Environment Template** is not defined, the **Available Desktops** tab is used to restrict the desktops to which the current profile gives access. By default all the desktops connected to the application are accessible.

☛ *To restrict the desktops accessible, see "Defining application desktops accessible to profile users", page 39.*

Terminology

The **Terminology** enables associating a terminology to the profile.

☛ *See "Associating a terminology with a profile", page 39.*

Available types

The **Available Types** tab enables definition of the specific objects available for the profile:

- Document category
- Business Document Pattern
- Report DataSet Definition
- Widget

☛ *See "Defining the object types available for a profile", page 40.*

Introduction to User Management

☛ *Only a user with Administrator type profile has management rights. In particular, he/she is the only user who can modify user characteristics.*

User management in the **Administration** application involves the following concepts:

- *users*:

  🕮 *A user is a person with a login.*

  - *persons*

    🕮 *A person is defined by his/her name and electronic mail address.*

  - *logins*

    🕮 *A Login uniquely defines a user or user group. It can be assigned to only one Person or Person Group.*

- *profiles*

  🕮 *A profile defines what a person can see or not see and do or not do in tools, and how he/she sees and can do it. The profile defines options, access rights to repositories and products, read/write and read-only rights on objects.*

- permissions:
  - *object UI access*

    🕮 *Object UI access defines user rights on creation, reading, update, and deletion on these objects and their tools. By default, object UI accesses have value *CRUD (C: create, R: read, U: update, D: delete, *: default value).*

  - *general UI access*

    🕮 *General UI access defines if tools are available or not. By default, general UI accesses have value *A (A: Available, *: default value)*

For information on:

- managing *data writing access*, see "Managing Data Writing Access", page 249
- managing *data reading access*, see "Managing Data Reading Access", page 275

The following points are detailed here:

- introduction:
  - "Users Delivered", page 27
  - "User: Definition", page 28
- properties:
  - "Person Properties", page 28
  - "Login Properties", page 30
- access:
  - "Accessing User Management and UI Access Management Folders", page 44
- characteristics:
  - "Viewing Person Characteristics", page 50
  - "Viewing Login Characteristics", page 51

Users Delivered

By default, at installation the following are created in the environment:

- persons indispensable to the system:
  - ***Administrator***, with Login "System"

    ☛ *The "Administrator" user cannot be deleted. It has no profile (it has all rights) and no password is assigned at installation.*

    ☛ *The "Administrator" user can create a first user with the "HOPEX Administrator" profile to manage repositories and users.*
  - ***MEGA Agent***, with Login "SysMA"

    ☛ *The "MEGA Agent" user cannot be deleted. The "MEGA Agent" user is used by the system to manage workflows. It has no profile (it has all rights) and no password is assigned at installation.*
- a person given by way of example:
  - ***Mega***, with Login "Mega"

    ☛ *The "Mega" user can be deleted (not recommended). The "Mega" user has the "HOPEX Administrator" profile, which allows to manage repositories and users. No password is assigned to "Mister Guide" at installation.*

User: Definition

For each environment, a user has:

- personal characteristics defined by his/her **Person**.

  ☛ *see "Viewing Person Characteristics", page 50.*

- a **login** which defines his/her connection identifier, his/her status and his/her authentication **HOPEX** mode. The login can also restrict the accessible products.

  ☛ *see "Login Properties", page 30.*

- a **user code** which enables naming of user associated files, for example the work repository.

  ☛ *see "Login Properties", page 30.*

- at least one **profile** assigned that determines the products (restricted by the products defined for the user login), applications, desktops, and repositories to which the user has access as well the access rights to UIs (permissions).
  By default the user does not have an assigned profile.

  ☛ *see "Profile Properties", page 22.*

  ☛ *see "Managing Profiles", page 33.*

  ☛ *see "Assigning a Profile to a Person", page 41.*

- **options**

  ☛ *see "Managing Options", page 323.*

Only a user with **HOPEX Administrator** profile (or with equivalent rights) can configure and modify user properties.

☛ *see "HOPEX Administrator profile", page 19.*

Person Properties

☛ *For information on a user, see "User: Definition", page 28*

☛ *To consult properties of a person, see "Viewing Person Characteristics", page 50.*

Name

The name of the person can include name and first name. It can comprise letters, figures and/or special characters. Format of the name of the person is free. It is recommended that the same format be used for all persons.

```
Example: DURAND Pierre
```

Image

You can download an image in .ico, .bmp, .gif or .png format up to a size of 30 MB.

E-mail address

The person e-mail address is useful, for distribution of reports (MS Word) for example.

It is mandatory for password change in Web mode and for receipt of questionnaires for example.

```
Example: pdurand@mega.com
```

Telephone number and initials

The telephone number and initials of the person are optional.

```
Example: +33102030405 / DP
```

Data language

The **Data language** attribute of the person is specific to Web applications. It enables definition of a specific data language for this user. It is the language in which data names are displayed by default, in case several languages are available.

☛ *By default, the data language is defined in the environment options for all users at installation (Options/Installation/Web application) via the **Data language** option.*

☛ *To define interface language, see "Managing Languages", page 334.*

Default library

The **Default Library** attribute enables attachment of objects created by the person in a library, when the creation context does not permit this.

Person reading access area and reading access area at creation

☛ *Information related to the reading access area are only visible when the **Activate reading access diagram** is selected in **Options** of the **Repository** of the **Environment**.*

Certain objects or modeling projects may be confidential or contain data (costs, risks, controls) that should be visible only to authorized users.

The **HOPEX** administrator can hide objects corresponding to this confidential data.

To implement a data confidentiality policy, objects must be organized in distinct sets. Each set of objects is a ***reading access areas***.

Each user is associated with a reading access area that determines objects the user can see. A user can only see objects located in his/her own reading access area or in the lower reading access areas.

☛ *For more details on reading access areas, see "Managing Data Reading Access", page 275.*

Person writing access area and writing access area at creation

☛ *Writing access management is available with the **HOPEX Power Supervisor** technical module.*

A writing access area is assigned to an object to protect it from inadvertent modifications. At creation, an object takes the writing access area of the user that creates it.

By default, a new user is connected to the only writing access area: "Administrator".

There is a hierarchical link between writing access areas: a user can only modify an object when he/she has the same writing access level as this object or a higher writing access area level.

☛ *See "Managing Data Writing Access", page 249.*

Login

The login of a person is a unique character string uniquely identifying the person that can connect. The person without a login cannot connect to ***HOPEX***.

```
Example: pdurand, pdd
```

☛ *For more details, see "Login Properties", page 30.*

Belongs to a Person Group

☛ *Managing person groups is a functionality available in **HOPEX** (Web Front-End). For more information on managing this functionality, see the Web HOPEX Administration - Supervisor guide.*

Login Properties

To:

- create a login, see "Creating the Login of a Person", page 55 or "Creating Users", page 52.
- consult login characteristics, see "Viewing Login Characteristics", page 51.
- configure a login, see "Defining the Login of a Person", page 56.

User code

The **User Code** is the short identifier (upper case, maximum length 6 characters) of the user that serves as the basis for private workspace naming.

This code is defined at creation of the user. To ensure data consistency, it should not be modified.

```
Example: PDD
```

Login Holder

The login holder is the person associated with the login.

```
Example: DURAND Pierre
```

Inactive user (Status)

Login status can be used to make a user inactive (value: Inactive). The user no longer has access to repositories, but trace of his/her actions is retained. The user can be easily reactivated (value: Active).

When you delete a user from the repository, the commands connected to this user become orphans and you lose part of the history saved in logs. With Inactive status, the user no longer has access to repositories, but the history of commands connected to the user is kept in logs.

Products accessible on the license (Command Line)

The **Command Line** field enables restriction of access of a user or profile to available products.

For more details, see "Products accessible on the license (Command Line)", page 22.

***If* a user is connected to a profile and the user and profile each have access to products restricted by the Command Line attribute, the products accessible to the user are at the intersection of the values of the Command Line attribute of the user and profile.**

Authentication mode

Default value of the **Authentication Mode** parameter on the user login is inherited at user creation from the **Authentication Mode** option defined in the options of the environment (**Options**/**Installation**/**User Management**).

See "Defining Default Authentication Mode", page 64.

Authentication mode of a user is by checking the user password. Available authentication modes are:

- **MEGA**
  Passwords are managed and stored in the **HOPEX** repository.
  This is default authentication mode.
  For more details, see "Authentication in HOPEX", page 63.
- **Windows**
  Passwords are managed and stored in Windows. This allows the user connected to Windows to be recognized automatically when he/she is

connected to **HOPEX** (Windows Front-End), not requiring entry of his/her password.

> ☛ ***Attention**: to connect to a **HOPEX (Web Front-End)** application, the user must enter his/her password.*

The list of users in your **HOPEX** environment is automatically synchronized with the list of users defined in your Windows network.

This authentication mode corresponds to unique authentication (SSO).

> ☛ *For more details, see "Windows Authentication", page 65.*

- **LDAP**
  Passwords are managed and stored in the LDAP server of the enterprise. The directory configuration is stored in options.

  The HOPEX user is authenticated at the LDAP server level.

  > ☛ *For more details, see "LDAP Authentication", page 66.*

- **Custom**
  This authentication is managed by an external authentication module or SSO. This authentication mode is specific to Web applications connection: users with this authentication mode cannot connect to **HOPEX** (Windows Front-End).

  > ☛ *See the technical article **Web connection overloading and configuration EN**.*

Windows identifier

> ☛ *This field only appears when the **Authentication Mode** is "Windows", see "Authentication mode", page 31.*

The **Windows Identifier** of a user enables connection of a **HOPEX** user to a Windows user, see "Associating a Windows user with a HOPEX user manually", page 65.

The user connected to Windows is automatically identified when connecting to **HOPEX** (Windows Front-End), without requiring a password. Windows handles user authentication and assures the security system.

> ☛ *To connect to a **HOPEX** application (Web Front-End), the user must enter his/her password.*

LDAP server

> ☛ *This field only appears when the **Authentication Mode** is "LDAP", see "Authentication mode", page 31.*

The **LDAP Server** is the server with which the **HOPEX** user is authenticated in LDAP authentication mode.

This server contains the LDAP directory in which the **HOPEX** user is registered.

MANAGING PROFILES

☛ *Profile management is only available with the **HOPEX Power Supervisor** technical module.*

Profiles are managed in the **HOPEX** administration applications: the **HOPEX Administration** application (Windows-Front-End) or the **HOPEX Administration** desktop (Web Front-End).

🕮 *A profile defines what a person can see or not see and do or not do in tools, and how he/she sees and can do it. The profile defines options, access rights to repositories and products, read/write and read-only rights on objects.*

☛ *See "Introduction to Profile Management", page 13.*

The following points are detailed here:

- "Introduction to Profiles", page 34
- "Creating a Profile", page 34
- "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35
- "Viewing Profile Characteristics", page 36
- "Configuring a Profile", page 37
- "Assigning a Profile to a Person", page 41
- "Performing a Mass Profile Assignment to Persons", page 42
- "Deleting a Profile", page 42

You can also:

- copy a profile
- explore a profile
- manage a profile
  - export a profile
  - compare and align profiles

To:

- modify profile options

  ☛ *see "Managing Options", page 323.*
- manage metamodel filters at profile level

  ☛ *see "Managing UI Access (Permissions)", page 233.*

Introduction to Profiles

A profile defines the function, in the business sense, of a person or person group in the enterprise

```
Example: Application Portfolio Manager, Enterprise
Architect).
```

A profile enables definition of the same connection rights to a set of users:

- applications accessible

  ☛ *See "Defining applications accessible to profile users", page 38.*

- desktops accessible

  ☛ *See "Defining application desktops accessible to profile users", page 39.*

- access rights to UIs (permissions).

  ☛ *See "Managing UI Access (Permissions)", page 233.*

- the same options.

  ☛ *See "Viewing Profile Characteristics", page 36.*

- access rights restricted to certain products

  ☛ *See "Products accessible on the license (Command Line)", page 31.*

  If a user already has access rights restricted by the Command Line attribute on his/her Login (see "Viewing Login Characteristics", page 51), the products accessible to this user are at the intersection of values of the Command Line attribute of the user login and profile.

You must assign each person at least one profile so that this person can connect to a **HOPEX** application.

☛ *By default, no profiles is assigned to a person.*

The assignment of a profile defines the following for each person:

- the repository concerned by the assignment
- the repository access rights
- (optional) the validity period of the assignment
- (optional, with read-only access to the repository) connection repository snapshot

  ☛ *See "Assigning a Profile to a Person", page 41.*

Creating a Profile

Users with the same profile share common characteristics (options, repository access rights, authorized products, reading and writing rights on objects).

To create a profile:

1. Open the **Profiles** management window.

   ☛ *See "Opening the profile management window", page 47.*

2. In the **Profile** tab, click **New** .
   A new profile appears in the list of profiles.

   ☛ *By default the **Name** of the profile is created in format "Profile-x" (x is a number that increases automatically).*

3. In the profile management window, right-click the profile and select **Properties** .
   The profile properties dialog box opens.
4. Select the **General** tab.
5. (Optional) In the **Name** field, modify the profile name.
6. Click **OK**.
   The new **Profile** appears in the list of profiles.
   You must:
   - define profile UI access

     ☛ *See "Managing UI Access (Permissions)", page 233.*

     ☛ *You can use configuration of UI access defined on an existing profile, see "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.*

   - configure the profile.

     ☛ *See "Configuring a Profile", page 37.*

Customizing an Existing Profile / Creating a Profile from an Existing Profile

You can create a profile by aggregation of existing profiles.

To customize a profile for which you do not have modification rights, you can create a new profile from this profile.

☺ ***To customize a profile provided by MEGA, MEGA recommends you create a new profile from this existing profile.***

To customize a profile for which you do not have modification rights:

1. Create a new profile.

   ☛ *See "Creating a Profile", page 34.*

2. In the properties dialog box of the new profile, select the **Used Profile** tab.
3. Right-click your **Profile** and select **Connect > Profile**.
4. (Optional) In the query field, enter the characters you want to find.
5. Click **Find** .
6. In the results list, select the profile you want to customize.

   ☛ *You can aggregate several profiles.*

7. Click **Connect**.
   The profile you have created inherits all accesses defined on the profile you have connected. You can customize these accesses.

Viewing Profile Characteristics

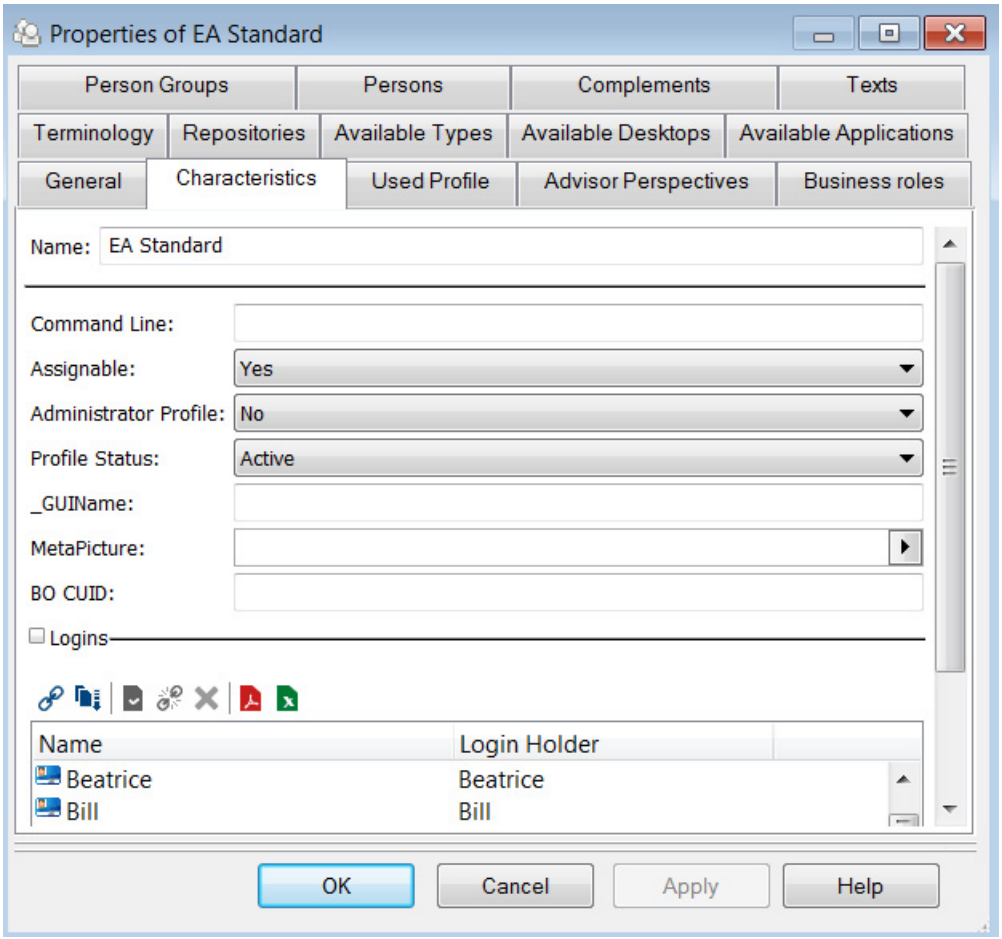
To view profile characteristics:

1. Open the **Profiles** management window.

   ☛ *See "Opening the profile management window", page 47.*

2. In the **Profile** tab, select the profile.
3. In the toolbar, click **Properties** .
   The profile **Properties** dialog box opens.
4. Select the **Characteristics** tab.

   ☛ *For detailed information on characteristics of a profile, see "Creating a Profile", page 34.*



☛ *See "Configuring a Profile", page 37.*

Configuring a Profile

From the profile properties window you can define:

☛ *See "Profile Properties", page 22.*

- products accessible to users with the current profile

  ☛ *See step 2.*

- if the profile is assignable or not

  ☛ *See step 3.*

- if the profile is an administrator profile or not

  ☛ *See step 4.*

- if the profile is provided at connection.

  ☛ *See step 5.*

- if the profile is active or not.

  ☛ *See step 6.*

- the profile display name in the interface.

  ☛ *See step 7.*

- the profile icon in the interface.

  ☛ *See step 8.*

- applications accessible to users of the profile

  ☛ *See "Defining applications accessible to profile users", page 38.*

- desktops accessible to users of the profile

  ☛ *See "Defining application desktops accessible to profile users", page 39.*

- the terminology associated with the profile

  ☛ *See "Associating a terminology with a profile", page 39.*

- object types available

  ☛ *See "Defining the object types available for a profile", page 40.*

- the sub-profiles used to define the profile.

  ☛ *See "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.*

From the profile properties window, you can also perform a mass profile assignment to persons, see *"Performing a Mass Profile Assignment to Persons", page 42.*

Configuring profile characteristics

To configure profile characteristics:

1. Access the properties of the profile.

   ☛ *See "Viewing Profile Characteristics", page 36.*

2. (Optional) In the **Command Line** field, enter the command defining products that can be accessed by users with the current profile.

   ☛ *See "Products accessible on the license (Command Line)", page 22.*

3. (Optional) In the **Assignable** field, modify the attribute value via the drop-down menu. By default, the profile is not assignable.

   ☛ *See "Assignable", page 23.*

4. (Optional) In the **Administrator Profile** field, modify the attribute value.

   ☛ *By default, the profile is not an administrator profile.*

   ☛ *See "Administrator profile", page 23.*

5. (Optional) In the **Profile Display** field, modify the attribute value.

   ☛ *By default the profile is provided at connection.*

   ☛ *See "Profile display", page 24.*

6. (Optional) In the **Profile Status** field, modify the attribute value.

   ☛ *By default, the profile is active.*

7. (Optional) In the **_GUIName** field, enter the profile name displayed in the interface.
8. (Optional) In the **MetaPicture** field, click the arrow and select **Query MetaPicture**.
   - In the query field, enter the characters you want to find and click **Find**.
   - In the results list, select the icon and click **OK**.

Defining applications accessible to profile users

☛ *To modify a profile provided by **MEGA**, you must create a new profile; see "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.*

So that a user of a profile can connect to an application, you must connect this application to the profile concerned.

☛ *See "Connection schema: Person - Profile - Application", page 14.*

All desktops connected to the application are then accessible. To enable access to only certain desktops of the application, see "Defining application desktops accessible to profile users", page 39.

☛ *See "Connection schema: Person - Profile - Application - Desktops", page 15.*

To define applications available for a profile:

1. Access the properties of the profile.

   ☛ *See "Viewing Profile Characteristics", page 36.*

2. Select the **Available Applications** tab.
3. In the toolbar, click **Connect** .
   The applications query tool appears.
4. (Optional) In the second field, enter the characters to find.
5. Click **Find** .
6. In the query results, select the application you want to connect.

   ☛ *To select more than one application, use the [Ctrl] key (Windows Front-End).*

7. Click **Connect**.
   The applications are connected to the profile.

Defining application desktops accessible to profile users

A user can connect to applications via customized desktops according to actions to be performed.

If an application contains several desktops, you can specifically define application desktops that are accessible to the concerned profile.

☛ *See "Connection schema: Person - Profile - Application - Specific Desktops", page 17.*

To do this, you must connect to the profile:

- the application containing the desktops.

  ☛ *See "Defining applications accessible to profile users", page 38.*

- the desktops to which you want users of the profile to connect.

  ☛ *The application desktops that are not connected to the profile are not accessible to users of the profile.*

  ☛ *To modify a profile delivered by **MEGA**, you must have rights to modify **HOPEX** data. Alternatively, you can create a new profile, see "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.*

To define application desktops available for a profile:

Prerequisite: The application accessible to users of the profile is defined.

☛ *See "Defining applications accessible to profile users", page 38.*

1. Access the properties of the profile.

   ☛ *See "Viewing Profile Characteristics", page 36.*

2. Select the **Available Desktops** tab.
3. In the toolbar, click **Connect** .
   The desktop query tool appears.
4. (Optional) In the second field, enter the characters to find.
5. Click **Find** .
6. In the query results, select the desktop you want to connect.

   ☛ *To select more than one desktop, use the [Ctrl] key (Windows Front-End).*

7. Click **Connect**.
   The desktops are connected to the profile.

Associating a terminology with a profile

A Terminology defines a set of terms used in a specific context instead of the standard term.

☛ *For information on creating and managing a Terminology, see **HOPEX Power Studio - Renaming HOPEX Concepts**.*

To associate a terminology with a profile:

1. Access the properties of the profile.

   ☛ *See "Viewing Profile Characteristics", page 36.*

2. Select the **Terminology** tab.
3. In the toolbar, click **Connect** .
   The business roles query tool appears.
4. (Optional) In the second field, enter the characters to find.

5. Click **Find** .
6. In the query results, select the terminology you want to connect.

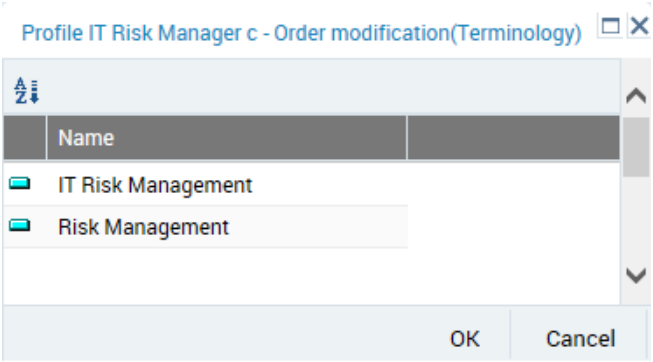
   ☛ *You can select several terminologies.*
7. Click **Connect**.
   The terminology is connected to the profile.

If you associate more than one terminology with the profile, you must define an order of priority for them.

To define the priority of the terminologies of a profile:

1. Access the properties of the profile.

   ☛ *See "Viewing Profile Characteristics", page 36.*
2. Select the **Terminology** tab.
3. In the toolbar, click **Reorganize** .
4. Drag and drop to place the priority terminology at the top.



In the example above, the terms of the Risk Management terminology are used when they are not defined in the IT Risk management terminology.

Defining the object types available for a profile

You can define which specific object types are available for a profile:

- document categories
- document models
- Report DataSet Definitions
- widgets

To define the object types available for a profile:

1. Access the properties of the profile.

   ☛ *See "Viewing Profile Characteristics", page 36.*
2. Select the **Available Types** tab.
3. Select **Available Objects**.
4. In the toolbar, click **Connect** .
   The object type query tool appears.

5. (Optional) In the query tool, in the first field, select the object type category.
6. (Optional) In the second field, enter the characters to find.
7. Click **Find** .
8. In the query result, select the object types to make available for the profile.
9. Click **Connect**.
   The object types selected are made available for the profile.

Assigning a Profile to a Person

A person may have several profiles.

A user must have at least one profile assigned to be able to connect to HOPEX.

Assigning a profile to a person defines:

- the profile assigned
- the repository concerned by the assignment
- the repository access rights concerned by the assignment
- (optional) a validity period of an assignment.
- (optional, with read-only access to the repository) connection repository snapshot

Repository Snapshot:

A repository snapshot defines repository state at a given moment.

The connection respository snapshot defines the state of the repository to which the users of a profile connect.

Repository snapshots are available for RDBMS repositories.

*The repository snapshot creation function is available with **HOPEX Collaboration Manager.***

To define a repository snapshot, a repository snapshot must have been previously created.

*To create a repository snapshot, see the **HOPEX Collaboration Manager - Repository Snapshots** guide.*

To assign a profile to a person:

1. Access the profile assignment management window.

   See "Opening the profile assignment window", page 49.

2. In the **Persons** pane, select the person to whom you want to assign a profile.
3. In the **Profile Assignment** pane, click **New** .
   The profile assignment creation wizard opens.
4. In the **Profile assigned** field, click the drop-down menu and select the profile you want to assign to the person.

   *To execute a filtered query on a profile, click the arrow and select **Connect Profile**.*

5. (optional) In the **Validity start date** field, use the calendar to define the start date of profile validity.
6. (optional) In the **Validity end date** field, use the calendar to define the end date of profile validity.
7. (optional) In the **Repository** field, click the drop-down menu and select the repository to which you want to assign the profile.

   ☛ *By default, the profile is assigned to all repositories.*
8. (optional) In the **Data access** field, click the drop-down menu to restrict access to read-only data.

   ☛ *By default, access to repositories is "read/write".*
9. (optional, with read-only access to the repository) In the **Connection Snapshot** field, select a connection repository snapshot.
10. Click **OK**.
    The profile is assigned to the person on the selected repository for the specified duration.

Performing a Mass Profile Assignment to Persons

You can perform a mass profile assignment to persons. In this case, the profile assignment to the person is defined automatically with a permanent validity on all repositories.

☛ *To modify the configuration of the assignment, see "Assigning a Profile to a Person", page 41.*

To perform a mass profile assignment to persons:

1. Open the profile properties dialog box.

   ☛ *See "Viewing Profile Characteristics", page 36.*
2. In the **Persons** tab, click **Connect** .
3. (Optional) In the query field, enter the characters to find.
4. Click **Find** to run the query.
   The list of persons who can be connected to the profile appears.
5. In the result list, select the persons you want to connect to the profile.

   ☛ *Use the [Ctrl] key to select more than one person at the same time.*
6. Click **OK**.
   The profile is assigned to all selected persons, on all repositories, without a validity limit.

   ☛ *To modify the configuration of the assignment, see "Assigning a Profile to a Person", page 41.*

Deleting a Profile

If you delete a profile that is the only profile assigned to a user, this user can no longer connect to HOPEX.

To delete a **Profile**:

1. Open the **Profiles** management window.

   ☛ *See "Opening the profile management window", page 47.*

2. In the **Profile** tab, select the profile you want to delete.

   ☛ *You can select more than one.*

3. Click **Delete** ⊗.
   The dialog box for deleting a profile opens.
4. Click **Delete**.
   The profile is deleted from the environment

ACCESS TO USER MANAGEMENT

See:

- "Accessing User Management and UI Access Management Folders", page 44.
- "Viewing Person Characteristics", page 50.
- "Viewing Login Characteristics", page 51.

Accessing User Management and UI Access Management Folders

This section describes how to access management windows for users and profiles.

See:

- "Accessing User Management and UI Access Management Folders", page 44
- "Opening the User Management window", page 45
- "Opening the profile management window", page 47

Accessing User Management and UI Access Management Folders

The **User management** folder contains all sub-folders for management of *users*, that is:

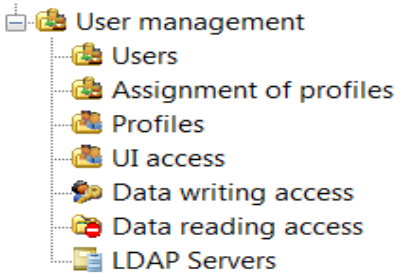
- *persons* and *logins*,
- *profiles*,
- *object UI access* and *general UI access*.

To access user management and UI access management folders:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **User management** folder of the environment.



From the **User management** folder of an environment, you can access management windows:

- of **Users**
  User characteristics are defined in the following tabs:
  - **Persons**
  - **Logins**

    ☛ *See "Opening the User Management window", page 45.*
- **Profile Assignments**

  ☛ *See "Opening the profile assignment window", page 49.*
- **Profiles**

  ☛ *See "Opening the profile management window", page 47.*
- of **UI access**
  Object UI access and general UI access are characteristics defined in tabs:
  - **Object UIs**
  - **General UIs**

    ☛ *See "Managing UI Access (Permissions)", page 233.*
- of **Data reading access**

  ☛ *See "Managing Data Reading Access", page 275.*
- of **Data writing Access**

  ☛ *See "Managing Data Writing Access", page 249.*
- of **LDAP servers**

  ☛ *See "LDAP Authentication", page 66.*

Opening the User Management window

Characteristics of *users* are managed in the **Users** window.

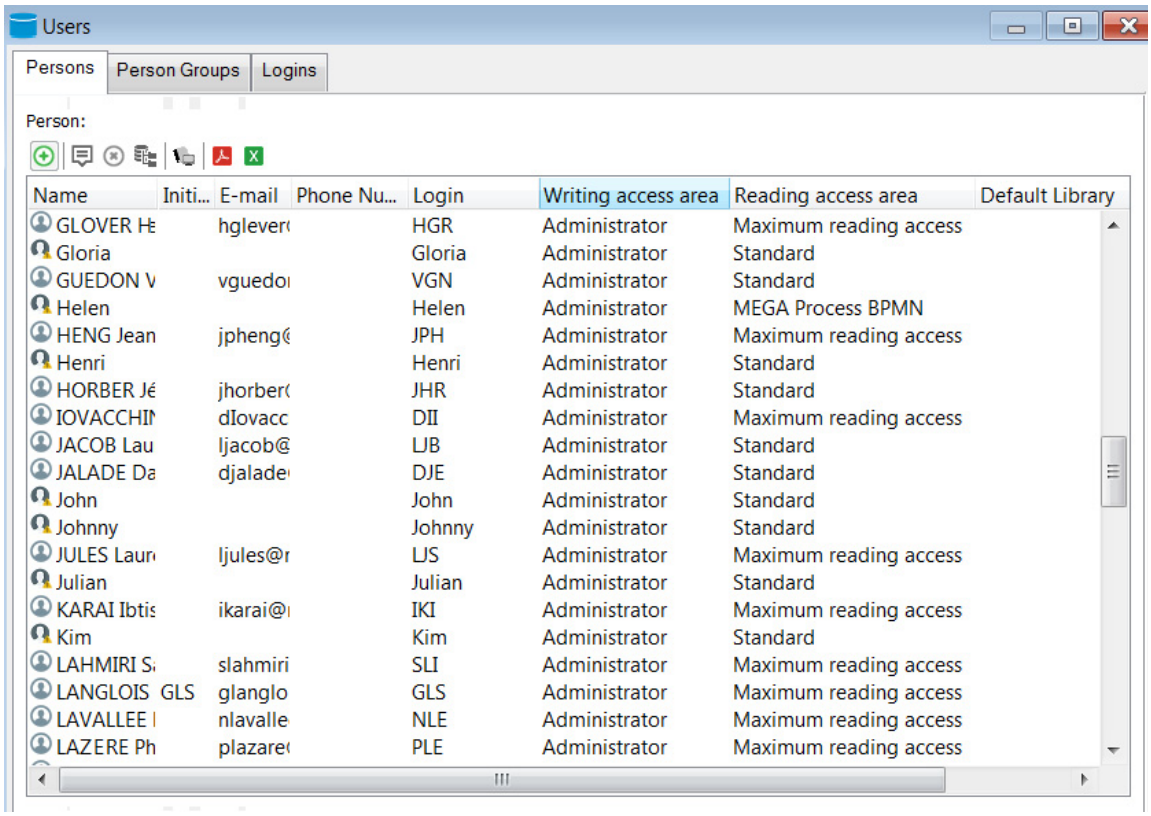
☛ *To manage user options, see "Managing Options", page 323.*

To manage users:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*
2. Expand the **User management** folder of the environment.

3. Right-click the **Users** folder and select **Manage**.
   The **Users** window opens.



The **Users** window contains tabs:

- **Persons**, listing all persons in the environment and detailing personal characteristics of each person.

  ☛ *See "Person Properties", page 28.*

- **Person Groups**, listing all person groups in the environment and detailing their characteristics.

  ☛ *Managing person groups is a functionality available in **HOPEX** (Web Front-End). For more information on managing this functionality, see the Web HOPEX Administration - Supervisor guide.*

- **Logins**, which lists all the users in the environment and details their characteristics.

  ☛ *See "Login Properties", page 30.*

From the **Users** management window you can:

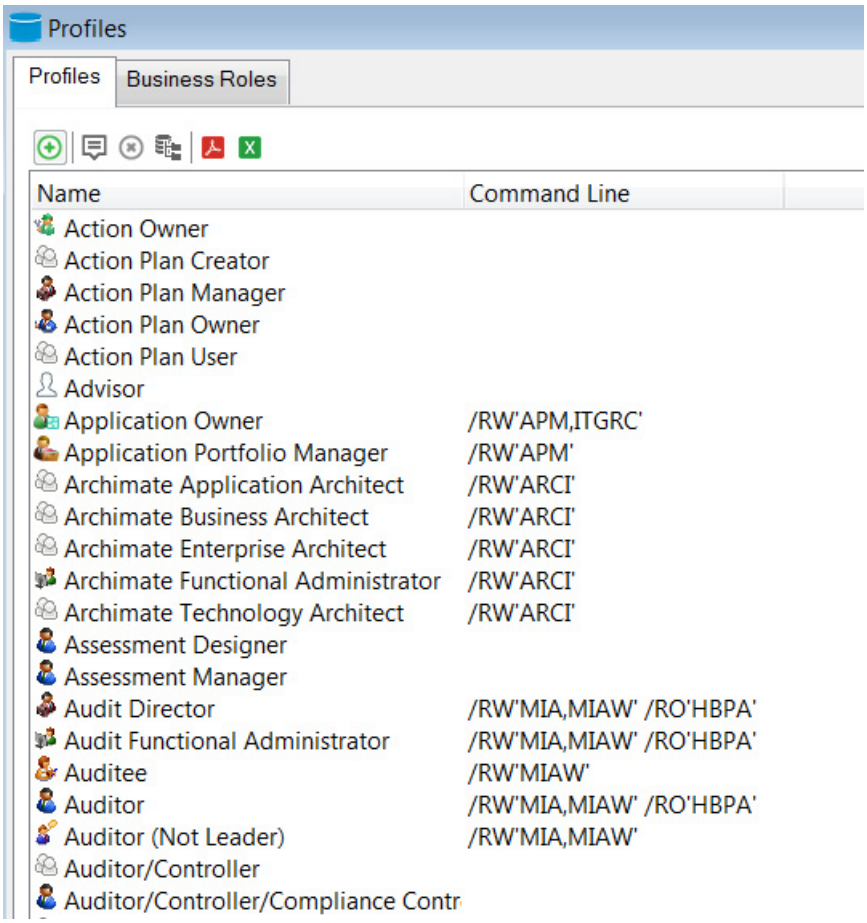
- create:
  - persons
    - ☛ *See "Defining a Person", page 53.*
  - user management
    - ☛ *See "Creating Users", page 52.*
- configure characteristics:
  - of a person
    - ☛ *See "Defining a Person", page 53.*
  - of a login
    - ☛ *See "Defining the Login of a Person", page 56.*
- delete users
  - ☛ *See "Deleting Users", page 58.*
- modify:
  - user properties
    - ☛ *See "Modifying User Properties", page 57.*
  - user passwords
    - ☛ *See "Creating or Modifying a User Password (Windows Front-End)", page 58.*
- access user options
  - ☛ *See "Managing Options", page 323.*

Opening the profile management window

To manage profiles:

1. Connect to **HOPEX Administration** and select the environment concerned.
   - ☛ *See "Connecting to an Environment", page 7.*
2. Expand the **User management** folder of the environment.

3. Right-click the **Profiles** folder and select **Manage**.
   The **Profiles** management window opens.



The **Profiles** window contains the following tabs:

- **Profiles**, which lists all the profiles in the environment and detailing their characteristics. For each profile, it details the products accessible (optional).

  ☛ *See "Managing Profiles", page 33.*

- **Business Roles**, which lists all the business roles in the environment used by functional administrators to manage them.

From the **Profiles** management window you can:

- create profiles

  ☛ *See "Creating a Profile", page 34.*

- customize an existing profile

  ☛ *See "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.*

- consult profile characteristics

  ☛ *See "Viewing Profile Characteristics", page 36.*

- configure profiles

  ☛ *See "Configuring a Profile", page 37.*

- perform a mass profile assignment to persons

  ☛ *See "Performing a Mass Profile Assignment to Persons", page 42.*

- delete profiles

  ☛ *See "Deleting a Profile", page 42.*

Opening the profile assignment window

To open the assignment of profiles window

1. Connect to **HOPEX Administration** and select the environment concerned.

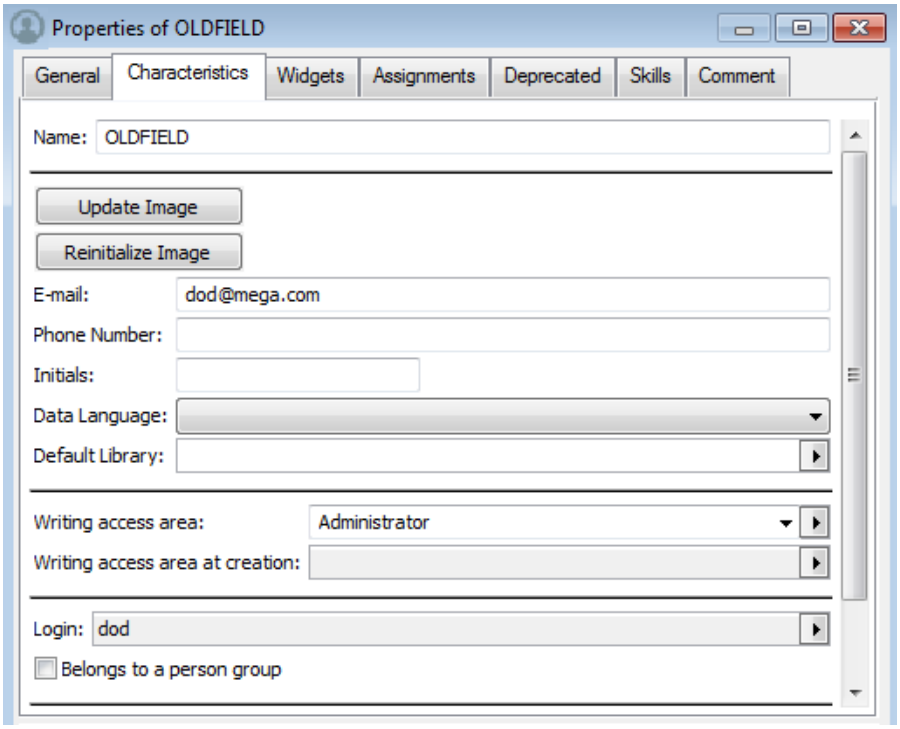
   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **User management** folder of the environment.
3. Right-click the **Assignment of profiles** folder and select **Manage**.
   The **Assignment of profiles** window appears.

From the **Assignment of profiles** management window, you can assign profiles to persons.

☛ *See "Assigning a Profile to a Person", page 41.*

Viewing Person Characteristics



The icon for a person is represented by:

- when the person is created (name and writing access area defined) but does not have a login.
- when the person has a login but is not fully configured (e-mail or profile assignment is not defined).
- when the person is configured as a **HOPEX** user:
  name, writing access area, login and e-mail address are specified and a profile is assigned to the person.

  ☛ *See "Defining a Person", page 53, "Creating Users", page 52 and "Assigning a Profile to a Person", page 41.*

To view person characteristics:

1. Open the **Users** management window.

   ☛ *See "Opening the User Management window", page 45.*

2. Select the **Persons** tab.
   The list of persons appears with the characteristics of each person.

   ☛ *You can modify these characteristics directly in this window (with a click in the corresponding field).*

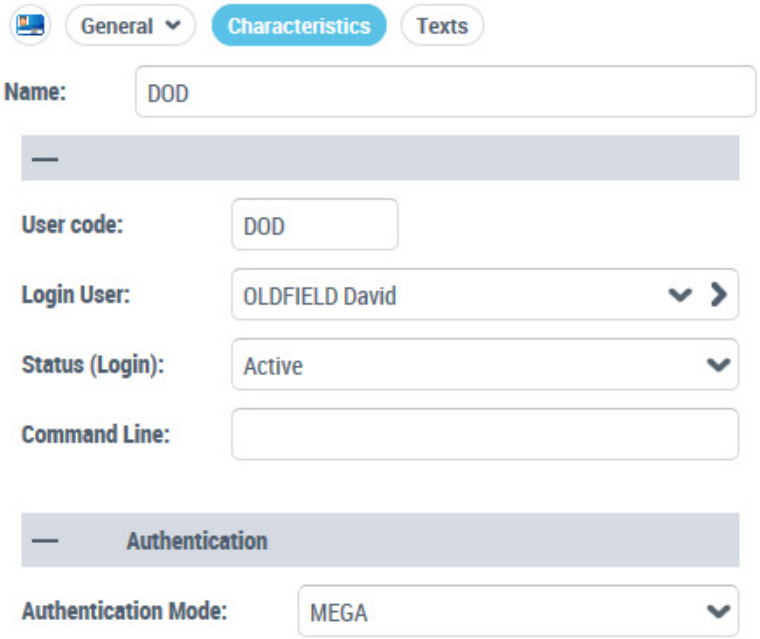
3. In the **Persons** list, select the person.
4. In the toolbar, click **Properties** .

5. From the **Properties** window for the person, click:
   - the **Characteristics** tab to define or modify the person properties.
     - ☛ *See "Person Properties", page 28.*
     - ☛ *See "Defining a Person", page 53.*
   - the **General** tab, **History** sub-tab to display the actions performed on the person.
   - the **Assignments** tab to display and assign profiles to the person.

Viewing Login Characteristics

☛ *For detailed information on characteristics of a login, see "Login Properties", page 30.*

☛ *To configure a login, see "Defining the Login of a Person", page 56.*



To view login characteristics:

1. Open the **Users** management window.
   - ☛ *See "Opening the User Management window", page 45.*
2. Select the **Logins** tab.
3. Select the **Login**.
4. In the toolbar, click **Properties** .
   The **Properties** dialog box of the selected login opens.

MANAGING USERS

For an overview of actions to be performed to create and define a user see "Actions to be Performed to Define a User", page 10.

The following points are covered here:

- configuration:
  - "Creating Users", page 52
  - "Defining a Person", page 53
  - "Creating the Login of a Person", page 55
  - "Defining the Login of a Person", page 56
  - "Modifying User Properties", page 57
- management:
  - "Managing User Inactivity", page 61
  - "Preventing User Connection", page 57
  - "Deleting Users", page 58
  - "Creating or Modifying a User Password (Windows Front-End)", page 58
  - "Exporting and Comparing Repository Users", page 59

Creating Users

☛ *Instead of creating users one by one, you can import a list of persons. This list can for example come from an LDAP server (see "Synchronization with a company directory", page 65).*

A user depends on an environment. To create a user, you must connect to the environment to which the user will be attached.

A user is a person with a login. To create a user, you must create a person with its login, or create the login of a person already created.

☛ *For detailed information on characteristics of a person, see "Person Properties", page 28.*

☛ *For detailed information on characteristics of a login, see "Login Properties", page 30.*

☛ *To import users from an LDAP directory, see "LDAP Authentication", page 66.*

To create a user:

1. Open the **Users** management window.

   ☛ *See "Opening the User Management window", page 45.*

2. In the **Users** window, select the **Persons** tab.
3. Click **New** ⊕.
   The **Creation of Person - Characteristics** dialog box opens.
4. In the **Name** field, enter the name of the person.

   `Example: DUBOIS Guillaume`

   ☛ *Remember to use the same format for all persons.*

5. (Optional, but recommended) In the **E-mail** field, enter the e-mail address of the person.

   ☛ *The e-mail address is necessary, for example, for distributing documents, receiving notifications and questionnaires, or when a Web user lost his/her password.*

6. In the **Login** field, enter the login of the person.

   ☛ *If you do not enter the Login, it will automatically take the value entered in the **Name** field.*

   ☛ *A **Login** is unique and can be assigned to only one Person or Person Group.*

   ☛ *A **Person** can have only one **Login**.*

   `Example: GDS`

7. (With the **HOPEX Power Supervisor** technical module) Using the drop-down menu in the **Writing Access Area** field, select the value of the writing access area of the user.

   ☛ *For more details on writing access, see "Managing Data Writing Access", page 249.*

8. (If required, with the **HOPEX Power Supervisor** technical module) Using the drop-down menu in the **Reading Access Area** field, select the value of the reading access area of the user.

   ☛ *By default at creation, the user is connected to "Standard" reading access area. This field only appears if reading access management has been activated, see "Managing Data Reading Access", page 275.*

9. Click **Finish**.
   The user is created (a login is associated with the person): the person appears in the list of persons and its login appears in the list of logins.

   ☛ *You must configure the login of the user, see "Defining the Login of a Person", page 56.*

   ☛ *By default, the new user does not have an associated password. To associate a password with the user, see "Creating or Modifying a User Password (Windows Front-End)", page 58.*

Defining a Person

📖 ***Person** represents a physical person or a system.*

☛ *For more information on properties of a person, see "Person Properties", page 28.*

From the profile properties window of a person, you can define:

- name of the person

  ☛ *See step 1.*
- image of the person

  ☛ *See step 2.*
- e-mail address of the person

  ☛ *See step 3.*
- telephone number and initials of the person

  ☛ *See step 4.*
- data language of the Web user

  ☛ *See step 5.*
- default library to store objects created by the person

  ☛ *See step 6.*
- writing access area of the user

  ☛ *See step 7.*
- reading access area of the user

  ☛ *See step 7.*
- login of the user

  ☛ *See step 8.*
- if the user belongs to a person group

  ☛ *See step 9.*

To configure a **Person**:

1. Access the properties of the person.

   ☛ *See "Viewing Person Characteristics", page 50.*
2. (Optional) To add or update the image of the person, click **Update Image**, select the image and click **OK**.

   ☛ *To delete the image, click **Reinitialize Image**.*
3. (Optional, but recommended) In the **E-mail** field, enter the e-mail address of the person.

   ☛ *The e-mail address is required, for example, to initialize the Web user password, for distributing documents, receiving notifications and questionnaires, or when a Web user lost his/her password.*
4. (Optional) Enter the **Phone Number** and the **Initials** of the person.
5. (Web specific, optional) In the **Data Language** field, you can define a specific data language for this user.
   - Click the arrow and select **Query Language**.
   - In the query wizard, select the data language (objects) and click **OK**.

     ☛ *If the field is not specified, the default data language is the interface language defined in environment options (**Options/Installation/User Management**: **Data Language**).*

     ☛ *See "Managing Languages", page 334.*
6. (Optional) In the **Default Library** field, click the arrow and select the default library in which objects created by the user are stored if the creation context does not define one.

7. (Optional, with the **HOPEX Power Supervisor** technical module) You can modify the values at the following levels:
   - user writing access via the drop-down menu in the **Writing Access Area** field.

     ☛ *By default, all users are connected to the only writing access area that exists: "Administrator". For more details on writing access, see "Managing Data Writing Access", page 249.*
   - user writing access at creation via the drop-down menu in the **Writing Access Area** field.
   - reading access via the drop-down menu in the **Reading Access Area** field.

     ☛ *This field only appears if reading access management has been activated, see "Managing Data Reading Access", page 275.*
   - reading access at creation via the drop-down menu in the **Writing Access Area** field.

     ☛ *This field only appears if reading access management has been activated, see "Managing Data Reading Access", page 275.*
8. So that the person can connect to **HOPEX**, the person must have a **Login**.

   ☛ *See "Creating the Login of a Person", page 55.*

   ☛ *See "Defining the Login of a Person", page 56.*
9. (optional) If necessary select **Belongs to a Person Group**
10. Click **OK**.
    The person is configured.

Creating the Login of a Person

To connect to **HOPEX**, a person must have a Login. When you create a person, his/her login is automatically created.

To create the login of a person:

1. Access the properties of the person.

   ☛ *See "Viewing Person Characteristics", page 50.*
2. In the **Login** field, click the arrow and select **Create Login**.
   The **Creation of Login** dialog box opens. The name of the login is already entered with the name of the login holder.
3. (Optional) In the **Name** field, modify the login name.

   ☛ *A login is unique; it can be assigned to one Person or one Person Group only.*

   ☛ *A **Person** can have only one **Login**.*

   `Example: GDS`
4. In the **User Code** field, enter the user code to be associated with the login.

   `Example: GDS`
5. Click **OK**.
   The login of the user appears in the **Login** field.

Defining the Login of a Person

From the Login properties window, you can:

☛ *See "Login Properties", page 30.*

- define the login name, the user code associated with the login and the login holder

  ☛ *See step 1.*

- modify user status (inactive)

  ☛ *See step 2.*

- restrict user access to certain products

  ☛ *See step 3.*

- modify user authentication mode

  ☛ *See step 4.*

To define the login of a person:

1. Display the **Characteristics** tab of the login properties.

   ☛ *See "Viewing Login Characteristics", page 51.*

   - The login **Name** and **User Code** attributes are already created, but you can modify these if necessary.

     *A **login** is unique and defined for a person or person group.*

     *The **User code** is the short identifier (upper case) of the user. It serves as a basis for naming user private workspaces.*

   - The **Login Holder** is the person associated with the login.

2. (Optional) Modify the **Status (Login)** field value, which defines if the user is active or not.

   ☛ *See "Inactive user (Status)", page 31.*

3. (Optional) In the **Command Line** field, define the products available to which the user has access.
   To restrict user access to products A and B, enter the command:
   /RW'<accessible Product A code>;<accessible Product B code>;<...>'

   For example: You have licenses for products **HOPEX Business Process Analysis, HOPEX IT Portfolio Management and other HOPEX products.** To authorize only **HOPEX Business Process Analysis** and **HOPEX IT Portfolio Management** modules to a user, enter:

   /RW'HBPA;APM'

   ☛ *To found out the product code, see "Access your list of available products", page 341 or on line documentation: **Concepts > Products**.*

   If a user is connected to a profile and the user and profile each have access to products restricted by the Command Line attribute, the products accessible to the user are at the intersection of the values of the Command Line attribute of the user (on his/her login) and profile.

4. (Optional) In the **Authentication Mode** field, click the arrow and modify the authentication mode. The default value is "MEGA".

   ☛ *See "Authentication mode", page 31.*

5. Click **Apply**.

Modifying User Properties

You can modify user properties. For each user you can modify properties of:

- person:
  - its name
  - image
  - e-mail address
  - telephone number
  - initials
  - data language
  - default library
  - writing access area
  - reading access area
  - group
  - profile assignments (connection)
  - object assignments (business roles)

    ☛ *See "Person Properties", page 28.*

    ☛ *See "Viewing Person Characteristics", page 50.*

    ☛ *See "Defining a Person", page 53.*
- login:
  - its name
  - user code

    To assure consistent actions history, the user code should not be modified.
  - status
  - accessible products (Command Line)
  - authentication mode

    ☛ *See "Login Properties", page 30.*

    ☛ *See "Viewing Login Characteristics", page 51.*

    ☛ *See "Defining the Login of a Person", page 56.*

Preventing User Connection

When you no longer want a user to connect to **HOPEX**, but want to retain trace of his/her actions, you must render the user inactive but not delete it from your repository.

To render a user inactive:

1. Open the **Characteristics** tab of the login properties dialog box.

   ☛ *See "Viewing Login Characteristics", page 51.*
2. In the **Status (Login)** field, select "Inactive".
3. Click **Apply**.

Deleting Users

When you delete a user from the repository, the commands connected to this user become orphans and you lose part of the history saved in logs. To remove a user but keep its history of commands, see "Preventing User Connection", page 57.

To delete a user:

1. Access the User Management page.

   See "Opening the User Management window", page 45.

2. In the **Persons** tab, select the person to be deleted and click **Delete** .

   You can select more than one.

   The **Delete Objects** dialog box opens.

3. (If necessary) In the **Delete** column, modify the deletion selection of a person and her/his login.
4. Click **Delete** to confirm deletion.
   The person and login are deleted from the repository.

   All traces of user actions are lost.

Creating or Modifying a User Password (Windows Front-End)

To manage the password of a Web user, see the HOPEX Administration - Supervisor guide:

By default, the user does not have a password.

*Only a user with **Administrator** type profile has management rights. He is the only user type that can create a user password. To grant administrator access rights to a user, see "Defining the Login of a Person", page 56.*

The password can be modified later:

- in **HOPEX**, by the user, from the **HOPEX** connection window.
- in **HOPEX Administration** , by the Administrator, notably when the user has forgotten his/her password.

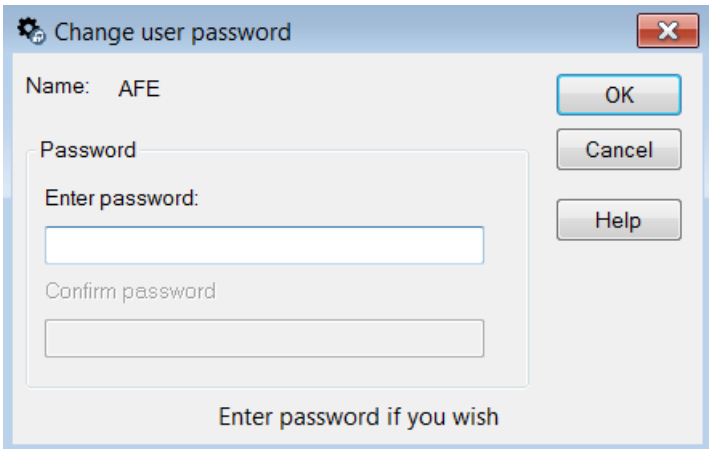
To create or modify the password of a user (Windows Front-End):

1. Open the **Users** management window.

   See "Opening the User Management window", page 45.

2. Select the **Logins** tab.

3. In the logins list, right-click the login concerned and select **Password**.
   The **Change User Password** dialog box opens.



4. Enter the password you want to assign to the user.

   ☛ *Do not use separation character (“!”, “$”, “@”, “:”, “;”).*

   Password confirmation entry is required.

   ☛ *If the two entries are not identical, the entry must be corrected.*

5. Click **OK**.
   The user password is saved.

Exporting and Comparing Repository Users

You can export and compare repository users. This can be useful for example to copy users and their characteristics associated with an environment into another environment.

To:

- export the log of a user:

  ☛ *See "Exporting HOPEX Objects", page 206.*

- compare users of a repository with another repository:

  ☛ *See "Comparing Environments", page 180.*

MANAGING USER OPTIONS

For specific requirements, you can modify default values of certain **Options** (see "Accessing Options", page 326).

Configuring Metamodel Access

With the **Metamodel Access** option (**Options > Repository** menu) , you can restrict the view of **HOPEX** objects or functions according to user skill level.

This option can be defined at environment, profile or user level according to the requirement.

Metamodel access levels are:

- **Beginner**
  For introduction to **HOPEX**. Only basic objects are visible. This level allows very simple modeling.
- **Intermediate** (default value)
  For standard use of **HOPEX**. Almost all object types, links and non-technical attributes are visible.
- **Advanced**
  For advanced use of **HOPEX**. All objects, links and non-technical attributes are visible, including those that require advanced skills for their use. Only object types and attributes which are present only for compatibility with previous versions are filtered. Certain technical object types are visible. The user can carry out simple customizations of the **HOPEX** platform.

  For example, this level enables access to:
  - certain navigation windows, with a restricted view of objects (**MetaStudio**, **Utilities**)
  - the **Administration** navigation window, which gives access to reading and writing access areas.
  - the **Repository Activity** (Navigation window in **HOPEX,** see "Viewing the Repository Update Log", page 86).
- **Expert**
  This level displays all object types, links, and attributes, as well as the abstract metamodel. All HOPEX platform customizations are available.

  Specify this access level only for a highly expert user or a particular profile (e.g.: HOPEX Customizer).

  This is the level for example that offers:
  - all functions of **HOPEX Power Studio** (**MetaStudio** navigation window)
  - report template creation (**Utilities** navigation window)
  - certain advanced commands (e.g.: update logfile export at particular intervals).

Authorizing Deletion of a Published Object

Users working in a public workspace can delete objects.

By default, users working in a private workspace are not authorized to delete dispatched objects, even if these have been created by the user wishing to delete.

The **Authorize dispatched object deletion from private workspace** allows the user to delete dispatched objects from a private workspace, irrespective of the creator.

This authorization complements object deletion rights defined elsewhere for the profile or user.

Authorizing HOPEX Data Modification

This option should only be selected in certain highly specific cases, at debugging operations or at MEGA request, and for a temporary period.

This option authorizes modification of the **HOPEX** metamodel or any other **HOPEX** technical object. Modifying a **HOPEX** object can generate errors at **HOPEX** upgrades, import of correctives, etc.

Specify this access level only for a highly expert user or highly advanced profile.

Managing User Inactivity

You can specify for how long user session time can remain inactive before closing.

This option can be useful for example for security requirements, or to ensure that all sessions are closed before starting a batch program.

By default, user inactivity management is not activated.

Activating/Deactivating user inactivity management

To activate/deactivate user inactivity management:

1. Access **Options** at the environment level.

   See "Accessing Options", page 326.

2. In the **Options** tree, select **Workspace**.
3. In the right pane:
   - to activate user inactivity management, select **Automatic Session Timeout**.
   - to deactivate user inactivity management, clear **Automatic Session Timeout**.

Managing user inactivity

User inactivity management is taken into account if the **Inactivity Management** option is selected.

☛ *See "Activating/Deactivating user inactivity management", page 61.*

To manage user inactivity:

1. Access **Options** at the environment level.

   ☛ *See "Accessing Options", page 326.*

2. In the **Options** tree, select **Workspace**.
3. In the right pane, select values for options:

   ☛ *Duration of inactivity requiring authentication must be less than that closing **HOPEX.***

   - **Period of inactivity requiring authentication**

     When this duration has been reached, the user receives a message requesting authentication.

     ☛ *This duration warns the user before he/she is disconnected if the **Duration of inactivity before closing HOPEX** option has been specified.*

   - **Duration of inactivity before closing HOPEX**

     When this duration has been reached, the user is disconnected and **HOPEX** closes without warning.

     ☛ *So that the user is warned of disconnection, the **Period of inactivity requiring authentication** must be specified and its value less than that of the **Duration of inactivity before closing HOPEX**.*

AUTHENTICATION IN HOPEX

Authentication is a process consisting of verifying that a person corresponds to his or her declared identity. In IT networks, authentication normally depends on a connection name and password.

Unique authentication, known as Single Sign On (SSO) or Unified Login, is a software solution that enables company network users to access all authorized resources in total transparency, on the basis of unique authentication at initial network access.

In this way, a single password enables access to all company applications and systems.

This solution offers several advantages, including:

- Greater security
  The user no longer has to remember several connection procedures, identifiers or passwords.
- Improved administrator productivity.
  HOPEX integrates into enterprise directories, which lightens administrator workload relating to password management.

The Single Sign On system used in **HOPEX** is based on standard security protocols natively integrated in Windows: Kerberos, SSO and LDAP. In addition, **HOPEX** Single Sign On complies with the following recognized standards:

- Windows Security Services
- C2-Level Security of the American Defense Department
- LDAP via ADSI
- Kerberos
- NTLM Authentication

For more details on single sign-on, see:

- document "Single Sign-On in Windows 2000 networks" at the following Web address:
  http://technet.microsoft.com/fr-fr/library/bb742456.aspx
- technical article ***Unified Login Security Management EN***.

HOPEX proposes the following authentication modes:

- **MEGA** authentication
- **Windows** authentication, which corresponds to Single Sign On.
- **LDAP** authentication
- **Custom** authentication, specific to Web applications connection only

> ☛ *See the technical article **Web connection overloading and configuration EN**.*

Defining Default Authentication Mode

Authentication can be:

- managed within the **HOPEX** platform (by default)
- delegated to a third party service

To select your authentication mode, **MEGA** recommends that you use authentication systems resistant to security attack:

- If your enterprise has an external authentication or SSO module, it is preferable to use the delegated authentication system.

  ☛ *See the **Web connection overloading and configuration EN** technical article.*

- If your enterprise has an LDAP authentication system, it is preferable to manage your authentication using an LDAP directory.

  ☛ *See "Defining default authentication mode to LDAP", page 64.*

- If you have no standard authentication system in your enterprise, you can use the MEGA authentication system managed in the **HOPEX** platform, less resistant to security attack.
  This is the authentication mode defined by default at installation, its value is "Standard".

  ☛ *See "Viewing default authentication mode", page 64.*

Viewing default authentication mode

In the environment options, you can consult and modify the default **Defined Authentication Mode**.

To view default authentication mode:

1. Access environment options.

   ☛ *See "Modifying options at environment level", page 326.*

2. In the options tree, expand the **Installation** folder and select **User Management**.
3. In the right pane, consult the value of the **Authentication Mode** option.
   By default at installation, "Standard" is the MEGA authentication mode.

Defining default authentication mode to LDAP

Users are managed in an LDAP directory and authentication is managed by the LDAP directory.

☛ *Authentication mode of users already created is not impacted. Only users created after the default authentication mode change are concerned.*

To define default LDAP authentication mode:

1. Access environment options.

   ☛ *See "Modifying options at environment level", page 326.*

2. In the options tree, expand the **Installation** folder and select **User Management**.
3. In the right pane, specify "LDAP" for the **Authentication Mode** option.

Defining user authentication mode

User authentication mode is defined on the login by the **Authentication Mode** parameter. The value of this parameter is inherited at user creation from the value of the **Authentication Mode** option defined in the environment options (see "Viewing default authentication mode", page 64).

To define user authentication mode, see "Defining the Login of a Person", page 56.

Windows Authentication

Synchronization with a company directory

Active Directory is a directory service designed principally for Windows environments.

Active Directory is a directory referencing persons (name, first name, telephone number, etc) and objects such as servers, printers, applications, databases, etc.

Active Directory enables inventory of all information concerning the network (users, machines and applications). Active Directory is at the heart of all network architecture and its purpose is to enable users to find and access any resource identified by the service.

Active Directory is based on standards DNS, LDAP, Kerberos, etc.

Associating a Windows user with a HOPEX user manually

You can connect a single **HOPEX** user to a Windows user.

To indicate the Windows identifier of a **HOPEX** user in the Administration application:

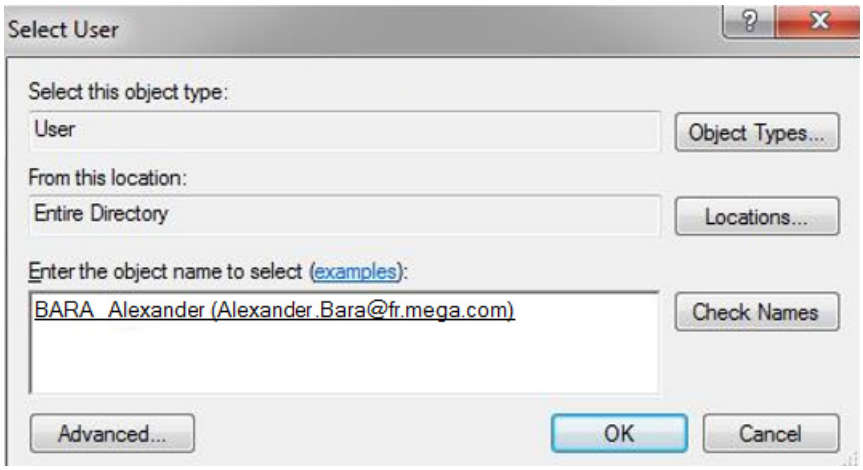
1. Open the properties dialog box of the user login.

   ☛ *See "Viewing Login Characteristics", page 51.*

2. Select the **Characteristics** tab.
3. In the **Authentication Mode** field drop-down list, select "Windows".
   The **Windows Login** field appears.
4. In the **Windows Login** drop-down list ▸, select **Select Windows User**.
   The **Select User** dialog box opens.
5. In the **Enter the object name to select** frame, enter the user name.

   ☺ *To find users with the help of a wizard, click **Advanced**.*

6. Click **Check Names**.



7. Click **OK**.
   The Windows identifier of the user is displayed.

Connection in case of unique authentication

(Windows Front-End) If the **HOPEX** user is:

- different from that identified by the Windows session opened on the workstation, the user must enter his/her own Windows password.
- the same as that identified by Windows session opened on the workstation, the user does not need to enter his/her password.

(Web Front-End) The user must enter his/her Windows password.

Single sign-on precautions

A system repository in which all users have been changed to Single Sign On connection mode (Windows) can no longer be opened outside the company in which the repository was created.

If you want the repository to be opened outside your company (by the **HOPEX** technical support team for example), ensure that at least one user remains in MEGA authentication mode.

LDAP Authentication

☛ *LDAP authentication is available only if you have technical module **HOPEX Power Supervisor**.*

An LDAP directory enables storage of user data of the enterprise.

HOPEX Administration allows you to create users authenticated at LDAP server level.

> ☛ *Only users (example: Administrator) with **HOPEX Administrator** or **User Management Administrator** profile can enter LDAP data, see "The Administration profiles provided", page 19.*

Configuring LDAP authentication

To configure LDAP authentication:

1. Create an LDAP server in **HOPEX Administration**.

   > ☛ *See "Creating an LDAP server", page 67.*

2. Specify parameters of your LDAP server.

   > ☛ *See "Configuring the LDAP server", page 68.*

3. (Optional) You can:
   - configure LDAP parameters

     > ☛ *See "Configuring an LDAP parameter", page 69.*

   - modify LDAP import parameters

     > ☛ *See "Modifying LDAP directory import content", page 70.*

When LDAP authentication has been configured:

- you can import persons from the LDAP directory.

  > ☛ *See "Importing persons from an LDAP server", page 71.*

- or you can manually map a **HOPEX** user group with a user group declared in your LDAP server.

  > ☛ *For more information on managing person groups, see the **HOPEX Administration - Supervisor Web** guide.*
  >
  > ☛ *When connecting to **HOPEX**, the authentication service uses the HOPEX Login and password of the user to authenticate the user with the list of available LDAP servers.*

Accessing LDAP server management

To access LDAP server management:

❱ From **HOPEX Administration,** in the **User Management** folder, right-click on **LDAP Servers** and select **Manage**.

> ☛ *See "Accessing User Management and UI Access Management Folders", page 44.*
>
> ☛ *The **LDAP Servers** folder is available only if you are connected with a user with the HOPEX Administrator profile (example: **Administrator**), see "The Administration profiles provided", page 19.*

Creating an LDAP server

The LDAP server is the server on which the LDAP directory is installed.

The LDAP directory can be an Active Directory directory.

To create an LDAP server:

1. Access LDAP server management.

   > ☛ *See "Accessing LDAP server management", page 67.*

2. In the LDAP server menu bar, click **New** ⊕.

3. In the creation of LDAP server dialog box, enter the **Name** of the LDAP server and click **OK**.
   The new LDAP server appears in the list of LDAP servers.
   You must configure the LDAP server, see "Configuring the LDAP server", page 68.

Configuring the LDAP server

LDAP server configuration is restricted to users with HOPEX Administrator or User Management Administrator profile.

To configure an LDAP server:

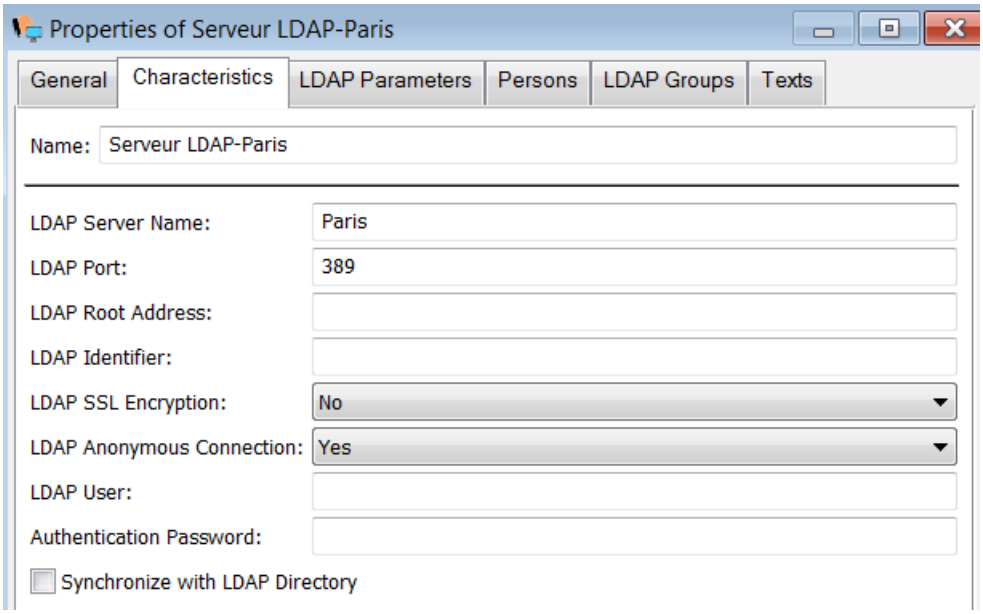
Prerequisite: the LDAP server is already created.

See "Creating an LDAP server", page 67.

1. Access LDAP server management.

   See "Accessing LDAP server management", page 67.

2. Select the new LDAP server and click **Properties** .



3. In the **Characteristics** tab, complete the following fields:
   - **LDAP Server Name**: name of the server hosting the LDAP directory.
   - **LDAP Port**: LDAP communication bridge

     `Example: 389`
   - **LDAP Root Address:** root address of LDAP server. This is an important attribute to limit query for a user in the LDAP directory or to address a particular forest.
   - **LDAP Identifier**: this is the LDAP attribute enabling unique identification of a user

     `Example: SAMAccountName, UID`
   - **LDAP SSL Encryption**: select **Yes** if you want LDAP directory connection to be SSL protocol encoded
   - **LDAP Anonymous Connection** : if you select **No**, you must specify the user via which LDAP directory connection will be made, as well as the user password

     ☛ *Only an administrator user can connect anonymously to an LDAP server.*
   - **LDAP User**: enter the identifier of the LDAP user used for LDAP directory connection. If connection is anonymous, this field should not be completed.

     ☛ *This user must have reading rights on data that **HOPEX** needs to access (example: LDAP person group, membership of a group in LDAP, e-mail in LDAP, etc.).*
   - **Authentication Password**: enter the password of the LDAP user used for LDAP directory connection. If connection is anonymous, this field should not be completed.
   - (if needed) select **Synchronize with LDAP Directory**, to synchronize LDAP parameters defined on the LDAP server (HOPEX) with the updates of the LDAP directory parameters.

     ☛ *Voir "Configuring an LDAP parameter", page 69*
4. Click **OK**.
   The LDAP server is configured.

   You can also:
   - configure an LDAP parameter, see "Configuring an LDAP parameter", page 69.
   - modify content of LDAP directory import, see "Modifying LDAP directory import content", page 70.

Configuring an LDAP parameter

An LDAP parameter is a parameter that exists in the LDAP directory and is associated uniquely with a **HOPEX** attribute.

Configuring an LDAP parameter is useful when importing persons from an LDAP directory. This configuration enables initialization of attributes (of the person or login created in **HOPEX**) corresponding to parameters with values stored in the LDAP directory.

```
Example: the "E-mail address" MetaAttribute of the person
is initialized with the "mail" LDAP parameter of the person
```

```
in the "Active Directory" LDAP directory (if mapping has
been carried out).
```

To configure an LDAP parameter:

1. Access LDAP server management.

   ☛ *See "Accessing LDAP server management", page 67.*

2. Select the LDAP server for which you want to configure an LDAP parameter and click **Properties** .
3. In the **LDAP Parameters** tab, click **New** .

   ☛ *The LDAP parameter enables pre-completion of characteristics of a person corresponding to the LDAP parameters.*

4. Enter the **Name** of the LDAP parameter (example: Mail), then click **Properties** .
5. (Optionally, access the "expert" metamodel) Select **Index on Persons**, so that the parameter value enables unique identification of a person. If a person in **HOPEX** has the same e-mail as a person defined in the LDAP directory, this person is reused (instead of creating a new person and risk duplicating the same person).
6. (Optionally, access the "expert" metamodel) Select **Is available for search** so that an e-mail address can be entered in the import entry area.

   ```
   Example: if you enter ctodd@mega.com, you should find Clara
   TODD.
   ```

7. In the **MetaAttribute** field, click the arrow and select **Connect MetaAttribute**.
8. Execute a query on the MetaAttribute (example: E-mail).
   When importing persons from the LDAP directory, the LDAP parameter (example: mail) will initialize the MetaAttribute (example: E-mail address).

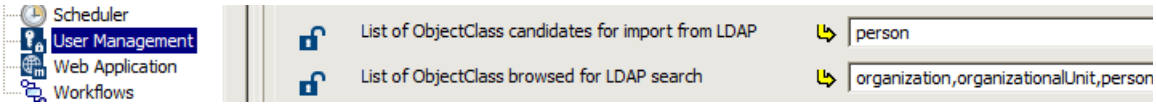
Modifying LDAP directory import content

You can modify LDAP directory import content:

- export candidate objects:
  This option enables definition of the type of objects to be imported from the LDAP directory.

  Default value: person.
- the list of objects browsed for LDAP query
  This option enables addition to the import of a particular person and/or persons of a team ("organization").

  Default value: organization,organizationalUnit,person



To define content of LDAP directory import:

1. Access the environment options management window.

   ☛ *See "Modifying options at environment level", page 326.*

2. In the options tree, expand the **Installation** folder and select **User Management**.
3. (Optional) In the right pane, modify the **List of ObjectClass candidates for import from LDAP** option.
   To import objects other than persons (default value), for example resources or org-units, specify this in this field. Objects should be separated by commas.

   Everything that is imported creates occurrences of persons with login.
4. (Optional) In the right pane, modify the **List of ObjectClass browsed for LDAP query** option.
   To add a person or organization to the import, enter the name of the person or organization (example: Quality) in the field.

   The result is the list of ObjectClass candidates for import from LDAP, that is, persons by default.

Importing persons from an LDAP server

The import of persons from an LDAP directory enables initialization of attributes (of the person or login created in **HOPEX**) corresponding to parameters with values stored in the LDAP directory.

☛ *See "Configuring an LDAP parameter", page 69.*

```
Example: the "E-mail address" MetaAttribute of the person
is initialized with the "mail" LDAP parameter of the person
in the LDAP file (if mapping has been carried out).
```

An LDAP parameter is a parameter that exists in the LDAP directory and is associated uniquely with a HOPEX attribute. For example, "mail" is a parameter of the "Active Directory" LDAP directory and can be associated with the e-mail of the person.

To import persons from an LDAP directory:

1. Open the user management window.

   ☛ *See "Opening the User Management window", page 45.*

2. In the **Persons** tab, click **Import From LDAP**.
3. The **LDAP Import Wizard** appears.
4. In the **LDAP Server** field, click the drop-down menu and select the server from which you want to import persons.

   ☛ *The LDAP server must be created, see"Creating an LDAP server", page 67.*

5. In the **Queried Element** field, enter the queried character string.

   ```
   Example: Support service.
   ```

6. Names resulting from the query are listed.
7. Select the persons you want to import.
8. Click **OK**.

Authentication and a user created on the fly (Web Front-End)

Creation of a user on the fly is not available for connection to HOPEX (Windows Front-End).

When a user has been created on the fly, the LDAP parameters can be used as indexing identifier (**Index on Person** attribute, see "Configuring an LDAP parameter", page 69) to check that a person with an attribute with the same value as the LDAP directory already exists in **HOPEX**.

Example of use:

Anne, responsible for sending questionnaires, wants to send a questionnaire. If one of the addressees does not exist:

- Anne can create the person (example: "Thomas KOCH" with e-mail "tkoch@mega.com")
- Anne cannot create the login of Thomas Koch since she is not an administrator.

When Thomas KOCH connects to **HOPEX** (Web Front-End), with tkh:

1. The authentication service authenticates tkh with the LDAP directory: the "mail" parameter exists and is indexing identifier type (**Index on Person** is selected),
2. The authentication service checks if a person already has this e-mail.

   Answer: Yes.

3. The authentication service creates the login associated with the person.

If Thomas KOCH has assignments to complete the questionnaire, he can connect to the application to complete this questionnaire.

MANAGING REPOSITORIES

A **HOPEX** repository constitutes the workspace in which modeling data is stored. Several users can connect to it and work simultaneously on the same project.

By default, any user of an environment can access all repositories of the environment. The **HOPEX** administrator can restrict these rights.

Repository management is carried out with the **HOPEX Administration** application.

The **HOPEX** repository server type has an influence on the functionalities available (for example: management of repository snapshots and collaborative workspace.

☛ *Certain recommendations may concern only RDBMS or GBMS type storage. Absence of a warning indicates that the recommendation is valid for all types of storage.*

This chapter covers points relating to working in a repository and the use of repositories:

✓ "Using Repositories", page 74
✓ "Managing Repositories", page 83
✓ "Optimizing Repository Access Performance", page 114
✓ "Referencing and Unreferencing a Repository", page 129

USING REPOSITORIES

A repository depends on an *environment*. Different policies of data distribution can be implemented. You can, for example, work on two repositories:

- a Development repository, which groups all projects
- a Production repository, which groups stable states of each project.

The following points are covered here:

- "Accessing Repositories", page 74
- "Repository Structure", page 75
- "Creating a Repository", page 78
- "Consulting and Modifying Repository Properties", page 79
- "Protecting the referencing of a repository", page 130
- "Consulting RDBMS Repository Performance", page 82
- "Generating a Health File for an RDBMS Repository", page 82
- "Accessing the Log of Repository Changes (.EMV file)", page 82

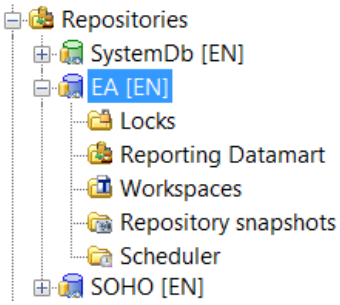
Accessing Repositories

To access a repository:

1. Connect to **HOPEX Administration** and select the environment concerned.

☛ *See "Connecting to an Environment", page 7.*

2. Expand the **Repositories** folder.
   The list of the repositories in the environment appears.



Each repository is represented by:

- an icon defining its storage type
- its installation language.

Repository storage type			
GBMS		***RDBMS***	
	MEGA proprietary format		ORACLE
			SQL Server

Repository Structure

The files enabling use of a repository are all in the same folder, the location of which is stored in the *system repository* (SystDb).

> *SystDb is a particular repository containing the metamodel and technical data (descriptors, Web site templates, queries, etc.). The metamodel and technical data are common to all repositories in the same environment. Definition of users and their rights are stored in this repository, essential for operation of the software.*

By default, a **HOPEX** repository is stored in GBMS format (**MEGA** proprietary format). You can choose another storage type for your **HOPEX** repository: an RDBMS format (relational database management system) such as Oracle or SQL Server.

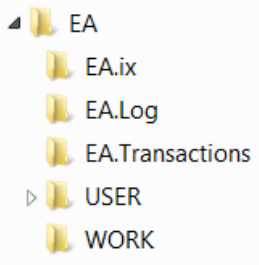
> *For more details on structure and the advantages of the different server types, see **RDBMS Repository Installation** deployment guide.*

A repository comprises "RepositoryName".XXX files of which type and format XXX depend on the repository server type:

	HOPEX repository storage type		
Information type	**GBMS**	**RDBMS**	
		Oracle	**SQL Server**
Semaphore of administration	.EMA (protects simultaneous accesses to the EMS)	N/A	N/A
Managing active private workspaces	.EMS	N/A	N/A
Repository evolution logfile (creation, update, etc.)	.EMV	.EMV	.EMV
Data	.EMB inseparable from .EMA, .EMS and .EMV files)	.EMO*	.EMQ*

*: this file does not contain data; it is a pointer to data stored on the relational database server.

For each repository, folders are created dynamically while the user is working:



- ***Approve***: contains validated reports (MS Word) detached from **HOPEX**.
- ***Data:*** contains all business documents created in **HOPEX**.
- ***Document***: contains all reports (MS Word) generated from **HOPEX**.
- ***RepositoryName.ix***: contains the result of indexing for quick search.

  ☛ *See "To allow the user to execute quick searches in HOPEX solutions, the repository must first be indexed.", page 80.*

- ***RepositoryName.Lock*** (GBMS repository): contains locks created on objects during user working. These files have .EMK extension.

  ☛ *See "Managing locks", page 161.*

- ***RepositoryName.Log*** (repository backup logfile): contains repository backup logfiles (enabled by default). Each private workspace is saved in logfile, .MGL format.

  ☛ *See "Backup logfile", page 83.*

- ***RepositoryName.Transactions*** (private workspace backup logfile: contains files linked to active private workspaces of repository users.

  ☛ *The presence of these files depends both on the server type used and on the options enabled for this repository.*

  These files are:

  - "CCC.MGL" files, where CCC is the code associated with the user (the **Backup logfile** option is enabled for the repository).

    ☛ *See "Backup logfile", page 83.*

  - "CCC.EMB" and "CCC.EMS" files (similar to those of the repository) containing data of the GBMS private workspace, where CCC is the code associated with the user.

- ***USER***: contains a folder for each user, in which all work files generated by the user are stored. It groups the files created by backup and extraction procedures, as well as control files.

  ☛ *Each folder is named "CCC", where CCC is the code associated with the user.*

- ***WORK***: contains work files created by administration operations carried out using the administration application.

Creating a Repository

You can create:

- an empty repository, then you can transfer data from another repository into it by importing updates or by restoring a backup file.
- (RDBMS specific) a repository already fed with data.

> ☛ *See **Installation and Deployment > RDBMS Repository Installation Guide > HOPEX RDBMS repositories specific administration actions > Restoring a HOPEX environment from formatted data > Restoring a data repository**.*

To create an empty repository:

1. Connect to **HOPEX Administration** and select the environment concerned.

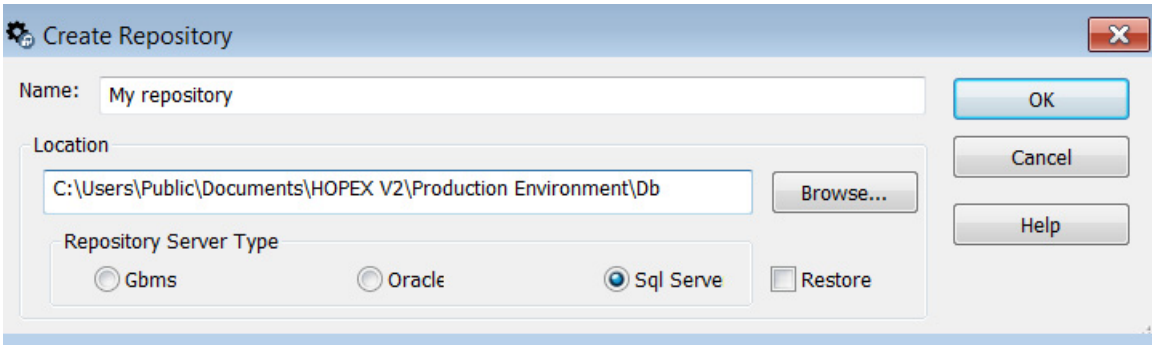
   > ☛ *See "Connecting to an Environment", page 7.*

2. Right-click the **Repositories** folder and select **New**.
   The **Create Repository** dialog box opens.
3. Enter the **Name** of the repository.
   The repository name should start with a letter and not exceed 60 characters. It can contain letters (A-Z), figures (0-9) and certain special characters (_$#).

   > ☛ *The name of an Oracle repository should not be an Oracle reserved word (see DBA Oracle).*

   > ☛ *The name of an SQL Server repository is prefixed by the environment name. This prefix is not included in the 60 characters limit.*

   > ☛ *If you use strange characters, you must be in the same system language as these characters so that they will be recognized by **HOPEX**.*



4. (Optional) By default, the repository is created in the **Db** folder of the selected environment; to modify this, click **Browse** and define another location.

5. Select the repository server type and click **OK**.

   ☛ *For more details on repository server type, see **RDBMS Repository Installation Guide**.*

   The repository folder and its files are created.

   ☺ *At creation of a repository, a sub-folder carrying the name of your new repository is automatically created in the **Db** folder of the environment folder to store corresponding repository files.*

   The repository created is empty. You can transfer data from another repository into it by importing updates or by restoring a backup file.

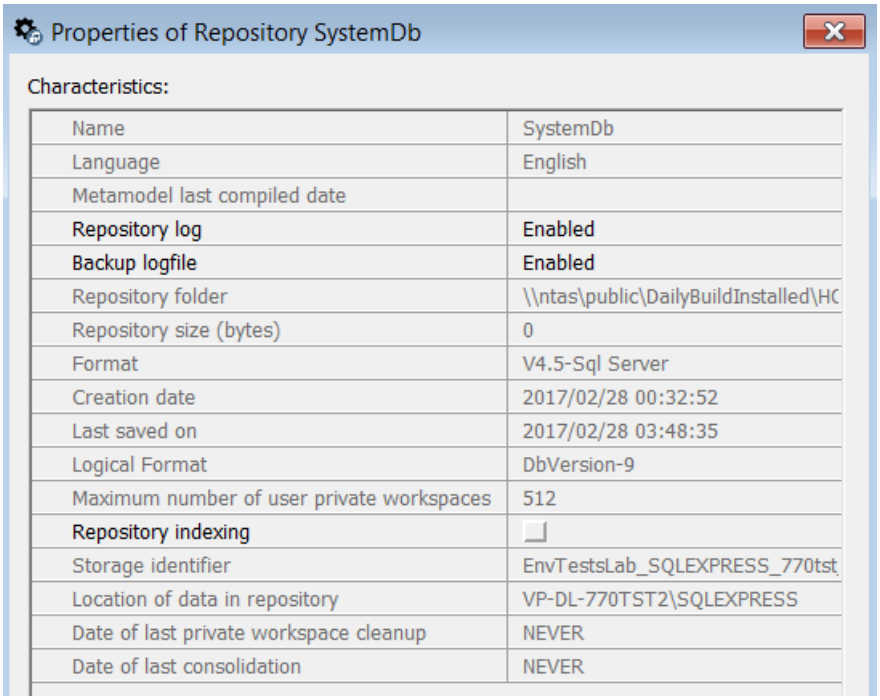
Consulting and Modifying Repository Properties

To consult and/or modify certain repository properties:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the concerned repository and select **Properties**.
   The **Repository Properties** dialog box opens.



From the **Repository Properties** dialog box, you can:

- click the row of one of the following characteristics to:
  - modify the **Name** of the repository
  - modify the repository **Language**

    Only if you need to modify the repository language right after its creation. Never change the repository language at any time, to avoid irreparable loss of object names.

  - enable/disable the **Repository log**

    For more details, see "Enabling the repository log", page 85.

  - enable/disable the **Backup logfile**
- consult its following characteristics:
  - **Metamodel last compiled date** (SystemDb repository specific)
  - location of the **Folder** in which the repository is stored
  - **Repository Size** presents the size of the .EMB file of the GBMS database.
  - **Format** is information used only for internal operation of **HOPEX** on GBMS repositories.
  - **Dates** of repository creation and last save
  - repository **Logic Format**.
  - **Date of last private workspace cleanup** and **Date of last consolidation** (RDBMS specific), which provide useful information for the maintenance plan of an RDBMS repository.

    See "Deleting RDBMS Repository Temporary and Historical Data", page 123.

- enable/disable **Repository Indexing**.

  See "Enabling and Customizing Repository Indexing", page 80.

Enabling and Customizing Repository Indexing

To allow the user to execute quick searches in **HOPEX** solutions, the repository must first be indexed.

Indexing runs automatically every 10 minutes (modifiable default value) with the indexing scheduler (RDBMS specific). This indexing is incremental (indexing modified objects only).

*To modify scheduler default configuration, see "Customizing the indexing scheduler (RDBMS)", page 81. For detailed information on the scheduler, see the technical article **HOPEX Power Studio - Scheduler**.*

*The initial indexing can take some time depending on the size of your repository (eg: more than 30 hours for a 2 GB repository, of which business documents constitute a major part), and can slow the performance of **HOPEX**. Remember to run this initial indexing when other **HOPEX** users are not connected.*

Take care to allow sufficient disk space before enabling indexing: statistically for a large repository (eg: 2 GB) of which business documents constitute a major part, the indexing size can represent twice the repository size.

Enabling/Disabling repository indexing for quick search

By default, repository indexing is not enabled.

To enable/disable repository indexing:

1. Access properties of the repository concerned.

   ☛ *See "Consulting and Modifying Repository Properties", page 79.*

2. Select/Clear **Repository Indexing** to enable/disable repository indexing.
3. Click **OK**.
   The scheduler (RDBMS specific) updates indexing every 10 minutes.

   ☛ *To modify scheduler default configuration, see "Customizing the indexing scheduler (RDBMS)", page 81.*

   Indexing is carried out for all of the languages installed.

   When indexing is completed, the "Repository Name".IX folder is created in the corresponding folder of the repository. This folder contains indexing results.

Indexing a repository manually

When the scheduler is unavailable (GBMS), or for initial indexing, the administrator can index the repository manually.

To index a repository manually:

☛ *To be able to manually index the repository, the repository indexing option must be selected, see "Enabling/Disabling repository indexing for quick search", page 81.*

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository to be indexed and select **Index for Quick Search**.
   Indexing is carried out for all of the languages installed.

   ☛ *For initial indexing, the "Repository Name".IX folder is created in the corresponding folder of the repository. This folder contains indexing results.*

Customizing the indexing scheduler (RDBMS)

You can modify indexing scheduler default configuration.

To modify indexing scheduler default configuration:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click **Scheduler** and select **Manage Triggers**.
3. Select the **System Triggers** tab.
4. Right-click **Indexing Automaton** and select **Update Scheduling**.
   The indexing scheduler configuration window appears.
5. Modify the configuration.

6. Click **OK**.

   ☛ *For detailed information on the scheduler, see the technical article **HOPEX Power Studio - Scheduler**.*

Consulting RDBMS Repository Performance

Before starting work in an RDBMS repository, **MEGA** recommends that you run the RDBMS diagnostic tool.

This tool indicates repository performance compared to optimized performance.

To run the RDBMS repository diagnostic tool:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the RDBMS repository concerned and select **RDBMS Administration > RDBMS Diagnostics**.

   ☛ *For more information on the RDBMS repository diagnostic tool, see deployment documentation: **RDBMS Repository Installation Guide**.*

Generating a Health File for an RDBMS Repository

HOPEX enables you to generate a file containing information on the fragmentation and the statistics of an RDBMS repository.

To generate a health file for an RDBMS repository:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the RDBMS repository concerned and select **RDBMS Administration > Information about fragmentation and statistics**. A report file is generated with the information returned by Oracle or the SQL Server on the repository health state.

Accessing the Log of Repository Changes (.EMV file)

The .EMV file contains repository changes (eg: creation, update).

To directly access the .emv file of a repository:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **See EMV File**.

MANAGING REPOSITORIES

☛ *For management operations specific to an RDBMS repository, see deployment documentation:* ***RDBMS Repository Installation Guide****.*

The following points are covered here:

- "Managing logfiles", page 83
- "Configuring the logging for an inter repository consolidation", page 85
- "Viewing the Repository Update Log", page 86
- "Exporting Updates", page 89
- "Deleting a Repository", page 89
- "Converting a Repository", page 90
- "Importing Libraries into a Repository", page 90
- "Repository Physical Backup", page 91
- "Checking and Physically Restoring a GBMS Repository", page 93
- "Reorganizing a repository", page 96
- "Repository Logical Backup", page 97
- "Duplicating a Repository", page 100
- "Initializing an Existing GBMS Repository", page 102
- "Updating a Repository", page 103
- "Viewing the Environment Report File", page 108
- "Viewing User Process Error Trace Files", page 110
- "Saving the Error Zip file for Diagnostics", page 112
- "Viewing Object History", page 112

Managing logfiles

At repository creation, by default:

- the backup logfile is **Enabled**
- the repository log:
  - (GBMS repository) is **Disabled**
  - (RDBMS repository) is **Enabled**

See:

- "Backup logfile", page 83
- "Backup logfile process", page 84
- "If you have a problem", page 84
- "Enabling the repository log", page 85

Backup logfile

The repository is configured so that changes made by users are saved simultaneously in the repository and/or in a specific file called the *backup logfile*.

Backup logfile process

When opening a private workspace, the backup logfile process is as follows:

Step	User A...	Result
1	...connect to **HOPEX.**	A private workspace opens. XXX.MGL logfile is created in folder \Db\RepositoryName\RepositoryName.Private Workspaces of the environment directory.
2	...dispatches work.	XXX.MGL logfile is consolidated (*): under YYYY-MM-DD_hh.mm.ss_XXX.mgl name in "\Db\RepositoryName\RepositoryName.Log\ folder
	...discards work.	There is no copy of XXX.MGL logfile.
3	...reconnect to **HOPEX**	A new private workspace opens. XXX.MGL logfile of folder RepositoryName.Transactions is reinitialized

XXX: Code of User A

YYYY-MM-DD: dispatch date (year-month-day)

hh.mm.ss: dispatch time (hour.minute.second)

(*): Consolidation consists of search and deletion of useless commands. For example:

```
When an object is created then deleted, the information is
not saved in the consolidated logfile.

When a comment is created, then modified several times, only
the final modification is saved in the consolidated
logfile.
```

☺ *To avoid overloading the* RepositoryName.Transactions directory, MEGA recommends that you regularly archive the old logfiles.

Modifications made by a user to the system repository are logged in the same way.

If you have a problem

In the event of a problem on the repository, you can restore the last repository backup and import backup logfiles saved in the RepositoryName.Log folder from this backup up to time of the problem.

☛ *It is highly recommended that you synchronize backup logfile archiving with repository backup.*

Dispatched modifications are logged in a new repository log.

In the event of a problem on a user private workspace, you can copy the user backup logfile saved in the RepositoryName.Log folder, delete the current private workspace and import the backup logfile in a new private workspace.

Enabling the repository log

By default, the repository log is disabled for a GBMS repository and enabled for an RDBMS repository.

To keep a history of the actions performed on the repository after dispatch, you must activate the repository log.

> **Enabling the repository log generates a large volume of objects. This volume can adversely affect HOPEX performance, particularly at dispatches. It is not recommended with a GBMS repository.**

To enable the repository log:

1. Open the **Properties** dialog box of the repository.

   See "Consulting and Modifying Repository Properties", page 79.

   The repository properties dialog box opens.
2. Click **Repository Log** line and select **Enabled**.

   The repository log is enabled.

The repository log lists all modifications made in the repository. It gives users a better understanding of actions dispatched in the repository from private workspaces.

Each time an action is executed, an occurrence of Change Item is created.

> *A **ChangeItem** is a MetaClass corresponding to a change made in a **HOPEX** repository.*

A repository log comprises **HOPEX** occurrences. These occurrences can be handled using **MEGA** APIs.

See "Viewing the Repository Update Log", page 86.

Configuring the logging for an inter repository consolidation

To improve performance you can define some MetaClasses or MetaAssociationEnds as non loggable.

See "Modifying MetaClass loggability", page 119.

Logging

Logging of updates enables:

- mainly to view the repository activity, i.e. actions performed on objects.

  ☛ *See "Viewing the Repository Update Log", page 86.*

  In that case, an incomplete (**Consolidate** command) or truncated (**Delete** command) log is functionally satisfactory.

  ☛ *See "Deleting a log or reducing the log size", page 115 step 4.*

  This is the default configuration.
- to transfer the commands performed from a repository to another.

  `Example: you want to transfer all the commands performed during the day from a development repository to a production repository.`

  In that case a complete log, including all the actions performed by the users, is necessary so that the inter repository *consolidation* is performed correctly. You must modify the default log behavior.

  ☛ *See "Modifying the log behavior", page 86.*

Modifying the log behavior

To include in your log the MetaClasses and MetaAssociations defined as non loggable, you must modify the log behavior.

To modify the log behavior:

1. Access the options.

   ☛ *See "Accessing Options", page 326.*
2. In the Options tree, select **Repository**.
3. In the right pane, for the **Log behavior** option, select "Logging all updates" value.

   All of the updates are included in the logs, including MetaClasses and MetaAssociations that are defined as non loggable.

   ☛ *See "Modifying MetaClass loggability", page 119.*

Viewing the Repository Update Log

You can view the history of the actions performed in the repository after dispatch in:

- **HOPEX Administration** application (Windows Front-End)

  ☛ *See "Viewing the Repository Update Log", page 87.*
- **HOPEX**(**Repository Activity** navigation window)

  ☛ *See "Displaying dispatches", page 88.*

  ☛ *See "Viewing the Repository Update Log", page 87.*
- the **Administration** desktop (Web Front-End)

  ☛ *See the **HOPEX Administration - Supervisor Web** guide, section "Displaying Updates Made in the Repository".*
- the object **History**

Viewing the Repository Update Log

To view the repository update log:

1. Connect to **HOPEX Administration** and select the repository concerned.

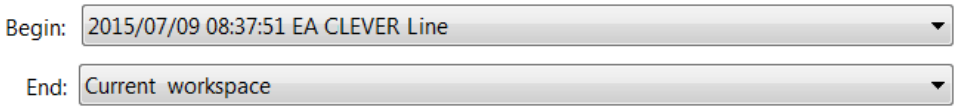
   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Repository Log > Open**.
   The **View repository updates log** appears.

   ☛ *From **HOPEX** (with access to the **Expert** metamodel level, see "Configuring Metamodel Access", page 60), select the **File > Properties** menu. In the dialog box that appears, select the **Update** tab. The log of the current private workspace is displayed by default.*

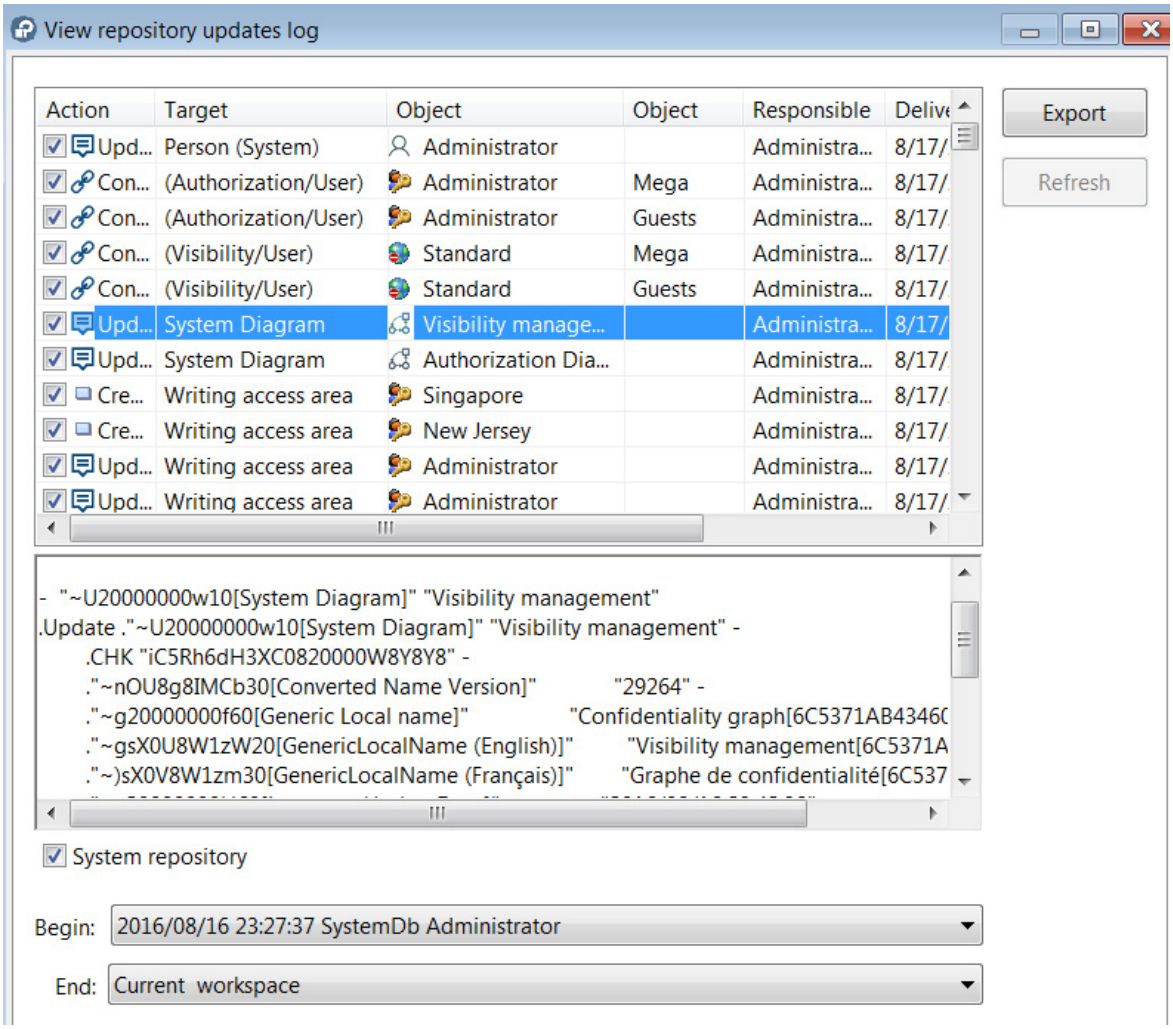
3. In the **Begin** and **End** fields, select the dispatch interval that you want to display.

   ☛ *You can view the repository log in its totality, or between two given dispatches. In the **Begin** drop-down list, the first element corresponds to the first dispatched private workspace of this repository.*



4. Click **Refresh**.
   The repository log appears as a list of actions displayed in chronological order.

5. Select (/Clear) **System Repository** to display (/hide) the system repository updates.



☛ *See "Exporting Updates", page 89.*

Displaying dispatches

To display in **HOPEX** private workspaces dispatched and the content of their updates:

☛ *To access the content of the **Repository Management** pane, you must have **Expert** metamodel access (see "Configuring Metamodel Access", page 60).*

1. Connect to **HOPEX**.

☛ *See "Connecting to HOPEX", page 136.*

2. In the **Repository Activity** Navigation window, expand the **Repository Dispatch** sub-folder.
   All dispatches performed on the current repository and the system repository are contained in the respective **<Repository Name>** and **SystemDb** folders.
3. Expand the repository folder concerned.
   Dispatches are sorted by day, week and month.
4. Expand the dispatch concerned.
   Each dispatch details the updates made to an object.
5. Expand the update concerned.
   In the properties of the object concerned, the **General > History** tab details the content of the dispatch in the form of a list of actions performed on the object concerned.

Exporting Updates

You can export updates between consecutive dispatches or not.

☛ *To export the repository log, see "Backing up the repository log", page 116.*

To export updates:

1. Access the repository update log.
   ☛ *See "Viewing the Repository Update Log", page 86.*
2. Select the updates to be exported.
3. Click **Export**.
4. Select the export format:
   - *.mgr: text format
   - *.xmg: MEGA XML format.
5. Click **Export**.
   You can open or save the export file.

Deleting a Repository

See:

- "Deleting a GBMS repository", page 89
- "Deleting an RDBMS repository", page 90

Deleting a GBMS repository

To delete a GBMS repository:

1. Connect to **HOPEX Administration** and select the repository concerned.
   ☛ *See "Accessing Repositories", page 74.*
2. Right-click the repository concerned and select **Delete**.
   A dialog box appears asking you for confirmation.
3. Click **Yes** to confirm deletion.

Deleting an RDBMS repository

To delete an RDBMS (SQL Server or Oracle) format repository, you need the appropriate administration rights.

- Oracle: you need at least an account equivalent to "system" to delete a schema.
- SQL Server: you need at least db_creator rights.

To delete an RDBMS repository:

1. Connect to **HOPEX Administration** and unreference the repository concerned.

   ☛ *See "Unreferencing a Repository", page 130.*

2. Ask the database administrator to delete the database or schema concerned from the RDBMS server.

Converting a Repository

Repository conversion is only necessary:

- at migration.

  ☛ *For more information on repository conversion, consult the technical note **How to migrate to HOPEX V2**.*

- on request from **HOPEX** support.

With **HOPEX** you can mass convert technical data. This possibility enables you to mass launch the procedure on all the repositories with no need to validate the execution for each repository.

Mass converting technical data

To mass convert technical data:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the environment folder.
3. Right-click the **Repositories** folder and select **Manage**.
4. In the **Action list** pane, select **Technical conversion**.
5. In the **Repository list** pane, select the repositories for which you want to convert technical data.
6. Click **Execute**.
   The procedure is automatically launched on each repository selected with no need to validate each execution.

   ☛ *A dialog box enables you to stop the procedure at any time. If needed, click **Cancel** to stop the procedure execution.*

Importing Libraries into a Repository

You can import the libraries you need for your work into a repository.

To import a library into a repository:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Object Management > Import**.
   The **Import HOPEX Data** dialog box opens.
3. Alongside the **Command File** field, click **Browse** ...
4. From the **HOPEX** installation folder, select in the **MEGA_Std** folder the desired Library (*.mol).
5. Click **Open**.
6. In the **Import HOPEX data** window, click **Import**.
7. When import is completed, click **Close**.

Repository Physical Backup

In event of a problem, you must have a valid and recoverable data backup.

A physical backup consists of copying the files of a repository from their original location to another one.

Data is stored at **HOPEX** environment level. Data to be backed up varies according to repository server type.

☛ *See "Repository Structure", page 75.*

The following points are covered here:

- "Backup frequencies", page 91
- "Elements to be backed up", page 92
- "Other elements to be backed up", page 93
- "Elements that could be useful to back up", page 93

Backup frequencies

For a **HOPEX** environment used by an active project, **HOPEX** recommends:

- daily backup of the environment
- backup before any major data update

  `Example: system database customization, data reprocessing, CP/RP update of HOPEX data.`

- that you keep:
  - daily backups of the last 30 days
  - monthly backups of the last 12 months

Whatever your repository server type, **MEGA** recommends cold backup (no **MEGA** user should be connected).

☛ *In SQL server type mode, hot backup is possible.*

Elements to be backed up

Identify environments that require regular backup.

```
Example: design environment.
```

From the environment folder, you must back up complete folders:

- **Db**
- **SysDb**
- **Mega_usr**

For repositories in format:

- GBMS
  Db and **SysDb** folders contain all EMA, .EMB, .EMS and EMV files of the system database and repositories.

  ☛ *The daily physical backup should include at least the .EMB, .EMS and .EMA files of the GBMS repository.*
- RDBMS
  Db and **SysDb** folders contain an .EMV file and (respectively) an .EMO .EMQ .EMY file that points to other folders that you must back up:
  - for Oracle:
    system database and repository schemas.
  - for SQL Server:
    system database and repository databases.

GBMS format repository recommendations

To obtain a valid backup, ensure that files are not accessible during backup:

1. Access environment options.

   ☛ *See "Modifying options at environment level", page 326.*
2. In the tree, select the **Options\Repository** folder.
3. In the right pane, for option **Authorize dispatch for the environment**, select "Prohibit".

Elements that can be excluded from the backup file

To save space and time it is not necessary to back up the complete content of the environment folder. You need not back up for example folders that contain:

- user work files
  These files are contained in the **SysDb\USER** and **Db\USER** folders.
- work files linked to the Administration application
  These files are contained in the **SysDb\WORK** and **Db\WORK** folders.
- lock files:
  - **Systemdb.Lock** contained in **SysDb** folder
  - **RepositoryName.Lock** contained in **Db\<RepositoryName>** folder

Other elements to be backed up

MEGA recommends that you back up folders concerning:

- configuration:
  the **Cfg** folder (in the **HOPEX** installation directory) containing the **megasite.ini** configuration file.
- licenses:
  the file containing licenses (.Must or .ELF).

(Optional) You can back up folders concerning:

- your java customization:
  lib_usr folder (in the **java** folder of the **HOPEX** installation directory)
- your installation customization:
  Mega_Usr folder
- your delivered data
  in the **HOPEX** installation directory, the following folders:
  - **Document**, which contains shared documents
  - **Intranet**, which contains the Web sites generated
  - **Approve**, which contains detached documents.

Elements that could be useful to back up

You may need to back up:

- private workspaces in progress
- technical data modifications in progress

To back up private workspaces in progress:

❱ copy the "RepositoryName.Transactions" folder in the repository folders tree.

> ☛ *The **USER** and **WORK** folders contain the work documents of users.*

To back up technical data modifications in progress:

❱ copy the **SysDb** folder and its sub-folders in the tree of the *system repository*.

Checking and Physically Restoring a GBMS Repository

This section describes how to:

- check a GBMS repository
- physically restore a GBMS repository.

> ☛ *A repository should be physically restored only in the event of system failure or to check that the backup is valid.*

Checking a GBMS repository

To facilitate the decision on repository reorganization, breakdown of volumes in the repository is presented during the check.

A repository check enables physical and logical verification of the repository on the main data storage file (.EMB extension). A repository check consists of:

- reading the file block by block
- checking the data
- checking all data indexing mechanisms contained in these data blocks.

As well as this physical integrity check, repository check gives a full technical report that can help the administrator decide whether or not to *reorganize* the repository to improve data access times (slow diagram loading, slow workspace opening, etc.).

To start the check:

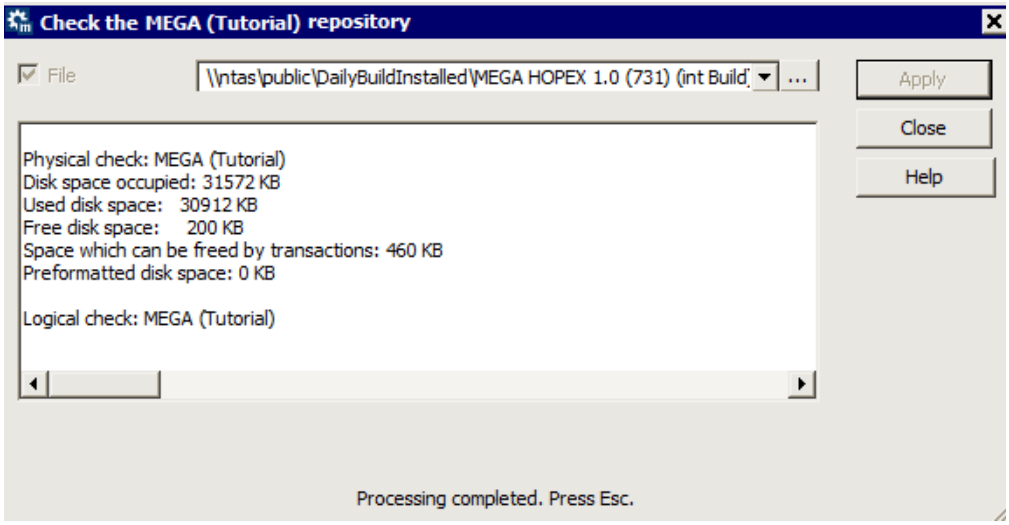
1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository and select **Check**.
   The **Repository Check** dialog box opens.
3. (Optional) Select the **File** option if you want to create a report file.
   By default when you run a check at level:
   - of a "Test" repository, the Testxxxx.txt file is generated in the Test folder (eg: MyEnvironment\db\test).
   - of the "systemdb" system repository, the SystemDb0000.txt file is generated in the sysDb folder (eg: MyEnvironment\SysDb).

   No writing access should occur during this process.

   This repository check does not check private workspaces. It does not detect defective private workspaces.
4. Click **Apply** to start checks.
   The report is displayed in the dialog box.



The following table explains the meaning of the different disk spaces used:

Type	Definition
Disk space occupied	Physical size of repository data file (.EMB). Sum of the used, free, reserved and freeable space. This space is not limited.
Disk space used	Disk space used by data limited to 4 GB.
Free disk space	Free disk space for consumption by progressive updates.
Freeable space by private workspaces	Space required for multi-user management and not compressible while older private workspaces remain (see "Managing workspaces", page 133).
Reserved space	Space temporarily reserved for updates without validation (generally zero).

From a physical standpoint, a repository is usually divided into four parts if it has never been reorganized:

- 80% usable space
- 10% or less free space
- 10% or less space that can be freed by private workspaces
- 0% preformatted space

☺ *We recommend that you reorganize the repository when the space occupied by its logical backup represents less than 3 times the disk size occupied by the repository itself or when space used by the data represents less than 20% of disk size occupied.*

For more information on repository reorganization, see "Reorganizing a repository", page 96.

Physical restore

A physical restore consists of copying previously saved repository files.

The following table shows different problems that may impact your repository. If any of these problems occur, you may have to *restore* your repository.

Cause	check
Problems of disk integrity	The following system tools can detect errors on your disk: - Run the Scandisk utility if Dos > V6 or Win95 - Run the server utility if server disk
Server saturation (lack of space, disk controller failure)	- Consult the server log. Other applications are also encountering errors.
Server disk controller is operating incorrectly	- Consult the server log. Other applications are also encountering errors.

Restore the *lock* file only if restoring recent files.

Reorganizing a repository

You can reorganize a repository:

- to optimize storage space and GBMS repositories performance.

  ☛ *This space increases rapidly when users begin accessing and updating the repository.*

- when you want to change storage type, for example: from GBMS repository to Oracle repository.

Before reorganizing a repository, you must check that there is no other active or passive private workspace on this repository, see "Workspace Administration", page 149.

The repository reorganization process is automated. It consists of:

1. physical backup of repository files (GBMS only).
2. logical backup of the repository, to obtain the *command file* which contains creation orders of repository objects and their links.
3. repository initialization.
4. repository restore by import of the command file in an empty repository.

Recommendations concerning GBMS repositories

Repository reorganization frequency depends on activity (dispatch frequency, number of users, functionalities used, etc.)

MEGA recommends that you:

- reorganize your repository:
  - twice a month for configurations of more than 100 users.
  - once every two months for smaller configurations.
- not forgetting to also reorganize the system repository SystemDb.

Reorganizing a repository

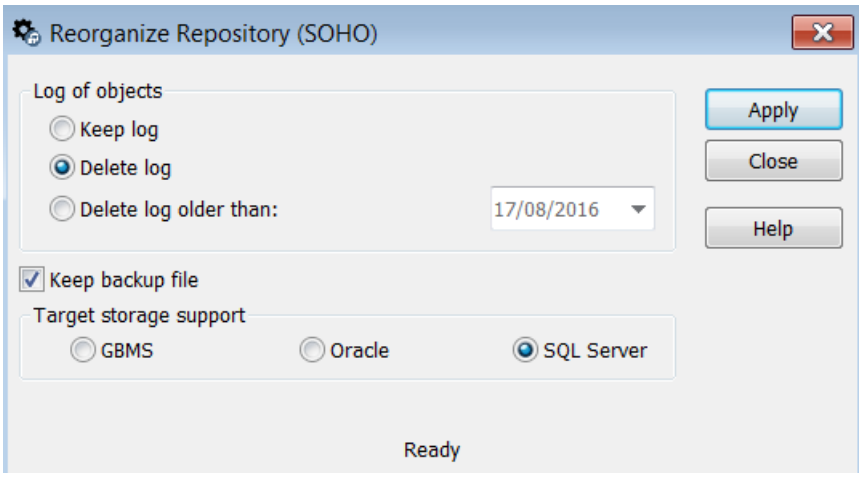
☺ *To improve reorganization times and **HOPEX** performance, remember to delete old log elements before repository reorganization.*

To reorganize a repository:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Reorganize**.



3. (Optional) In the **Log of objects** frame, you can select the actions to be executed on the log at reorganization.
4. (Optional) If you do not want to keep the repository log, clear **Keep backup file**.

   ☛ *The log is generated in the work folder WORK of the repository, in format Bkp_<YYYY-MM-dd_HH.mm.ss>_<repository name>.mgr*

   ☛ *By default, the target server type is the same as the repository server type.*
5. Click **Apply**.
   When the repository has been reorganized, a reorganization completed message is displayed.

Repository Logical Backup

Before you make a repository backup, dispatch all current private workspaces so that their changes are included in the backup. To view private workspaces active on your repository, use the private workspaces administration tool, see "Workspace Administration", page 149.

Logical *backup*:

- creates a *command file* that allows you to reconstruct the repository by update of an empty repository.
- analyzes all repository content.
- is safer than a physical backup since it checks that all data in the repository is readable.
- can be used as a long term archive or to merge repositories.

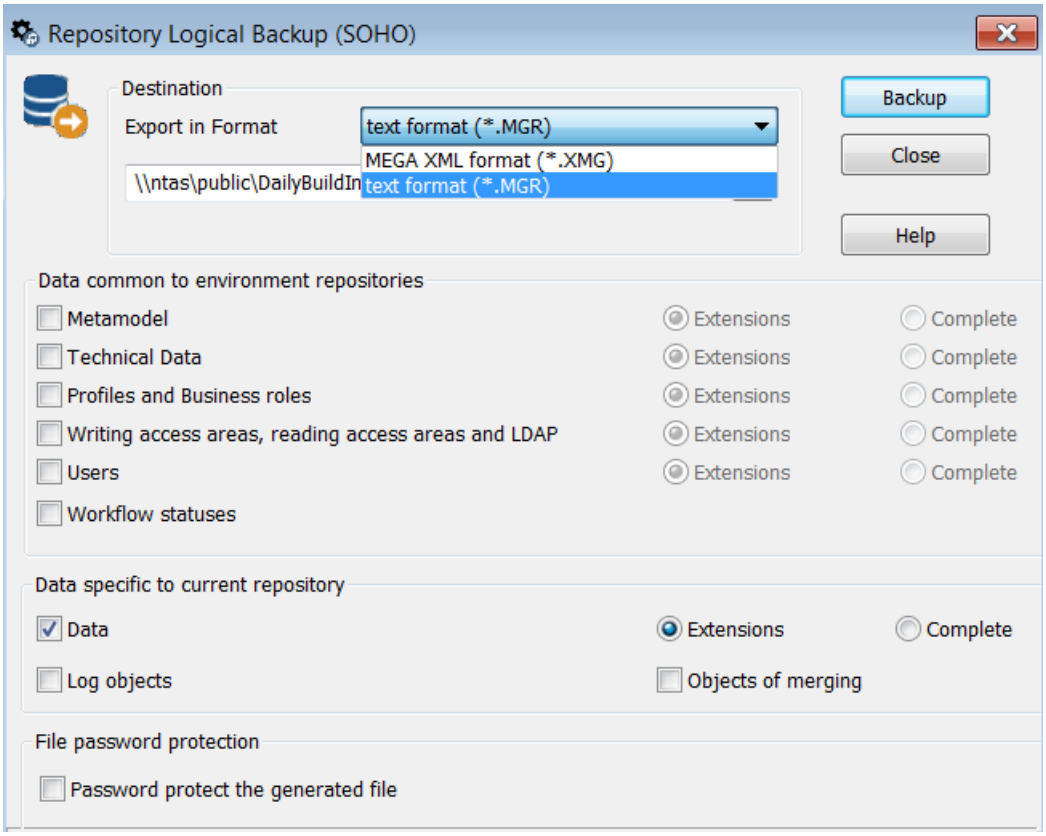
  You can execute updates on the repository after starting logical backup by dispatching a private workspace. Note,

however, that these modifications are not included in the backup.

☺ *You can temporarily prohibit users from updating the repository by clearing the **Authorize Dispatch** option in the environment or user options.*

To execute a logical backup of a repository:

1. Connect to **HOPEX Administration** and select the repository concerned.
2. Right-click the repository concerned and select **Logical Backup**.
   The **Repository Logical Backup** dialog box opens.



3. In the **Destination** frame, select the export format of the backup file:
   - **text format** (*.MGR).

     ☺ *For more details on .MGR file syntax, see "Command File Syntax", page 307.*
   - **MEGA XML format** (*.XMG)
     This format is reserved for exchange of data between **HOPEX** and other applications. It includes commands or data (objects and links).

This format cannot be used to extract the metamodel or technical data.

☺ *For more details on MEGA XML data exchange format, see technical article **MEGA Data Exchange XML Format EN**.*

☛ *Environment options enabling configuration of **HOPEX** data export (XMG encoding, default export format, etc.). See "Managing Options", page 323.*

4. (Optional) In the **Destination** frame, click **Browse** ... to browse the folder tree and modify the name and/or location of the backup file.
By default, the backup creates a file "RepositoryName.mgr" in the WORK *work folder*.

☛ *We recommend that this backup be made to a physical device other than the device on which the repository is located. Selection of a different logical device, such as a different partition on the same disk, does not protect your backup in the event of disk failure.*

5. Select the type of data you want to save.

☛ *The **Extensions** buttons correspond to data created by users.*

Carry out a complete backup only if technical support asks you to do so.

☺ *You can choose what you want to save: extensions common to environment repositories and data specific to current repository. It is recommended that you save all these elements in separate specific files, with names indicating their contents.*

☛ *For a standard logical backup, simply select the **Data** and its **Extensions**.*

In the frame **Data common to environment repositories**, select the data type of the system repository to be saved:

- **Metamodel** allows extraction of the *metamodel*, if the standard metamodel has been modified.
- **Technical Data** allows extraction of data such as *descriptors*, *queries*, and *report templates (MS Word)*.

  A complete logical backup of the repository including the Technical Data can take time and occupies considerable space.

- **Profiles and Business roles** allows extraction of created profiles and business roles (those not provided by MEGA).
- **Writing access areas, reading access areas and LDAP** allows extraction of created writing and reading access areas (those not

provided by MEGA) and LDAP parameters (parameters, servers, groups).

- **Users** allows extraction of created users (persons, person groups, logins).
- **Workflow statuses** enables extraction of workflows (workflow instances, transitions and statuses, tasks, validations, requests for change).

In the **Data specific to current repository** pane:

- **Data** allows extraction of repository data. This data includes assignment of business roles to persons.
- **Log objects** allows you to include object histories in the extraction of **HOPEX** data.

  ☛ *For more information on object logs, see "Viewing Object History", page 112.*

- **Objects of merging** allows you to export technical objects resulting from merging objects (_TransferredObject).

  ☛ *For further information on merging objects, see "Merging Two Objects", page 225.*

6. (If needed) In the **File password protection** pane, select **Password protect the generated file**.

   ☛ *The password is asked at file import.*

7. Click **Backup** to start backup.
   During execution of backup, a series of messages keeps you informed of progress.
   The backup report is displayed in the **Report** area.

   ☛ *For more information on this file, see "Viewing the Environment Report File", page 108.*

To update a repository, import the repository backup file.

☛ *For more details on importing a command file, see "Updating a Repository", page 103.*

Duplicating a Repository

See:

- "Duplicating a GBMS repository", page 100
- "Duplicating an Oracle repository", page 101

Duplicating a GBMS repository

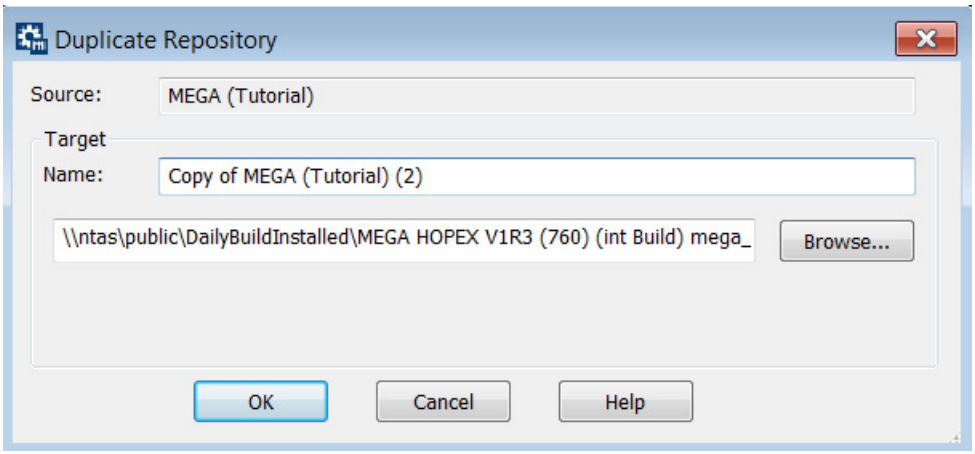
Prerequisite: before duplicating a repository, check that there is no other active or passive private workspace on this repository, see "Workspace Administration", page 149.

To duplicate a GBMS repository:

1. Connect to **HOPEX Administration** and select the GBMS repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Duplicate**.
   The **Duplicate a Repository** dialog box appears.



   The name of the **Source** repository is indicated.
3. In the **Name** field of the **Target** frame, enter the name of the repository to be created.
4. (Optional) Click **Browse** to select the location where the repository will be created.
5. Click **OK**.

Duplicating an Oracle repository

> ☛ *For detailed information on duplicating an environment and an RDBMS repository, see the **RDBMS Environment Duplication** technical article. See also deployment documentation: **RDBMS Repository Installation Guide**.*

Prerequisite: before duplicating a repository, check that there is no other active or passive private workspace on this repository, see "Workspace Administration", page 149.

To duplicate an Oracle repository:

1. Connect to **HOPEX Administration** and select the Oracle repository concerned.

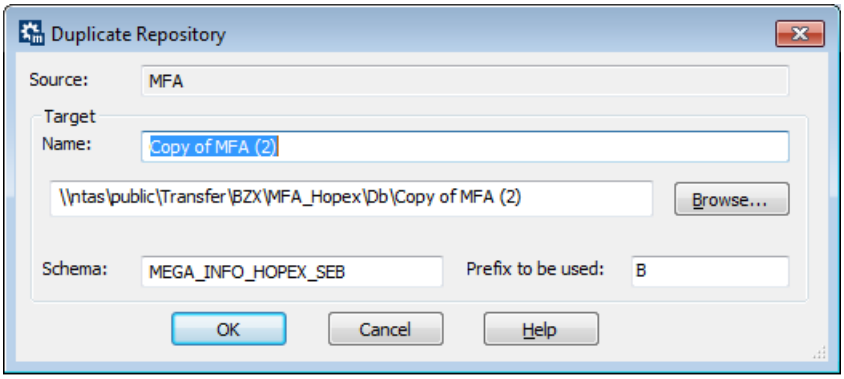
   > ☛ *See "Accessing Repositories", page 74.*

2. Right-click the RDBMS repository concerned and select **RDBMS Administration > Duplicate Oracle Storage**.
   The **Duplicate a Repository** dialog box appears.
   The name of the **Source** repository is indicated.
3. (Optional) In the **Name** field of the **Target** frame, modify the name of the repository to be created.
4. (Optional) Click **Browse** to select the location where the repository will be created.
5. (Optional) In the **Schema** field, modify the schema.

   > ☛ *The source schema is used by default. You should use a different schema for a different repository.*

6. (Optional, if you have kept the name of the source schema) In the **Prefix to be used** field, enter a prefix to distinguish the source schema from the target schema.

   ☛ *The prefix must start with an alphabetic character.*



7. Click **OK**.
8. In the connection window, enter the password to access Oracle.
9. Click **Test connection**.
   You must validate the test before continuing the procedure.
10. Click **Test GRANTs**.
    You must validate the test before continuing the procedure.
11. Click **OK**.
    The repository is duplicated.

Initializing an Existing GBMS Repository

To reinitialize a GBMS repository:

1. Connect to **HOPEX Administration** and select the GBMS repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Initialize**.
   This empties the repository of its data so that a backup can be reinserted, for example.

   A message box asks for confirmation.
3. Click **Yes** to confirm reinitialization.

   This operation deletes all repository data irreversibly.

   ☺ *Initialization of the GBMS repository, as well as repository reorganization, keeps the absolute identifier of the processed repository. This can be useful if you use the absolute identifier of your repository in a macro or query.*

   Repository reinitialization does not keep repository configuration options. You must update configuration of the newly-initialized repository.

Updating a Repository

Importing command files

You can update a repository by importing the command file produced by the repository backup tool, export of an object or any other means of command file production.

You can import two types of command file into a **HOPEX** repository:

- **text format** (.MGR).

  ☛ *For more details on .MG* file syntax, see "Command File Syntax", page 307.*

- **MEGA XML format** files. These files have .XMG extension and contain commands or data (objects and links).

  ☛ *For more details on MEGA XML data exchange format, see technical article **MEGA Data Exchange XML Format EN**.*

To import a command file:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

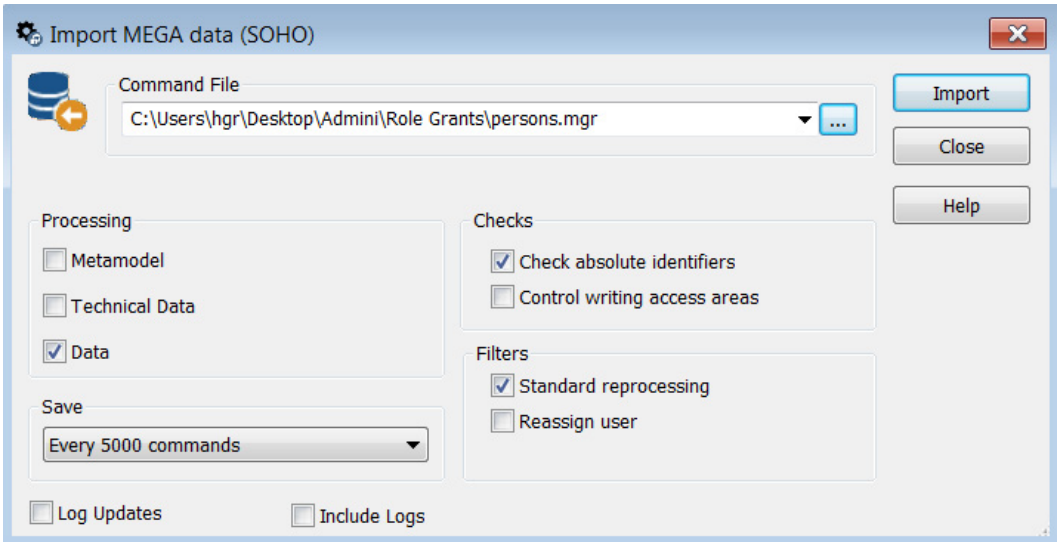
2. Right-click the repository concerned and select **Object Management > Import**.
   The **Import HOPEX Data** dialog box opens.

   ☛ *To import a command file from **HOPEX**, select **File > Import > HOPEX File**.*

3. In the **Command File** frame, click **Browse** ... to browse the folders and select the backup file.

4. Select the types of **Processing** to be executed:
   You can update:
   - the **Metamodel** (repository structure)
   - the **Data** (most frequent case)
   - the **Technical Data** (*descriptions*, *requests*, as well as *users*).

     ☛ *If the file includes commands that do not match the type you have selected, these commands are ignored.*



5. Select the **Save** frequency of the modifications.

   ☛ *Note that there is no optimal save frequency:*

   - ***Standard*** *frequency saves at each "Validate" command in the command file and at the end of the file. This type of frequency is useful when the command file has been written by a user.*
   - ***At end*** *is generally sufficient if the file is not very large.*
   - ***At end if no reject encountered*** *saves the changes only if no rejects were encountered.*
   - ***Never*** *is used to carry out tests before the effective update, for example for syntax checking.*
   - ***Every 5000 commands****: each save is quite long. You can speed things up or slow them down by saving* ***every 100, 200, 500, 1000*** *or* ***5000 commands****.*

   ☛ *Large files may cause memory problems when updating. To avoid such problems, you should decrease the intervals between saves.*

6. In the **Checks** frame, the checks to be carried out are selected automatically, based on the file extension:
   - **Check Absolute Identifiers** is not selected in the case of a command file that does not come from a **HOPEX** repository.
   - **Control writing access areas** is selected when the **HOPEX Power Supervisor** technical module is available on the site, ensuring that

the user who executed the update has the corresponding writing access in the repository.

> ☛ *For command files with the MGR extension (repository backup), absolute identifiers are included in the imported objects and writing access levels are maintained.*
>
> ☛ *For command files with the MGL extension (log extraction or backup logfile), the absolute identifiers are included in the imported objects. The writing access levels are maintained if the updates are consistent with the writing access diagram for the environment.*
>
> ☛ *These controls are not carried out if the user level is "Administrator", this enables the data restorations.*

7. In the **Filters** frame, select the import behavior to be applied:
   - **Standard Reprocessing** changes creation of an already existing object into a modification, or into creation of an object of the same name preceded by a number if their absolute identifiers are different.
   - **Reassign User** ignores the writing accesses contained in the imported file. All elements in the imported file are given the same writing access level as the user executing the import. This is useful when you have the **HOPEX Power Supervisor** technical module. The creator and modifier names are replaced with the name of the user executing the import.

     > ☛ *It is recommended that you enable this option when the import file comes from an environment where the writing access diagram is not the same as the one for the environment where this file is being imported.*

8. (Optional: **Windows Front-End**) Select the **Log Updates** option if you want to update the *repository log*, if this log will be exported to another workstation without the file being imported.

   > **This option is an advanced operation, MEGA recommends that you contact MEGA support before selecting this option.**

9. Selecting **Include Object Logs** allows you to also import object histories.

   > ☛ *For more information on object logs, see "Viewing Object History", page 112.*

   The options you select are controlled by the software, based on the file extension and the standard processing to be applied. If your choices are not consistent with the file extension, a message box informs you of this fact and its possible consequences.

   > ☛ *For more details on the main causes of rejects, see "Dispatch Conflicts", page 141 and "Rejects When Dispatching", page 142.*

10. Click **Import**.
    The report window appears showing the import progress.

    The **Processing** frame details the number of commands accepted and rejected.

When the import contains errors:

- a reject report file is generated.

  > ☛ *See "Viewing rejects", page 106.*

- an execution report file is available.

  > ☛ *See "Viewing the import execution report file", page 107.*

Viewing rejects

To view the rejects (or errors) recorded during the import of the command file:

1. Import the command file

   ☛ *See "Importing command files", page 103.*

2. Click the **Report File** button.

   ☛ *The contents of the report file depend on import options. For more details on importing a command file, see "Managing Options", page 323.*

Case of a text file import (MGR, MGL)

The report file appears and details all the rejects.

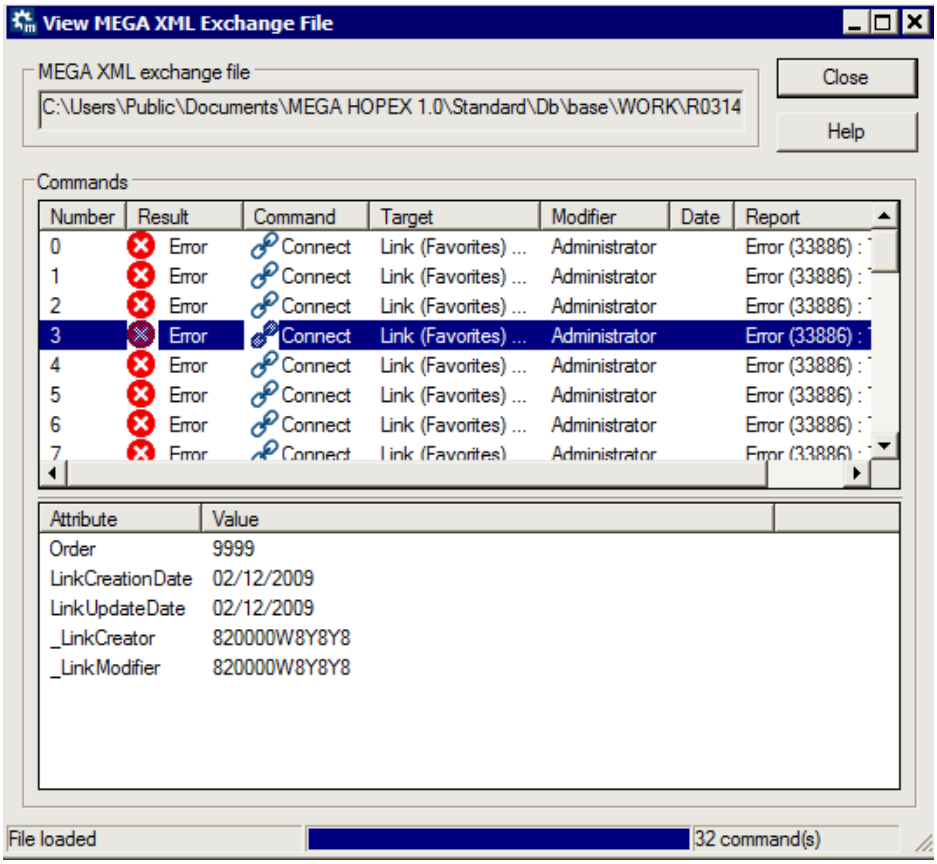
```
- Execution     : (Import) 2013/03/07 17:22:05 18:2
- Input File    : C:\Users\hgr.NTAS\Desktop\DiagrammeAuthentification.mgr
- Description   :
- Reject File   : \\ntas\public\DailyBuildInstalled\MEGA HOPEX 1.0 (731) (int Build)
mega_msi_2010\731-3506.Us_VM\Demonstration\db\MEGA (Tutorial)\WORK
\R0307000.MGR
- Environnement : \\ntas\public\DailyBuildInstalled\MEGA HOPEX 1.0 (731) (int Build)
mega_msi_2010\731-3506.Us_VM\Demonstration
- Base          : MEGA (Tutorial)
- User          : 0000000044444444

- Err Code: 1008481 ErrorLevel: 2  Line: 61 (Offset: 5877)
- A value is required for the 'Object Availability' attribute.

- Execution       : Extraction (2012/02/14 17:25:11)
- File exported   : C:\Users\hgr.NTAS\Desktop\DiagrammeAuthentification.mgr
- Environment     : C:\Users\Public\Documents\MEGA 2009 SP5\Demonstration
- DataBase        : Adventure
- User            : User
- ***********************************************************************************************


- Root objects:
-     Authentification_Hopex
```

Example of reject file at MGR file import

Case of a MEGA XML import (Windows Front-End)

The view window for the report file appears and details the **Commands** contained in the imported file and the **Result** of execution of each command:

- ***Accepted***: the command is accepted
- ***Rejected***: the command is rejected
- ***Warning***: the command is accepted, but contains errors



Viewing the import execution report file

To view the import execution report:

1. Import the command file

   ☛ *See "Importing command files", page 103.*

2. Click **Report**.
   The report details the number of commands analyzed, executed and rejected for each command type.

Viewing the Environment Report File

The environment report file, MegaCrdYYYYmm.txt (where YYYY and mm represent year and month of creation) indicates all administration operations (backup, export, restore, controls, etc.) carried out in the environment. The report file is stored in the user work folder associated with the environment system repository: SYSDB\USER\XXX\XXX.WRI where XXX is the user code.

After you have executed backup and restore operations on a repository, you can view the environment *report file*.

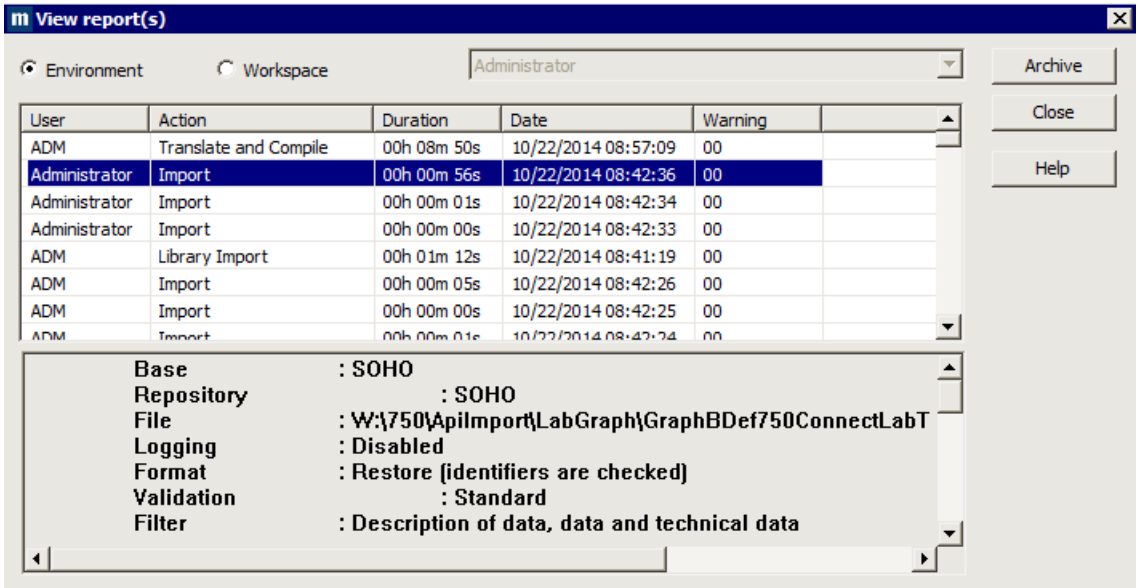
Viewing the environment report file

To open the environment report view window:

1. Connect to **HOPEX Administration** and select the environment concerned.

   See "Connecting to an Environment", page 7.

2. Right-click the environment and select **Reports > Open**.



This dialog box also allows you to view report files of each user of the environment. Indicated for each action are:

- **User** who executed the operation.
- **Action** executed: dispatch, import, installation, translation, export, derivation, extraction, backup, update, user diagram extraction, user diagram import, repository translation, repository or environment check,

repository creation, repository deletion, log initialization, object protection, etc.
- **Date** the operation began.
- **Duration** of the operation.
- return code in the **Warning** column:
  - Error level 0: no error.
  - Error level 2: minor errors such as attempts at creating already existing objects or links.
  - Error level 4: the actions involved nonexistent objects or links.
  - Error level 8: a system error was encountered during the update.
  - Error level 16: update aborted because of a problem such as insufficient disk space.

when you select an action, the lower frame displays:
- the repository concerned
- the file used
- details of the action.

Copying the environment report file

To copy and reinitialize the environment report file:

1. In the **View Report(s)** window (see "Viewing the Environment Report File", page 108), click **Archive**.
   A confirmation request message appears.
2. Select **Yes**.

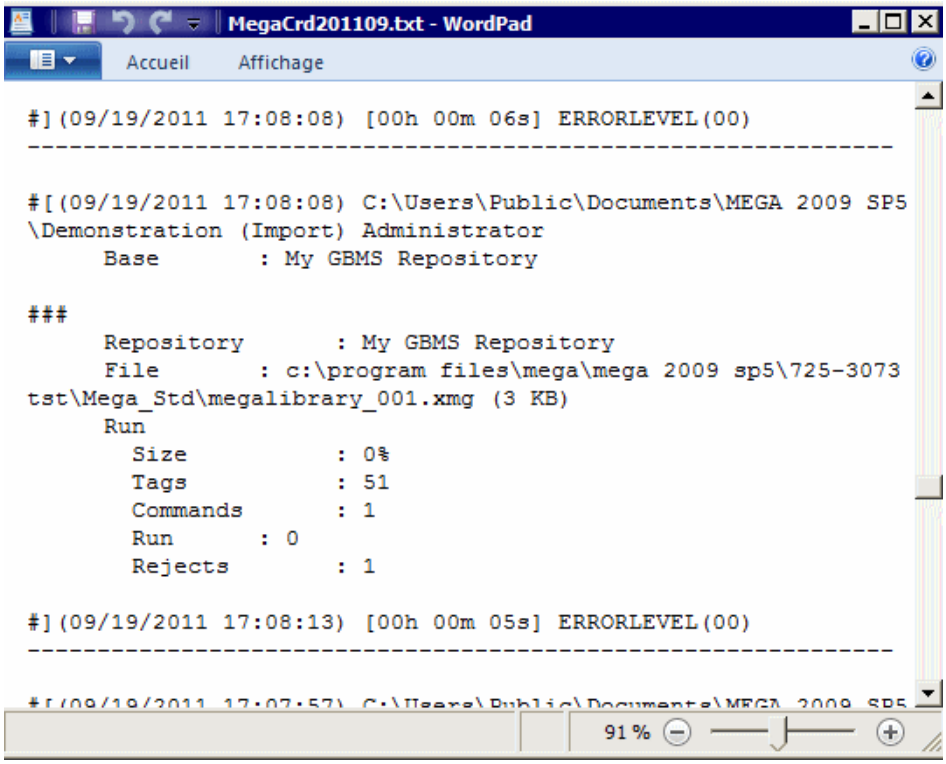
   ☛ *You should reinitialize this file (or archive with the network version) from time to time.*

Opening the environment report file

To open the "MegaCrdYYYYmm.txt" file with Wordpad:

1. From the explorer, localize the "MegaCrdYYYYmm.txt"" file in the environment folder.

2. Right-click the file and select **Open with > Wordpad**.



Viewing User Process Error Trace Files

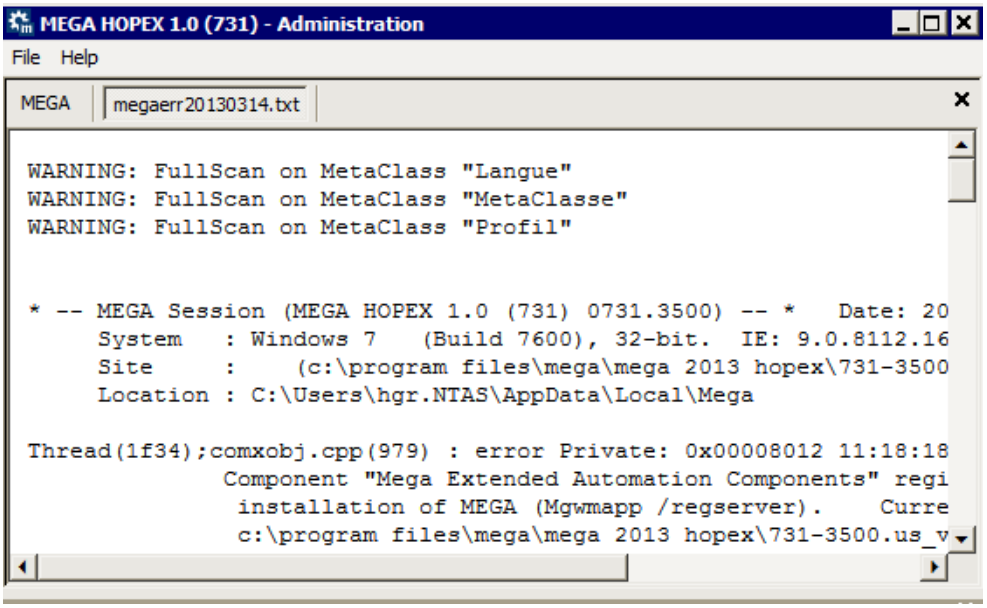
Trace files of errors in user processes contain information on operations executed and possible anomalies.

If there is a problem with a repository, this file will help the **MEGA** Research Center to analyze it.

Each *trace file* has the following characteristics:

- format: *.txt
- name: megaerrYYYYmmDD
  where YYYYmmDD represent the year, month and day when the file was generated

  Example: file Megaerr20110204.txt was generated on 04 February 2014).
- information on errors:
  - date and time of the error
  - action that produced the error
  - associated error message



You can open the trace file from:

- **HOPEX Administration**
- **HOPEX Server Supervisor**
- **HOPEX**

Opening the trace file from HOPEX Administration

To open the trace file from **HOPEX Administration**:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*
2. In the navigation tree, right-click **Workstation** and select **Trace File > Open**.
   The file is opened in the main pane of the window.

Opening the trace file from the HOPEX Server Supervisor tool

To open the trace file from the **HOPEX Server Supervisor** tool:

1. (Prerequisite) The **HOPEX Server Supervisor** tool is started, see "Starting HOPEX Server Supervisor", page 190.
2. In your workstation system tray, right-click **HOPEX Server Supervisor** and select **Mega Logs > Open Daily Logs**.

Opening the trace file from HOPEX

To open the trace file from **HOPEX**:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*
2. In the **HOPEX** menu bar, select **Help > About HOPEX**.
   The **About HOPEX** dialog box appears.
3. In the dialog box that opens, click **System Information**.
4. In the **System Information** dialog box, select **Error Log > Edit**.

Saving the Error Zip file for Diagnostics

The **HOPEX Server Supervisor** tool allows you to save a zip file containing information required by the **MEGA** Research Center to help in repository problem diagnostics.

This zip file of errors contains in particular the trace files of errors related to user processes and/or SSP (megaerrYYYYmmDD.txt and ssperrYYYYmmDD.txt).

Prerequisite: **HOPEX Server Supervisor** is started, see "Starting HOPEX Server Supervisor", page 190.

To save the error zip file for diagnostics:

1. In your workstation system tray, right-click **HOPEX Server Supervisor** and select **Logs > Daily Logs manager**.
2. Click **Zip**.
3. Specify a saving location and enter the zip file name.
4. Click **Save**.

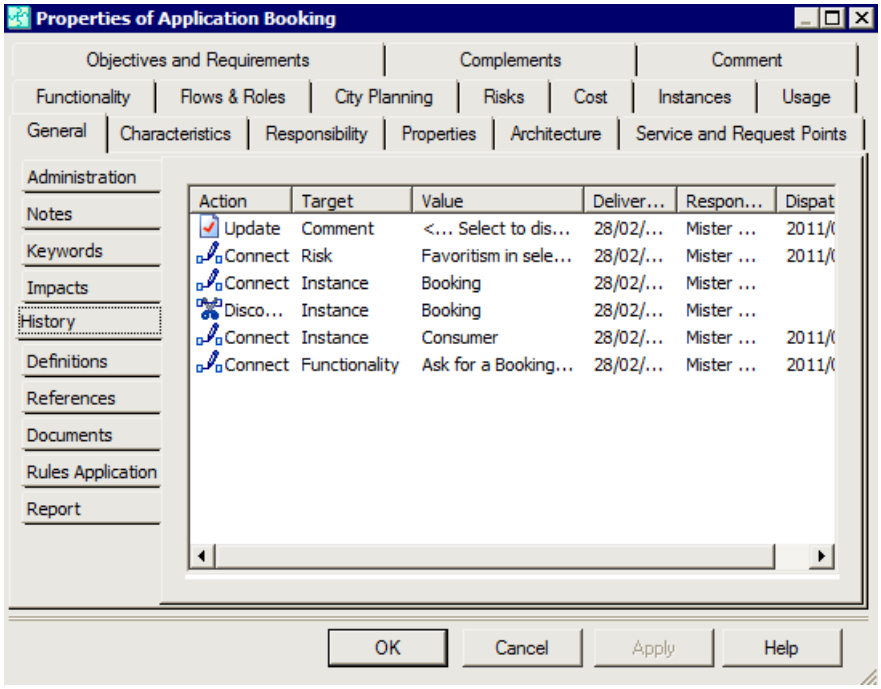
Viewing Object History

The object history is a different view of the repository log: instead of browsing actions performed by a user in a dispatch, the object history shows actions carried out from an object for all dispatches and users.

☛ *The repository log must be enabled so that the object history can be supplied, see "Enabling the repository log", page 85.*

In the properties dialog box of an object, the **History** subtab of the **General** tab gives an overview of actions on each object in the repository. At each update

concerning the object (Create, Modify, Connect, Disconnect), the corresponding action is added to the list.



Optimizing Repository Access Performance

To optimize **HOPEX** performance, you must also remember to optimize your repository size. You must:

- reduce the log size

  ☛ *See "Managing Log Size", page 115.*

- increase the cache size

  ☛ *See "Increasing RDBMS cache size (memory)", page 120.*

- delete temporary data and history data (RDBMS repository) regularly

  ☛ *See "Deleting RDBMS Repository Temporary and Historical Data", page 123.*

- perform regular maintenance tasks of RDBMS repositories

  ☛ *See "Performing regular maintenance tasks (RDBMS repository)", page 124.*

- reduce quantity of status indicators

  ☛ *See "Managing Status Indicators", page 122.*

- reduce quantity of locks

  ☛ *See "Managing GBMS Repository Locks (Windows Front-End)", page 125.*

- clean up repository

  ☛ *See "Cleaning up a Repository", page 125.*

- configure anti-virus actions

  ☛ *See "Configuring the Anti-Virus According to HOPEX Data", page 126.*

- reorganize repository

  ☛ *See "Reorganizing a repository", page 96.*

Managing Log Size

> ☛ *Managing the log size is only necessary if you have enabled the repository log, see "Enabling the repository log", page 85.*

To reduce the log size, you can:

- delete or consolidate all the log commands earlier than a selected date

  > ☛ *See "Deleting a log or reducing the log size", page 115.*

- (SQL Server) select and delete log elements, earlier than a selected date

  > ☛ *See "Deleting log elements to reduce the log size (SQL server)", page 118.*

- modify MetaClass loggability

  > ☛ *See "Modifying MetaClass loggability", page 119.*

Log size management frequency

You must reduce the log size:

- every month, for configurations of less than 50 users.
- every week, for configurations of more than 50 users.

Deleting a log or reducing the log size

Prerequisite: before deleting your log (complete or partial deletion), MEGA recommends that you back up it.

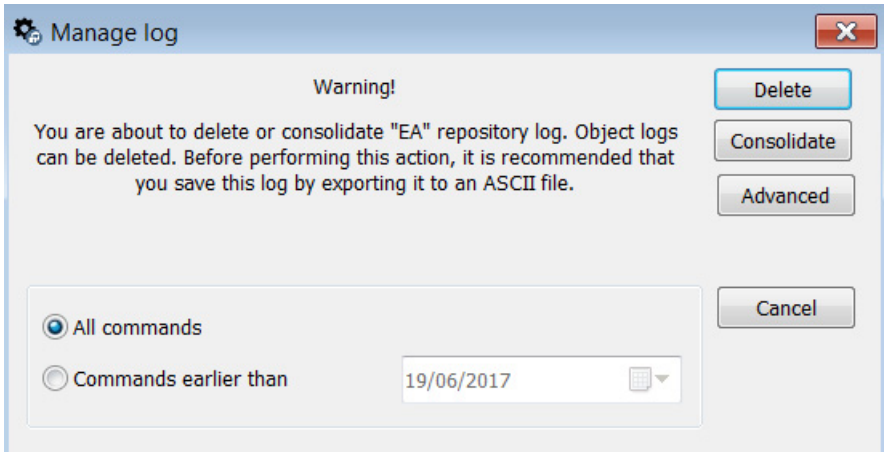
> ☛ *see "Backing up the repository log", page 116.*

To delete a log or to reduce its size:

1. Connect to **HOPEX Administration** and select the repository concerned.

   > ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Repository Log > Manage Repository and Object Log**.
   The **Manage Log** dialog box opens.

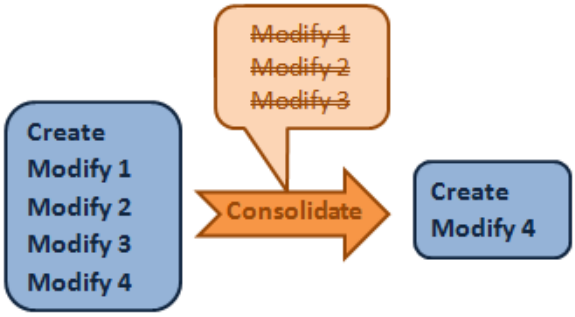


3. In the **Manage log** window, define the commands to be deleted. Select either:
   - **All commands** or
   - **Commands earlier than** and select the date using the drop-down menu calendar.
4. Click either:
   - **Delete** to delete all the commands contained within the selected time interval.

     ☛ *(GBMS, Oracle) If you did not follow the recommendation regarding log management frequency, this action may take a very long time, see "Log size management frequency", page 115.*

     ☛ *(SQL Server) For a more specific deletion, see "Deleting log elements to reduce the log size (SQL server)", page 118.*
   - **Consolidate** to delete only the intermediate commands contained within the selected time interval.

     ☺ *You can consolidate the latest information.*



Backing up the repository log

To back up the repository log you have to export the repository log in a file and save it.

This file can be exported in:

- **logfile text** format (.mgl).
  The format of the exported file name is "LOGmmdd.mgl", where "mmdd" represents logfile export date month and day.
- **XML MEGA format** (.xmg)
  The exported file is in the form of an XML file containing commands or data (objects and links).

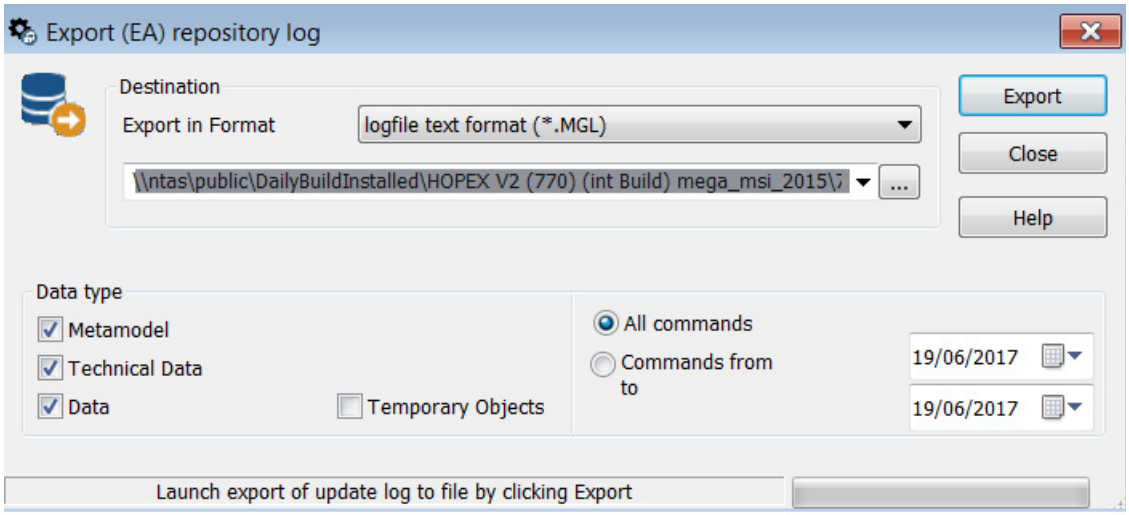
To avoid backup duplications and save backup time and size, select **Commands from** "selected date" **to** "selected date" and select a starting date corresponding to the end date of your last backup, and select the ending date you require.

To backup the repository log in the form of command file:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Repository Log > Export**.
   The **Export repository log** dialog box opens.



3. Select export format.
4. (Optional) By default the file is saved in **db\<repository name>WORK** folder. If needed, modify the name and folder proposed by default to save the export file. The **Browse** button ... allows you to browse the folder tree and select the folder in which the file will be placed.
5. In the **Data Type** frame, select the type of exported modifications:
   - **Metamodel** if you want to extract the metamodel from the system repository. This is useful if the standard metamodel has been modified.
   - **Technical Data** if you want to include in your file the changes made to data such as descriptors and queries.
   - **Data** if you only want to export changes made to repository data, particularly the workspaces.
   - **Temporary Objects** - these objects are created when executing requests, consulting objects (stored in the history), etc. Usually you do not need to export these objects.
6. To export the commands that correspond to a defined period only, select **Commands from** / **to** and specify the relevant dates.
7. Click **Export**.
   The **Execution Report** appears.
8. Click **OK**.
   The file is exported and saved in the specified folder.

Deleting log elements to reduce the log size (SQL server)

With an SQL Server repository storage, you can delete log elements more specifically: you can delete only log elements relating to specific MetaClasses, earlier than a selected date.

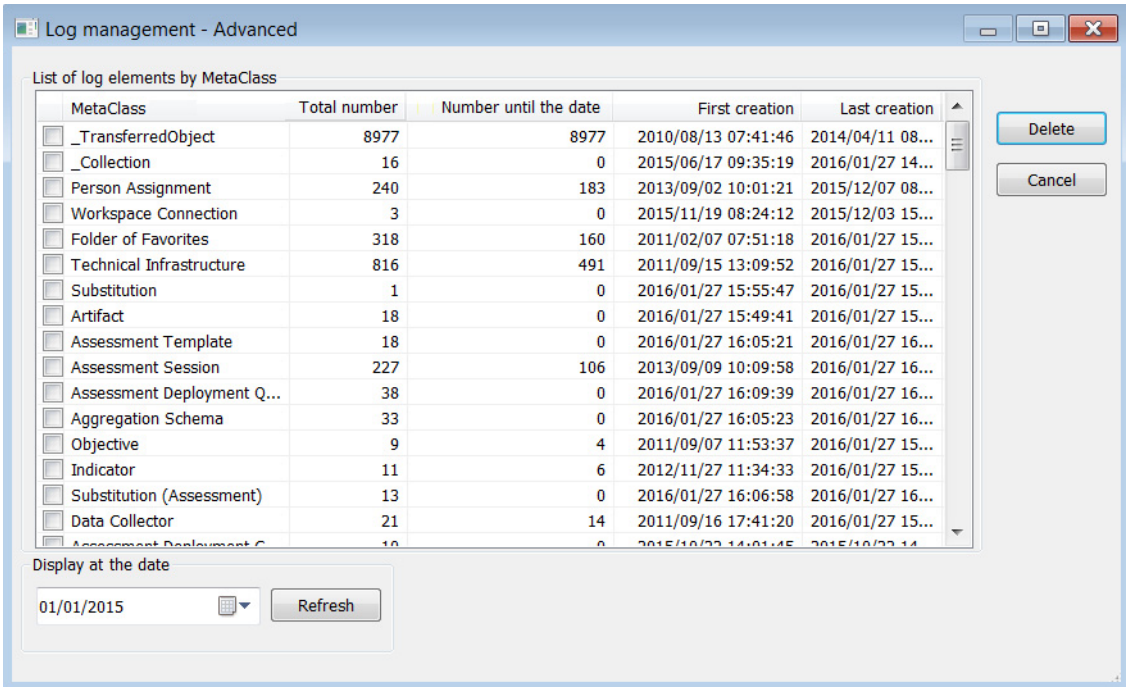
> ☛ *If you do not want to have to delete log elements you are not interested in, see "Modifying MetaClass loggability", page 119.*

To delete log elements regarding specific MetaClasses:

1. Connect to **HOPEX Administration** and select the repository concerned.

   > ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository and select **Repository Log > Manage Repository and Object Log**.
   The **Manage Log** dialog box opens.
3. Click **Advanced**.
   The **Log Management - Advanced** window shows the log element number by MetaClass and specifies the first and last creation date.



MetaClass	Total number	Number until the date	First creation	Last creation
_TransferredObject	8977	8977	2010/08/13 07:41:46	2014/04/11 08...
_Collection	16	0	2015/06/17 09:35:19	2016/01/27 14...
Person Assignment	240	183	2013/09/02 10:01:21	2015/12/07 08...
Workspace Connection	3	0	2015/11/19 08:24:12	2015/12/03 15...
Folder of Favorites	318	160	2011/02/07 07:51:18	2016/01/27 15...
Technical Infrastructure	816	491	2011/09/15 13:09:52	2016/01/27 15...
Substitution	1	0	2016/01/27 15:55:47	2016/01/27 15...
Artifact	18	0	2016/01/27 15:49:41	2016/01/27 15...
Assessment Template	18	0	2016/01/27 16:05:21	2016/01/27 16...
Assessment Session	227	106	2013/09/09 10:09:58	2016/01/27 16...
Assessment Deployment Q...	38	0	2016/01/27 16:09:39	2016/01/27 16...
Aggregation Schema	33	0	2016/01/27 16:05:23	2016/01/27 16...
Objective	9	4	2011/09/07 11:53:37	2016/01/27 15...
Indicator	11	6	2012/11/27 11:34:33	2016/01/27 15...
Substitution (Assessment)	13	0	2016/01/27 16:06:58	2016/01/27 16...
Data Collector	21	14	2011/09/16 17:41:20	2016/01/27 15...

4. In the **Display at the date** pane, select the date until which you want to delete the log elements.
5. Click **Refresh**.
   The **Number until the date** column shows for each MetaClass the log element number until the selected date.

   > ☺ *Click the **Number until the date** column header to sort the MetaClasses, with the most populated at the top.*

6. In the **MetaClass** column, select the MetaClass for which you want to delete the log elements until the selected date.

   ☛ *You can select several MetaClasses.*

7. Click **Delete**.
   Log elements are deleted.

Modifying MetaClass loggability

To reduce the number of objects generated in logs, you can modify logging of MetaClasses you do not want to track.

☛ *In case of inter-repository consolidations, see "Modifying the log behavior", page 86.*

To modify MetaClass loggability:

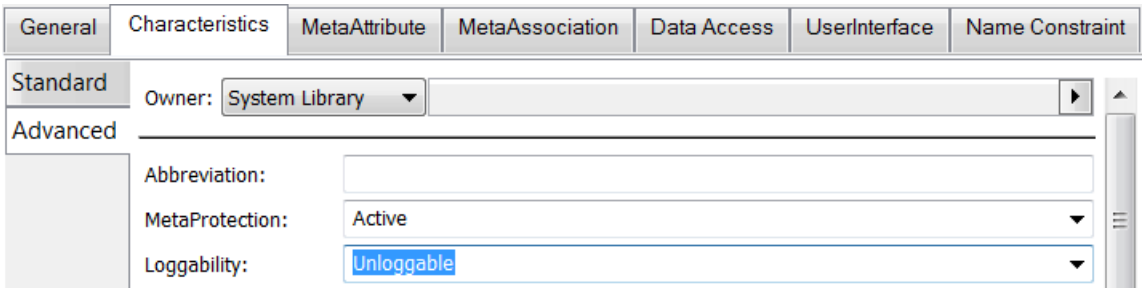
1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. Open the **MetaStudio** navigation window and expand folders **MetaClass > MetaModel**.

   ☛ *Alternatively click **Explore** and use explorer to find the **MetaClass** or **MetaAssociation** object.*

3. Open the properties dialog box of a **MetaClass** or **MetaAssociation**.
4. In the **Characteristics > Advanced** tab, for the loggability attribute, select **Unloggable** value.



Occurrences created, updated or deleted in a private workspace are dispatched, but certain commands are not available in object histories or in repository activity.

Managing cache in RDBMS environments

The RDBMS local cache avoid multiple requests when multiple users are on the same repository view. Access for the following users is speed up.

By default the RDBMS local cache is activated on SystemDb repository only:

- You must configure your anti-virus accordingly.

  ☛ *See "Configuring the Anti-Virus According to HOPEX Data", page 126.*

- You can:
  - increase RDBMS cache size (memory).

    ☛ *See "Increasing RDBMS cache size (memory)", page 120.*

  - extend the RDBMS local cache option to all the repositories or disable the option.

    ☛ *See "Extending or disabling RDBMS local cache files", page 121.*

RDBMS local cache files are generated in .mgc format. They are stored in **RDBMS Data** sub-folder accessible from: <ProgramData>\MEGA\HOPEX <HOPEX version>\Cache\ folder.

☛ *To modify the **HOPEX** cache location, see "Modifying HOPEX cache location", page 120.*

Increasing RDBMS cache size (memory)

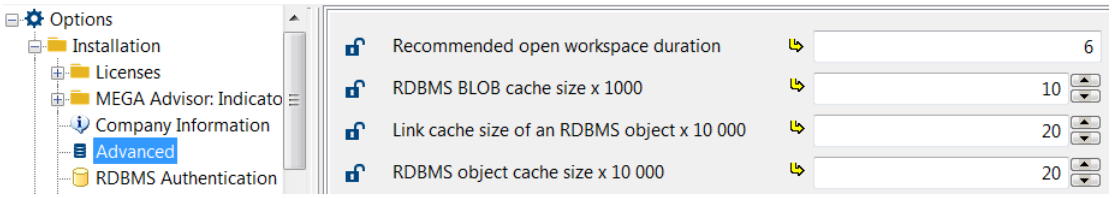
In RDBMS storage case, with repositories including a large amount of objects, we advise you to increase the size of RDBMS caches. The larger your cache space, the fewer the network exchanges and the better your **HOPEX** performance.

To increase cache size:

1. Connect to **HOPEX Administration** and select the environment in which the repository is referenced.

   ☛ *See "Connecting to an Environment", page 7.*

2. Right-click the environment and select **Options > Modify**.
3. In the environment options, select **Options > Installation > Advanced**.



4. Increase cache sizes depending on your memory space.

Modifying HOPEX cache location

By default, **HOPEX** caches are stored in <ProgramData>\MEGA\HOPEX <HOPEX version>\Cache\ directory sub-folders:

- Compiled data
- Pictures
- RDBMS data

If needed you can change **HOPEX** cache folder location.

To modify the **HOPEX** cache folder location:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. Right-click the Site (**HOPEX**) and select **Options > Modify**.
3. In the environment options, select **Options > Installation > Advanced**.
4. In **Path of HOPEX cache** field, enter a location.
   This location indicates the location where **HOPEX** must read and write its cache data.

   ☛ *For this location select a local drive that has a good access time.*

Extending or disabling RDBMS local cache files

By default RDBMS local cache is enabled on SystemDb repository only. According to your infrastructure performances, you can extend the storage to the other repositories or disabled the storage.

☛ *You must be an advanced user to modify this option.*

Modifying the scope of the RDBMS local cache files depends on:

- the latency between HOPEX application server and RDBMS databases, and
- Reading/Writing performances of HOPEX application server local disks.

Local disk performance (reading/writing)	Low latency (< 2 ms)	2 ms < Latency < 5 ms	High latency (5 ms <)
Good (SSD, physical infrastructure)	SytemDb only	SytemDb only	All repositories
Fair (virtual infrastructure)	Disabled	SytemDb only	SytemDb only

To modify the default scope of RDBMS local cache files:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. Right-click the Site (**HOPEX**) and select **Options > Modify**.
3. In the environment options, select **Options > Installation > Advanced**.
4. Modify the **Scope of HOPEX-RDBMS local cache** option value according to your infrastructure performances.

   ☛ *Default value: "SytemDb only".*

Cyphering RDBMS local cache data

Cyphering of RDBMS local cache option enables cyphering data included in the RDBMS local cache. Selecting this option decreases the benefit of RDBMS local cache activation. In that case, balance the benefit of storing the RDBMS local cache files with **Scope of HOPEX-RDBMS local cache** option activation.

To cypher the data of the RDBMS local cache files:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. Right-click the Site (**HOPEX**) and select **Options > Modify**.
3. In the environment options, select **Options > Installation > Advanced**.
4. Select **Cyphering of HOPEX-RDBMS cache** option.

Managing Status Indicators

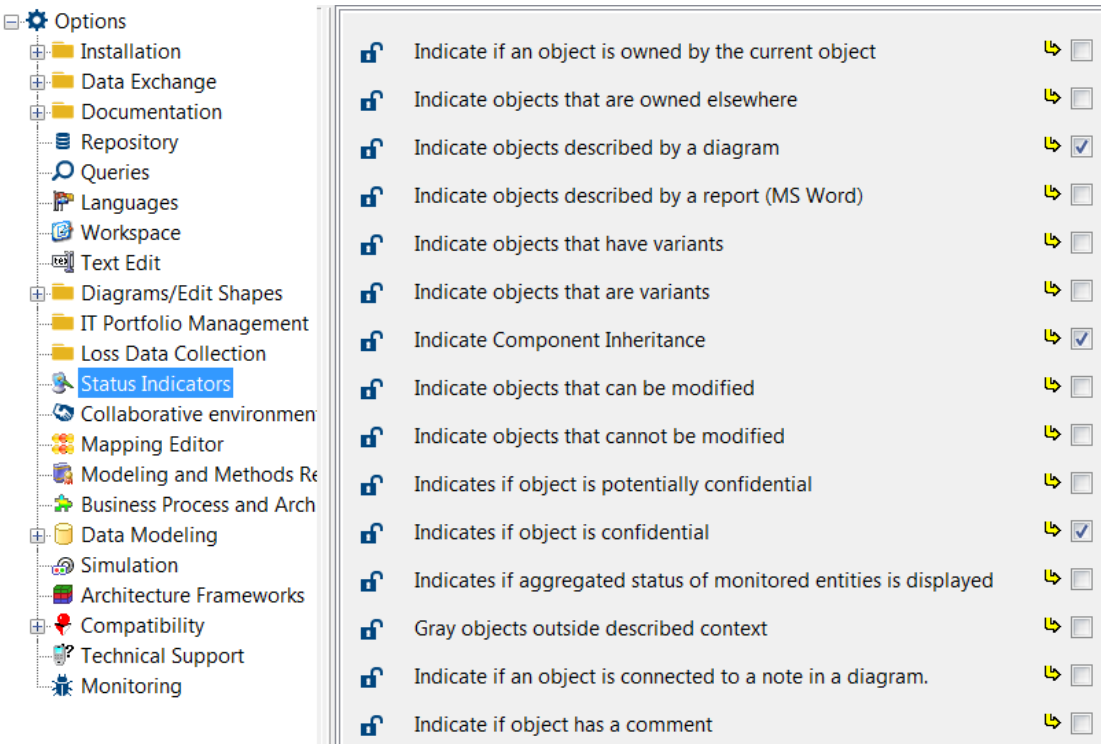
Use of status indicators generates large quantities of queries on repositories. Select only those indicators you require.

To modify indicator backup selection:

1. Connect to **HOPEX Administration** and select the environment in which the repository is referenced.

   ☛ *See "Connecting to an Environment", page 7.*

2. Right-click the environment and select **Options > Modify**.
3. In the **Environment Options** dialog box, select **Status Indicators**.



4. In the right pane, select only those indicators you require.
5. Click **OK**.

Deleting RDBMS Repository Temporary and Historical Data

To prevent repository size increase and keep optimized performances, you should regularly delete data of the completed private workspaces of **HOPEX** users.

MEGA recommends that you delete all RDBMS repository temporary and historical data:

- every week for less than 10 users
- every evening if you have more than 10 users.

To delete the temporary and historical data of an RDBMS repository, you (or your database administrator) must include the following procedures in your regular maintenance tasks:

> ☛ *To perform these procedures, see **RDBMS Repository Installation Guide** deployment guide.*

- SP_CLEAN_MEGA_DATABASE

> ☛ *To consult the date of last private workspace cleanup (last execution of this procedure), see "Consulting and Modifying Repository Properties", page 79.*

- SP_CONSOLIDATE_MEGA_DATABASE

> ☛ *To consult the date of last consolidation (last execution of this procedure), see "Consulting and Modifying Repository Properties", page 79.*

On an exceptional basis, with **HOPEX** you can perform these deletion tasks on several repositories in one action. This possibility enables you to mass launch these procedures on all the repositories with no need to validate the execution for each repository.

See:

- "Mass deleting temporary data of private workspace", page 123
- "Mass deleting repository historical data", page 124

Mass deleting temporary data of private workspace

To mass delete private workspace temporary data:

1. Connect to **HOPEX Administration** and select the environment concerned.

   > ☛ *See "Connecting to an Environment", page 7.*

2. Expand the environment folder.
3. Right-click the **Repositories** folder and select **Manage**.
4. In the **Action list** pane, select **Private workspace temporary data deletion**.

   > ☛ *To mass delete repository historical data, select also **Repository historical data deletion**.*

5. In the **Repository list** pane, select the repositories for which you want to delete temporary data.

6. Click **Execute**.
   The cleaning is automatically launched on each repository selected with no need to validate each execution.

   ☛ *A dialog box enables you to stop the procedure at any time. If needed, click **Stop** to stop the procedure execution.*

   Private workspace temporary data is deleted on all the selected repositories.

   ☛ *If you selected the option, repository historical data is also deleted on all the selected repositories.*

Mass deleting repository historical data

To mass delete historical data:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the environment folder.
3. Right-click the **Repositories** folder and select **Manage**.
4. In the **Action list** pane, select **Repository historical data deletion**.

   ☛ *To mass delete repository historical data, select also **Repository historical data deletion**.*

5. In the **Repository list** pane, select the repositories for which you want to delete repository historical data.
6. Click **Execute**.
   The consolidation procedure is automatically launched on each repository selected with no need to validate each execution.

   ☛ *A dialog box enables you to stop the procedure at any time. If needed, click **Stop** to stop the procedure execution.*

   Repository historical data is deleted on all the selected repositories.

   ☛ *If you selected the option, private workspace temporary data is also deleted on all the selected repositories.*

Performing regular maintenance tasks (RDBMS repository)

With RDBMS repositories you must include a regular maintenance plan. You (or your database administrator) should, for each repository (SystemDb repository included), perform the following maintenance tasks regularly (at least once a week):

- rebuild indexes
- update the statistics
- (optional) shrink the logs regarding the policies
- perform the stored procedures

  ☛ *See "Deleting RDBMS Repository Temporary and Historical Data", page 123.*

  - SP_CLEAN_MEGA_DATABASE
  - SP_CONSOLIDATE_MEGA_DATABASE

Managing GBMS Repository Locks (Windows Front-End)

In a GBMS repository, a file is associated with each lock. When private workspaces are deleted manually, certain files may persist despite lock deletion.

A folder with more than 1000 files:

- is costly to manage for Windows
- slows down operation of **HOPEX**.

To delete locks that persist, for example after deletion of a user private workspace:

- Delete locks for a user whose private workspace has been manually deleted.

☛ *See "Managing locks from HOPEX Administration", page 163.*

Cleaning up a Repository

During modeling work, handling different objects results in creation of temporary objects (temporary queries, etc.). It is probable that these objects will subsequently be unnecessary. To avoid unnecessarily increasing repository size, you can delete these temporary objects.

To clean up a repository:

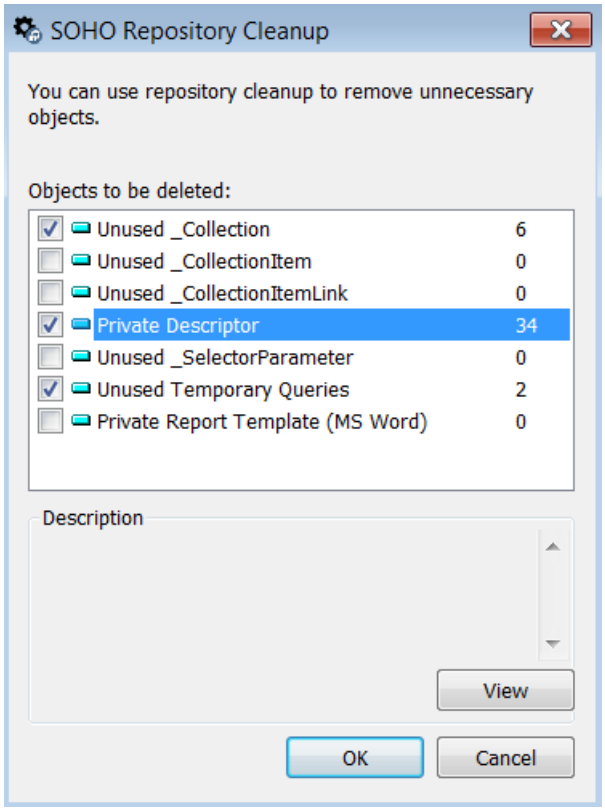
1. Connect to **HOPEX Administration** and select the repository concerned.

☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Object Management > Repository cleanup**.

   ☛ *Alternatively, connect to **HOPEX** and in the menu bar, select **File > Properties**. In the properties of the repository to which you are connected, select **Characteristics**.*

   The **Repository Cleanup** dialog box opens.



3. (Optional) To view the content of an object group, click the object group name and then click **View**.
4. Select the object groups to be deleted and click **OK**.

   ☛ *By default, all the object groups that include objects are selected.*

   Selected object groups are deleted from the repository.

Configuring the Anti-Virus According to HOPEX Data

To optimize anti-virus activity and to avoid unnecessary slowdown, you must configure what should be managed or not by the anti-virus.

You must exclude the following files from anti-virus scanning:

	Access mode	Location	File type
Program files external to HOPEX Files used by **HOPEX** to display user interface elements in Web or Windows interfaces	***Read***	Program Files (x86) \ MEGA\ MEGA-STD and sub-folders	Proprietary format: ***.mgs*** (vectorial shapes in diagrams) Public formats: ***.gif***, ***.ico*** and ***.bmp***
Quick search indexes Files used by the workstation or server executing the search	***Reading*** by the workstation or node executing the search ***Writing*** by the Administration station and by the server programming index creation	<Environment folder>\Db\ <Repository name>\ <Repository name>.ix	Proprietary format: ***.ix*** (indexes) Public formats: ***.log*** and ***.dat***
Data cache Files used by **HOPEX** to improve its performance In RDBMS environment, **RDBMS data** cache subfolder is automatically populated, see "Managing cache in RDBMS environments", page 119	***Read/Write***	<ProgramData>\MEGA \HOPEX <version HOPEX>\cache	Public format: ***.mgc***
Import/export files Files generated or read during import/export/logical backup processes	***Reading*** by the import function ***Writing*** by export and logical backup functions	Selected by the user	Proprietary formats: ***.mgr*** or ***.xmg***

	Access mode	Location	File type
Logfiles Files generated by **HOPEX** for each repository when the **Backup Logfile** option is activated. Files generated by the Export command	***Writing***	<Environment folder>\Db\<Repository name>\<Repository name>.transaction or <Repository name>.Log	Proprietary format: ***.mgl***
MUST license Note: if the anti-virus does not allow such a configuration, allow ***Reading/Writing*** for all of the files on the license oath and sub folders.	***Reading/Writing*** of the files on MUST license path and under this path	<Installation folder>\Cfg\megasite.ini gives the MUST License path: [must license] path=	Public format: ***.ini*** Proprietary formats: ***.tnk*** and ***.usr***
	Reading of the files on MUST license path and under this path		Proprietary format: ***.must***

REFERENCING AND UNREFERENCING A REPOSITORY

Referencing and unreferencing a repository is carried out via the drop-down menu of the repository.

See successively:

- "Referencing a Repository", page 129
- "Unreferencing a Repository", page 130
- "Protecting the referencing of a repository", page 130

Referencing a Repository

If you have moved or copied a repository without using MEGA move or restore commands, you will have to reference the repository so that the environment recognizes it.

> ☛ *To reference a repository in another environment, the two metamodels must be identical.*

To do this:

1. Connect to **HOPEX Administration** and select the environment in which you want to reference the repository.

   > ☛ *See "Connecting to an Environment", page 7.*

2. Right-click the **Repositories** folder and select **Create reference**.
3. (If the repository is password-protected) Enter the repository password and click **OK**.

   > ☛ *See "Protecting the referencing of a repository", page 130.*

4. Several options are proposed, select for example a **Repository in MEGA format**.

   > ☛ *You can also reference a repository from SQL Server or Oracle. For more details on **HOPEX** repository server type, see the **HOPEX** deployment guide.*

   The HOPEX **Repository Selection** dialog box opens. This dialog box allows you to create a reference for a new repository in the environment.

5. Indicate where the .EMB (or .EMO, .EMQ, .EMY) file for the repository is located. The repository name is automatically indicated and cannot be modified.

The repository is accessible exactly as repositories created in the normal way.

> ☛ *you must save and then restore a repository to move it from one environment to another. The users and metamodel of the two environments must be defined identically so that transfer occurs without reject.*

> ☛ *To also copy these objects, import the missing part of the metamodel. One way of doing this is to upgrade the site or the environment. Then, import the rejected commands if you used the backup-restore procedure.*

> ☛ *Check that the repository is not simultaneously referenced in two different environments. Compatibility errors may occur if the environments are not identical.*

Unreferencing a Repository

You can delete a repository reference from an environment. This action does not delete the repository.

If you delete a reference of a password-protected repository. you must know this password to reference the repository again.

To delete a repository reference:

1. Connect to **HOPEX Administration** and select the repository concerned.

   See "Accessing Repositories", page 74.

2. Right-click the repository of which you want to delete the reference and select **Delete Reference**.
   A message requests confirmation.

   If the repository is password-protected, you must know this password to reference the repository again.

3. Click **OK** to delete the repository reference.
   The repository reference is deleted, and can be created in another environment on condition that the metamodel is identical.

   A repository must be referenced in only one environment. It is important to check that the reference for your repository was deleted in its original environment.

Protecting the referencing of a repository

You can password-protect the referencing of a repository.

See:

- "Adding a referencing password to a repository", page 130
- "Modifying/Canceling a repository password", page 131

Adding a referencing password to a repository

To password-protect the referencing of a repository:

1. Connect to **HOPEX Administration** and select the repository concerned.

   See "Accessing Repositories", page 74.

2. Right-click the repository and select **Protect with a password**.
   The **Password Protect Repository** dialog box appears.
3. In the **Password** field, enter a password and confirm it.

   The password must contain at least eight characters.

4. In the **Security question** pane:
   - enter a **Question**.
   - enter the **Answer** to the question.

5. Click **OK**.
   The repository password is requested on creation of a repository referencing.

Modifying/Canceling a repository password

To modify or cancel a repository password:

1. Connect to **HOPEX Administration** and select the repository concerned.
   ☛ *See "Accessing Repositories", page 74.*
2. Right-click the repository concerned and select **Password Protect**.
   The verification window for the password appears.
3. In the **Enter the password of the current repository** field, enter the repository password.
4. Click **OK**.
5. In the **Password** pane:
   - To modify the existing password, enter a password and confirm it.
     ☛ *The password must contain at least eight characters.*
   - To cancel the password, leave the fields empty.
6. Click **OK**.

MANAGING WORKSPACES

Workspaces are managed by the administrator.

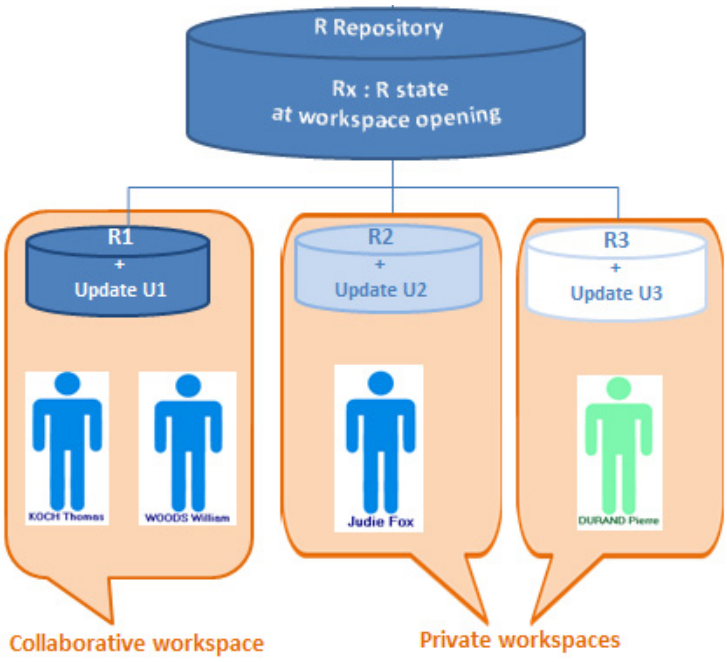
The following points are covered here:

- ✓ "Private Workspace Principle", page 134
- ✓ "Using Your Private Workspace", page 136
- ✓ "Workspace Administration", page 149
- ✓ "Private Workspace Life: Example", page 152
- ✓ "Managing Updates", page 155
- ✓ "Managing locks", page 161

PRIVATE WORKSPACE PRINCIPLE

In a traditional management application, the user cannot control the opening duration of his/her workspace: the end of a data entry corresponds to a definitive save of his/her work.

With **HOPEX** the user controls management of his/her workspace: opening, closing, dispatch, refresh.



Private Workspace

When a user connects to a **HOPEX** desktop (**Windows Front-End**) and to certain **Web Front-End** desktops, he/she opens a *private workspace.* This private workspace is a temporary view of the repository (repository snapshot) at the moment of user connection.

The user then sees:

- initial repository snapshot objects of his/her visibility scope
- updates he/she has executed on these objects.

The user decides when he/she wants to integrate his/her repository updates and make them visible to other users. To do this, he/she dispatches modifications.

☛ *See "Dispatching Your Work", page 140.*

The user controls opening duration of his/her private workspace.

> ☛ *The private workspace is open for each user on both his/her repository and on the system repository. A private workspace always exists on system repository even if the user is not using any of the system repository objects.*

Several users can have private workspaces open in parallel. In his/her private workspace, the user is independent of updates carried out simultaneously by other users in their respective private workspaces.

> ☛ *Locks inform the user of objects modified by others. See* *"Managing locks", page 161*.

The user can also update his/her private workspace with the updates of other users. To do this, the user refreshes his/her private workspace.

> ☛ *See* *"Refreshing Data", page 143*.

HOPEX allows several users to work at the same time.

Collaborative Workspace

> ☛ *This functionality is available with an RDBMS format repository only.*

The user can also share his/her private workspace with other users before dispatching his/her modifications and making public his/her work to all other users. To do this, the user creates a **Collaborative Workspace** from his/her private workspace.

> ☛ *See the **HOPEX Common Features** guide, "Working in a Collaborative Workspace" section.*

A user can, in parallel:

- have a private workspace
- be the owner of several collaborative workspaces
- be invited to participate in as many collaborative workspaces as he/she wishes.

USING YOUR PRIVATE WORKSPACE

A private workspace is a temporary view of the work repository allocated to a user before the user dispatches his/her work. This view of the repository is only changed by modifications made by the workspace user, independently of concurrent modifications made by other users. This private workspace exists until it is refreshed, dispatched or discarded.

Note that a private workspace is kept when the user disconnects from the repository, unless the user indicates otherwise.

The following points are detailed here:

- "Connecting to HOPEX", page 136
- "Saving Sessions", page 138
- "Workspace Properties", page 138
- "HOPEX Repository State Changes", page 139
- "Dispatching Your Work", page 140
- "Dispatch Conflicts", page 141
- "Rejects When Dispatching", page 142
- "Dispatch Report", page 143
- "Refreshing Data", page 143
- "Conflicts When Refreshing", page 145
- "Discarding Work", page 145
- "Exiting a Session", page 146
- "Workspace Administration", page 149
- "Viewing Updates", page 155
- "Exporting Your Private Workspace Log", page 157

Connecting to HOPEX

When you connect to **HOPEX**, you can:

- create a private workspace (if you do not already have one).

  ☛ *You can only have one private workspace open in the same environment.*

  ☛ *The private workspace is open for each user on both his/her repository and on the system repository. A private workspace always exists on system repository even if the user is not using any of the system repository objects.*
- resume work in your private workspace
- resume work in a collaborative workspace

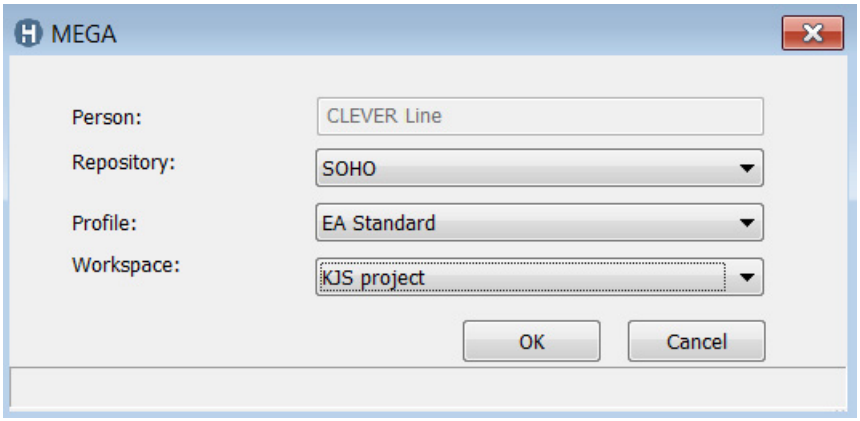
  ☛ *This option is available with **HOPEX Collaboration Manager** only.*

To connect to **HOPEX**:

1. Start the **HOPEX** application.
   The authentication dialog box appears.
2. In the **Login** field, enter your login.
3. (Optional) In the **Password** box, enter your password if required.

4. In the **Environment** field, select your work environment.
5. Click **OK**.
   You are authenticated, your name appears in the **Person** field.
6. In the **Repository** field, select your work repository.

   ☛ *If you already have a private workspace open, the repository is automatically selected and this field is grayed. To change repository, you must first dispatch or discard your current private workspace.*
7. In the **Profile** field, select the profile with which you want to work.
   If you have **HOPEX Collaboration Manager**, go to step 9.
8. Click **LOGIN**.
   A private workspace is created and your desktop opens.



9. If:
   - you do not have a collaborative workspace available, the **Workspace** field is not available. Click **OK**.

     A private workspace is created and your desktop opens.

     ☛ *If you already have a private workspace open, you should connect to it. If you want to change profile or repository, you must close the private workspace that is open.*
   - you have at least one collaborative workspace available, in the **Workspace** field, select **Access Private Workspace** or select the collaborative workspace to which you want to connect, or select **Create Private Workspace** (if one has not already been created). Click **OK**.

     ☛ *A user has at most one private workspace in progress in an environment, but can have in parallel several available collaborative workspaces.*

     Your desktop opens.

A private workspace comprises a set of files located in a sub-folder of the repository:

"<EnvironmentName>\DB\<RepositoryName>\<RepositoryName>.Transactions\xyz.*"

where "xyz" represents the user code.

Note that a private workspace cannot be separated from its repository (these files cannot be used independently).

Saving Sessions

A session is the period during which a user is connected to a repository. A session begins when the user establishes connection and ends when he/she exits HOPEX. Sessions and private workspaces can overlap. When you dispatch, refresh or discard a private workspace, a new private workspace is created in the same session. Conversely, a user can keep his/her private workspace when exiting a session.

To save modifications you have made in your *session* since the last save:

- In the **HOPEX** menu bar, click **Save** .

  These modifications are not saved in the repository. To save your modifications in the repository, you must dispatch these modifications, see "Dispatching Your Work", page 140.

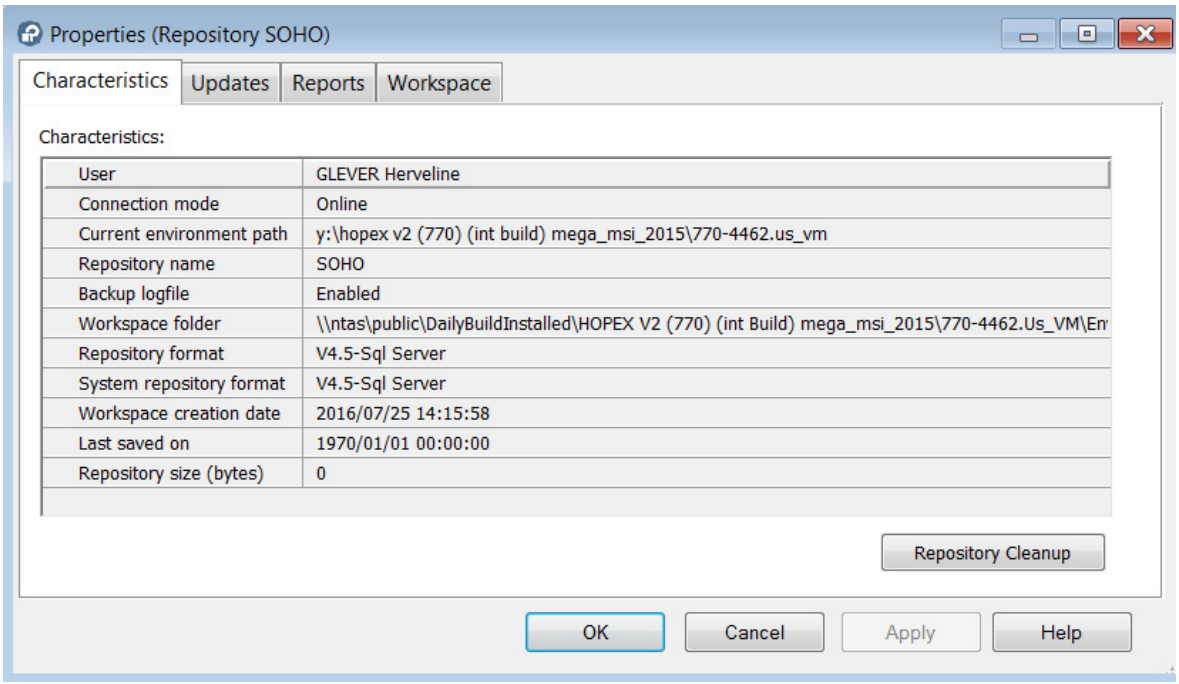
Workspace Properties

To consult your workspace properties:

1. Connect to **HOPEX**.

   See "Connecting to HOPEX", page 136.

2. In the **HOPEX** menu bar, select **File > Properties**.
   The properties dialog box of your workspace appears.



The workspace properties dialog box provides the following information on the current workspace:

- current **User**
- information on the current repository: its **Name**, **Backup logfile**, **Format**, **Workspace creation date**, **Last saved on**, **size**
- **Repository Cleanup** option.

☛ *For more information on repositories, see "Managing Repositories", page 73.*

HOPEX Repository State Changes

The integrity of the repository is assured by successive changes in its state.

☛ *See example "Private Workspace Life: Example", page 152.*

When repository updates are executed in a private workspace, they are only visible to other users when the user dispatches his/her work.

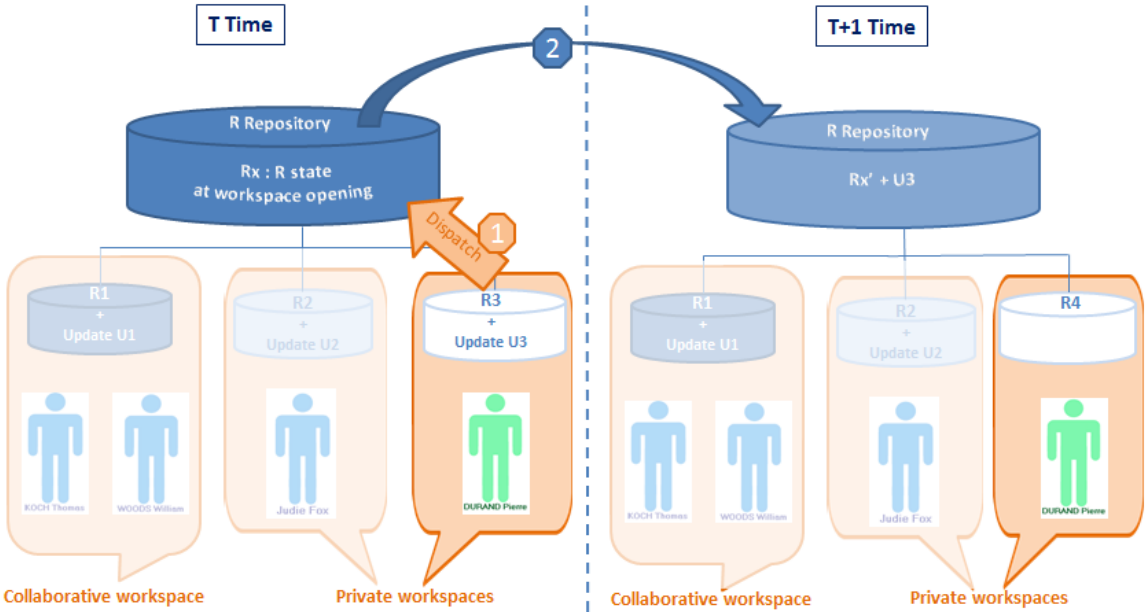
The repository changes from state N to state N+1 with memorization of new data related to the previous situation (state N).

If the user does not save updates, the changes made since the last valid state are forgotten. The repository remains in state N. Note that **HOPEX** repository state changes are managed automatically.

A workspace opens on the latest states of the repository and system repository.

Dispatching Your Work

Dispatch consists of making public the work carried out from a private workspace, or the work of participants in a collaborative workspace.



Dispatch allows:

- a user to make available to other users the modifications he/she has made to the repository.
- users of a collaborative workspace to make available to other users the modifications they have made to the repository.
- other users to have these updates available when they open a new workspace, whether this be after dispatch, refresh or discard of their current private workspace.

Dispatch:

- executes an update of the **HOPEX** repository and the system repository.
- creates a new workspace for the user containing all updates since creation of his/her previous private workspace.

Note that only one user can dispatch at a time. When several users dispatch their work at the same time, a dialog box appears asking the user if he/she wants to queue his/her dispatch. This allows the user to exit **HOPEX** without having to wait until the works from other queued private workspaces are dispatched.

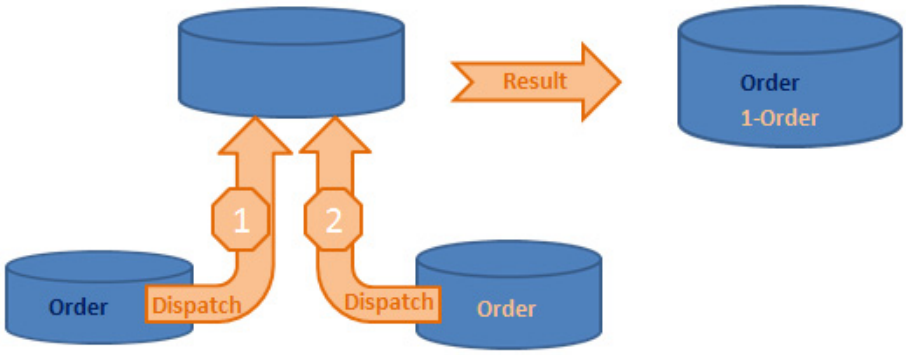
☛ *See "Dispatch Conflicts", page 141.*

Dispatch Conflicts

The dispatch process automatically manages most conflicts that may arise when several users make updates.

Creation of duplicated objects

When duplicate objects are created, a prefix is added before the name of the second object.



Two users create an object with the same name. The first to dispatch his/her work actually creates the object. The second user to dispatch his/her work creates an object with a prefixed name.

This is indicated in the dispatch report.

The administrator or the users can then decide to rename one of these objects if they are actually different, or combine them into one object if they are in fact the same object.

☛ *See "Merging Two Objects", page 225.*

Deletion of already deleted objects or links

As the deletion has already been performed, nothing happens. There is no mention of this in the dispatch report.

Links related to a renamed object

This happens when a user modifies an object that in the interim has been renamed by another user, or tries to link something to this object. The new one is kept and the changes are executed normally. The object can be found using its *absolute identifier*. Note that this does not create a reject, but the dispatch report indicates that the object was renamed.

📖 *An absolute identifier is a string of characters associated with each object in the repository. This string is computed using the date and time the session was opened, the number of objects created since the session started, and the object creation date (in milliseconds). An*

absolute identifier provides a unique way to identify an object in the repository, so that even if the object is renamed, all the links to it are retained.

Rejects When Dispatching

There are normally no rejects when dispatching of updates carried out in a private workspace. Most conflicts are managed automatically.

Rare cases of rejects are listed in the *rejects file*.

When updating a repository (importing, restoring, dispatching), a reject file is created in order to store rejected commands. Rejected commands are stored with the reason for which they were rejected. These are found in the "MegaCrd.txt" file located in the environment folder.

Change in writing access values between opening and dispatching a private workspace

If you have access to the writing access management function, you can, for example, protect an object in the repository while a user is deleting it in his/her private workspace. When the user dispatches his/her work, he/she is no longer allowed to delete the object and the deletion is rejected.

Rename/create collisions

A user renames an object in his/her private workspace, for example, from "Customer" to "Customer***s***". Then another user dispatches his/her private workspace in which he/she created an object having the same name "Customer***s***".

When the first user dispatches his/her private workspace, since the "Customer***s***" object already exists, the object "Customer" cannot be renamed "Customer***s***". The rename command will therefore be rejected.

Verifying link uniqueness

A user creates a link for which there is a uniqueness check. For example, he/she indicates that the "Order" message is sent by the "Customer" org-unit. In the meantime, another user indicates in his/her private workspace that the "Order" message is sent by the "Customers" org-unit. When the second user dispatches his/her private workspace, the link is rejected if the uniqueness control imposes that a message can be sent by only one object.

*For more information on MetaAssociation uniqueness check, see the **HOPEX Power Studio** - Imposing MetaAssociation Uniqueness.*

Attribute uniqueness (other than name)

Other attributes besides name may also be checked for uniqueness. If two users give two different objects the same value for this attribute, the second update will be rejected.

Updating a deleted object

If a user has made changes to an object in his/her private workspace and the object has been deleted by another user, the updates are rejected. The **Connect** and **Change** commands concerning this object are also rejected. All these rejects are listed in the private workspace report file.

Dispatch Report

The report file can be accessed from **HOPEX Administration**.

To view the report file:

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Right-click the environment and select **Reports > Open**.
   A new window allows you to view report contents.

   ☛ *See "Viewing the Environment Report File", page 108 for more details on these dialog boxes.*

The error level is indicated in the dispatch report.

- Error level 0: no error.
- Error level 2: minor errors such as attempts at creating already existing objects or links.
- Error level 4: the actions involved nonexistent objects or links.
- Error level 8: a system error was encountered during the update.
- Error level 16: update aborted because of a problem such as insufficient disk space.

  ☛ *These error levels are the same as those used for manual file imports.*

  ☛ *When manually importing a file, rejects concerning the creation of already existing objects or links can be filtered out using the Reprocess option.*

Objects that were renamed are also listed in the report.

Refreshing Data

A user can see modifications dispatched by other users of this repository without dispatching his/her own modifications. To do this, the user refreshes his/her data.

A user can refresh his/her data:

- in his/her private workspace
  The system creates a new private workspace, into which the *private workspace log* of the user's previous modifications is automatically imported.

  📖 *The private workspace log contains all modifications made by a user in his/her private workspace. It is applied to the repository at*

dispatch, then automatically reinitialized. This log is stored in the EMB private workspace file.

Refreshing allows a user to incorporate repository and system repository changes made by other users, without dispatching current work.

- in a collaborative workspace.
  The system then creates a new collaborative workspace for all participants in the collaborative workspace, into which is automatically imported the collaborative workspace log containing modifications previously made by participants.

  ☛ ***MEGA** recommends that you warn other participants before executing refresh.*

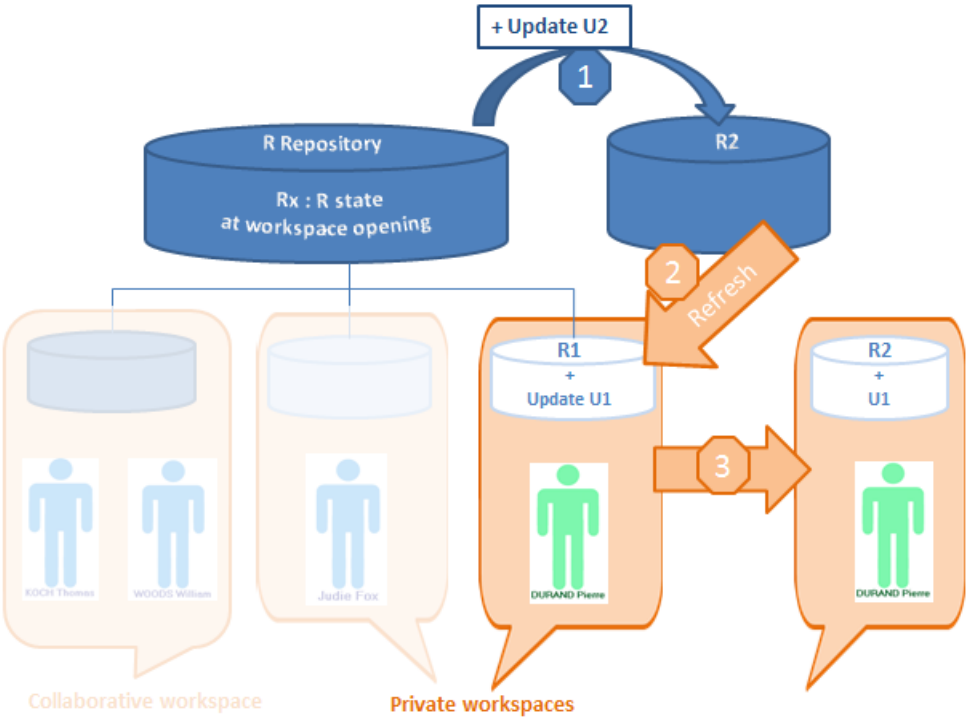
  Refreshing allows a user to incorporate repository and system repository changes made by other users, without dispatching current work.

Refreshing a private (or collaborative) workspace.

- does not update repository or system repository state.
- does not unlock objects modified in the private workspace.

☛ *see "Managing locks", page 161.*

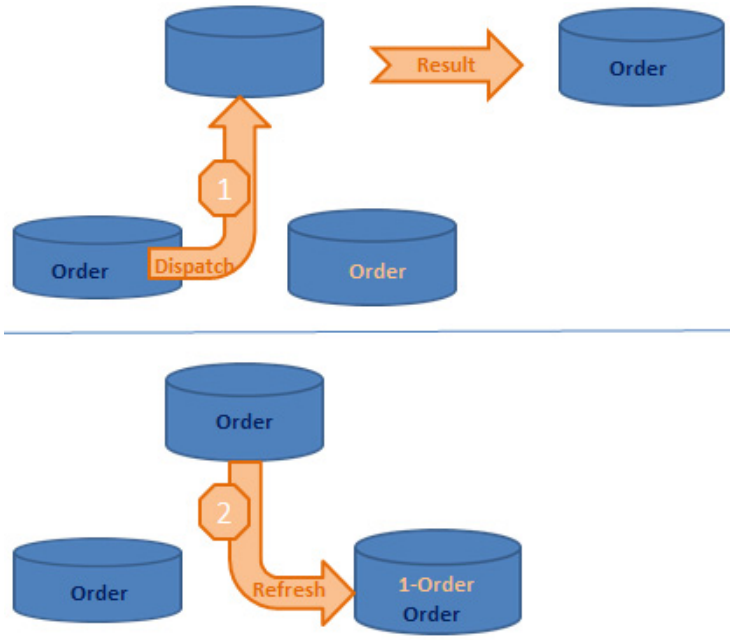
When a user resumes work on his/her private space that has lasted longer than the limit set by the administrator (the default is 6 days), **HOPEX** proposes that the user refreshes or dispatches his/her work.



Conflicts When Refreshing

Conflicts when refreshing are the same as when dispatching, but they apply to the private workspace only.

☛ *For more details on the main causes of rejects, see "Dispatch Conflicts", page 141 and "Rejects When Dispatching", page 142.*



As is true in dispatching, if two objects are created with the same name, the second object name is prefixed:

```
The second "Order" object is renamed "1-Order".
```

Discarding Work

Discarding a workspace (from a private or collaborative workspace) cancels all modifications made since the last dispatch. *Discard* of work causes loss of work carried out since opening of the private or collaborative workspace, including modifications to the desktop. A warning message reminds the user of this. Use discard with care.

Discarding work from a private workspace

In **HOPEX**, to discard your private workspace:

1. (Optional) It is advisable to export the private workspace before confirming the discard, see "Elements that could be useful to back up", page 93.
2. In the **HOPEX** menu bar, select **File > Discard**.

> ☛ *You can also discard your private workspace at disconnection, see "Exiting a Session", page 146 (choose not to dispatch modifications).*

Discarding work from a collaborative workspace

Only the collaborative workspace **Owner** can discard the collaborative workspace.

> ☛ *See the **HOPEX Common Features** guide, section "Working in a Collaborative Workspace".*

In **HOPEX**, to discard your collaborative workspace:

1. (Optional) It is advisable to export the collaborative workspace before confirming discard, see "Elements that could be useful to back up", page 93.
2. In the **HOPEX** menu bar, select **File > Workspace > Discard**.

Exiting a Session

When you exit **HOPEX**, you close your session. From:

- your private workspace you can:
  - save in the repository the modifications you have made in your private workspace
  - keep the modifications you have made in your private workspace

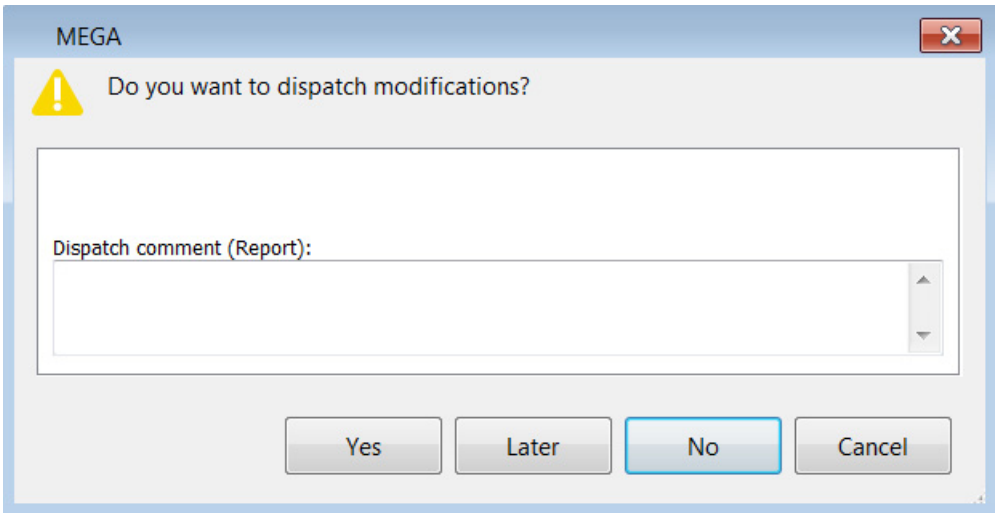
    > ☛ *These modifications will remain awaiting validation, subsequent modification, or deletion.*
  - cancel modifications you have made.
- a collaborative workspace you can:
  - keep modifications you have made

    > ☛ *These modifications are saved in the collaborative workspace. These modifications are not saved in the repository until the collaborative workspace is closed.*
  - cancel modifications you have made.

Exiting a session from a private workspace

In **HOPEX** (Windows Front-End), to exit your work *session*:

1. In the **HOPEX** menu bar, select **File > Exit**.
   The **HOPEX** exit dialog box appears.



2. (Optional) In the **Dispatch comment (Report)** frame, enter a comment to remind you of modifications made in your private workspace.
3. Select your **HOPEX** exit mode.
   - **Yes**
     Modifications you have made in your private workspace are saved in the repository.

     ☺ *In order to work effectively as a team, it is recommended that you dispatch frequently and regularly. Other users can update their own private workspace without dispatching their work (menu **File > Refresh**).*

     ☛ *This exit mode also allows the user to select a different repository the next time he/she logs in.*
   - **No**
     All modifications you made since your last dispatch will be lost. You can use this option if you want to view data quickly and exit without impacting the repository.

     ☛ *Modifications to your desktop are also lost.*
   - **Later**
     This option allows you to keep your changes without impacting the repository. You can open your session later and continue working but other users are not yet seeing the changes you have made.

     ☛ *Click **Cancel** to not exit your private workspace.*

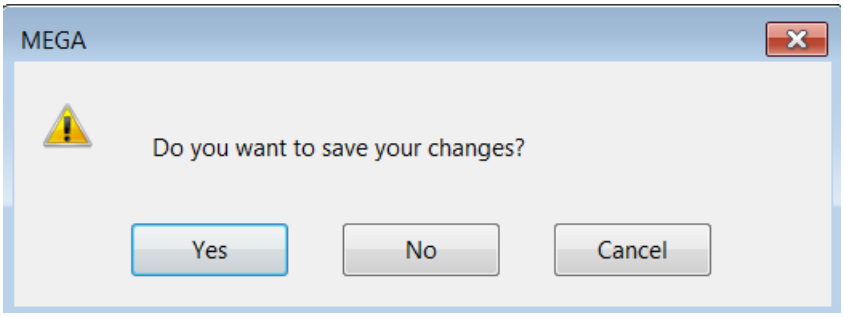
Exiting a session from a collaborative workspace

Exiting **HOPEX** from a collaborative workspace is the same whether you are its owner or not.

For as long as the collaborative workspace is not closed, participants can exit and rejoin the collaborative workspace at any time.

In **HOPEX** (Windows Front-End), to exit your work *session*:

1. In the **HOPEX** menu bar, select **File > Exit**.
   The **HOPEX** exit dialog box appears.



2. Click:
   - **Yes** to save your modifications in the collaborative workspace.
     You will be able to continue your modifications in a subsequent work session.
     These modifications are not saved in the repository. Users not participants in the collaborative workspace do not see these modifications.
   - **No** to cancel your modifications in the collaborative workspace.
     Your modifications are not saved in the collaborative workspace, but the latter remains available to carry out other updates.

     ☛ *Click **Cancel** to remain in your collaborative workspace.*

WORKSPACE ADMINISTRATION

You can view the list of current workspaces and their characteristics (owner, delay, status).

See:

- "Accessing workspace management", page 149
- "Deleting a Workspace", page 151

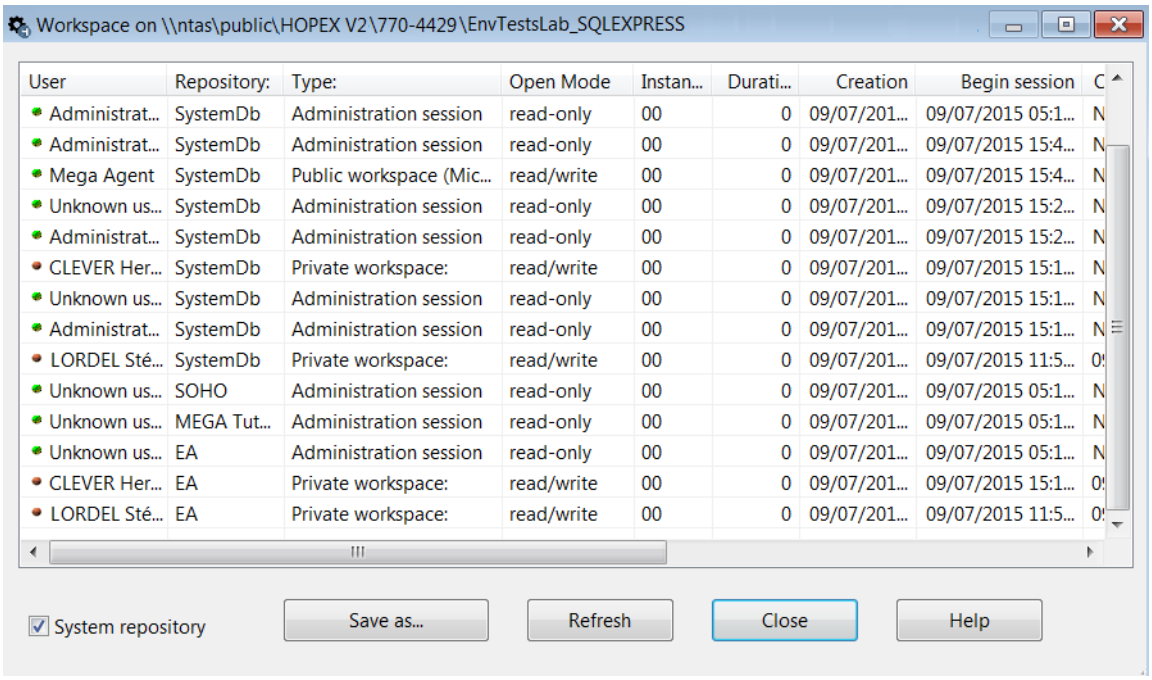
Accessing workspace management

To access the list of current workspaces in an environment:

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **Repositories** folder.
3. Expand the folder of the repository concerned, right-click **Workspaces** and select **Manage**.
   The management page for workspaces currently in progress in the environment appears.



User	Repository:	Type:	Open Mode	Instan...	Durati...	Creation	Begin session	C
Administrat...	SystemDb	Administration session	read-only	00	0	09/07/201...	09/07/2015 05:1...	N
Administrat...	SystemDb	Administration session	read-only	00	0	09/07/201...	09/07/2015 15:4...	N
Mega Agent	SystemDb	Public workspace (Mic...	read/write	00	0	09/07/201...	09/07/2015 15:4...	N
Unknown us...	SystemDb	Administration session	read-only	00	0	09/07/201...	09/07/2015 15:2...	N
Administrat...	SystemDb	Administration session	read-only	00	0	09/07/201...	09/07/2015 15:2...	N
CLEVER Her...	SystemDb	Private workspace:	read/write	00	0	09/07/201...	09/07/2015 15:1...	N
Unknown us...	SystemDb	Administration session	read-only	00	0	09/07/201...	09/07/2015 15:1...	N
Administrat...	SystemDb	Administration session	read-only	00	0	09/07/201...	09/07/2015 15:1...	N
LORDEL Sté...	SystemDb	Private workspace:	read/write	00	0	09/07/201...	09/07/2015 11:5...	0
Unknown us...	SOHO	Administration session	read-only	00	0	09/07/201...	09/07/2015 05:1...	N
Unknown us...	MEGA Tut...	Administration session	read-only	00	0	09/07/201...	09/07/2015 05:1...	N
Unknown us...	EA	Administration session	read-only	00	0	09/07/201...	09/07/2015 05:1...	N
CLEVER Her...	EA	Private workspace:	read/write	00	0	09/07/201...	09/07/2015 15:1...	0
LORDEL Sté...	EA	Private workspace:	read/write	00	0	09/07/201...	09/07/2015 11:5...	0

4. (Optional) Select **System Repository** to view system repository workspaces and the user connected to **HOPEX Administration**.

The private workspace dialog box details the following for each workspace:

> ☛ *To sort workspaces according to a column, click the header of the corresponding column.*
>
> ☺ *You can also arrange columns in the order you want. To do this, click the header of the column to be moved and, holding down the mouse button, move it to the required position.*

- its state:
  - the green button indicates that the workspace is open (private workspace) or in consultation (public workspace)
  - a red button indicates that the workspace is passive or that the user is disconnected.
- the **User** who created it and his/her **Code**.

> ☛ *In the case of collaborative workspaces, only the collaborative workspace owner appears.*

- the **Repository** to which it relates
- the **Type** of workspace:
  - "Private Workspace":
    The user can modify data. His/her updates are kept in his/her private workspace until dispatched.
  - "Collaborative Workspace":
    The participants of a collaborative workspace can, depending on their access level, modify the data. Their updates are kept in the collaborative workspace until the owner of this workspace decides to dispatch them.
  - "Administration Session":
    The user is Administrator type. The user cannot modify data.
  - "Public Workspace":
    The user can modify data. His/her updates are immediately visible to all other users.
- the **Open Mode** of the workspace, for example:
  - "read/write" when a session is open.
  - "read-only" when the user is in consultation only.
  - no value, if the private workspace is passive (the user has saved his/her session but is not currently connected to **HOPEX**).
  - no value if the user is in offline mode
- its **Duration** in days
- its **Creation** date and time
- the **Start** and **End** dates and times of the last **Session**
- the **Location** where it is stored
- its **Dispatching Delay**, which indicates its state compared to the current state of the repository (number of saves since the private workspace began, whether manual or automatic).
  Each time the repository is updated, the difference between the private workspace view and the repository state increases. This difference is measured by the private workspace delay. Dispatching or refreshing the private workspace reduces the delay to zero.

> ☺ *It is recommended that you refresh your workspace if its delay is greater than 50.*

Deleting a Workspace

The **HOPEX** administrator can delete a private workspace when this is passive.

The result is equivalent to discarding it.

To delete a workspace:

1. Open the workspace management dialog box.

   ☛ *See "Accessing workspace management", page 149.*

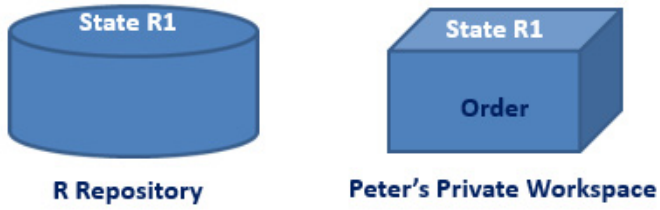
2. Right-click the workspace concerned and select **Delete**.

   When a workspace is deleted, the workspace in the work repository and that open on the system repository are deleted. Use private workspace deletion with care.

Private Workspace Life: Example

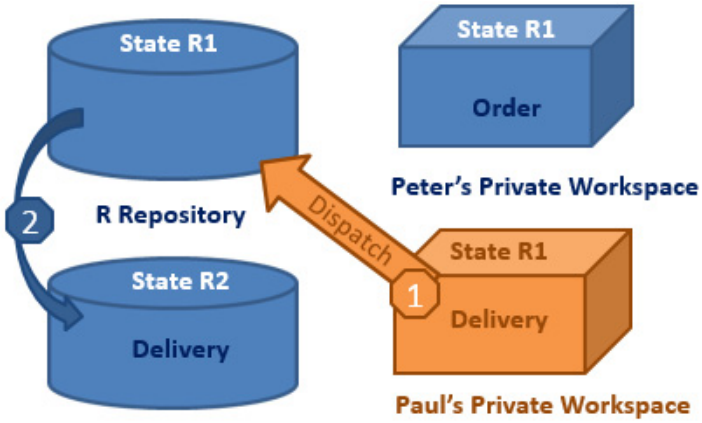
To illustrate how private workspaces work, the following is an example of some steps in the work of several users:

Private Workspace 1



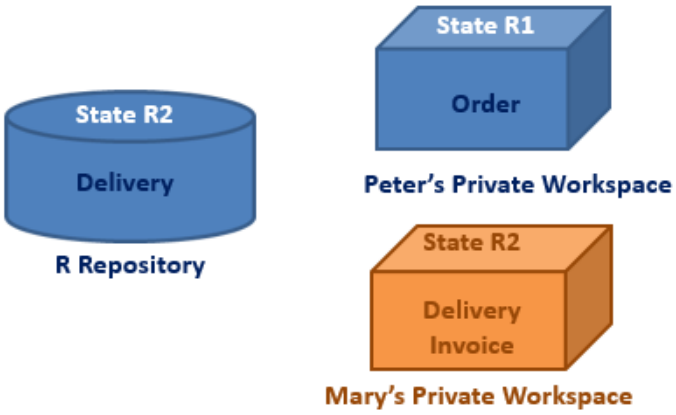
- Peter opens a private workspace, his repository view is state "n" (R1).
- He creates the "Order" message, which he links to the "Customer" org-unit.
- In parallel, Paul dispatches his work...

Private Workspace 2



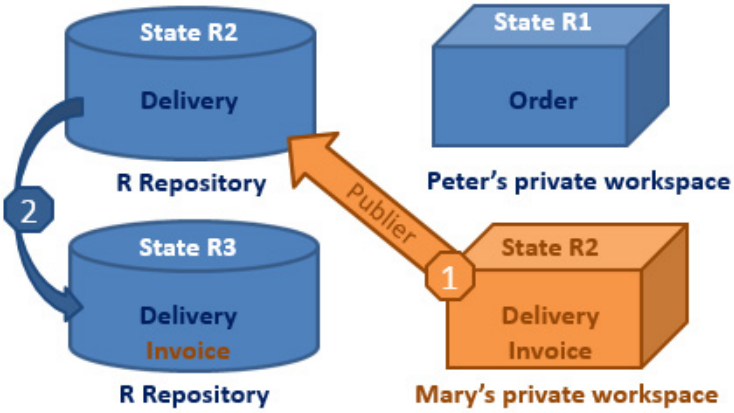
- Paul dispatched his work, which included the creation of the "Delivery" message, linked to the "Customer" org-unit.
- Paul's dispatch changes the repository to state "n+1" (R2).
- This new message is not seen from Peter's private workspace...

Private Workspace 3



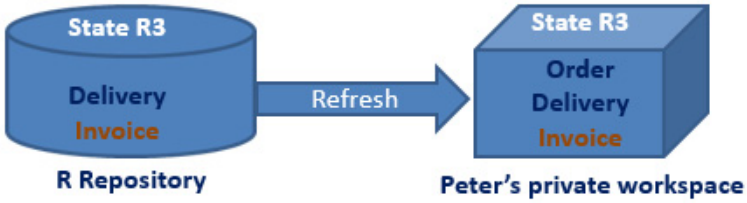
- Mary opens a new private workspace, her repository view is state "n+1" (R2).
- Mary creates the message "Invoice" connected to the "Customer" org-unit...

Private Workspace 4



- Mary dispatches her work.
- The repository passes to state "n+2" (R3).
- Peter's view is still in state "n" (R1).
- Peter refreshes his private workspace...

Private Workspace 5



- Peter has refreshed his private workspace.
- His view now corresponds to state "n+2" (R3).
- He can now see the "Customer" org-unit with the additions made by Paul and Mary.
- Peter dispatches his work...

Private Workspace 6

- When Peter, Paul, and Mary have dispatched their work, all the modifications they have made are visible in state "n+3" (R4) of the repository.

MANAGING UPDATES

During their modeling work, users make additions to a **HOPEX** repository within their private workspace: they create objects, links between objects, diagrams, etc. Updates corresponding to user actions can be viewed in detail. You can back up all modifications made to a repository from a private workspace in a private workspace log, which can be exported in the form of a command file.

The following points are detailed here:

- "Viewing Updates", page 155
- "Exporting Your Private Workspace Log", page 157
- "Private Workspaces and Repository Size", page 159

Viewing Updates

You can view updates:

- in your private workspace
- in the repository

Private workspaces

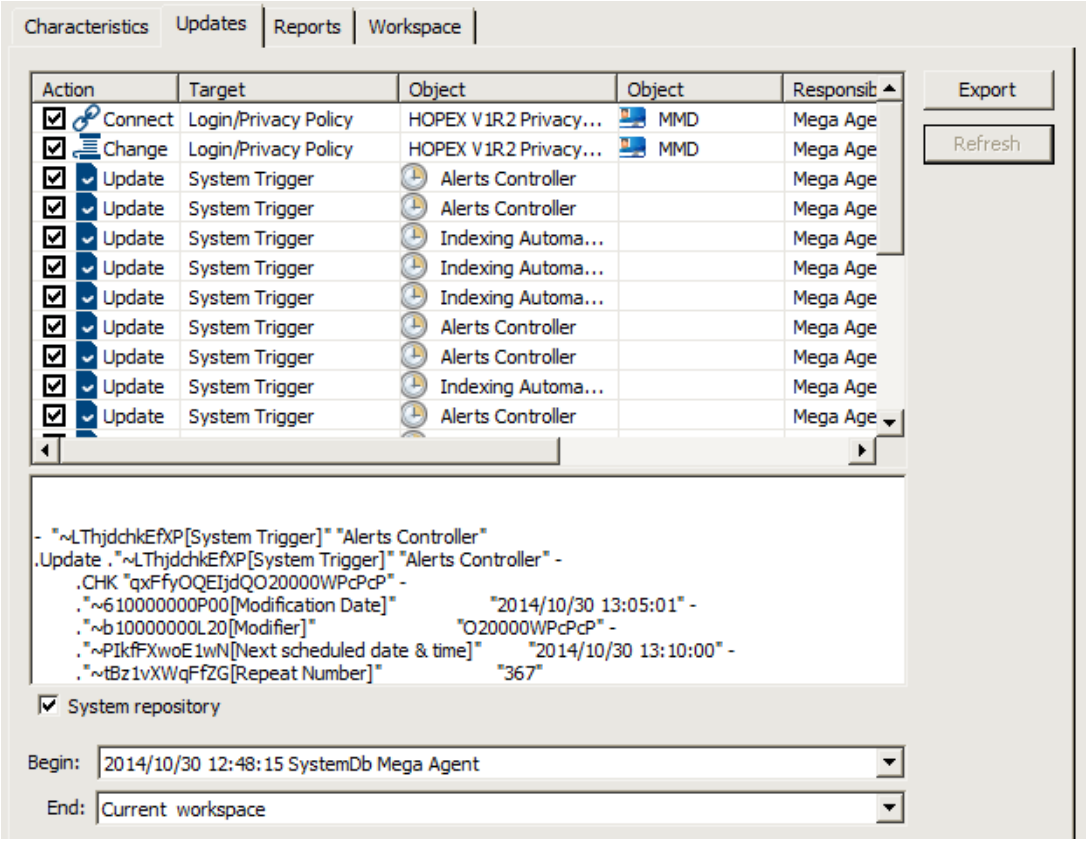
To view updates you have made in your private workspace:

1. In the **HOPEX** menu bar, select **File > Properties**.

   ☛ *This process may take some time if there are many updates.*

   The properties dialog box of your private workspace appears.

2. Select the **Updates** tab.



The **Properties** dialog box shows the commands you have executed in your private workspace or in your collaborative workspace. For each command, the following are indicated:

- the **Date** the command was executed.
- **Action** executed:

  `Example: "Create", "Connect", "Update".`

  ☛ *See "Command File Syntax", page 307 for more information on operators.*
- Object Type concerned in the **Target** column.
- the name of the **Object** concerned
- the name of the second **Object** concerned in the case of a "Connect", "Disconnect" or "Change" action.
- when you select the line of a command, the complete text of the update appears (lower frame of dialog box).

To sort the updates list:

❱ Click the column header.

To copy a command from the logfile (to paste it in the notepad for example):

1. Select the command.

2. In the lower frame of the dialog box, select the command text and simultaneously press keys <CTRL> + <C>.

To display only those commands concerning objects stored in the system repository (for example Person):

- Select the **System repository** check box.

In the logfile dialog box, the **Export** button enables export of selected commands in MGL or XMG format.

> ☛ *For more details on private workspace log export, see "Exporting Your Private Workspace Log", page 157.*

Dispatches

> ☛ *See "Viewing the Repository Update Log", page 86.*

The **Repository Dispatches** folder contains private workspaces dispatched in the current repository and in the system repository. Dispatches are filed by day, week and month.

To display private workspaces dispatched and the content of their updates:

1. From (Windows Front-End) **HOPEX**, in the navigation window **Repository Activity> Repository Dispatch**, right-click the dispatch and select **Properties**.

   > ☛ *To access the content of the **Repository Activity** navigation window, you must have **Expert** metamodel access (see "Configuring Metamodel Access", page 60).*

   Private workspaces dispatched on the current repository and system repository are contained in the repository folder.
2. Select the **Updates** tab.

Exporting Your Private Workspace Log

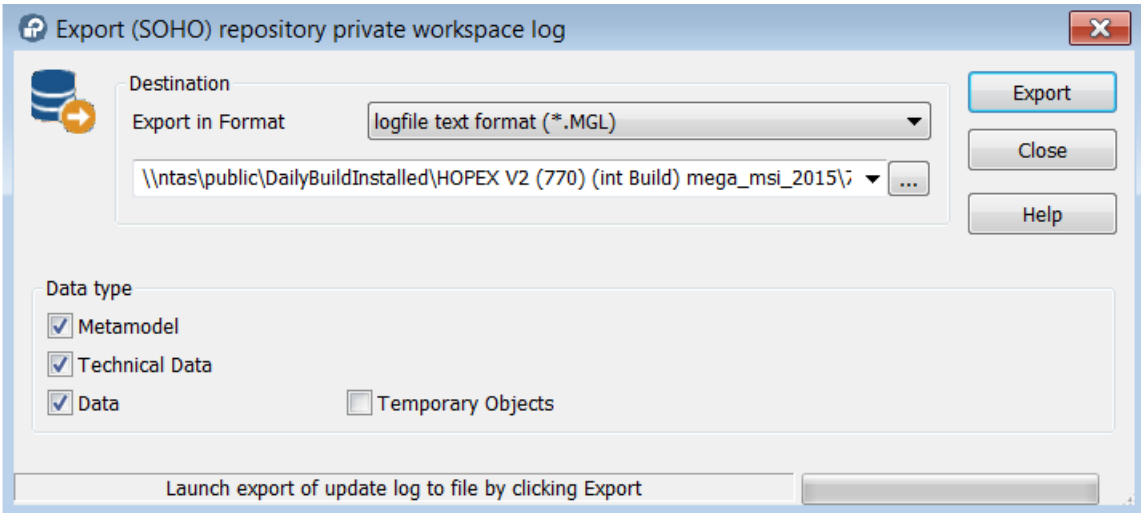
You can create an export file (*private workspace log*) and save it in your *work folder*.

The export file can be exported in format:

- **logfile text** (.mgl).
  Name format of the exported file is "LOGmmdd.mgl", where "mmdd" represents logfile export date month and day.
- **XML MEGA** (.xmg)
  The exported file is in the form of an XML file containing commands or data (objects and links).

To export the work done in the current private workspace in the form of a command file:

1. From the **HOPEX** menu bar, select **Tools > Manage > Export Updates Log**.
   The **Export Repository Private Workspace Log** dialog box opens.



2. Select export format.
3. (Optional) If necessary, modify the data export file name and save folder proposed as default. The **Browse** button ... allows you to browse the folder tree and select the folder in which the file will be placed.
4. In the **Data Type** frame, select the type of exported modifications:
   - **Metamodel** if you want to extract the metamodel from the system repository. This is useful if the standard metamodel has been modified.
   - **Technical Data** if you want to include in your file the changes made to data such as descriptors and queries.
   - **Data** if you only want to export changes made to repository data, particularly the workspace.
   - **Temporary Objects** - these objects are created when you execute requests, when you consult objects (stored in the history), etc. Generally you will not need to export these objects.
5. (Optional) If you want to export only those commands that correspond with a defined period, select **Commands from** / **to** and specify the dates.
6. Click **Export**.
   The **Execution Report** appears.
7. Click **OK**.
   The file is exported and saved in the specified folder.

The user can subsequently import this logfile, for example into a new private workspace.

Private Workspaces and Repository Size

Private workspace life

A private workspace gives a user a frozen view of a repository.

When the repository is modified by other dispatched private workspaces, this private workspace keeps the view it had when created.

Since the data corresponding to these views is kept in the repository, its size grows, and may become disproportionate to the actual repository contents.

☛ *See "Dispatching Your Work", page 140 and "Refreshing Data", page 143.*

Private workspace monitoring

More than one user can connect to a single repository via network share. The first time a user connects to the repository, he/she opens a private workspace. This private workspace ends only when the user dispatches, discards, or refreshes his/her modifications, and not when simply disconnecting from the **HOPEX** repository.

☛ *See "Refreshing Data", page 143 and "Discarding Work", page 145.*

In his/her private workspace, changes made by the user are saved in a temporary space (file for GBMS repository and data for RDBMS repository) dedicated to data of his/her private workspace. The repository is updated only when the user dispatches these changes.

☛ *See "Dispatching Your Work", page 140.*

All data accessed by a user is "frozen" for the duration of the private workspace.

Example:

```
For example, if an object is renamed in the reference
repository after the private workspace is opened, the user
sees the previous name unless the private workspace is
refreshed. However, users connecting after the modification
has been dispatched will have a view reflecting the most
recent state of the repository.

If other users connected before you dispatch your updates,
and are accessing the same data as you, they will have an
obsolete view of the data until they refresh their private
workspace. Note that when you dispatch your updates, your
view is refreshed automatically.
```

This means that data being accessed by user A and modified at the same time by user B is duplicated. It is stored in its initial state for each user private workspace; if a user makes a change, the previous state is stored along with the new one.

When the updates are dispatched and all users accessing the updated data have refreshed their private workspaces, the previous state is no longer required.

If a user does not want to dispatch his/her private workspace, refreshing it allows the user to avoid a large increase in the **HOPEX** repository size.

The administrator can set the maximum duration of a private workspace. If your private workspace exceeds the duration defined by your administrator, then each time you establish a connection, a message box appears asking you to dispatch or refresh your private workspace.

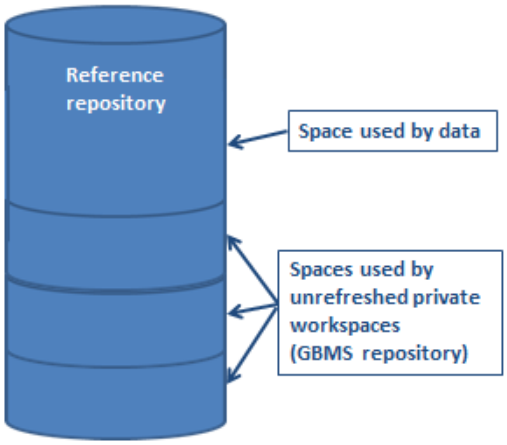
You can audit private workspaces with the **HOPEX Administration** application.

A private workspace that began before significant modifications were made may cause a considerable increase in repository size, since an image of the previous state of the modified elements is kept until the private workspace is refreshed.

The default option when a user disconnects is indicated in the user configuration. It is defined for the entire environment.

The greater the number of private workspaces and the longer their duration, the more the volume of the temporary space (file for GBMS repository and data for RDBMS repository) dedicated to private workspace data will increase. This volume can be reduced by repository backup or deletion, new repository creation or logical backup restoration.

For more information on reorganization of repositories, see "Reorganizing a repository", page 96.



MANAGING LOCKS

When several users are connected to the same repository simultaneously, there may be conflicts when they modify the same object in their different private workspaces.

See:

- "Principle", page 161
- "Managing Locks on Objects", page 163

Principle

With the network version, concurrent accesses to objects can be checked using *locks*.

Preventing conflicts

As soon as a user modifies an object, a lock is placed on this object. Another user can thus only view this object. This user cannot access a new update until the first user dispatches his/her private workspace, and he/she refreshes his/her private workspace. This prevents conflicts between the state of the object in the repository and the obsolete view of the second user.

Deleting a lock or unlocking an object

Lock management is automatic. You do not normally need to delete locks or unlock objects.

The few exceptions are due to abnormal operations, for example when a private workspace has been deleted from the private Workspace management window, or at desynchronization of clocks.

> ☛ *For more details on clock synchronization, see "Clock Synchronization", page 162.*

When a lock is deleted, another user can modify the object without dispatching or refreshing his/her private workspace.

> ☛ A user can delete locks placed on his/her private workspace since its creation.

When a user dispatches his/her work, the lock (his/she had placed on an object) is unlocked but not deleted. The lock is deleted only when all other private workspaces, which were created before the lock was unlocked, are closed. A private workspace is closed when it is dispatched, discarded, or refreshed.

Clock Synchronization

When lock management is active, workstation and environment clocks must be synchronized.

> **If clock times differ by more than 5 minutes when you connect to HOPEX, an error message appears and your work is saved at your workstation time. This situation can create consistency problems when dispatching your work.**

To be able to start **HOPEX**, you must synchronize the clocks:

Times relating to the network object locking system are always expressed in GMT0. For network workstations on which executables are decentralized, ensure that clocks are synchronized with a reference server. To do this, it is possible to use the "Net-Time" command or a Web clock synchronization service.

- With Windows NT and 2000, this can be done using a network command such as “Net time \\Workstation name/set” with LAN Manager or Windows NT.
- With Windows XP, setting is carried out from the Windows **Control Panel**, **Date/Time** icon.

> *For more information on clock synchronization, see Windows online Help.*

Details on Lock Operating

HOPEX only indicates that objects are locked when their attributes are modified (unlike links for example).

Warning on unlocking

If you attempted to use an object that was locked, a message alerts you as soon as the object is freed for you to use again.

Diagrams

There are two types of locking applied to diagrams

- ***The diagram has simply been viewed and not modified***: as soon as the first user closes the diagram it can be opened by a second user.
- ***The diagram has been modified***: as for classical locking, the second user must wait until the diagram has been dispatched by the first user and therefore unlocked.

Managing Locks on Objects

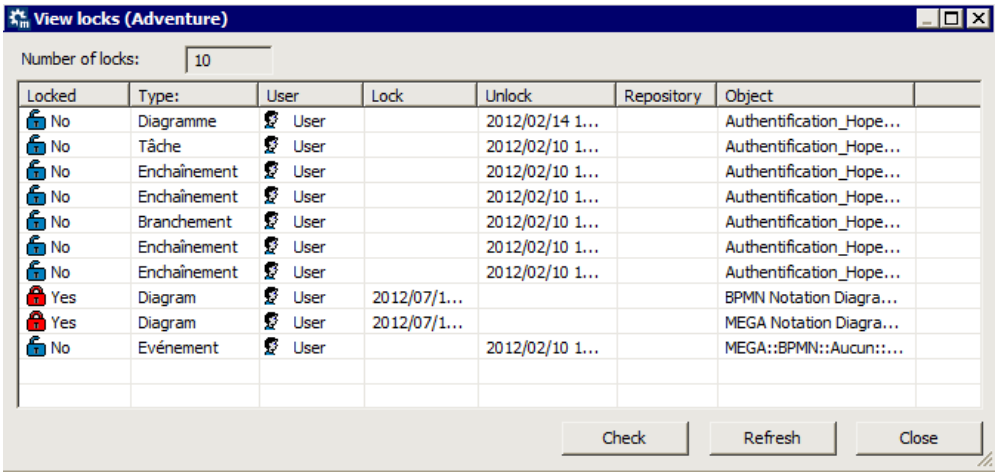
The lock management window can be accessed from:

- **HOPEX Administration**

  ☛ *"Managing locks from HOPEX Administration", page 163*

- **HOPEX**

  ☛ *"Managing locks from HOPEX", page 164*



Locked	Type:	User	Lock	Unlock	Repository	Object
No	Diagramme	User		2012/02/14 1...		Authentification_Hope...
No	Tâche	User		2012/02/10 1...		Authentification_Hope...
No	Enchaînement	User		2012/02/10 1...		Authentification_Hope...
No	Enchaînement	User		2012/02/10 1...		Authentification_Hope...
No	Branchement	User		2012/02/10 1...		Authentification_Hope...
No	Enchaînement	User		2012/02/10 1...		Authentification_Hope...
No	Enchaînement	User		2012/02/10 1...		Authentification_Hope...
Yes	Diagram	User	2012/07/1...			BPMN Notation Diagra...
Yes	Diagram	User	2012/07/1...			MEGA Notation Diagra...
No	Événement	User		2012/02/10 1...		MEGA::BPMN::Aucun::...

The **Display Locks** window displays the locks created since the creation of the oldest private workspace. The following information is provided for each lock:

- the lock state (**Locked**):
  - **Yes**: When a user locks an object, other users can only view it, even if they dispatch their work.
  - **No**: the object is locked when its modifications are dispatched. Other users must dispatch their work or refresh their private workspaces before they can modify the object.

    ☛ *To unlock an object or delete a lock, see "Managing locks from HOPEX Administration", page 163*
- the name of the **Object** concerned
- the **Type** of object concerned
- the **User** who owns the lock
- the date and time (GMT0) of the **Lock**, and, if applicable, **Unlock**.
- the name of the **Repository** to which the lock relates.

Managing locks from HOPEX Administration

To manage locks from the **Administration** application:

1. Connect to the environment concerned.

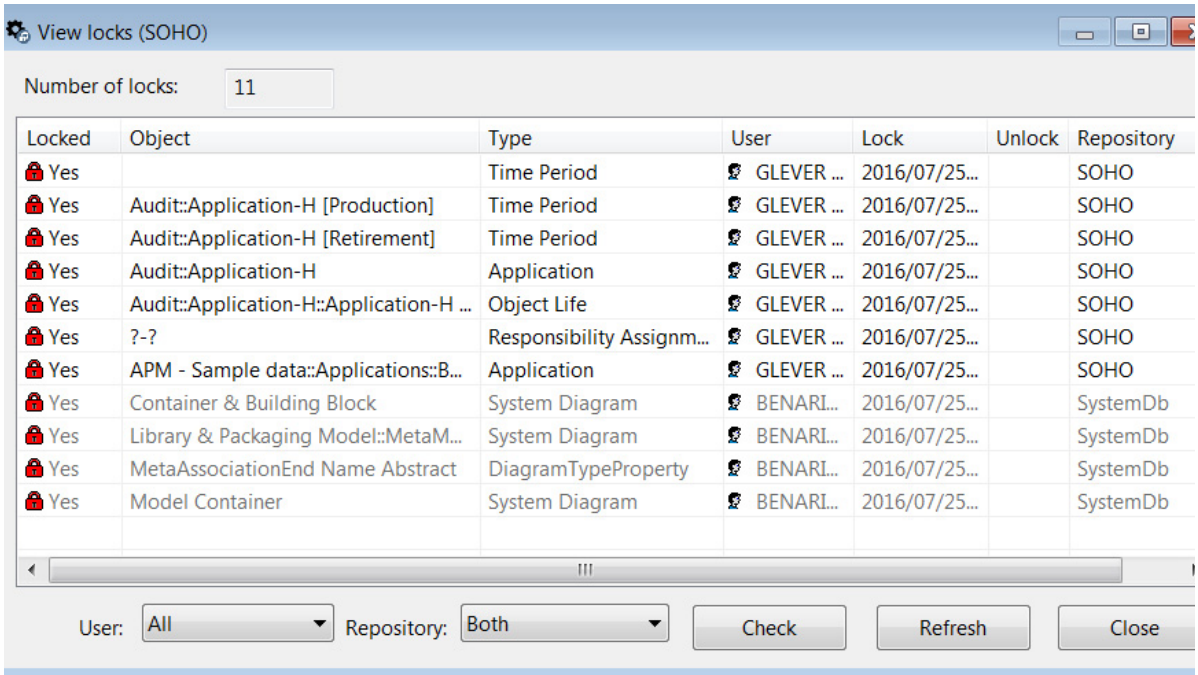
   ☛ *See "Connecting to an Environment", page 7.*
2. Expand the **Repositories** folder, then that of the repository concerned.

3. Right-click the **Locks** folder and select **Manage**.
   The **View locks** dialog box opens.
4. To:
   - unlock an object, right-click the lock concerned and select **Unlock**.
   - delete an object, right-click the lock concerned and select **Delete**.

Managing locks from HOPEX

To consult locks from **HOPEX**:

1. In the **HOPEX** menu bar, select **Tools > Manage > View Object Locks**.
   The **View locks** dialog box opens.



   The locks appear grayed, except that of the current user.
2. To:
   - delete an object, right-click the lock concerned and select **Delete**.
   - unlock an object, right-click the lock concerned and select **Unlock**.

To:

- sort locks as a function of a column, click the column header.

  ☺ *You can also arrange columns in the order you want. To do this, click the header of the column to be moved and, holding down the mouse button, move it to the required position.*

- refresh lock display, click **Refresh**.
- sort the display according to **User**, select:
  - **Current** to view your locks only.
  - **Other users** to view locks of other users only.
  - **All** to view your locks and those of other users.
- sort the display according to **Base**, select:
  - **Current Repository** enables display of locks on the current repository.
  - **System Repository** enables display of locks on the system repository.
  - **Both** enables display of locks on current and system repositories.

MANAGING ENVIRONMENTS

When multiple users are working together across a network, it may be useful to create several work environments. Another reason for creating a new environment is to provide the administrator with an environment where report templates (MS Word), queries, etc. can be modified and tested without interfering with users' work.

Environment management functions are used when several environments are available and they need to exchange data or repositories.

The following points are covered here:

- ✓ "Environment Structure", page 168
- ✓ "Using Environments", page 169
- ✓ "Moving and Referencing an Environment", page 171
- ✓ "Deactivating an environment (RDBMS)", page 173
- ✓ "Customizing Environments", page 174
- ✓ "Comparing Environments", page 180

ENVIRONMENT STRUCTURE

An environment groups a set of users, the repositories on which they can work, and the system repository. It is where user private workspaces, users, system data, etc. are managed.

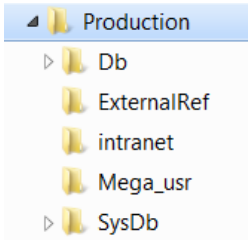
An *environment* comprises:

- A configuration file.
- A system repository (see "System Repository", page 168)
- Shapes, if they are different from the site standard shapes.
- A "Db" folder for the repositories used in this environment.

In most cases, there is only one environment per site. It is possible to define several environments when the *site* includes repositories on which users work with different resources (such as specific report templates (MS Word) or metamodel extensions).

All users must have full access rights to the resources of the environments and repositories (creation and deletion of files and folders).

The folder structure of an environment is as follows:



- ***Db***: contains the environment repositories

  Each repository is located in a folder carrying its name. See "Repository Structure", page 75.

- ***ExternalRef***: contains external references of the environment
- ***intranet***: contains the Web sites generated in the environment
- ***Mega_usr***: contains shapes specific to the environment
- ***SysDb***: the system repository

System Repository

The system repository (SysDb) contains the configuration required to run **HOPEX**:

- the metamodel, constituting the structure of repositories
- programmed *queries* and macros
- the elements to produce outputs: report templates, *report templates (MS Word)*, Web site templates
- users and their rights.

A system repository exists for each environment, automatically created in the **Sysdb** folder at creation of the environment.

USING ENVIRONMENTS

The following points cover certain basic operations that can be executed on an environment:

- "Creating an Environment", page 169
- "Copying a GBMS Environment", page 170
- "Deleting a GBMS Environment", page 170

Creating an Environment

To create an environment:

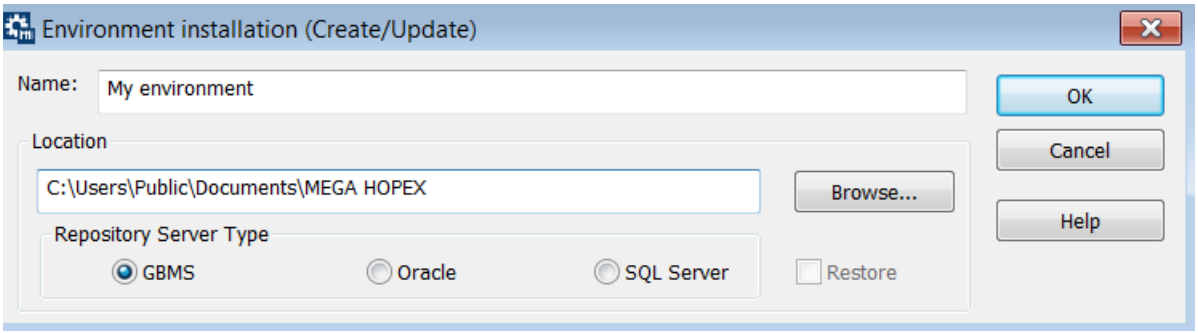
1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. In the navigation tree, right-click **Environments** and select **New**.
3. Enter the **Name** of the new environment.

   ☛ *The environment name must respect Windows folder naming constraints.*

4. (Optional) To modify the default environment location, click **Browse**.



5. Select the repository server type for the new environment.

   ☛ *Possible server types are: GBMS (**MEGA**) and RDBMS (Oracle and SQL Server). For more details on **HOPEX** repository server type, see the **HOPEX** deployment guide.*

6. Click **OK**.
   Creation and installation of the new environment are started.

   ☛ *(GBMS) The SystemDb repository is rebuilt and environment creation takes several minutes.*

   (RDBMS) The SystemDb repository is built dynamically and environment creation can take several hours.

   A directory is created. It hosts the environment structure.

   ☛ *When you create a new environment, it is automatically referenced, and it appears in the list of environments.*

   ☛ *The environment is identified by the complete path of its root directory. It is therefore essential that all users see the network resource in which it is located under the same name.*

Copying a GBMS Environment

Prior to this operation, check that there is no private workspace active on the environment.

To copy an environment:

- With Windows explorer, copy the following files and folders:
- The environment root directory, which contains:
  - the environment system repository
  - all or some of the repositories that it references
  - the shared reports (MS Word) folder, "Document"
  - the "MEGA_Usr" folder containing the custom shapes and style sheets.
- Directories of repositories that are not stored under the environment root.

At the time of this operation, it can be useful to delete work folders of users and administrators.

When the operation is completed, you should check in each repository that the reports (MS Word) and Web Sites point to the correct network locations.

Deleting a GBMS Environment

Prior to this operation, check that there is no private workspace active on the environment.

To delete an obsolete environment with all its repositories:

1. Delete the reference for the old environment.

   See "Deleting a Reference to an Environment", page 172.

2. From the Windows explorer, delete the root directory of the environment and the folders of repositories that will not be hosted under this root.

Moving and Referencing an Environment

During administration of **HOPEX**, you may need to move an environment. When an environment is moved, you must create a new reference for it to be able to use it.

When you delete an environment, it is recommended that you delete its reference. It will then no longer appear in the list of environments available in the corresponding **HOPEX** site.

For more details on these options, see:

- "Moving an Environment", page 171
- "Referencing an Environment", page 171
- "Deleting a Reference to an Environment", page 172

Moving an Environment

When you need to move an environment, such as when you have to place it on a different drive:

1. Copy the root folder of the environment in its new location.
2. Delete the reference for the old environment.
3. Create a reference for it at its new location.
4. Carry out the same operations for each of the repositories not hosted in the environment structure.
5. Check that the reports (MS Word) and Web sites of each repository point to the correct files.

Precautions to be taken:

- Verify that there is enough free space in the destination folder.
- It is not necessary to dispatch private workspaces before moving the environment. They will be moved with the environment.
- No user should be connected during movement of environments. No private workspace should be active.

Referencing an Environment

When an environment has been moved by the user, it is necessary to create a reference for this environment so the site will recognize it.

To create a reference to an environment:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. In the navigation tree, right-click **Environments** and select **Create Reference**.
3. In the dialog box that opens, select the folder in which the environment to be referenced is located.

4. Click **OK** to validate.
   The new environment reference is created and appears in the list of available environments.

Deleting a Reference to an Environment

To delete a reference to an environment:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. In the navigation tree, right-click the desired environment and select **Delete Reference**.
   A confirmation request appears.
3. Select **Yes**.
   The environment reference is deleted and no longer appears in the list of available environments.

DEACTIVATING AN ENVIRONMENT (RDBMS)

In an RDBMS environment, for administration requirements, you can prevent users from connecting to **HOPEX**.

You can block all users or Web users only.

The users already connected are not disconnected and can continue to work.

To deactivate an environment:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. In the navigation tree, right-click the environment (RDBMS) that you want to make inactive and select **Environment Activity** then:
   - **Inactive environment**: only the **Administration (Windows Front-End)** application is accessible. Users cannot connect to **HOPEX** (Windows Front-End and Web Front-End).
   - **Active environment with Windows-Front-End only**: access to **HOPEX (Web Front-End)** is blocked.

CUSTOMIZING ENVIRONMENTS

Administration tasks may involve you in modifying configuration of an environment so that it meets your particular requirements.

With **HOPEX Administration** you can perform the following actions:

- "Backing Up Environment Customizations", page 174
- "Restoring Environment Customizations", page 176
- "Compiling an Environment", page 177

Backing Up Environment Customizations

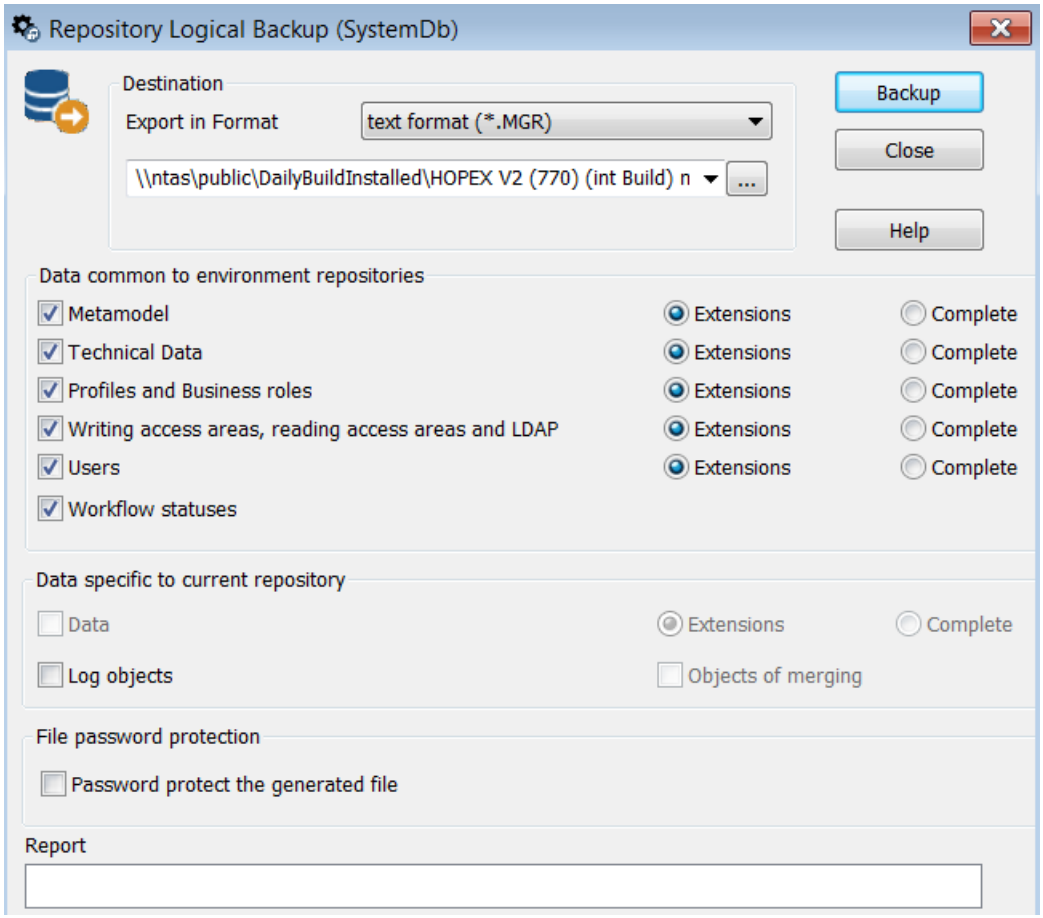
The following operations allow you to back up modifications made to the standard environment so that these can be imported into a new environment. This backup contains modifications you have made to *report templates (MS Word)*, Web site templates, *descriptors*, *queries* etc. It also contains the list of users and their configuration if this exists. You can also save any extensions to the metamodel.

To back up customizations:

1. From **HOPEX Administration**, connect to the environment concerned.

☛ *See "Connecting to an Environment", page 7.*

2. In the **Repositories** folder, right-click the system repository (SystemDb) and select **Logical Backup**.
   The **Repository Logical Backup** dialog box opens.



3. In the **Destination** frame, select the export format of the backup file:
   - **text** (.MGR)
   - **XML MEGA** (.XMG)

   ☛ *This format cannot be used to extract the metamodel or technical data. This format is reserved for exchange of data between **HOPEX** and other applications. It includes commands or data (objects and links).*

4. (Optional) In the **Destination** frame, click **Browse** ... to browse the folder tree and modify the name and/or location of the backup file.

   ☛ *By default, the backup is saved in the "SystemDb.mgr" file in the \SysDb\WORK\ folder of the repository.*

5. Select the type of data you want to save.

   ☛ *The **Extensions** button corresponds to data created by users.*

   Carry out a complete backup only if technical support asks you to do so.

   ☺ *It is recommended that you save the extensions of the metamodel and technical data in different files.*

   In the frame **Data common to environment repositories**, select the data type of the system repository to be saved:

   - **Metamodel** if you want to extract the metamodel from the system repository. This is useful if the standard metamodel has been modified.
   - **Technical Data** allows extraction from the system repository of data such as *descriptors*, *queries*, and *report templates (MS Word)*.
   - **Profiles and Business roles** allows extraction of created profiles and business roles (those not provided by MEGA).
   - **Writing access areas, reading access areas and LDAP** allows extraction of created writing and reading access areas (those not provided by MEGA) and LDAP parameters (parameters, servers, groups).
   - **Users** allows extraction of created users (persons, person groups, logins).
   - **Workflow statuses** enables extraction of workflows (workflow instances, transitions and statuses, tasks, validations, requests for change).

   In the **Data specific to current repository** pane:

   - **Log objects** allows you to also save object logs.

   ☛ *For more information on histories, see "Deleting a Repository", page 89 and "Viewing Object History", page 112.*

6. (If needed) In the **File password protection** pane, select **Password protect the generated file**.

   ☛ *The password is asked at file import.*

7. Click **Backup** to start backup.
   A series of messages keeps you informed of backup progress.
   The file is saved in the selected format (*.mgr or *.xmg) or in *.mgz format if you have selected the protection option.

Restoring Environment Customizations

You can restore environment customizations that you have backed up.

☛ *See "Backing Up Environment Customizations", page 174.*

To restore environment customizations:

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Under the **Repositories** folder, right-click the repository concerned and select **Object Management > Import**.

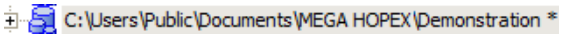
   ☛ *See "Updating a Repository", page 103.*

Compiling an Environment

The metamodel and technical data must be compiled after migration or customization. This is to check configuration of the environment concerned. When compilation has been completed, processing for all users of this environment is speeded up.

HOPEX can operate in "interpreted" (not compiled) mode but with reduced performance.

In the **HOPEX Administration** navigation tree, an asterisk after the environment name indicates that the metamodel or/and technical data (excluding permissions) of this environment is/are in "interpreted" (not compiled) mode.



Metamodel compilation includes in parallel translation in the current language. You can also translate the metamodel into another language.

Prerequisite:

1. Ask **HOPEX** users to exit their application:
   - mandatory for **Web Front-End** users
   - recommended for **Windows Front-End** users.
2. Start the **HOPEXMega Server Supervisor** tool.

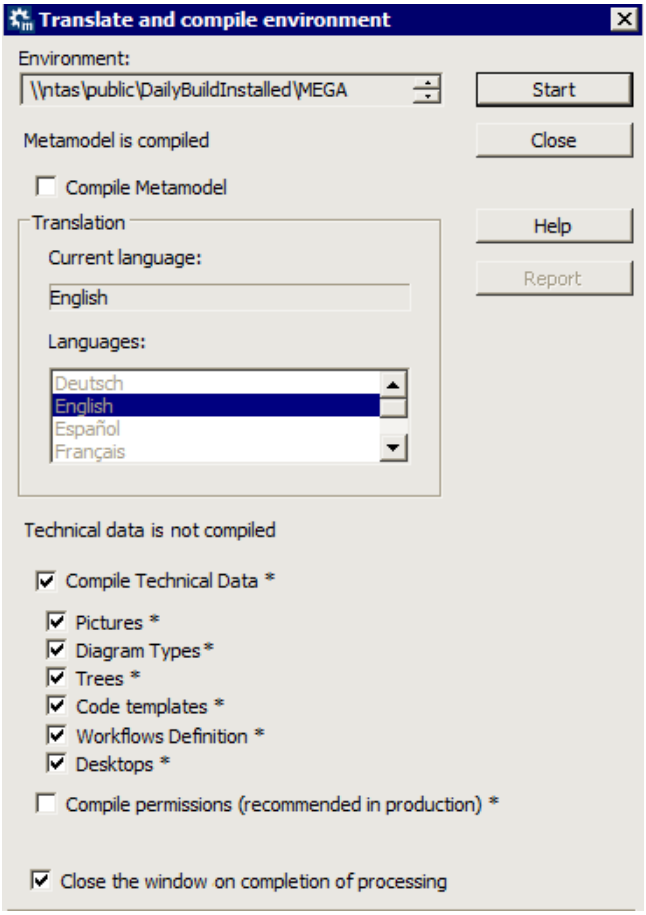
   ☛ *See "Starting HOPEX Server Supervisor", page 190.*
3. In the identification area of your workstation, right-click **HOPEX Server Supervisor** and select **System > Stop HOPEX Processes Services and Web Application**.

To translate and compile the metamodel and/or compile technical data:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. In the navigation tree, right-click the desired environment and select **Metamodel > Translate and Compile**.
   The **Translate and compile environment** dialog box opens.



   * : the asterisk indicates interpreted (not compiled) mode.

   In the **Translation** frame, the **Current Language** field indicates the current language of the system repository.

3. If the metamodel is not compiled, keep **Compile Metamodel*** selected.
4. (If **Compile Metamodel*** is selected) In the **Translation** frame, in the **Languages** list of the system repository, select the target translation language.

   `Example: "English"`

5. If technical data is not compiled, keep **Compile Technical Data*** selected.

   If technical data and metamodel are not both compiled, you must also keep Compile Metamodel* selected.

   By default, all technical data (images, diagram types, trees, code templates, workflow definitions, desktops) is selected.

6. If you are in a production environment, keep the **Compile Permissions*** option selected; otherwise, you can erase the selection. This compilation improves **HOPEX** loading times.

   ☛ *Compilation of permissions can take some time (more than an hour) and is recommended only in a production environment. The fact of not compiling permissions (permissions interpreted mode) has no impact on correct operation of **HOPEX**.*

7. Keep **Close the window on completion of processing** option selected.

   ☛ *This option enables automatic closing of the **Translate and Compile Environment** window when compilation is completed, allowing **HOPEX (Windows Front-End)** users to resume their work.*

8. Click **Start** to run compilation and translation.

   ☛ *If **HOPEX (Windows Front-End)** users have remained connected, they are blocked during processing.*

   Metamodel and/or technical data compilation (excluding permissions) takes several minutes.

   If you selected metamodel compilation with a different target language, after execution the system repository is available in the new language.

9. When compilation processing is completed, **HOPEX** (Windows Front-End) users can resume their work.

   ☛ *If at step 7 the **Close the window on completion of processing** option was not selected, click **Close** to close the **Translate and Compile Environment** window and allow **HOPEX (Windows Front-End)** users to resume their work.*

10. In the identification area of your workstation, right-click **HOPEX Server Supervisor** and select **System > Restart HOPEX Processes Services and Web Application**.

    ☛ ***HOPEX (Web Front-End)** users can connect to Web applications.*

COMPARING ENVIRONMENTS

The environment comparator enables evaluation of the differences between two environments and production of an update file that enables switching between one and the other. It enables detection of all changes to an environment or focuses only on certain aspects of customization by carrying out a comparison by topic. You therefore have available a tool enabling detection of metamodel extensions.

This comparator only functions if both environments have the same level of metamodel, meaning that they must share the same list of languages and the same MetaClass and MetaAttribute definitions.

☛ *This tool is written using **MEGA** APIs, and it is essential that the **HOPEX** version is correctly referenced in the registry.*

See:

- "Comparison Types", page 180
- "Environment Comparator Configuration", page 181
- "Running the Environment Comparison", page 182

Comparison Types

You can run an analysis:

- standard (selected by default)
  It is a complete comparison of the two environments.
- by topic
  If you do not wish to carry out standard analysis due to lack of time, or if you need to compare a certain object type only.

Standard analysis

Both environments must be compiled in the same language.

☛ *Standard analysis is an extremely long process: it takes 8 - 12 hours.*

Standard Analysis consists of comparing:

- **Metamodel**:
  The metamodel analysis phase spots metamodel changes. If these changes are significant and concern addition of a language or modification of **HOPEX** "core" data, comparison will not continue.
- **Technical data**:
  The technical data analysis phase detects all customizations of the system repository, excepting the metamodel.

This analysis produces:

- a report file (**XML report**)
- an update command file (Upgrade **MGR file**)
  This command file applied to the source environment allows us to obtain the target environment.

Analysis by topic

You can carry out comparison by topic:

- **Metamodel** to compare **HOPEX** core metamodel elements:
  - MetaClasses
  - MetaAssociations
  - MetaAttributes
- **Diagram settings** to compare all diagram configuration elements.

  `Example: DIAGRAMTYPEFIELD, MODELTYPELINK, DIAGRAMTYPEPARAM.`
- **Report templates (MS Word)** to compare Report Templates (MS Word).

  ☛ *This option does not cover comparison of report template (MS Word) descriptors and queries.*
- **Descriptors** to compare descriptors.
- **Queries**to compare queries.
- **Web supplies** to compare the main components of Web sites: Site Web template.

  ☛ *Web site descriptors and queries are not analyzed.*
- **MOKA supplies** to compare the main components of MOKA supplies:
  - Project type
  - Phase
  - System diagram
  - Deliverable
  - Project context
- **Generators** to compare the main components of generators:
  - Generator
  - Generation rule
  - Generation kinematics

  ☛ *Generator descriptors and queries are not analyzed.*

This comparison produces a report covering only the differences detected (**XML report**).

Environment Comparator Configuration

See:

- "Configuring the comparison", page 182
- "Configuring the XML Report File", page 182

Configuring the comparison

You can alter comparator behavior depending on your knowledge of modifications carried out in your environment. Avoiding unnecessary comparisons significantly reduces total analysis time.

You can compare:

- translations (**Compare Translation**)

  ☛ *Clearing this option deactivates comparison of translations and reduces processing time by around 20%.*

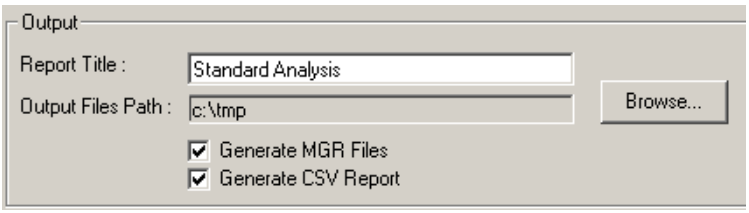
- comments (**Compare Comments**)

  ☛ *Clearing this option deactivates comparison of object comments and reduces processing time.*

Configuring the XML Report File

To configure the XML report file, you can modify:

- its name in the **Report Title** field.
- its path in the **Output Files Path** field.



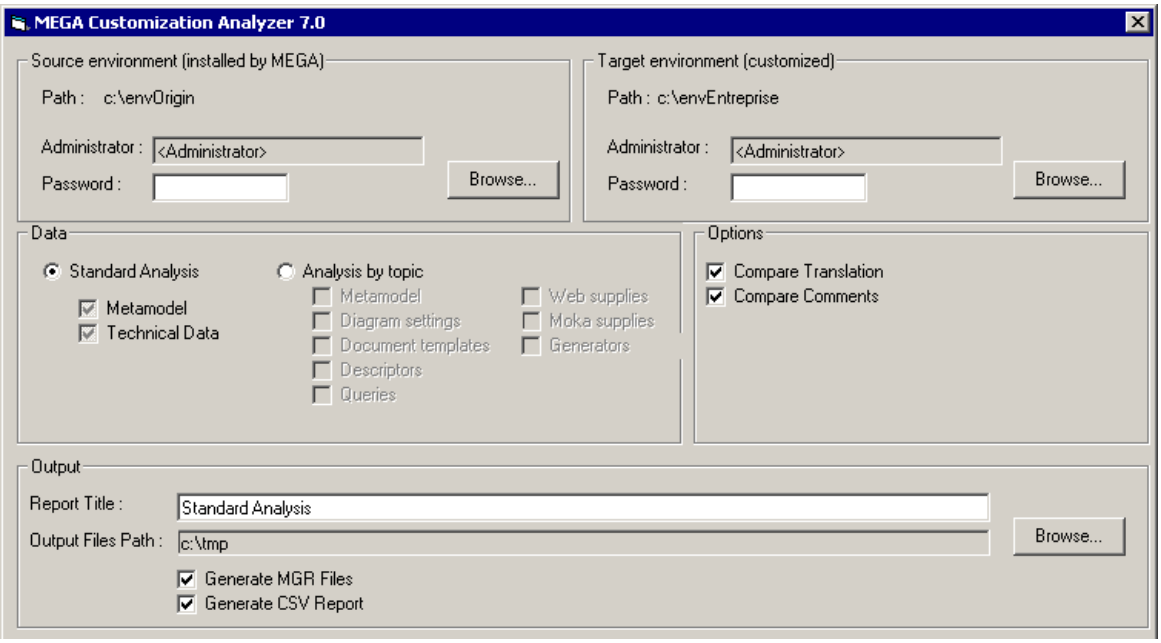
The XML comparison file opens automatically on completion of processing.

Running the Environment Comparison

To compare environments:

1. In the **HOPEX** site directory, open **Utilities** folder.

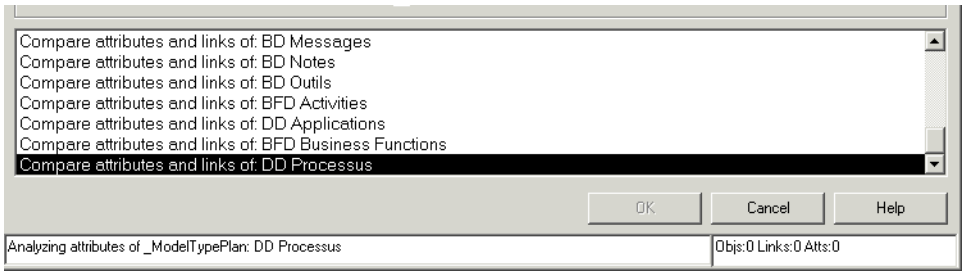
2. In the **MEGA Customization Analyzer** folder, double-click "mgwenvc.exe" file .
   The **MEGA Customization Analyzer** application opens.



If you have modified your work environment (Target environment) by importing corrective files of type .mol provided by MEGA, you must also apply these to the source environment. In this way the comparison will check out only your modifications.

3. In the **Source environment** pane, click **Browse** to define the **Path** of the source environment installed by **HOPEX**.
4. Click **OK**.
   The source environment **Path** is displayed.
5. In the **target environment** pane, click **Browse** to define the **Path** of the target (or customized) environment.
6. Click **OK**.
   The target environment **Path** is displayed.
7. (Optional) The environment standard administrator is selected. If you have deleted this, enter the name of your administrator and password if necessary.
8. Select data to be compared.
   ☛ *See "Comparison Types", page 180.*
9. Define comparison parameters.
   ☛ *See "Environment Comparator Configuration", page 181.*

10. Click **OK**.
    Messages indicate progress of comparison.



The XML report file is generated:

Depending on configuration, it includes a list of objects, links, characteristics and translations, indicating the differences detected. You can change results sort order simply by clicking the different columns.

MANAGING EVENTS

HOPEX provides a supervision tool (**Hopex Server Supervisor**) that enables management of events.

The following points are covered here:

- ✓ "Introduction to supervision", page 186
- ✓ "Supervision tool: HOPEX Server Supervisor", page 190
- ✓ "Supervising events", page 195
- ✓ "Events to be Monitored (Production Server)", page 200

INTRODUCTION TO SUPERVISION

The **Hopex Server Supervisor** supervision tool is used to collect messages from **HOPEX** applications (**Windows Front-End** and **Web Front-End**). These messages include indicators to ensure that **HOPEX** is operating correctly.

```
Example: information or error messages.
```

A message corresponds to a supervision event, which has been previously coded in the executable. A client cannot create a new event.

The following points are covered here:

- "Prerequisites to Supervision", page 186
- "Supervising Events", page 186
- "Supervision files", page 187
- "Supervision configuration file: MegaSite.ini", page 187

Prerequisites to Supervision

Before starting supervision function, you must check the following points:

- MegaSSP web application is up and running.
  MegaSSP web application collects all of the messages.
- SSP service is started.
  The SSP stores messages in the supervision log file.
- The Megasite.ini file is configured.

☛ *See "Supervision configuration file: MegaSite.ini", page 187.*

Supervising Events

A supervision event can include about fifty pieces of information.

```
Example: "Event infos" information is a json (or a text)
that can provide details regarding the event context. The
json structure depends on each event.
```

To consult and analyze a supervision event, see "Consulting a supervision event file", page 197.

Event types

Events are sorted by type

- A (Action): user action
- W (Warning): alert
- E (Error): erreur
- S (Snapshot): process snapshot

A type A event is characterized by:

- a start, which corresponds to its creation.
- an end, which corresponds to the moment the message is actually sent. Only one message is sent at the end. It summarizes the indicators of the process for the event duration.

Type W and type E events are immediate.

Type S events summarize the indicators of the process since the last sent snapshot.

Supervision files

Supervision files include supervision events.

Each supervision files represents a day.

Supervision file name format: sspsprvs<MM-DD-YY.txt.

```
Example: SSPSPRVS02-10-17.TXT
(supervision file for the 10th of February 2017)
```

To find the supervision file location, see "Finding the supervision file location", page 193.

Supervision configuration file: MegaSite.ini

Configuration of some of the supervision behaviors is performed in the MegaSite.ini file (<**HOPEX** installation directory>\Cfg) in [Supervision] section.

In addition, Supervision components need to communicate with the SSP and access the [SSP] section content.

```
[Supervision]
StateInterval=<time interval>
Filter=<Filter>

[SSP]
url=<SSP url>
```

MegaSite.ini	Description
[Supervision]	Supervision parameter section
StateInterval	Time interval (in millisecond) between two supervision events of snapshot type. Minimum value: 1000 By default this parameter value corresponds to 3' (180000) (do not modify this parameter)
Filter	Enables definition of executables to be supervised. Other executable supervision is deactivated. If not specified, there is no filtering and all of **HOPEX** executables are supervised. Example: `Filter=AM` Displays only **HOPEX Windows Front-End** (code A) and its Administration (code M) events. See "Executable code", page 188. To modify the filter, see "Modifying processes to be supervised (MegaSite.ini filter)", page 192
[SSP]	
Url	SSP url If the url is not specified, supervision is deactivated.

Executable code

Each application is associated with a code:

Windows Front-End :

- A: Administration
  (mgwmapp.exe /DesktopAppGbm.Administration)
- M: HOPEX
  (mgwmapp.exe)
- N: Automation
  (API mgwmapp.exe/Automation)

Web Front-End :

- R: Session holder
  (mgwspro.exe)
- T: site
  (mgwmapp.exe)
- O: SSP environment holder
  (mgwspro.exe)
- I: SSP site
  (mgwmapp.exe)

SUPERVISION TOOL: HOPEX SERVER SUPERVISOR

The **Hopex Server Supervisor** tool enables reading supervision files.

The **Hopex Server Supervisor** tool gives access to calculated/filtered views of supervision events.

You can activate/deactivate the supervision of certain processes to filter messages. Only selected executables send messages.

☛ *See "Supervision configuration file: MegaSite.ini", page 187.*

See:

- "Starting HOPEX Server Supervisor", page 190
- "Extend HOPEX Server Supervisor functionalities", page 191
- "Modifying processes to be supervised (MegaSite.ini filter)", page 192
- "Finding the supervision file location", page 193
- "Modifying the supervision file location", page 194

Starting HOPEX Server Supervisor

To start the **Hopex Server Supervisor** tool:

1. In the **HOPEX** installation folder, expand the **Utilities** folder, then **HOPEX Server Supervisor**.
2. Right-click **Hopex Server Supervisor** and select **Execute as administrator**.

   The **Hopex Server Supervisor** icon appears in the system tray of your workstation.

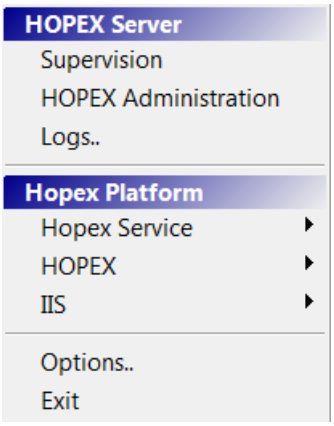
   ☛ *The green button on the icon indicates that the SSP is ready and that IIS is started, else the button is red:* .

Extend HOPEX Server Supervisor functionalities

By default, at **HOPEX** installation, **Hopex Server Supervisor** menus are minimum.

To extend access to **Hopex Server Supervisor** functionalities:

1. In your workstation system tray, right-click **Hopex Server Supervisor** .

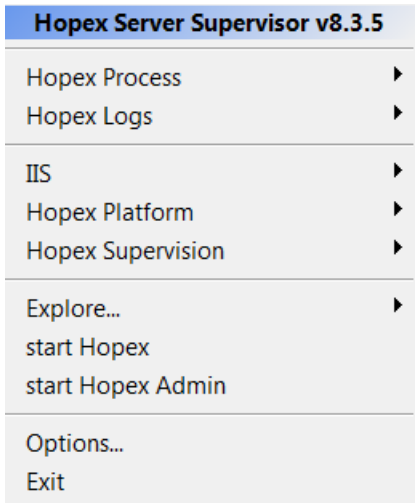


2. Click HOPEX Server .
   The **Command** field is displayed.

3. Enter "swl 1" and press "Enter".

All of **Hopex Server Supervisor** functionalities are available.

☛ *To come back to a minimum display, perform step 2, enter "swm" and press "Enter".*



Modifying processes to be supervised (MegaSite.ini filter)

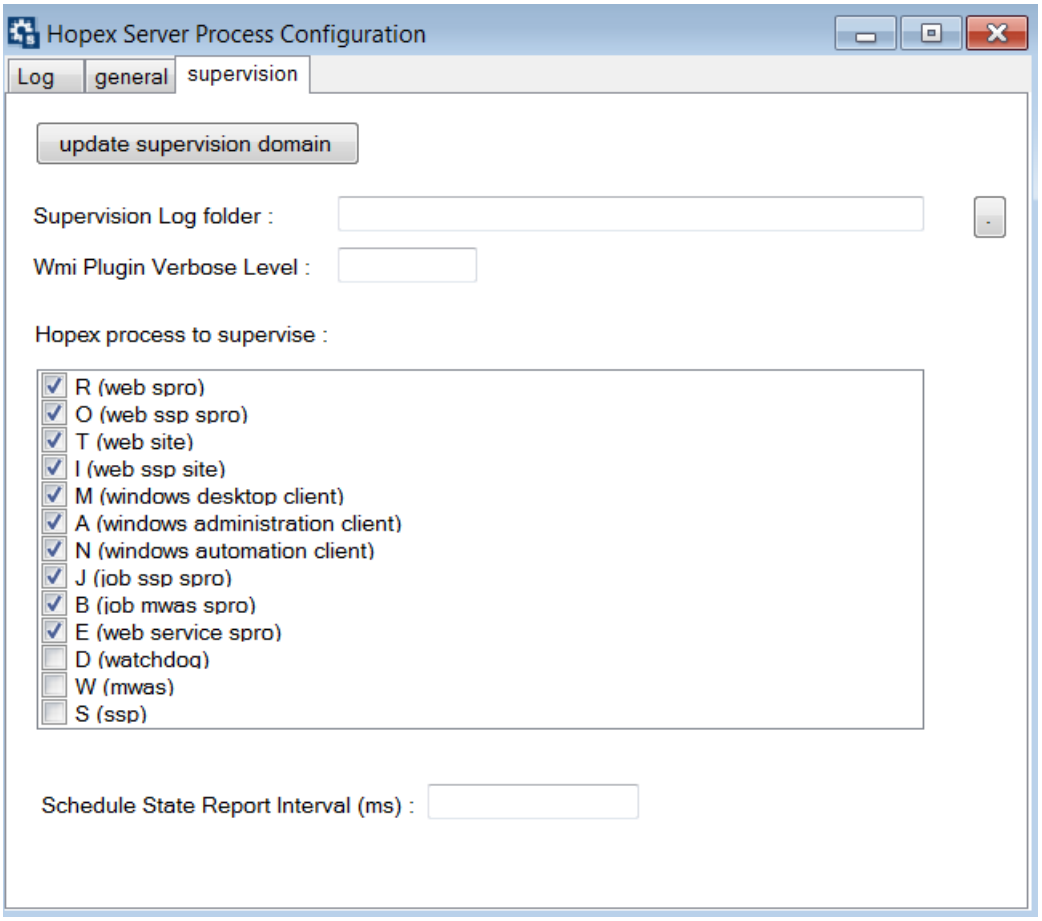
Hopex Server Supervisor enables to configure the **Filter** section of MegaSite.ini file:

☛ *For information on MegaSite.ini file, see "Supervision configuration file: MegaSite.ini", page 187*

To configure the **Filter** section of MegaSite.ini file:

1. In your workstation system tray, right-click **Hopex Server Supervisor** and select **Hopex Supervision > Supervision configuration**.

2. From **HOPEX Server Process Configuration** window, **Supervision** tab, select the processes you want to supervise.



Your modifications are immediately taken into account in MegaSite.ini file.

Finding the supervision file location

To open the folder where the supervision files are stored:

1. Open **Hopex Server Supervisor** in extended configuration.

   ☛ *See "Extend HOPEX Server Supervisor functionalities", page 191.*

2. From **Hopex Server Supervisor**, select **Hopex Logs > Daily Logs Manager**.
3. Right-click the daily log line (sspsprvs<mm-dd-yy>.txt) and select **Open Folder**.

   ☛ *To modify the supervision file location, see "Modifying the supervision file location", page 194.*

Modifying the supervision file location

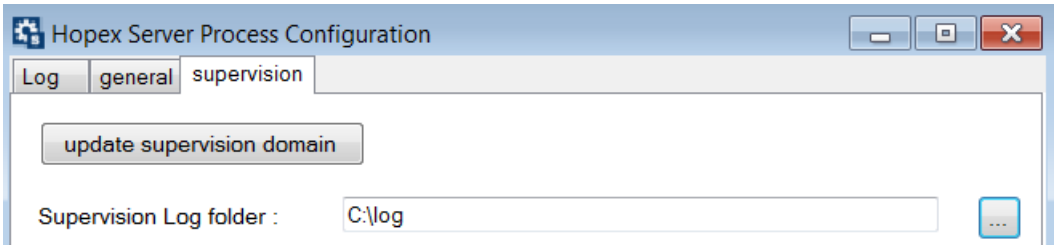
To modify the supervision file location:

1. Create the directory where you want the supervision files to be stored.

   `Example: c:\log`

2. From **Hopex Server Supervisor**, select **Hopex Supervision > Supervision configuration**.
3. Select the **Supervision** tab.
4. In the **Supervision Log folder** field, use the browse button ... to define the directory path where the supervision files are stored.

   `Example: c:\log`



These modifications are immediately taken into account. Supervision files are stored in the specified directory (e.g.: c:\log).

SUPERVISING EVENTS

Event supervision is performed from the **Supervision** tool of **Hopex Server Supervisor**.

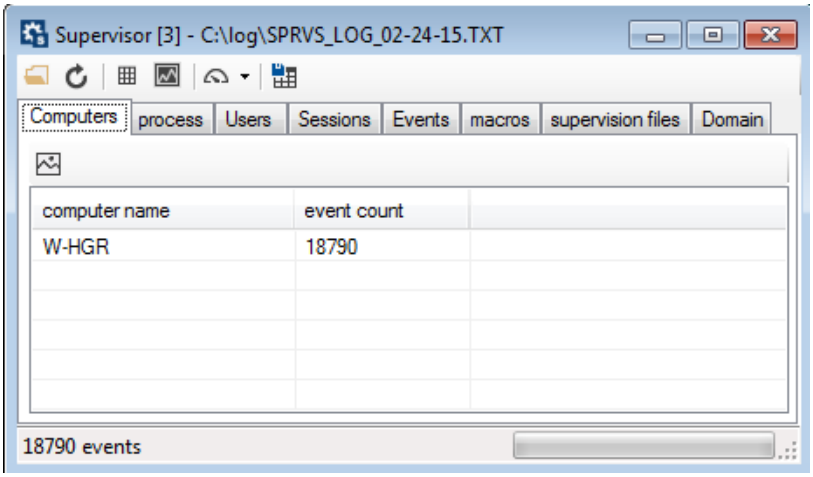
See:

- "Supervision tool", page 195
- "Consulting a supervision event file", page 197
- "Actions from an event supervision window", page 199

Supervision tool

The **Hopex Server Supervisor Supervision** tool opens and analyzes **HOPEX** event files.

☛ *To launch* **Supervision** *tool, see "Consulting a supervision event file", page 197.*



For detailed information on event files, see:

- "Supervising Events", page 186
- "Supervision files", page 187.

Supervision tool toolbar

From the **Supervision** tool toolbar, click:

- to open one or several specific supervision files
- to refresh calculated view data of the current supervision file
- to open the set of snapshots that have been created on the different servers.
  A consolidated snapshot gives an application calculated view over the last three minutes.
- to view the load state, object consumption on the set of supervised processes.
- to load a supervision domain (reference domain - Standard) so that comparisons can be made.
- export data in CSV format.

Supervision tool tabs

In the **Supervision** tool, events are grouped by calculated view that gives access to the corresponding list of prefiltered events. Views correspond to the following tabs:

- **Computers**
  This tab shows the list of servers used and its associated event number.
- **process**
  This tab shows the set of supervised processes of the set of servers.
- **Users**
  This tab shows the list of users who logged on the application.
- **Sessions**
  This tab shows the list of current or past sessions of all the servers or workstations supervised.
- **Events**
  This tab shows the list of HOPEX events that have been recorded.
- **macros**
  This tab enables to view macros, for which execution time is high.
- **supervision file**
  This tab shows analyzed supervision files.
- **Domain**
  This tab enables to show activity synthesis according to the supervision domain

Consulting a supervision event file

To consult an event of a supervision file:

1. From your workstation system tray, right-click **Hopex Server Supervisor** and select **Supervision**.

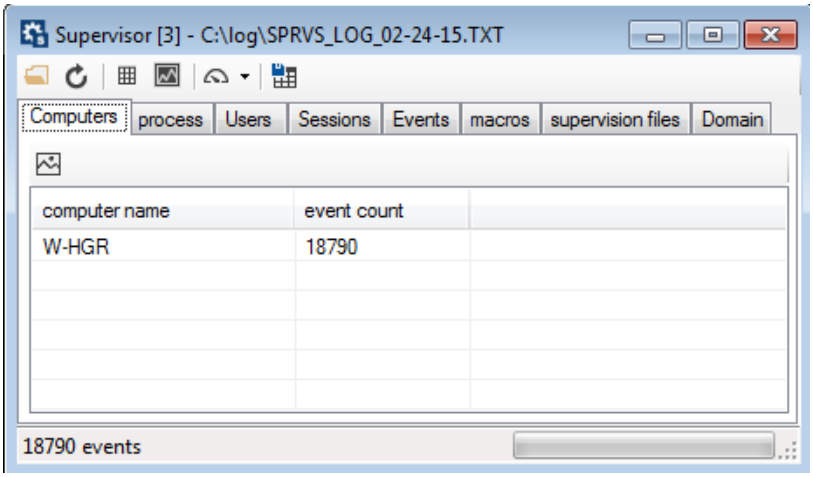
   ☛ *If **Hopex Server Supervisor** is in extended configuration (see "Extend HOPEX Server Supervisor functionalities", page 191), select **Hopex Supervision > Supervision**.*

   The current supervision file opens.

   ☛ *To open another (or several) supervision file , if **Hopex Server Supervisor** is in extended configuration select **Hopex Supervision > Supervision from file** and select the files.*

   *Alternatively, from **Supervision chart** toolbar, click **Open Supervision data file** and select the files.*

   See "Finding the supervision file location", page 193.



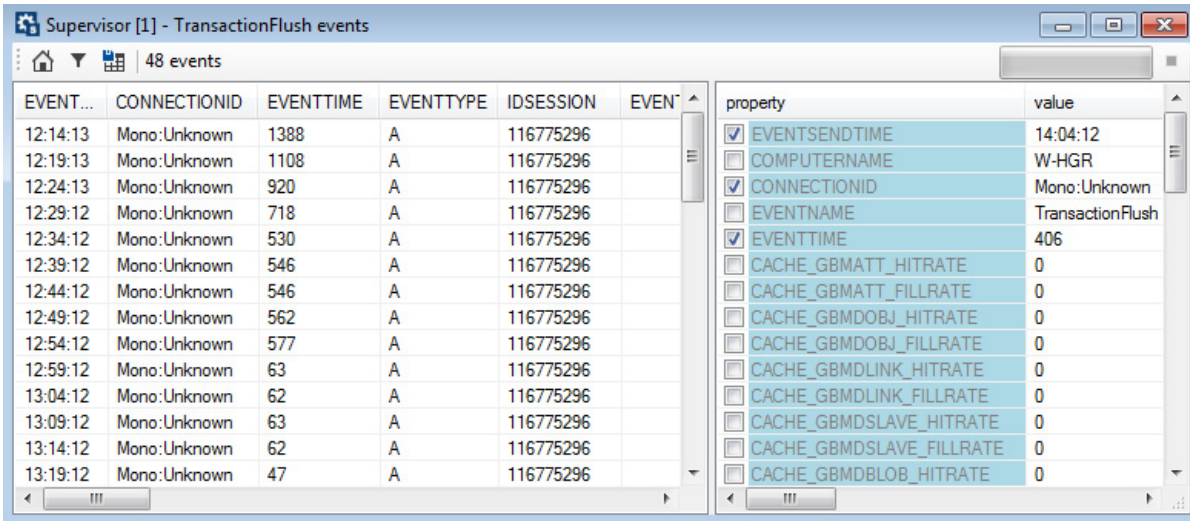
2. (optional) If you are on the current supervision file, click **Refresh** anytime you want to immediately refresh the view data.

   ☛ *See "Supervision tool tabs", page 196.*

3. Click the tab regarding the view you want to consult:

   ☛ *See "Supervision tool tabs", page 196.*

4. Double-click the line of the view you want to consult the events. Events regarding this view are displayed in table format.



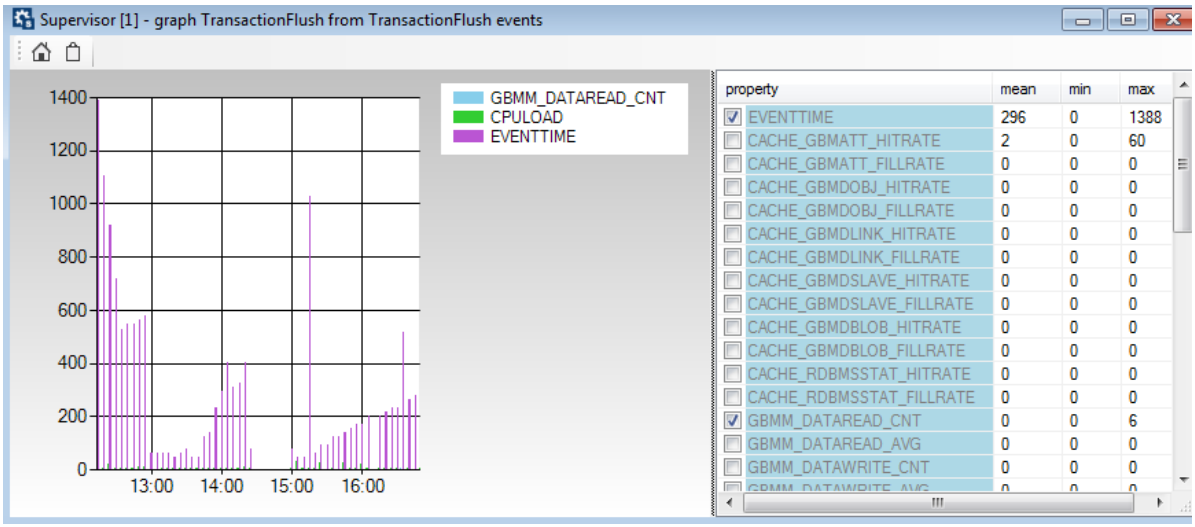
Each line represents an event of which characteristics are detailed.

☛ *See "Actions from an event supervision window", page 199.*

5. (optional) In the **property** pane, select the indicators (columns) you want to be displayed in a table column.
6. (optional) Right-click the event for which you want to display a graph and select **show graph for <event name>**.

☛ *You cannot get a graph for an Error type or Warning type event.*

The graph is displayed.



7. In the **property** pane, select the indicators you want to be displayed in the graph.
   The graph is calculated on the set of the prefiltered original view, taking into account the event selected.
8. (optional) Click **copy graph image to clipboard** to copy the graph image in the clipboard.

Actions from an event supervision window

From the supervision window, which lists the events in table format, you can:

- access the window of the source **Supervision chart** tool.

  Click **Supervisor Home** .
- create filters to filter displayed events.

  Click **Filters** .
- export current events in .txt format.

  ☛ *Example: to import events in Excel.*

  Click **Export the current filter view as supervision format** .
- interrupt data downloading.

  ☛ *Example: when experiencing long downloading times.*

  Click **Interrupt loading** .

EVENTS TO BE MONITORED (PRODUCTION SERVER)

To keep your HOPEX production server in the best operating conditions, here are the events you have to monitor:

- "Events: Login and Authentication", page 200
- "Events: Configuration Management", page 201
- "Events: Workspace Activity", page 202
- "Events: Repository Connections", page 202
- "Events: Service Execution", page 204

In the following tables, each event **Type** column indicates the importance level of the event as follows:

- severity ++: requires an immediate action
- severity ++: requires a rapid analysis
- severity -: requires your attention
- severity --: informative

Events: Login and Authentication

Events you have to monitor in a login and authentication context are listed in the following table:

Event	Type (severity)	Context / JSON	Description / Content
LoginPasswordChangeFailure	Alert (+)	***Password update***	***Error during a password update***
		"Login Name":	Name of the login
LoginAuthenticationFailure	Alert (-)	***Authentication***	***Failure in the authentication process***
		"Login Name":	Name of the login
		"Failure reason":	Error message
LicenseLoginFailure	Alert (+)	***License***	***Error while obtaining a license token***
		"Failure reason":	Error message

Events: Configuration Management

Events you have to monitor in a configuration management context are listed in the following table:

Event	Type (severity)	Context / JSON	Description / Content
GraphCompile	Action (--)	***Writing access diagram compilation***	***Should never happen on a production server*** Except after the update of a new writing access hierarchy definition
MetaDataCompile	Action (--)	***Environment compilation***	***Should never happen on a production server*** Except after the update of new customization
CompiledDataReset	Alert (+)	***Meta or technical data modification***	***Should never happen on a production server*** If this event is raised on a server, it means that a person is carrying out MetaModel extensions directly on the server
		`"User":`	`User responsible for the update`
		`"CompiledDataUpdated":`	`Modified object`
		`"CompiledDataResetOrigin":`	`Executed command.` This information is written during the upgrade or import process
GraphUpdate	Alert (+)	***Writing access diagram modification***	***Should never happen on a production server*** Except after the update of a new writing access hierarchy definition

		`"GraphUpdated":`	`"Authorization"`
			if the writing access diagram has been updated
			`"Visibility"`
			if the reading access diagram has beenupdated

Events: Workspace Activity

Events you have to monitor in a workspace activity context are listed in the following table:

Event	Type (severity)	Context / JSON	Description / Content
TransactionFlushFailure	Alert (++)	***Workspace flush***	***Unable to validate data update***
		"Failure reason":	Error message
TransactionDispatchFailure	Alert (++)	***Workspace publication***	***Failure in the authentication process***
		"Failure reason":	Error message

Events: Repository Connections

Events you have to monitor in a repository connection context are listed in the following table:

Event	Type (severity)	Context / JSON	Description / Content
RepositorySessionConnect	Action (--)	***Connection (session)***	***This event enables to monitor the connection times***
		"GBMSESSIONOPEN-MODE":	One of the following values: GBMSESSION_OPENMODE_READ-WRITE GBMSESSION_OPENMODE_READON-LY_PHYSICAL GBMSESSION_OPENMODE_READON-LY_LOGICAL GBMSESSION_OPENMODE_READON-LY_REALTIME GBMSESSION_OPENMODE_READ-WRITE_REALTIME
		"Login":	IdAbs of the login
		"User":	IdAbs of the user
		"Profile":	IdAbs of the profile
		"Role":	IdAbs of the business role
		"pathDB":	Path of the SystemDb reposi-tory of the Environment

Event	Type (severity)	Context / JSON	Description / Content
RepositorySessionConnectFailure	Alert (+)	**Connection**	**Error during the session connection phase**
		"GBMSESSIONOPENMODE":	One of the following values: GBMSESSION_OPENMODE_READWRITE GBMSESSION_OPENMODE_READONLY_PHYSICAL GBMSESSION_OPENMODE_READONLY_LOGICAL GBMSESSION_OPENMODE_READONLY_REALTIME GBMSESSION_OPENMODE_READWRITE_REALTIME
		"Login":	IdAbs of the login
		"User":	IdAbs of the user
		"Profile":	IdAbs of the profile
		"Role":	IdAbs of the business role
		"pathDB":	Path of the SysDb repository of the Environment
		"Failure reason":	Error message
RepositoryOpenFailure	Alert (+)	**Repository opening phase**	**Error detected when opening a repository**
		"RepositoryName":	Name of the repository
		"Failure reason":	Error message
RepositoryCloseFailure	Alert (-)	**Repository closing phase**	**Error detected when closing a repository**
		"RepositoryName":	Name of the repository
		"Failure reason":	Error message
RepositorySessionReconnect	Alert (++)	**The connection with the database server has been lost**	**HOPEX is trying to reconnect automatically to the server, each try generates an alert (5 by default)**
		"RepositoryServerName":	Name of the repository

Events: Service Execution

Events you have to monitor in a service execution context are listed in the following table:

Event	Type (severity)	Context / JSON	Description / Content
ScheduledJobExecutionFailure	Alert (+)	**Job execution**	**Error when running a scheduled job**
		`"Failure reason":`	`Error message`
SchedulerError	Error (+)	**Scheduler service start**	**Unable to run the scheduler service due to a lack of license or other reason (see msg)**
		`"Msg":`	`Error message`
ProcessException	Error (++)	**Important exception occurred**	**Important error occurred when running HOPEX**
ERQLExcessiveInvocationTime	Alert (-)	**An ERQL query executing more than 5000 GBM commands**	**Optimizing the query manually**
		`"QueryId": or "QueryName":`	Query identifier
		`"GBM Get count":`	Number of readings generated by the query
		`"GBM Get count":`	Number of searches generated by the query

MANAGING OBJECTS

The following points are covered here:

- ✓ "Exporting HOPEX Objects", page 206
- ✓ "Protecting Objects", page 216 (function available with **HOPEX Power Supervisor**)
- ✓ "Comparing and Aligning Objects Between Repositories", page 219 (function available with **HOPEX Power Supervisor)** or **HOPEX Collaboration Manager**)
- ✓ "Merging Two Objects", page 225 (function available with **HOPEX Power Supervisor**)
- ✓ "Querying Isolated Objects", page 228 (function available with **HOPEX Power Supervisor**)
- ✓ "Querying Objects to be Translated", page 230
- ✓ "Importing a Solution Pack in HOPEX", page 232
- ✓ "Managing UI Access (Permissions)", page 233 (function available with **HOPEX Power Supervisor**)
- ✓ "Managing Shapes", page 248

EXPORTING HOPEX OBJECTS

Export of an object with propagation enables creation of a consistent set allowing transfer of part of the repository to another repository. For example, export of **HOPEX** objects from a library includes objects present in the library and their dependent objects.

The following points are detailed here:

- "Export", page 206
- "Exporting Objects", page 206
- "Viewing Objects Before Export", page 213

Export

You can export common modeling objects as well as configuration and parameterization objects.

```
Examples: report templates (MS Word), queries, metamodel
extensions, users.
```

To access these objects, you must select the extended metamodel options in your configuration. This option is available in user options, from the **Repository** icon.

☛ *See "Managing Options", page 323.*

Export uses the propagation mechanism, which can be configured using perimeters.

☛ *For detailed information on propagation mechanism, see the **HOPEX Power Studio - Perimeters** technical article.*

Propagation steps in organizational process (major) export:

1. For an organizational process (major) you also export its operations (minor).
2. For an operation you export the event messages or result messages (minor) of the operation (major)
3. Propagation continues step by step until all links have been explored.

   ☛ *Export takes account of link types: for certain link types, search in depth of other links stops.*

   All keywords are also exported.

Exporting Objects

You can export **HOPEX** objects from:

- **HOPEX Administration**
- **HOPEX**

☛ *To export objects from the **Administration** desktop (Web Front-End) see the HOPEX Administration - Supervisor Web guide.*

You can export objects in the following formats:

- **text**
  The exported file is in the form of an .MGR file.

  ☛ *For more details on .MGR file syntax, see "Command File Syntax", page 307.*

- **XML MEGA**
  The exported file is in the form of an .XMG file containing commands or data (objects and links).

  ☛ *For more details on MEGA XML data exchange format, see technical article "MEGA Data Exchange XML Format 70".*

- **Excel**

  ☛ *See the **HOPEX Common Features** guide, "Exchanging Data With Excel" chapter.*

You can password-protect the export file generated. The exported file has the .mgz format and can only be imported by entering the password you defined.

See:

- "Exporting HOPEX objects", page 207
- "Exporting a HOPEX object from the object", page 210

Exporting HOPEX objects

You can export objects from **HOPEX Administration** or from your **HOPEX** desktop.

To export **HOPEX** objects:

1. Access the repository from which you want to export objects.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Object Management > Export Objects**.

   ☛ *Alternatively, from your **HOPEX** desktop, select **File > Export > MEGA objects**.*

   The **Export HOPEX Objects** dialog box opens.

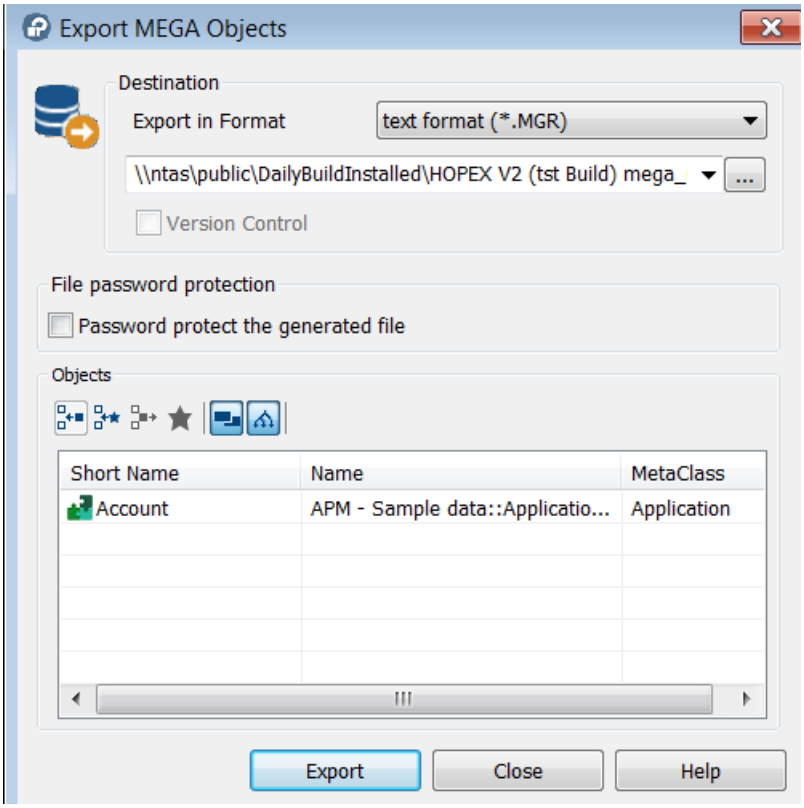
3. In the **Export in Format** field, select the export file format.
   Several options enable object export configuration.

   ☛ *See "Managing Options", page 323.*

4. (Optional) Enter a different name and folder if the default values are not suitable.

   ☛ *Button ... enables you to browse the folder tree and select where the export file will be located.*

5. (If needed) In the **File password protection** pane, select "Password protect the generated file".



6. In the **Objects** frame, click **Add Objects to List** .

   ☺ *To simplify the query, click* **Add objects to list from favorites** .

   The query dialog box appears.
7. Start the query and select the appropriate objects in the result window.
8. Click **OK**.

   The selected objects are added to the **Export HOPEX Objects** dialog box list, preceded by their type.

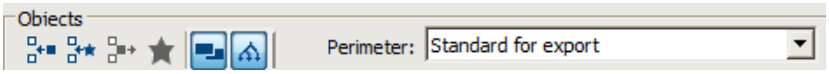
   You can carry out this procedure several times, allowing you for example to export objects of different types.

   ☛ *In the event of an error, click* ***Remove objects from list*** *to delete an object from the list.*

9. In the **Objects** group box, by default two export configuration options are proposed:
   - **Include Objects of Merging**, which allows you to export technical objects resulting from merging objects (_TransferredObject).

     ☛ *For further information on merging objects, see "Merging Two Objects", page 225.*
   - **Propagate**, which allows you to export listed objects together with their dependent objects.

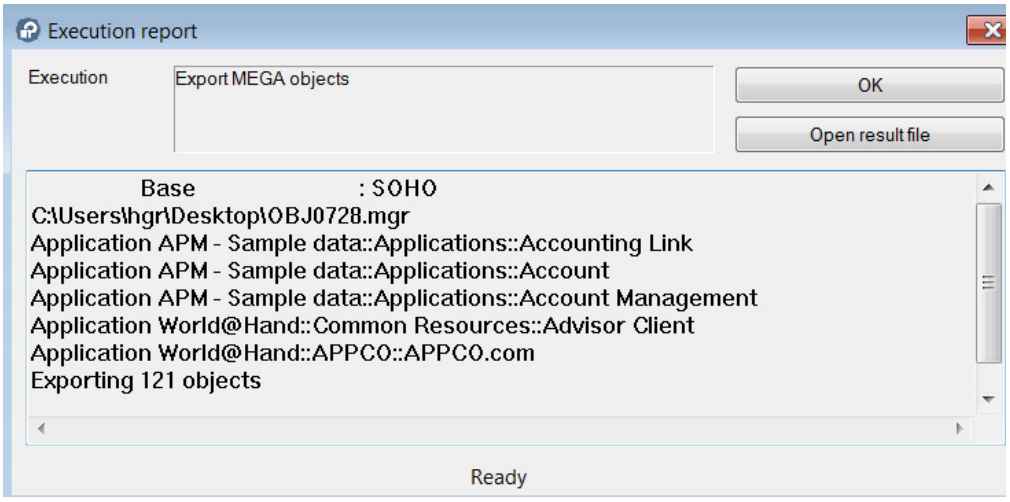
     ☛ *To view objects before export, see "Viewing Objects Before Export", page 213*
10. (Optional) By default, the export perimeter is as defined in the properties of the Export tool. To modify the export default perimeter, you must have previously activated export perimeter selection, see "Activating the export perimeter selection option", page 212.
    In the **Objects** frame, select the **Perimeter** of export using the drop-down menu.



11. When selection is complete, click **Export**.
12. (If step 5 you selected the password protect file option) In the **File protection** dialog box, enter a password, confirm it and click **OK**.
    The export process begins.

    ☛ *To interrupt export during execution, click **Cancel**.*

    During export, the name and type of the objects being exported appear at the bottom of the dialog box, together with duration of export. On completion of export, the **Execution Report** displays the number of exported objects.



☛ *See "Viewing an export file", page 210.*

Exporting a HOPEX object from the object

You can export an object from your **HOPEX** desktop.

To export a **HOPEX** object from the **HOPEX** desktop:

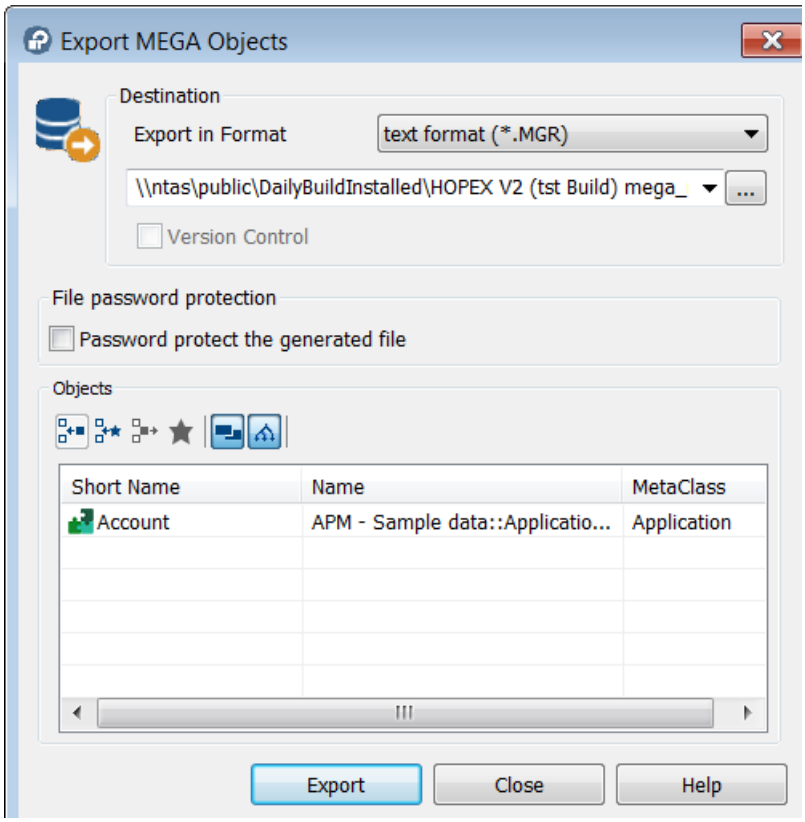
1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. In the **Main Objects** navigation window, right-click the object you want to export and select **Manage > Export**.

   ☛ *You can also export **HOPEX** objects by selecting **File > Export > HOPEX Objects** in your workspace.*

   The **Export HOPEX Objects** dialog box opens.



   In the **Objects** frame, the object to be exported is already selected.

3. Refer to the procedure "Exporting HOPEX objects", page 207.

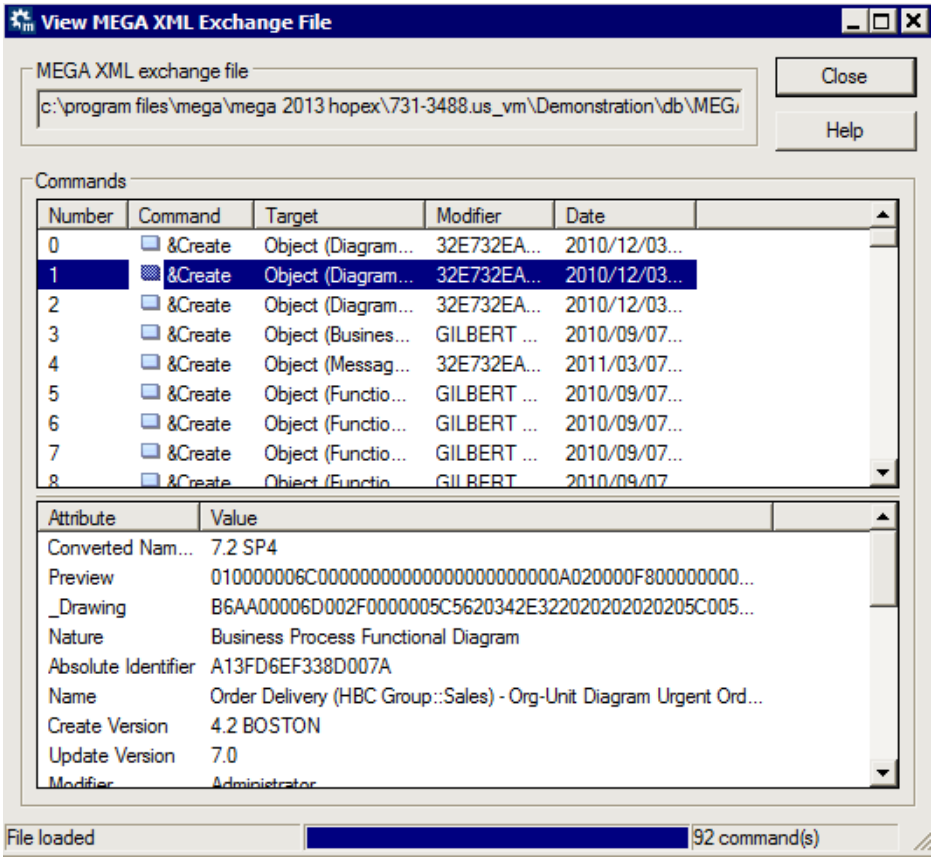
Viewing an export file

To view the exported file:

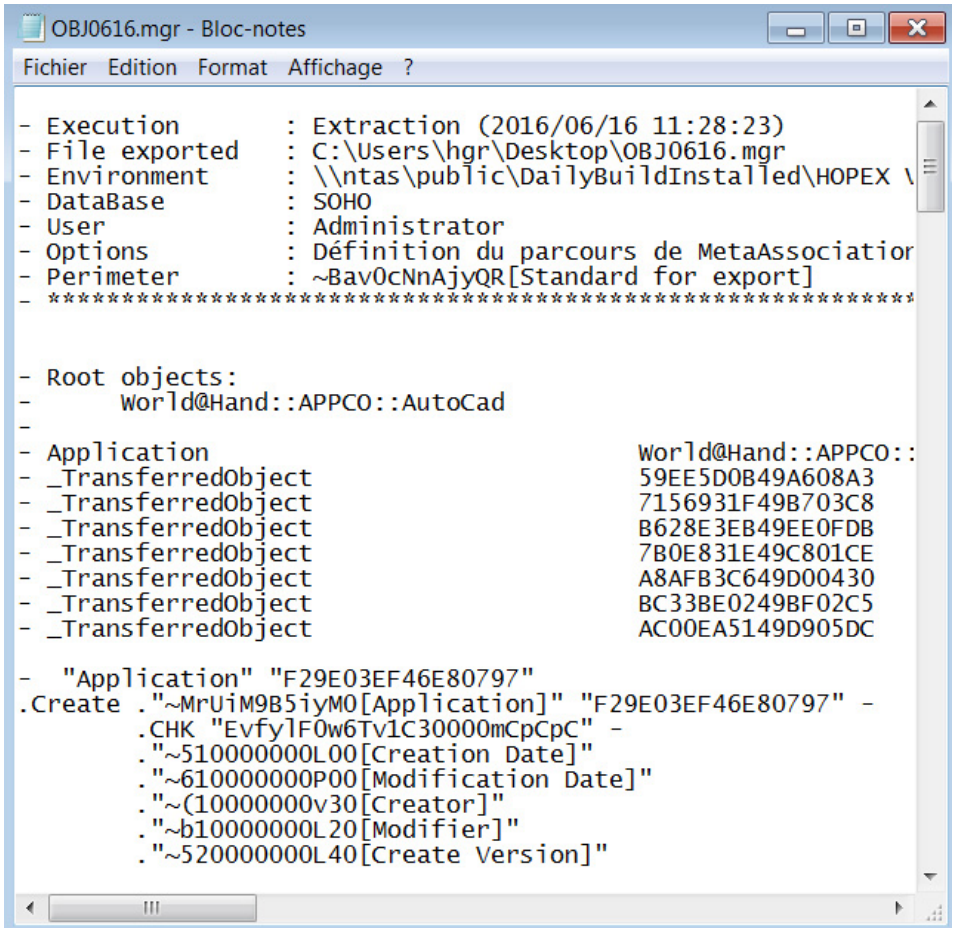
1. Export objects.

   ☛ *See "Exporting Objects", page 206.*

2. In the export **Execution Report** dialog box, click **Open result file**
   In the case of:
   - an export in **MEGA** XML format, the view MEGA XML exchange file dialog box appears.



   This file is presented in the form of a table showing a list of commands. Corresponding to each command ("Create", "Connect", etc.) there is a an object, object modification date and name of the last modifier.
   - export in text format, the Notepad dialog box appears:



This file lists all objects in text format.

The exported file can then be imported into another repository.

☛ *For more details on updating a repository by command file import, see* *"Updating a Repository", page 103*.

Activating the export perimeter selection option

At the time of export, to be able to select export perimeter, you must activate the **Activate export perimeter selection** option:

1. From your **HOPEX** workspace, select the **Tools > Options** menu.
2. In the HOPEX **Data Exchange > Export > Files** option group:**Generic Options**, select the **Activate export perimeter selection** option.

Viewing Objects Before Export

☛ *Viewing of objects is available with the **HOPEX Power Supervisor** technical module.*

Viewing models with a perimeter allows preview of the operation result before its execution, and modification of the operation if required. You can therefore:

- see default impact of the applied perimeter, see "Viewing objects", page 213.
- modify behavior of the perimeter according to the different types of links browsed from the root object, see the **MetaStudio** guide.

In the case of object export, other objects (connected to the root object) are also exported - they are determined by behavior of the "***Standard for export***" perimeter related to links existing around the root object.

☺ *Before exporting one or several **HOPEX** objects, you may find it useful to view all objects that will be exported. These can be objects you have selected as well as those deduced by the propagation mechanism.*

☛ *For detailed information on perimeters, see the **HOPEX Power Studio** - **Perimeters** technical article.*

Enabling the view option

To view objects that will be exported, you must enable the **View objects before export** option:

1. From your **HOPEX** workspace, select **Tools > Options**.
2. In the HOPEX **Data Exchange > Export > Files** option group:**Generic Options** option group, select the **View objects before export** option.

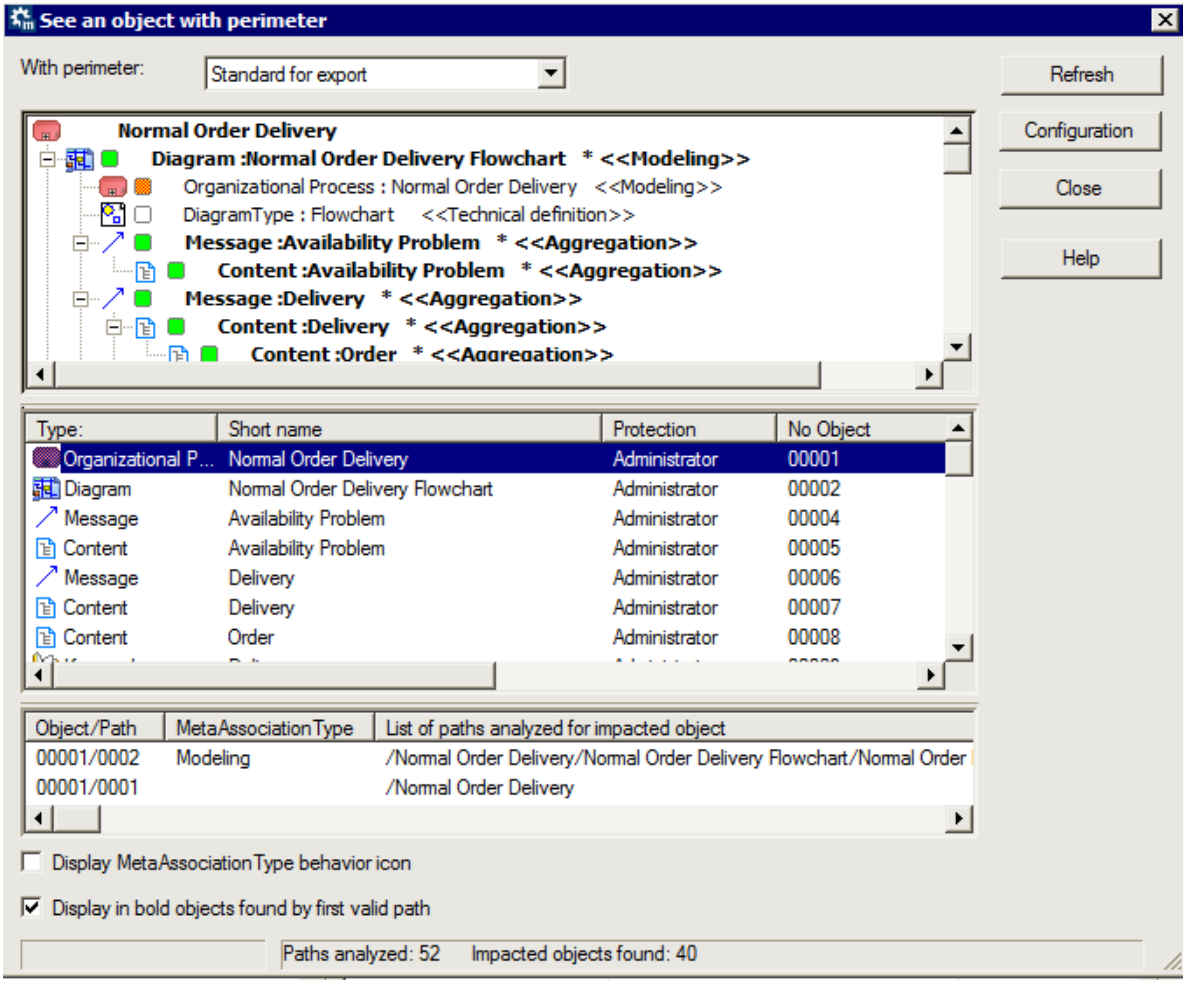
Viewing objects

To view objects that will be exported:

1. Select menu **Tools > Options**.
2. In the HOPEX **Data Exchange > Export > Files** option group:**Generic Options**, select the **View objects before export** option.

   The **View** button appears in the **Objects** frame.

3. In the **Export HOPEX Objects** dialog box, select an object in the list and click **View** .
   The object detailed view window appears.



The ***See an object with perimeter*** window presents all exported objects in two ways (result of propagation applied to the root object):

> ☛ *To view the impact of other perimeters on the object concerned, select another **Perimeter** in the drop-down list.*

- The top frame presents, in tree form, the objects that will be exported with the root object. For each object it details:
  - the propagation behavior defined by the "***Standard for export***"perimeter for the link browsed.
    The behavior of this operator depends on the various link types and will determine the objects that will also be exported.
  - the corresponding link type (for example ***Modeling***).
  - the propagation type (identified by an icon) that will be executed on the object.

Table: Description of propagation behaviors

Icon	Value	Propagation description
	Deep	***Recursive complete propagation***: Takes into account this link and the opposite object only. Propagation continues.
	Standard	***Simple propagation***: Takes into account this link and the opposite object only. Propagation stops.
	Link	***Limited propagation:*** Takes into account this link but not the opposite object. Propagation stops.
	Abort	***No propagation:*** Does not take into into account this link or the opposite object. No propagation:

You can customize display of these results by selecting:

- **Display MetaAssociationType behavior icon**, which presents propagation behavior defined for the MetaAssociationType.
- **Display in bold objects found by first valid path**.

- the middle table lists objects that will be exported with the root object. For each object it details:
  - the number of paths linked to the object.
  - the comment associated with the object.
  - the bottom table details all paths by which the object selected in the middle table has been found, together with the corresponding link type (MetaAssociationType).

To locate an object/path in the tree:

❯ From one of the tables, right-click the object/path you want to locate and select **Find in tree**.

The **Configuration** button accesses the perimeter configuration tool.

☛ *For detailed information, see how to configure a perimeter in the **HOPEX Power Studio - Perimeters** guide.*

PROTECTING OBJECTS

The object protection function is available with the **HOPEX Power Supervisor** technical module.

An object has the writing access level of the user who created it. The writing access level of an object can be modified:

- directly
- by its dependence on another object (project, process, etc.) by propagating the authorization level for that object. This dependence may be indirect.

☛ *See "Viewing Objects Before Export", page 213 for more details.*

See:

- "Accessing the Object Protection Management Window", page 216
- "Assigning a Writing Access Area to an Object", page 217
- "Propagating Authorizations Between Linked Objects", page 218

Accessing the Object Protection Management Window

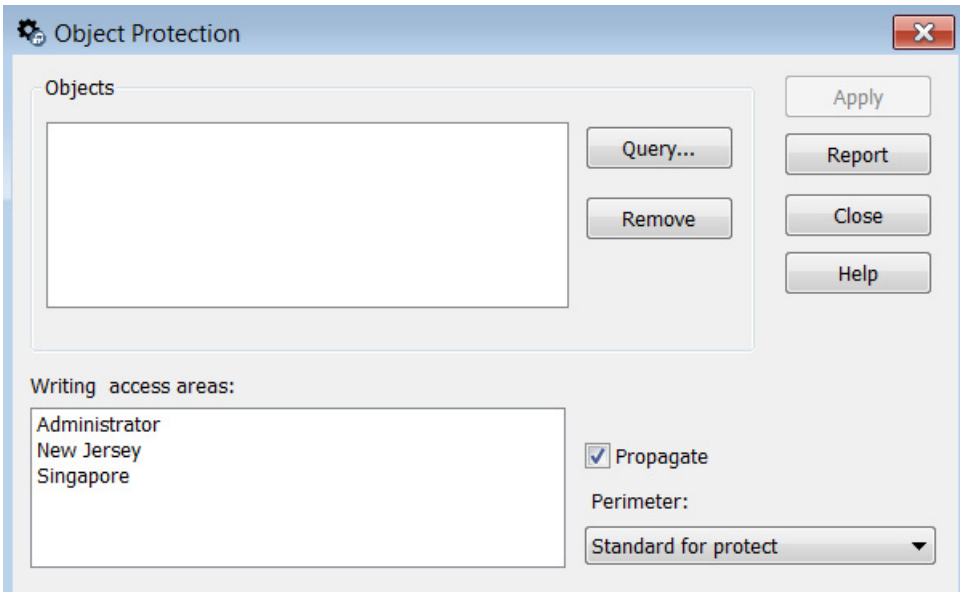
To access the object protection management window:

1. From **HOPEX Administration**, connect to the environment concerned.

☛ *See "Connecting to an Environment", page 7.*

2. Expand the **Repositories** folder.

3. Right-click **SystemDb** and select **Object Management > Protect Objects**.
   The **Protect objects** dialog box opens.



The **Writing Access Areas** list presents the writing access area of the user accessing the dialog box, and if appropriate, its dependent access areas (it is not possible to assign writing access areas higher than its own writing access area).

Writing access areas are identified by level.

When you open this dialog box, its **Objects** list is empty.

☛ *See "Assigning a Writing Access Area to an Object", page 217.*

☛ *See "Propagating Authorizations Between Linked Objects", page 218.*

Assigning a Writing Access Area to an Object

To assign a writing access area to an object:

1. Access the **Object Protection** window.

   ☛ *See "Accessing the Object Protection Management Window", page 216.*

2. In the **Writing Access Areas** list, select a writing access area.
3. Click **Query** to start a query.

4. Run the query, select the appropriate objects from the results, and click **OK**.
   The selected objects are added to the **Protect objects** dialog box list. Their types are also displayed.

   ☛ *You can carry out this procedure several times, allowing you for example to protect objects of different types.*

   ☛ *In event of error, select the unwanted object in the list and click **Delete**.*
5. Select the objects in the list.
6. When selection is complete, click **Apply**.
   The objects are assigned the selected protection level.

   ☛ *You can select some objects in the list and assign one authorization to these, then select other objects and assign a different writing access level without executing a new query.*

Propagating Authorizations Between Linked Objects

You can automatically assign writing access level of an object to objects linked directly or indirectly to it.

1. Access the **Object Protection** window.

   ☛ *See "Accessing the Object Protection Management Window", page 216.*
2. Click **Query** to select the objects concerned.
3. In the **Writing Access Areas** list, select a writing access area.
4. Select the **Propagate** check box.
5. Click **Apply**.
   Objects dependent on those for which propagation is requested are protected with the same writing access level.

   ☛ *For more details, see "Viewing Objects Before Export", page 213.*

   ☛ *Note that this type of propagation may modify the authorizations for objects shared by several projects or diagrams.*

COMPARING AND ALIGNING OBJECTS BETWEEN REPOSITORIES

> ☛ *The object compare and align feature is available with the **HOPEX Power Supervisor** or the **HOPEX Collaboration Manager** technical module.*

HOPEX enables comparison and alignment of:

- two complete repositories
- objects in different repositories
- objects of the public repository with those of the current private workspace.
- a file and a repository (or repository objects)
- two repository archived states

> ☛ *The objects compared must not be in the same private workspace.*

See:

- "Compare and Align Principle", page 219
- "Compare and Align Warnings", page 220
- "Comparing and Aligning Two Repositories", page 220

Compare and Align Principle

The principle of comparing and aligning objects between repositories is as follows:

1. ***Extraction***
   The selected objects and any linked objects are extracted from the two repositories, browsing links according to **HOPEX** principles of object extraction.

   Comparison
   The two sets of data are compared on the basis of *absolute identifiers* of the objects they contain.
2. ***Comparison result***
   A window displays the results of the comparison. You can also generate a report and a command file in this window.

   > ☛ *The page showing differences displays a maximum of 1000 lines. If the list of differences is greater than 1000 lines, a message prompts you to either ignore this limit and display all the lines (in this case, the list may take some time to load) or not.*
3. ***Alignment***
   The upgrade command file is imported in the target repository.

Compare and Align Warnings

You must be aware of the following points before alignment and selection of the user executing alignment.

Repository log

The repository log lists all modifications made in the repository. It gives users a better understanding of actions executed in a repository in private workspaces. Each time an action is executed, an occurrence of Change Item is created.

☛ *For more details on repository log, see "Managing logfiles", page 83.*

The repository log is not transferred from one repository to the other: a newlog is created in the target repository. Object history is not therefore kept.

Users

The creator/modifier of an object in the target repository is the user executing the alignment.

The date of creation of an object is the date on which alignment was executed.

Reading (confidentiality) and writing access levels

Writing and reading access levels are taken into account during the comparison and during the alignment.

☛ *For more details on writing and reading access management, see "Managing Data Reading Access", page 275 and "Managing Data Writing Access", page 249.*

To perform a comparison and an alignment, you must have reading access (if reading access management is activated) and maximum reading access for all objects in the repository.

☛ *Reject files are generated on completion of alignment. To delete files: in environment options **Options > Data Exchange > Import/ Export Synchronization > MEGA**, select the option **Delete files produced at compare/align on completion of processing**.*

Comparing and Aligning Two Repositories

☛ *Before comparing and aligning, see "Compare and Align Warnings", page 220*

To compare and align two repositories:

1. Connect to **HOPEX**.

☛ *See "Connecting to HOPEX", page 136.*

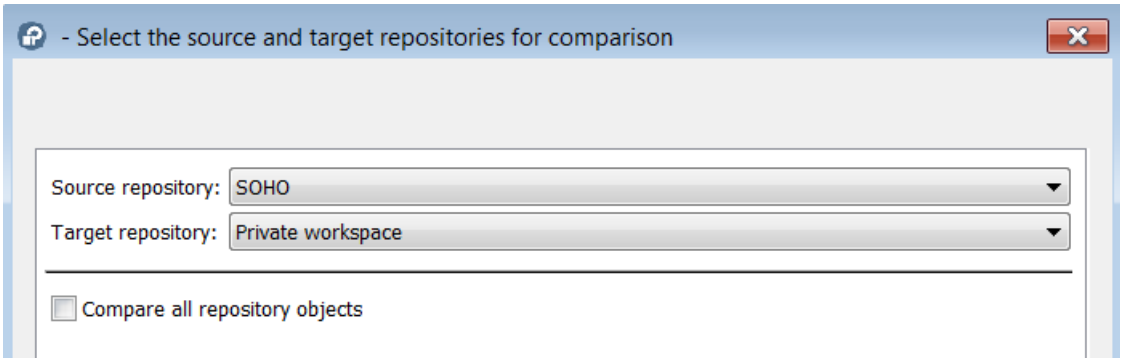
☛ *You can also, from **HOPEX Administration**, in the environment concerned, right-click the repository concerned and select **Compare and Align**.*

2. In the **HOPEX** menu bar, select **Tools > Manage > Compare and Align**.

   ☛ *You can also right-click the object and select **Manage > Compare and Align**.*

   The object comparison wizard opens.
3. Indicate if you want to compare:
   - two repositories
   - two current repository archived states (RDBMS repository only)
   - a file and a repository
4. Click **Next**.
5. Select:
   - the **Source repository**
   - the **Target repository**, which is the repository to be updated.

     ☛ *It can be a private workspace of the repository.*



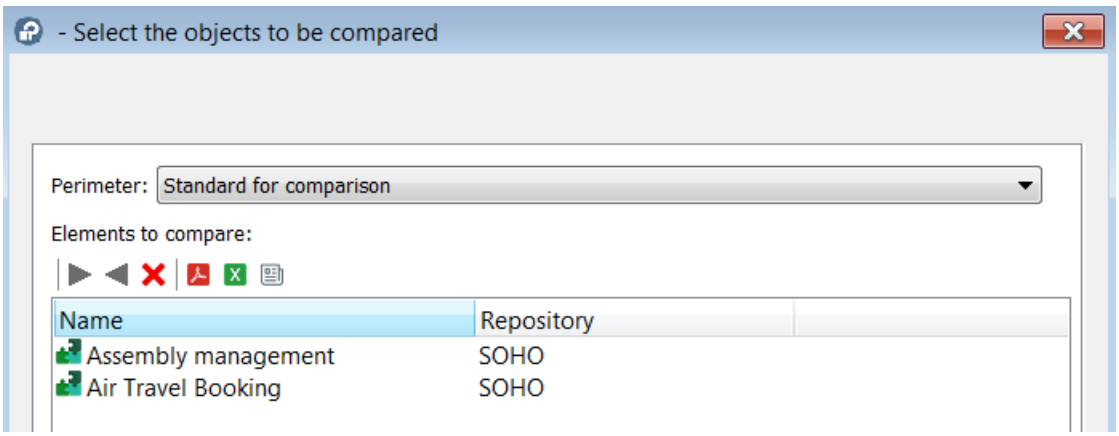
6. (Optional) If required, you can choose to **Compare all repository objects**.

   ☛ *Processing of this option can take some time.*
7. Click **Next**.
   The dialog box for selection of objects to be compared opens.
8. In the **Perimeter** field, select the perimeter type (by default **Standard for Comparison**)

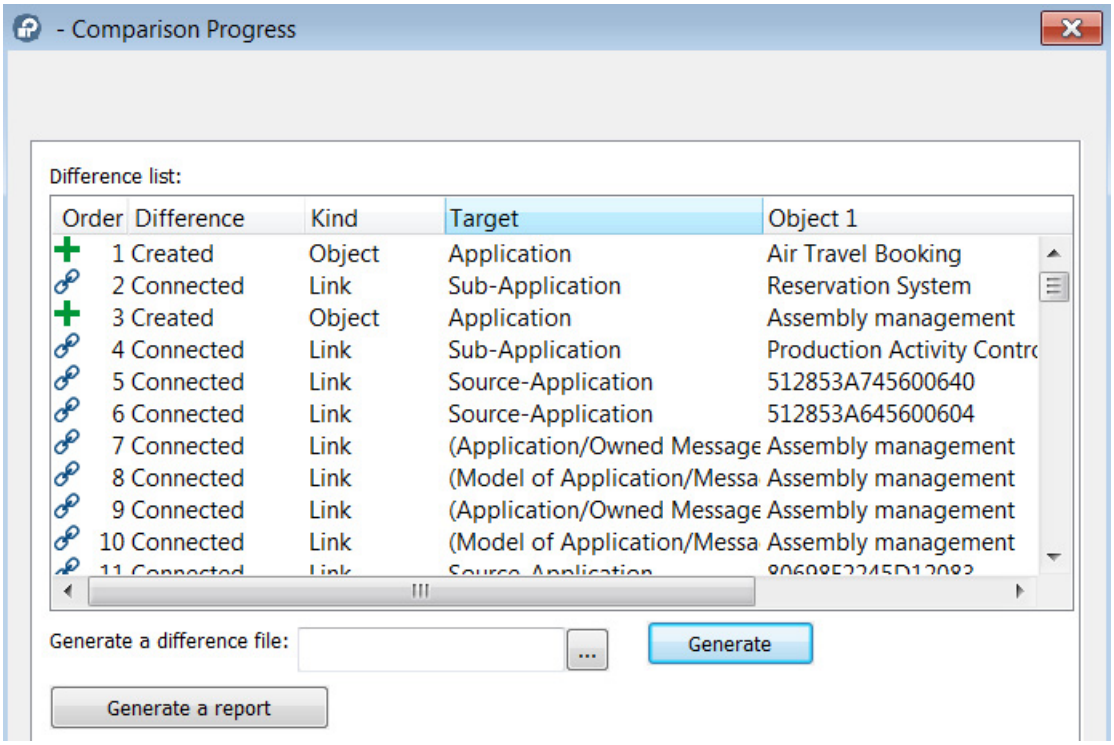
   ☛ *For detailed information on perimeters, see the **HOPEX Power Studio - Perimeters** technical article.*

9. In the **Elements to compare** pane, select:
   - ▶ to add objects from the source repository, or
   - ◀ to add objects from the target repository.

     ☛ *If you have opened the comparison wizard from an object, this object is automatically added in the list of objects to be compared.*



10. Click **Next**.
    The **Comparison Progress** window opens. It presents the differences between compared objects and their modifications.



The **Difference** column presents differences by update category:

- ***Created***: objects not existing in the target repository.
- ***Deleted***: objects existing in the target repository but not in the source repository.

  ☛ *Deletion commands of compare and align can be generated in a separate file. To do this, activate the corresponding option in **Options > Data Exchange > Import/Export Synchronization > MEGA**.*

- ***Modified***: objects of which characteristics, including name, have been modified.
- ***Connected***: links, between two objects, that do not exist in the target repository.
- ***Disconnected***: links existing in the target repository but not in the source repository.
- ***Changed***: links for which a characteristic has been modified.

The **Type** column presents differences by type.

11. In the **Generate a difference file** field (.mgr format):
    - Click ... and enter the name and location of the comparison file, then click **Save**.
    - Click **Generate** to generate the .mgr file that contains the list of differences detected.

12. (Optional) Click **Generate Report** to generate the report (.pdf format) which contains:
    - the list of differences detected
    - statistics
13. Click **Next**.

    Differences are imported in the target repository.

    The target repository is aligned with the source repository.

    ☛ *An alignment file with the content of differences (align-YYYY-MM-DD-hh-mm_555.mgr) is automatically saved in folder <Environment Name>\Db\<Repository Name>\USER\<User Code>.*

    If the alignment contains rejects, click **Display Rejects** to open the alignment rejects file (format .mgr).

    A rejects file is automatically saved in folder <Environment Name>\Db\<Repository Name>\USER\<User Code> (rejects file-reject-YYYY-MM-DD-hh-mm_555.mgr). This file is empty if alignment does not contain rejects.
14. Click **Finish**.

MERGING TWO OBJECTS

The object merge feature is available with the **HOPEX Power Supervisor** technical module.

When you merge two objects, you obtain a single object by transferring the *characteristics* and *links* from one object to the other. The source object is deleted. It is therefore recommended that merges be done in a new private workspace, so you can discard the changes if the result is not satisfactory.

You must have administration rights to merge two objects.

Choice of the objects to be merged

The **Target** object is the reference object that will be merged with the **Source** object. By default:

- its characteristics are not modified
- merging proposes addition of source object links.

The **Source** object is the object of which:

- you want to reuse certain characteristics or certain links
- characteristics and links will be transferred to the **Target** object.

When the link is to be a unique link (e.g., for sub-typing where the type is unique), the link of the target object is kept by default.

When merging is completed, the source object is deleted.

*You can **Explore** objects using the corresponding command. It can also be used to explore their links.*

Merging Two Objects

To merge two objects:

1. Connect to **HOPEX**.

   "Connecting to HOPEX", page 136.

2. In the menu bar, select **Tools > Manage > Merge Two Objects**.
   The first step of the **Merge Objects** wizard appears: **Object selection**.

   To have the right to merge two objects, you must have the right to delete objects.

3. In the **Object** pane, select the object type to be merged.

   See "Choice of the objects to be merged", page 225.

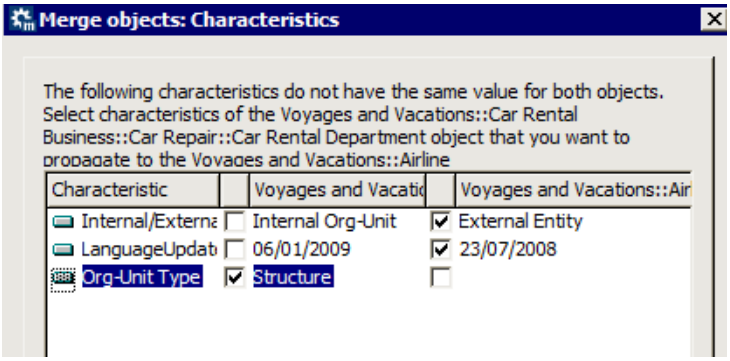
4. In the **Source** field, click the arrow and select the source object.
5. In the **Target** field, click the arrow and select the target object.

6. Click **Next**.

   ☛ *If the target and source objects are the same, the **Next** button is disabled.*

   The second step of the **Merge Objects** wizard appears: **Characteristics**. It presents the differences found in characteristic values of both objects.

7. Select the source object characteristics you want to transfer to the target object. Characteristics that remain selected in the target object are kept.

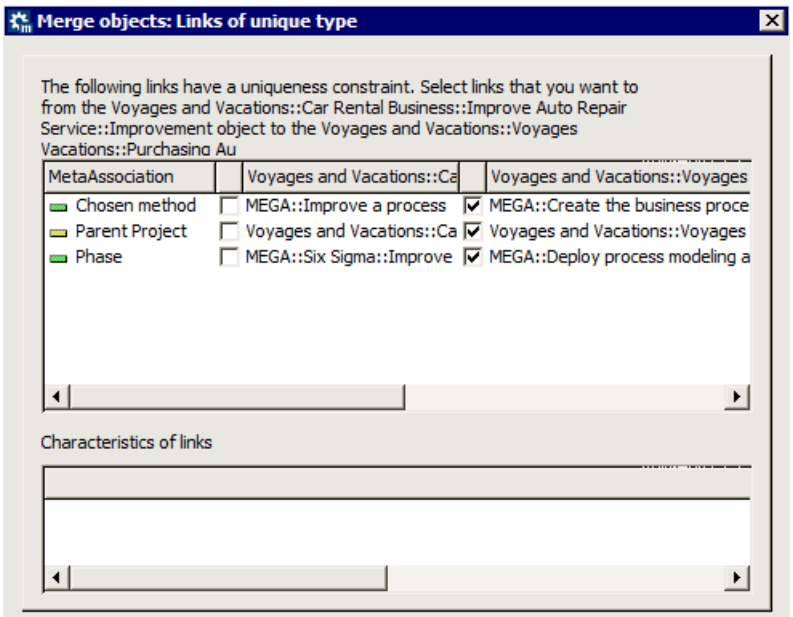


8. Click **Next**.

   The third step of the **Merge Objects** wizard appears: **Unique links** (those that can only exist once for a given object).

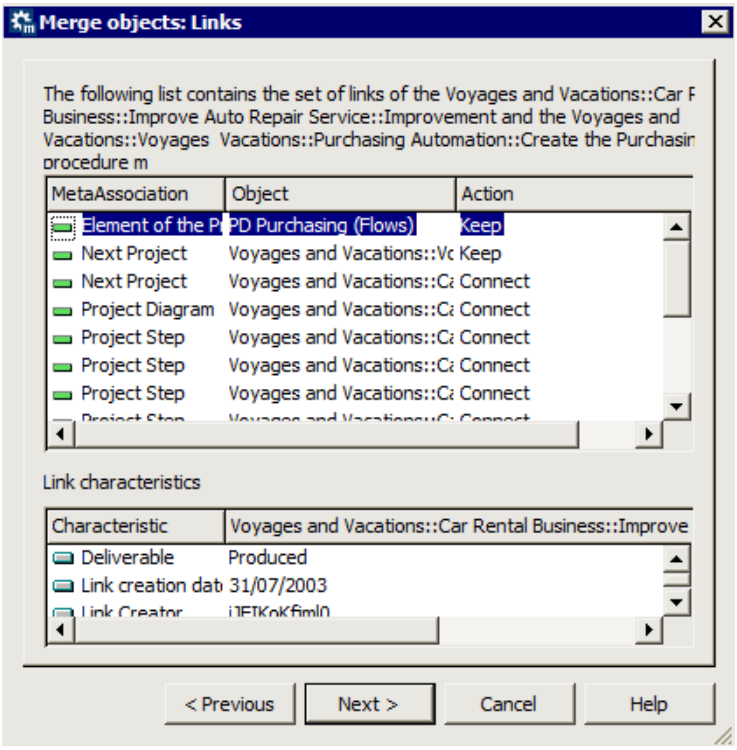
   `Example: a message can have only one super-type.`

   ☛ *This step appears only if the objects to be merged have unique links connecting them to different objects.*



9. Select the links to be transferred.

10. Click **Next**.
    The **Merge Objects** wizard presents: **Links**.



- When the link does not exist for the target object, the default is to connect the target object (**Action** : "Keep"). You can select the **Do not connect** command so as not to transfer the link.

You can **Keep** existing links, or **Disconnect** them.

- When the two objects are linked to the same object by the same link, it is possible to **Not change characteristics** of the link for the target object, or to **Copy characteristics** of the link for the source object. In this case, you can indicate for each characteristic whether to use the value of the source object, or keep the value of the target object.

11. Click **Next**.
12. Click **Finish** to start merging.
    The gauge indicates progress of the operation.

    When merging has been completed, the source object no longer exists and the selected *characteristics* and *links* have been transferred to the target object.

    ☛ *"_TransferredObject" temporary merge objects are created on this occasion. Merge objects of a repository can all be exported at export of **HOPEX** objects.*

Querying Isolated Objects

The Query isolated objects function is available with the **HOPEX Power Supervisor** technical module.

Isolated objects are objects unconnected to other objects, and are therefore almost certainly not used.

Querying isolated objects

1. From **HOPEX Administration**, connect to the environment concerned.

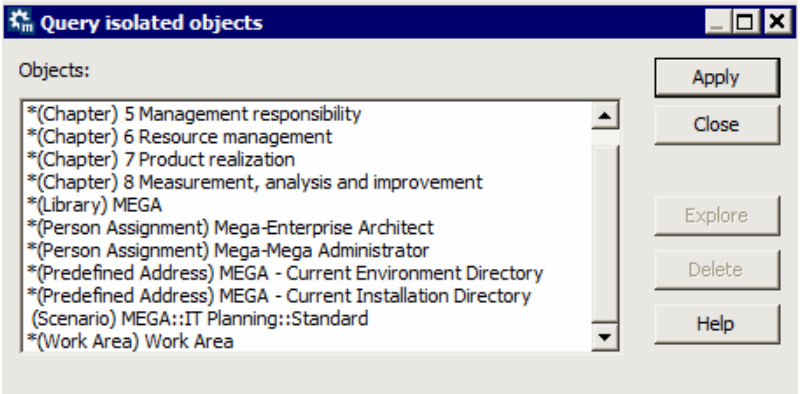
   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **Repositories** folder.
3. Right-click a repository and select **Object Management > Query Isolated Objects**.

   ☛ *Alternatively, in your **HOPEX** workspace, select **Tools > Manage > Query Isolated Objects**.*

   The **Query Isolated Objects** dialog box appears.

4. Click **Apply** to run the query.
   The system scans the repository, and lists the **Objects** not connected to higher level objects in the hierarchy, preceded by their type.



> ☛ *Objects not preceded by an asterisk are not linked to any other object and can be deleted.*
>
> ☛ *Objects preceded by an asterisk are connected only to lower objects in the hierarchy. An example is an org-unit that sends messages but does not appear in any diagram.*
>
> ☛ *Querying for isolated objects involves searching the entire repository and for this reason the operation may take some time.*

The **Explore** button allows you to explore links around the selected object.

The **Delete** button allows you to delete selected objects.

> **It is normal to find projects, processes, applications and keywords in the list. Do not delete these types of object.**
>
> **In the "Extended metamodel" configuration, objects of type _deskitem are listed. These are user workspaces: Do not delete them.**

Querying Objects to be Translated

In the case of multilingual use of **HOPEX**, you can query objects to be translated in the repository.

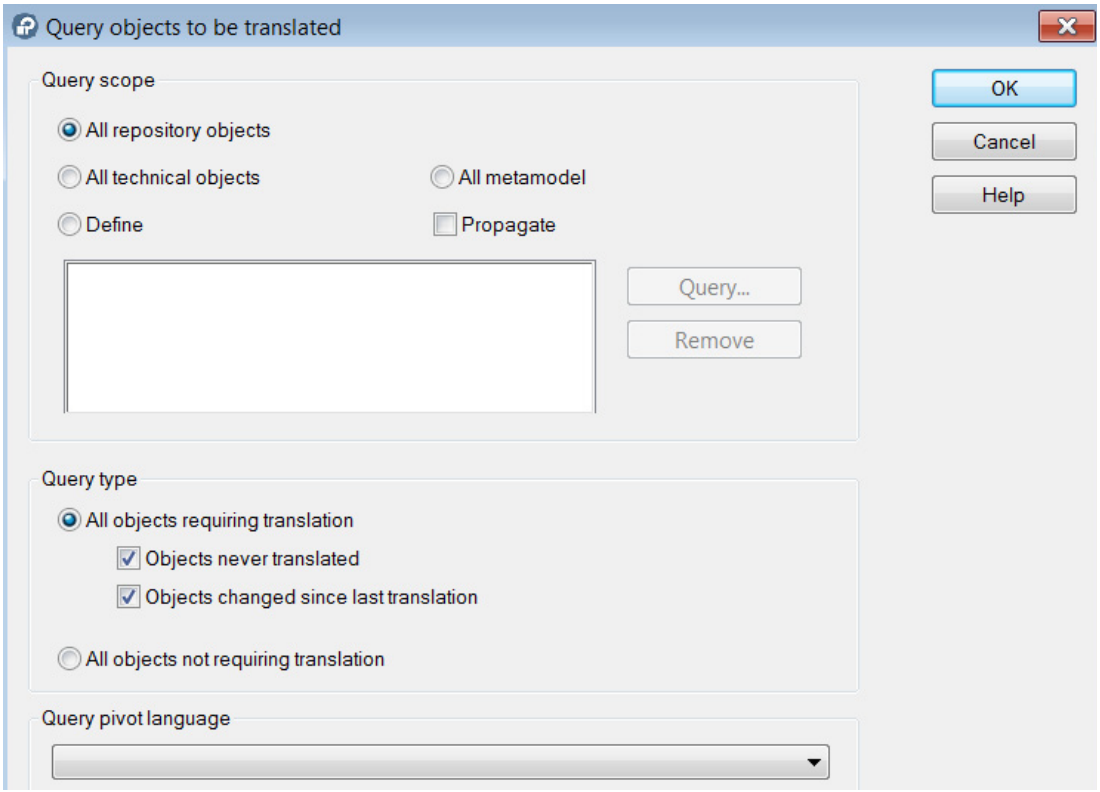
> ☛ *For more details on translation of objects, see the **HOPEX Common Features** guide, in the "Handling repository objects" chapter, the "Using **HOPEX** in a Multilanguage Context" section.*

To query objects to be translated:

1. Connect to **HOPEX**.

   ☛ *"Connecting to HOPEX", page 136.*

2. In the menu bar, select **Tools > Manage > Query Objects to be Translated**.
   The **Query objects to be translated** dialog box opens.



3. In the **Query scope** pane, define the query perimeter of the objects not translated:
   - on **All repository objects**
   - on **All technical objects**
   - on **All metamodel**
   - on specific items: select **Define** and click **Query**.

4. By default the search is performed on **All objects requiring translation**.
   You can refine your query by clearing one of the options:
   - **Objects never translated**
   - **Objects changed since last translation**

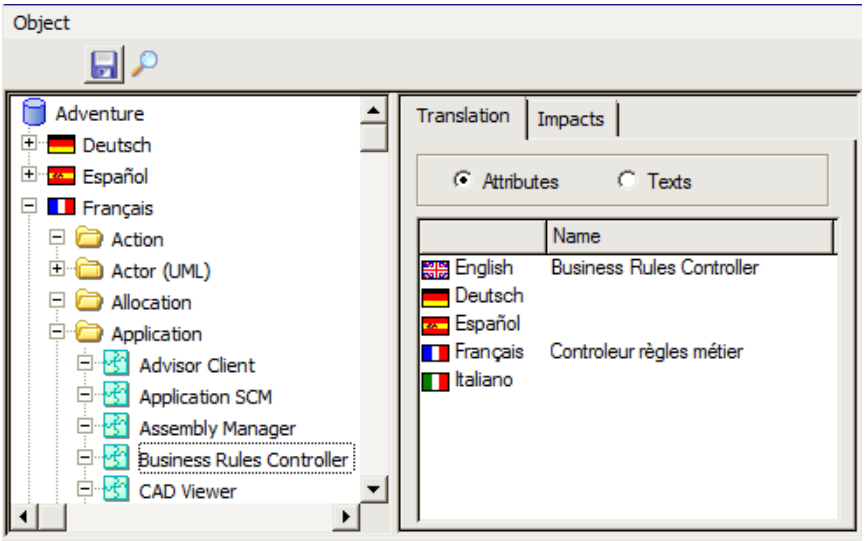
   ☛ *If needed, you can perform the query on **All objects not requiring translation**.*

5. Click **OK**.
   The query starts.

   ☺ *Click **Stop** to interrupt the query in progress.*

   The **List of objects to be translated** dialog box opens. It indicates objects to be translated, classified in the menu tree by language and object type.

6. Enter translation of the object names directly in this dialog box.



☛ *When you want to change language for display of the **HOPEX** interface, you must close then reopen a session. The names of menus and tabs are then refreshed.*

IMPORTING A SOLUTION PACK IN HOPEX

Prerequisite: the Solution Pack that you want to import is decompressed (**HOPEX** installation folder > **Utilities > Solution Pack**, double-click the Solution Pack to extract it).

To import a Solution Pack in a HOPEX repository:

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **Repositories** folder.
3. Right-click the repository and select **Object Management > Import Solution Pack**.
   The **Solution Pack Import** dialog box appears.
4. Select the Solution Pack(s) that you want to import.
5. Click **OK**.
   The **Import MEGA Data XML** dialog box displays import progress.
   The selected Solution Packs are imported into the repository.

MANAGING UI ACCESS (PERMISSIONS)

☛ *UI access management is only available with the **HOPEX Power Supervisor** technical module.*

To modify profile UI access, you must have modification authorization rights on this profile.

To modify a profile for which you do not have modification rights, you can create a new profile from this profile, see *"Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35*.

UI access is managed at profile level.

You can manage:

- *object UI access*

  *Object UI access defines user rights on creation, reading, update, and deletion on these objects and their tools. By default, object UI accesses have value *CRUD (C: create, R: read, U: update, D: delete, *: default value).*

  ☛ *For information on management of workflow UI Access, see the **HOPEX Collaboration Manager - Workflows** guide ("Managing Workflows" Chapter, "Managing workflow authorizations" section, "Configuring permissions on objects" paragraph).*

- *general UI access*

  *General UI access defines if tools are available or not. By default, general UI accesses have value *A (A: Available, *: default value)*

Opening the UI Access Management Window (Permission)

The **UI Access** dialog box (Windows Front-End) of an environment enables management of UI access for the complete environment and for each profile.

To modify UI access of a profile, you must have modification authorization rights on this profile.

To modify a profile supplied by MEGA, you must create a new profile, see "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35.

- **Object UIs**, which details for the selected profile its access to UI of objects and its access to tools specific to these objects.

  ☛ *See "Object UI Access Values", page 234.*

  ☛ *See "Managing UI Access", page 235.*

- **General UIs**, which details for the selected profile its access to general UIs.

  ☛ *See "Object UI Access Values", page 234.*

  ☛ *See "Managing General UI Access", page 245.*

To open the UI Access Management Window (Windows Front-End)

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **User management** folder of the environment.
3. Right-click the **UI access** folder and select **Manage**.
   The dialog box for managing **UI Access** opens.
4. Click the tab:
   - **Object UIs**
   - **General UIs**

Object UI Access Values

Object UI access enables definition of user permissions on the selected metamodel.

- Preceding the value of a permission, the character:
  - * indicates that the value is directly inherited from the default value.
  - - indicates that the value is inherited from an element hierarchically higher in the same profile or sub-profile.
- Value empty means that the user has no permission on the element. The element is not visible to the user.
  When a MetaClass is hidden to a user, it is not available in the repository.

  ```
  For example, if the "Package" MetaClass is hidden for a
  user, this user cannot use packages in modeling work since
  this object type is not accessible in the interface.
  ```

See:

- "MetaClass occurrence access permissions", page 234
- "MetaAssociationEnd access permissions", page 235
- "MetaAttribute access permissions", page 235
- "Permissions on a tool", page 235

MetaClass occurrence access permissions

By default, the access permission on occurrences of a MetaClass takes value *CRUD:

- C: Create
- R: Read
- U: Update
- D: Delete

An access permission on occurrences of a MetaClass can take combinations of values:

- **R**: read occurrences of the MetaClass
- **CRU**: create, read and update occurrences of the MetaClass
- **CRUD**: create, read, update and delete occurrences of the MetaClass
- **RU**: read and update occurrences of the MetaClass
- **RUD**: create, read, update and delete occurrences of the MetaClass

MetaAssociationEnd access permissions

By default, the access permission on a MetaAssociationEnd takes value *CRUD :

- C: Connect
- R: Read
- U: Update
- D: Disconnect
- M: Mandatory

A permission on a MetaAssociationEnd can take combinations of values:

- R
- CRU
- CRUD
- RU
- RUD

MetaAttribute access permissions

By default, access permission on a MetaAttribute takes value: *RU.

- R: Read
- U: Update
- M: Mandatory

A permission on a MetaAttribute can take combinations of values:

- R: the MetaAttribute is visible
- RU: the MetaAttribute is visible and modifiable
- RUM: the MetaAttribute is visible, modifiable and mandatory

Permissions on a tool

A tool can be available or not.

By default, availability on a tool is: *A.

The permission on a tool can take value:

- A: the tool is available
- <empty>: the tool is not available

Managing UI Access

To modify UI access on an object for a given profile, you must have modification authorization rights on this profile.

*For information on management of accesses to user interface workflows, see the **HOPEX Collaboration Manager - Workflows** guide.*

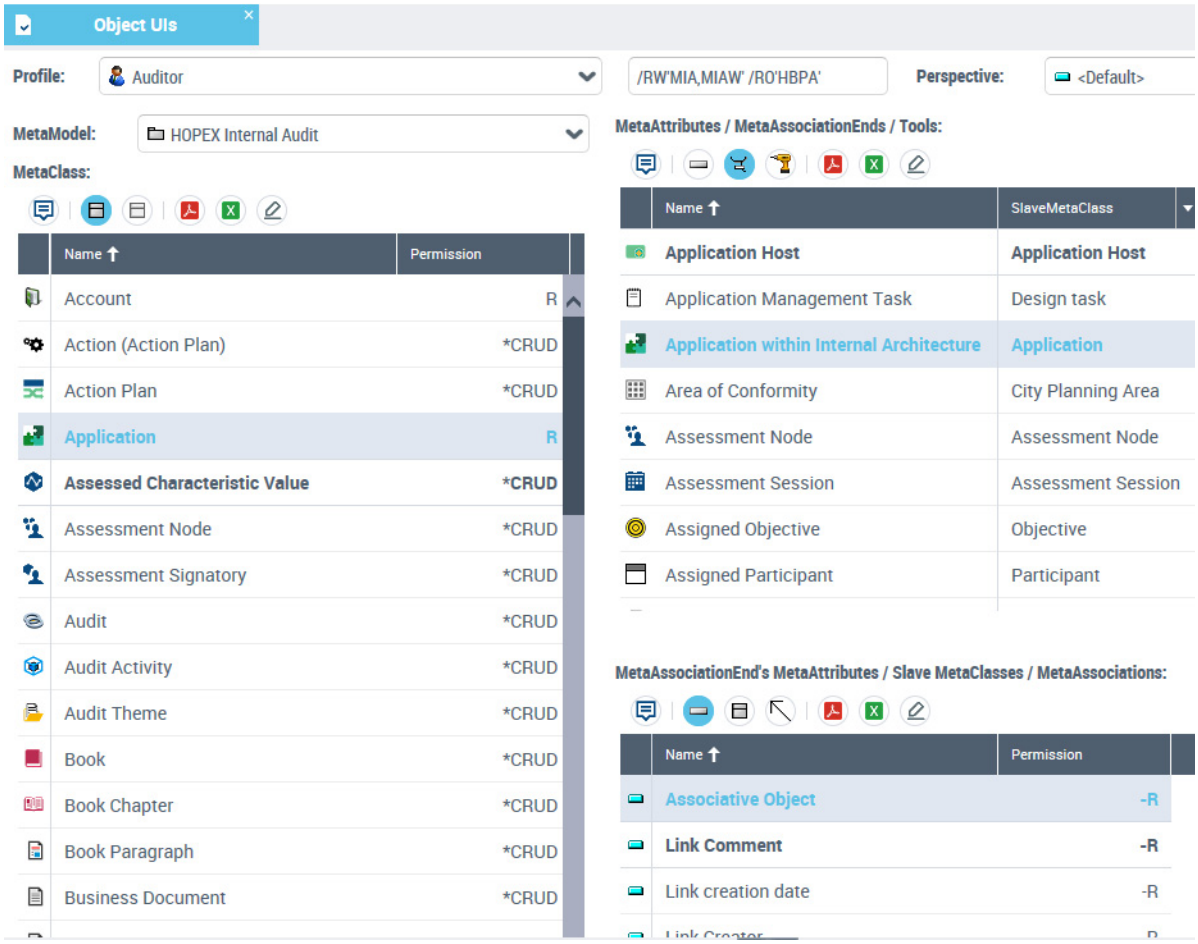
For a new profile, access permissions on an object of this profile are by default:

- inherited from the Default profile, if the profile is not an aggregation of profiles (in profile parameters, the profile does not contain sub-profiles,

see "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35).

- inherited from permissions defined on owned profiles, if the profile is an aggregation of profiles (in profile parameters, the profile contains one or several profiles, see "Customizing an Existing Profile / Creating a Profile from an Existing Profile", page 35).

☛ *See "Rules on permissions at profile aggregation", page 243.*



In the **Object UIs** tab:

- the **Profile** field enables definition of the profile for which you want to define access permissions.
- the **MetaModel** field enables filtering of MetaClasses displayed in the **MetaClass** frame according to the selected MetaModel.
  - "All" value lists all existing MetaClasses.
  - <Extensions> value lists all MetaClasses that are not stored in standard MetaModels (HOPEX products)

To define access permissions on objects, see:

- "Modifying access permissions on occurrences of a MetaClass for a profile", page 237.
- "Modifying access permissions of MetaAttributes of a MetaClass for a profile", page 239.
- "Modifying access permissions to tools of a MetaClass for a profile", page 240.
- "Modifying access permissions of a link around a MetaClass for a profile", page 241.
- "Modifying access permissions on links around a MetaClass for a profile", page 242.

Modifying access permissions on occurrences of a MetaClass for a profile

To modify access permissions on occurrences of a MetaClass for a profile:

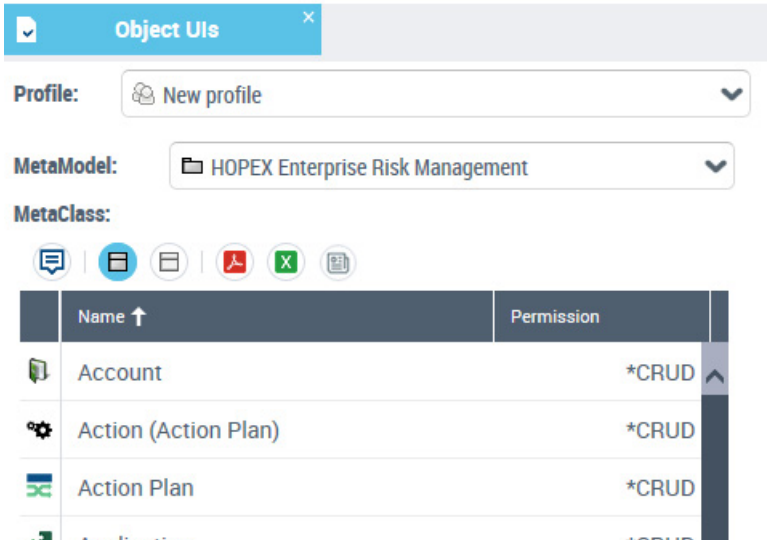
1. Access the UI access management window and select the **Object UIs** tab.

   ☛ *See "Opening the UI Access Management Window (Permission)", page 233.*

2. In the **Profile** field, select the profile using the drop-down menu.

   ☛ *The <Default> profile defines default access permissions of each MetaClass, MetaAttribute, MetaAssociationEnd and tool.*

3. In the **MetaModel** field, select the MetaModel concerned.
   In the **MetaClass** frame, the listed MetaClasses are filtered according to the selected MetaModel.

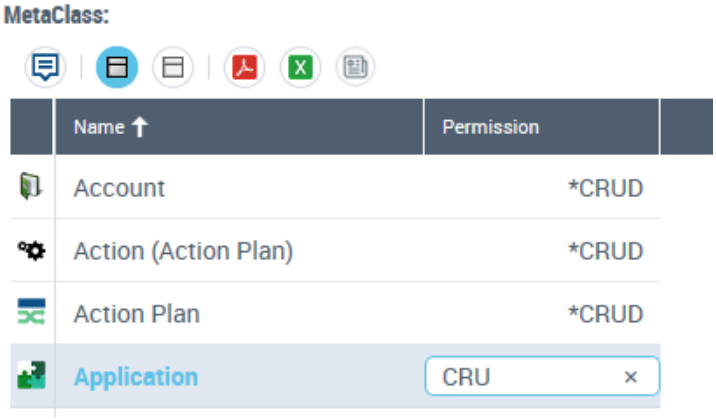


4. In the **MetaClass** frame, select the MetaClass for which you want to modify configuration of access permissions.

   ☛ *By default, its configuration is that inherited from the <Default> profile.*

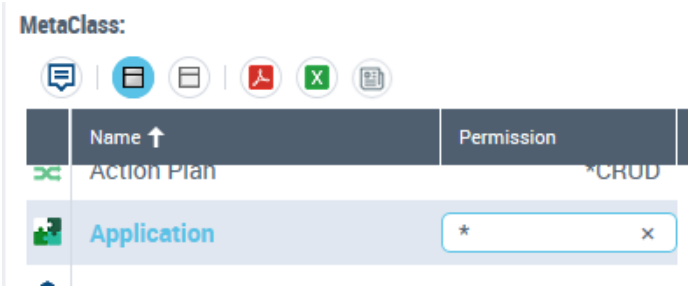
5. In the **Permission** field, enter the new value.

☛ *See "MetaClass occurrence access permissions", page 234.*



6. Press "Enter".
   The value of the MetaClass permission is modified.
   In the **MetaAttributes/MetaAssociationEnds/Tools** frame, the values of permissions of elements of the MetaClass are also modified.

   ☛ *To return to the default value of the permission on the MetaClass, enter the character *.*



   ☛ *To obtain information on inheritance of the value, enter the character ?.*

   You can also modify the MetaAttributes/MetaAssociationEnds/Tools of a MetaClass, see:

   - "Modifying access permissions of MetaAttributes of a MetaClass for a profile", page 239.
   - "Modifying access permissions to tools of a MetaClass for a profile", page 240.
   - "Modifying access permissions of a link around a MetaClass for a profile", page 241.
   - "Modifying access permissions on links around a MetaClass for a profile", page 242.

Modifying access permissions of MetaAttributes of a MetaClass for a profile

To modify access permissions of MetaAttributes of a MetaClass for a profile:

1. Access the UI access management window and select the **Object UIs** tab.

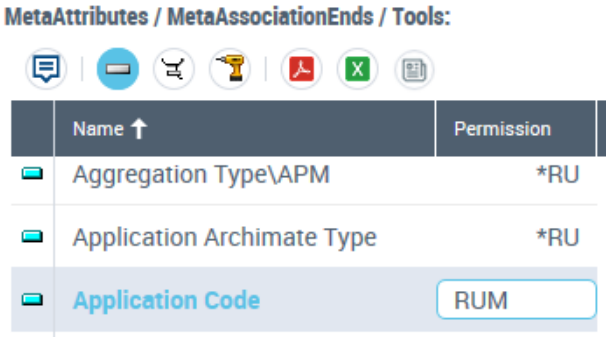
   ☛ *See "Opening the UI Access Management Window (Permission)", page 233.*

2. In the **Profile** field, select the profile using the drop-down menu.

   ☛ *The <Default> profile defines default access permissions of each MetaClass, MetaAttribute, MetaAssociationEnd and tool.*

3. In the **MetaModel** field, select the MetaModel concerned.
   In the **MetaClass** frame, the listed MetaClasses are filtered according to the selected MetaModel.
4. In the **MetaClass** frame, select the MetaClass concerned.
5. In the toolbar of the **MetaAttributes/MetaAssociationEnds/Tools** frame, click **MetaAttribute** .
   The MetaAttributes of the MetaClass are listed.
6. Select the MetaAttribute for which you want to modify permissions.
7. In the **Permission** field, enter the new value.

   ☛ *See "MetaAttribute access permissions", page 235.*



8. Press "Enter".
   The value of the MetaAttribute permission is modified.

   ☛ *To return to the default value, enter the character *.*

   ☛ *To obtain information on origin of an inherited value, enter the character ?.*

Modifying access permissions to tools of a MetaClass for a profile

A tool can be available or not.

To modify access permissions to tools of a MetaClass for a profile:

1. Access the UI access management window and select the **Object UIs** tab.

   ☛ *See "Opening the UI Access Management Window (Permission)", page 233.*

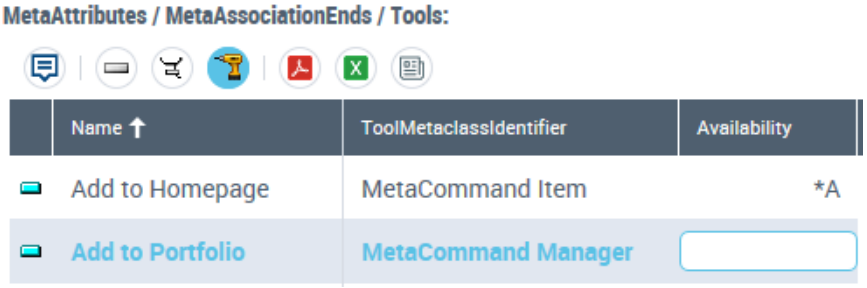
2. In the **Profile** field, select the profile using the drop-down menu.

   ☛ *The <Default> profile defines default permissions of each MetaClass, MetaAttribute, MetaAssociationEnd and tool.*

3. In the **MetaModel** field, select the MetaModel concerned.
   In the **MetaClass** frame, the listed MetaClasses are filtered according to the selected MetaModel.
4. In the **MetaClass** frame, select the MetaClass concerned.

5. In the toolbar of the **MetaAttributes/MetaAssociationEnds/Tools** frame, click **Tools** .
6. Select the tool for which you want to modify access permissions.
7. In the **Permission** field, enter the new value.

   ☛ *See "Permissions on a tool", page 235.*



8. Press "Enter".
   The value of the tool access permission is modified.

   ☛ *To return to the default value, enter the character *.*

   ☛ *To obtain information on inheritance of the value, enter the character ?.*

Modifying access permissions of a link around a MetaClass for a profile

To modify access permissions of a link around a MetaClass for a profile:

1. Access the UI access management dialog box and select the **Object UIs** tab.

   ☛ *See "Opening the UI Access Management Window (Permission)", page 233.*
2. In the **Profile** field, select the profile using the drop-down menu.

   ☛ *The <Default> profile defines default permissions of each MetaClass, MetaAttribute, MetaAssociationEnd and tool.*
3. In the **MetaModel** field, select the MetaModel concerned.
   In the **MetaClass** frame, the listed MetaClasses are filtered according to the selected MetaModel.
4. In the **MetaClass** frame, select the MetaClass concerned.
5. In the toolbar of the **MetaAttributes/MetaAssociationEnds/Tools** frame, click **MetaAssociationEnd** .
6. Select the MetaAssociationEnd for which you want to modify link access permissions.

7. In the **Permission** field, enter the new value.

   ☛ *See "MetaAssociationEnd access permissions", page 235.*

MetaAttributes / MetaAssociationEnds / Tools:

Name	SlaveMetaClass	Link Permission
Action	Action (Action Plan)	*CRUD
Action Plan	Action Plan	*RUD
Aggregation of	Application	*CRUD

8. Press "Enter".
   The value of the link access permission is modified.

   ☛ *To return to the default value, enter the character *.*

   ☛ *To obtain information on inheritance of the value, enter the character ?.*

   See also "Modifying access permissions on links around a MetaClass for a profile", page 242.

Modifying access permissions on links around a MetaClass for a profile

You can modify access permissions on:

- the link according to the MetaClass accessed via the link
- one of the MetaAttributes of the link
- one of the MetaClasses accessed via the link

```
Example: You can grant rights to connect (but not to create)
an IT Service to an Application via this same link.
```

To modify access permissions on links around a MetaClass for a profile:

1. Select the MetaAssociationEnd.

   ☛ *See "Modifying access permissions of a link around a MetaClass for a profile", page 241, steps 1 to 6.*

2. In the menu bar of the **MetaAttributes of MetaAssociationEnds/ Slave MetaClasses/MetaAssociations**, click **MetaAttribute**, **MetaClass**, or **MetaAssociation**.
3. In the list, select the MetaAttribute, MetaClass or MetaAssociation concerned.
4. In the **Permission** field, enter the new value.

   ☛ *See "MetaAttribute access permissions", page 235.*

   ☛ *See "MetaClass occurrence access permissions", page 234.*

5. Press "Enter".
   The value of the access permission is modified.

   ☛ *To return to the default value, enter the character *.*

   ☛ *To obtain information on origin of an inherited value, enter the character ?.*

Rules on permissions at profile aggregation

When a profile aggregates several sub-profiles, its permissions are defined by the addition of permissions defined on its sub-profiles.

```
Example:

Profile 1 is the aggregation of sub-profiles 1.1 and 1.2.

If the permission on an object A of sub-profile 1.1 has
value CR, and that of sub-profile 1.2 has value RUD, then
the value of this permission on object A for profile 1 is
CRUD.
```

Attention to default values

A permission value with * means that this value is the default permission value and that it has not been specifically defined. Only those values specifically defined are taken into account in aggregation.

```
Example:

Profile 1 is the aggregation of sub-profiles 1.1 and 1.2.

If the permission on an object A of sub-profile 1.1 has
value *CRUD, and that of sub-profile 1.2 has value R, then
the value of this permission on object A for profile 1 is R.
```

Managing Data Access Dynamically

Writing and reading access diagrams define data access statically. A person sees objects belonging to his/her reading access area, and can modify objects belonging to his/her writing access area.

☛ *See "Managing Data Writing Access", page 249, "Managing Data Reading Access", page 275.*

You can define dynamic rules for reading or writing data access.

Dynamic rule:

- defines for a person, his/her reading or writing access rights on a given object
- can be based on characteristics of:
  - an object
  - of a person
  - an object or person
- can be associated with one or several profiles

Creating permission rules

A permission rule is defined by a macro. A permission rule can define reading or writing access rights to an object.

To create a permission rule:

1. From the **HOPEX** explorer, click **Create** ✚.
2. Select **Data Access Rule** and click **OK**.

3. In the **Creation of Data Access Rule** dialog box, enter a **Name** for the rule and click **OK**.
4. Access properties of the rule.
5. In the **Characteristics** tab, in the **Macro** field, click the arrow and connect the macro.
6. In the **Data Access Type** field, select the data access type (**Reading** or **Writing**).
   In the **User Profile** frame, if no profile is connected to the rule, the rule applies to all profiles.

   ☛ *See "Associating a permission rule with a profile", page 244.*

Associating a permission rule with a profile

☛ *To associate a dynamic permission rule to an object, you must have rights to modify **HOPEX** data, see "Authorizing HOPEX Data Modification", page 61.*

To associate a permission rule with a profile:

1. Open permission rule properties.

   `Example: "Action Plan - Writing"`
2. Click the **Characteristics** tab.
3. In the **User Profile** frame, click **Connect** and select the profile with which you want to associate the permission rule.

Associating a permission rule with an object

☛ *To associate a dynamic permission rule to an object, you must have rights to modify **HOPEX** data, see "Authorizing HOPEX Data Modification", page 61.*

To associate a permission rule with an object:

1. Open object properties.

   `Example: MetaClass "Risk"`
2. Select the **Data Access** tab.
3. In the **Data Access Rule** frame, click **Connect** and select the rule you wish to associate with the object.

Generating a Report on Permissions by Profile

A report allows you to generate the detail of permissions for a given workflow.

Generating the report

To generate this report:

1. In the menu bar of **HOPEX** (Windows Front-End), select **Tools > Profile and Permission Management > Profile Permissions Report**.
   A wizard opens.

2. (Optional) In the **Report File Name** field, modify the name and/or location in which to save the report by default. By default this is your user folder.

   ☛ *See "Repository Structure", page 75.*

3. Select report parameters:
   - In the **MetaModel** field, select the MetaModel concerned.

     ☺ *For a more rapid result, do not select <All>, but just the metamodel concerned.*

   - in the **Profile** frame, click **Add Profile** (you can add several).

     ☺ *For a more rapid result, do not select a large number of profiles.*

4. Click **OK**.
   The report is generated as an Excel worksheet.

   ☛ *Generation can take some time, depending on the parameters you have selected.*

Report content

All MetaClasses of the selected metamodel appear in the report.

Presented for each MetaClass are:

- in rows: all MetaAttributes, Tools, MetaAssociations (and MetaAttributes of MetaAssociations) of the MetaClass.
- in columns: permissions for all selected profiles.

  ☛ *For improved readability, missing permissions are replaced by _.*
  *For example: *RU is replaced by *_RU_.*

Managing General UI Access

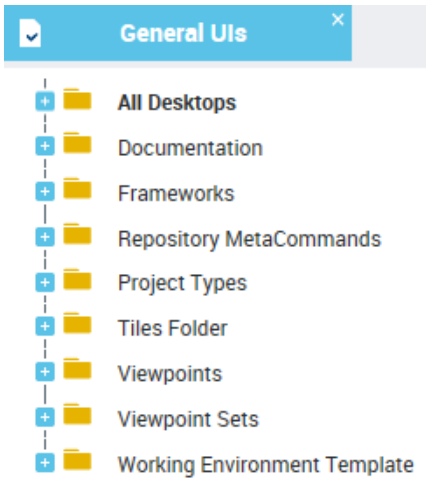
You can manage general UI access for a profile. General UIs are classified by category:

- desktop
- command category
- command group
- general command
- properties page
- tree

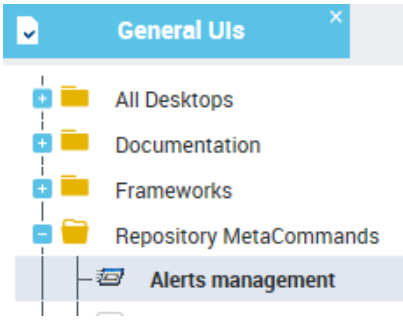
To manage general UI access:

1. Access the UI access management dialog box and select the **General UIs** tab.

   ☛ *See "Opening the UI Access Management Window (Permission)", page 233.*

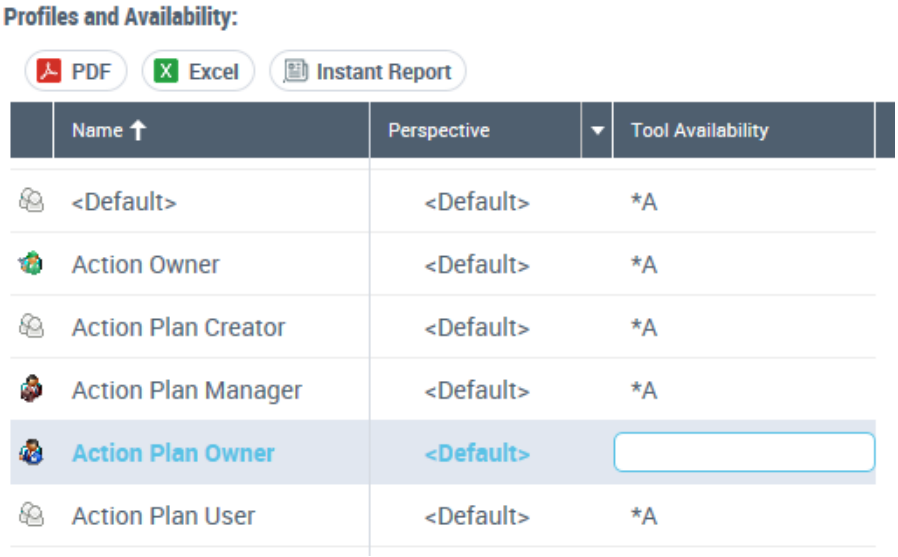


2. In the **SystemDb** tree, expand the folder of the category concerned.
3. In the list, select the tool concerned.



4. In the **Profiles and Availability** frame, select the profile for which you want to modify access on the tool.

5. In the **Tool Availability** field, enter the availability value.



6. Press "Enter".
   The value of tool availability is modified.

   ☛ *To return to the tool availability default value, enter the character *.*

   ☛ *To obtain information on origin of an inherited value, enter the character ?.*

MANAGING SHAPES

HOPEX provides several sets of shapes used to represent the various objects in diagrams.

HOPEX Administration, allows you to consult the list of shapes:

- shapes customized by users
- shapes supplied by **MEGA**
  Each set of shapes (5.1 / 5.2 / 2005 / 2009) contains three shape categories:
  - Art
  - Background
  - Method

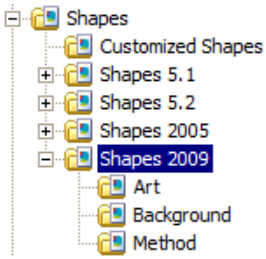
Accessing Shapes

To access shapes:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. Expand the **Shapes** folder.
3. Expand the sub-folder of the desired shapes.



4. Right-click the category of the desired shapes and select **Manage**.
   A table appears listing the images of this category and their location.

MANAGING DATA WRITING ACCESS

HOPEX Administration is provided with tools required for writing access management.

This chapter explains how to create a *writing access diagram* and how to customize its characteristics.

The following points are covered here:

- ✓ "Introduction to writing access management", page 250
- ✓ "Opening the Writing Access Diagram", page 254
- ✓ "Compiling the Writing Access Diagram", page 255
- ✓ "Defining Writing Access Areas", page 256
- ✓ "Customizing Writing Access Area Management", page 263
- ✓ "Managing Users from the Writing Access Diagram", page 269
- ✓ "Customizing Writing Access Diagram Display", page 272

INTRODUCTION TO WRITING ACCESS MANAGEMENT

☛ *Managing writing access areas is available with the **HOPEX Power Supervisor** technical module only.*

The administrator declares the users and defines the writing access.

The structure of *writing access* is defined in the *writing access diagram*. There is a hierarchical link between writing access.

☺ *Implementation of this function does not replace a structured management of projects. To operate, the writing access diagram must model itself on organization of projects.*
Clear functional breakdown of your projects simplifies management of operational follow-up of writing accesses.

Users

user

A user is a person with a login.

At creation of an environment, two users are declared by default. These two users have administration rights to manage repositories and users:

- the "Administrator" person, with Login "System" (without password)

  ☛ *The "Administrator" user cannot be deleted. It has no profile (it has all rights). It is recommended that you define a password to restrict use of the "Administrator" user code.*

- The "Mega" person, with Login "mega" (without password)

You must declare other users who will access repositories.

If several environments are defined for a site, users must be defined in each of these environments. To do this, export the user diagram from the reference environment, and import it into each of the other environments.

Do not manually create a user with the same name in other environments. If you do, the user will have a different absolute identifier in each environment, and you would actually have created different users with the same name.

☺ ***MEGA** recommends that you create the repositories before defining the users, so you can declare the user access rights when they are created.*

To access a repository, a user must identify himself/herself. Then, depending on user writing access area, he/she is able to modify the repository.

User groups

A person can belong to one or more groups. A user group is a group of persons with a login.

Persons belonging to a group:

- depend on the same environment.
- share the same connection characteristics defined on the **login** of the group.
- share the assignments defined for the group:
  - (Web Front-End) persons accumulate their assignments and those defined for the group.
  - (Windows Front-End) the assignments defined for the person are ignored.
- connect to the application
  - (Web Front-End) with the group **login**
  - (Windows Front-End) with their **login**, but with access rights to the repository defined on the **login** of the group.

☛ *see "Login Properties", page 30.*

By default at creation of an environment, a group of users is created:

- the "Guests" person group, with Login "Guest".

Writing Access Areas

Each user or group of users is connected to a writing access area. It is the person or person group that carries the writing access area.

Each object is connected to a writing access area.

☛ *At creation, the object inherits the writing access area of the person who created it.*

MEGA delivers by default the "Administrator" writing access; this writing access area:

- cannot be deleted.
- is the highest writing access area level; it does not depend on any other writing access area. In principle it should be reserved for repository administration.
- is the writing access area to which "Administrator", "Mega" and "Guests" are connected.
  When the writing access diagram has been installed, **MEGA** recommends that you change the writing access area levels of "Mega" and "Guests". It is not desirable that default users have such extensive rights.

All other writing access areas depend on at least one writing access area.

Writing access areas are interconnected by hierarchical links. This is a strict hierarchy, with no circular dependencies: a writing access area cannot be declared at a higher level than the writing access area on which it depends, either directly or via a succession of dependencies.

A user can modify an object connected to his/her writing access area or to a hierarchically lower writing access area.

The writing access area of an object can be modified by the administrator:

- by specifically changing the object writing access area
- when modifying the writing access area of another object (project, process, diagram, etc.) if the propagation option is enabled.

Writing Access Diagram

There is one writing access diagram per environment.

☛ *If several environments use the same protection configuration, the same user diagram must be used in all these environments.*

Rules

Installation of a writing access diagram diagram must respect the following rules so as to minimize user management costs:

- Any object must be modifiable by users that may need to modify it, without administrator intervention.
- The administrator should intervene only in exceptional circumstances.

Use

The writing access diagram is used similarly to a diagram. The persons, person groups and writing accesses are handled just like standard objects:

- Creation and modification of the name, etc., are done in the same way as for standard objects.
- Be careful however when you delete a writing access.

☛ *See "Deleting Writing Access Areas", page 259.*

Avoid using the Cut command for a person or person group, as this can result in errors in the writing access diagram if the person or person group is not deleted from the repository or not linked to a writing access.

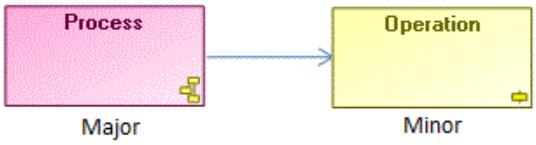
Link Orientation: Major and Minor Objects

When two objects are linked, one object is *major* and the other *minor*.

You cannot delete a minor object if you do not have writing access on the major object.

The major object in the link is the one whose nature changes with the presence or absence of the link. For example a process, defined as a succession of operations, is modified if you remove an operation. The process is then major for the link. If the objects are protected, you must have the correct authorization for modifying the major object in order to create or delete the link.

The minor object in a link is the one whose nature is not modified or only slightly modified by presence or absence of this link. For example, removing an operation from a process does not change characteristics of this operation. Therefore the process is minor in the link.



In the above example, you must have writing access on the org-unit (the major object) to disconnect or delete the message (minor object).

Opening the Writing Access Diagram

☛ *To access the writing access diagram, you must have a license for the **HOPEX Power Supervisor** technical module.*

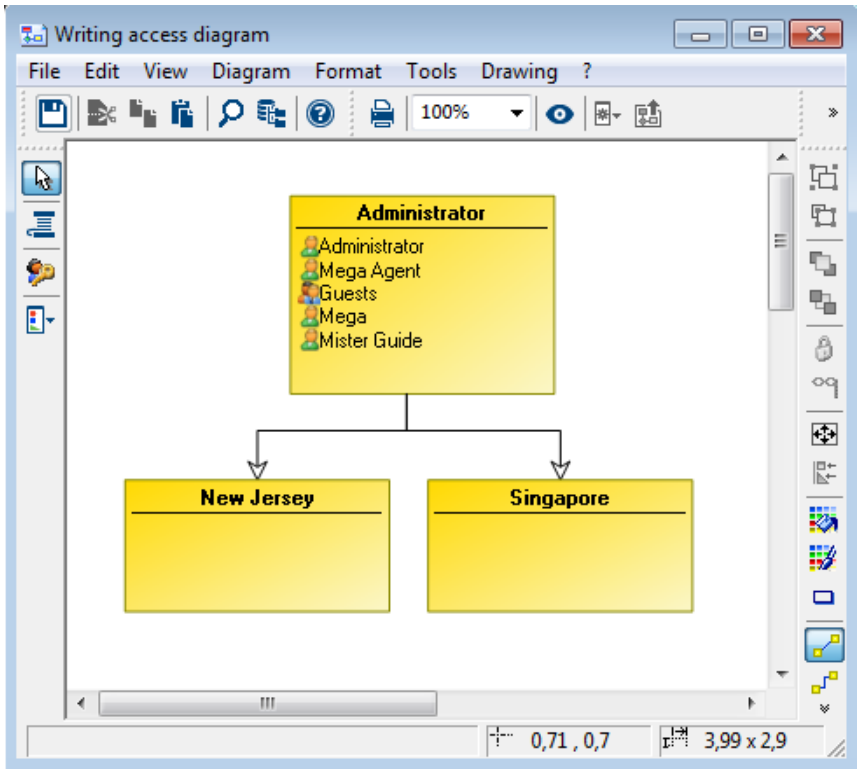
Opening the Writing Access Diagram (Windows Front-End)

To open the writing access diagram:

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. In the **User Management** folder, right-click **Data Writing Access** and select **Open Diagram**.
   The diagram opens in a new window.



COMPILING THE WRITING ACCESS DIAGRAM

Running writing access diagram compilation assures consistency of behavior of **HOPEX** with declarations of the diagram.

If the diagram is not compiled, there is a risk that certain users will be able to update objects that are normally protected.

When modifying the writing access diagram, so as to warn of rejects due, for example, to writing access restrictions or deletions before compilation it is recommended that:

- all changes made on the user workstations should be uploaded to the administrator workstation, or
- all private workspaces dispatched.

To compile the writing access diagram from the **Administration** application:

1. From **HOPEX Administration**, connect to the environment concerned.

   See "Connecting to an Environment", page 7.

2. Expand the **User management** folder of the environment.
3. Right-click the **Data Writing Access** folder and select **Compile**.
   When compilation is complete, a message indicates whether the operation was successful or whether the diagram contains errors.

   The most frequent errors are:

   A writing access area (other than '"Administrator") is not attached to any other.

   A person or person group is not attached to any authorization.

Defining Writing Access Areas

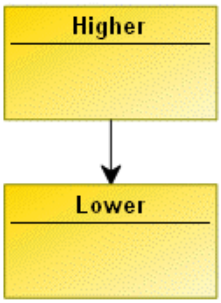
The writing access diagram is available if you have the HOPEX Power Supervisor technical module. With this diagram, you can create users and manage their writing access rights for repositories and product functions. By default only one writing access area is defined, named "Administrator". Attached to it are the "Administrator" and "Mega" persons. This is the highest writing access level and it should normally be reserved for repository management. You cannot delete this writing access area.

The *writing access diagram* enables creation of writing access areas.

Creating a Writing Access Area

To create a writing access area:

1. In the diagram insert toolbar, click icon **Writing Access Area** then click in the diagram.
2. Enter the name of the writing access area.
   Dependency of writing access access areas is determined by creation of a "lower" link which starts from the higher writing access area to the lower writing access area.



Defining Writing Access Area Persons or Person Groups

To define persons or person groups of a writing access area:

1. From **HOPEX Administration**, open the writing access diagram.
   See "Opening the Writing Access Diagram", page 254.
2. Right-click the writing access area and select **Properties**.
3. Select the **Users** tab and click **Connect**.

4. In the query tool, click the arrow in the first field and select the target (*Access Area Member*, Person or Person Group).

   Access area member groups all persons and person groups belonging to an access area. This area defines objects that can be accessed by the person or person group.

5. (optional) In the second field, enter the character string to be queried.
6. Click **Find**.
7. In the results list, select the required access area member and click **Connect**.

   Press the [CTRL] key to select several members simultaneously.

   The person or person group you have connected appears in the list of access area members of the selected writing access area.

Defining a Writing Access Area at Creation

To assign to all objects created by a user a writing access area different from the writing access area of this user, you must associate a writing access area at creation with the user concerned.

To define a writing access area at creation of a user:

1. Open the properties dialog box of the person.
2. Select the **Characteristics** tab.
3. In the **Writing Access Area at Creation** field, select the required writing access.

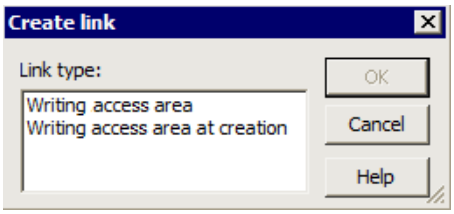
   See "Person writing access area and writing access area at creation", page 29.

To define a writing access area at creation of a user:

1. From **HOPEX Administration**, open the writing access diagram.

   See "Opening the Writing Access Diagram", page 254.

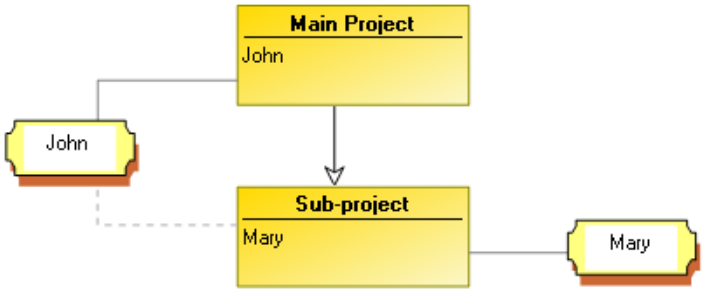
2. If this is not already done, place the person and the writing access area concerned in the diagram.
3. Draw a link between the person and the required writing access area. The **Create Link** dialog box opens.



4. Select **Writing Access Area at Creation** and click **OK**.

   *Value "None" of **Writing Access Area at Creation** signifies that the user creates objects in the same writing access area as that to which he/she belongs.*

When the writing access area at creation has been created, it is represented by a dotted line link between the person and the writing access area in the writing access diagram.



In the above example, user John has:

- a writing access area of level "main project"
- a writing access area at creation of level "sub-project"

Objects created by John can therefore be modified by Mary.

Modifying Writing Access Areas of Objects

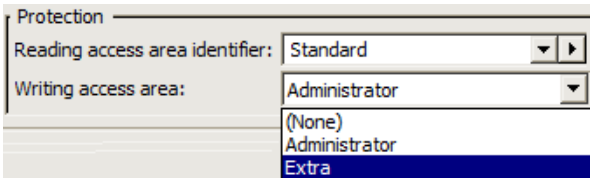
If you have a writing access area of level higher than or equal to that of an object, you can modify the writing access area of this object in the object properties dialog box.

To modify writing access area of an object:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. In your **HOPEX** desktop, open the object properties window, select the **General** tab, then the **Administration** sub-tab.
3. In the **Protection** pane, in the **Writing Access Area** field, select a writing access area via the drop-down menu.



4. Click **OK**.

Modifying Writing Access Areas of an Object Group

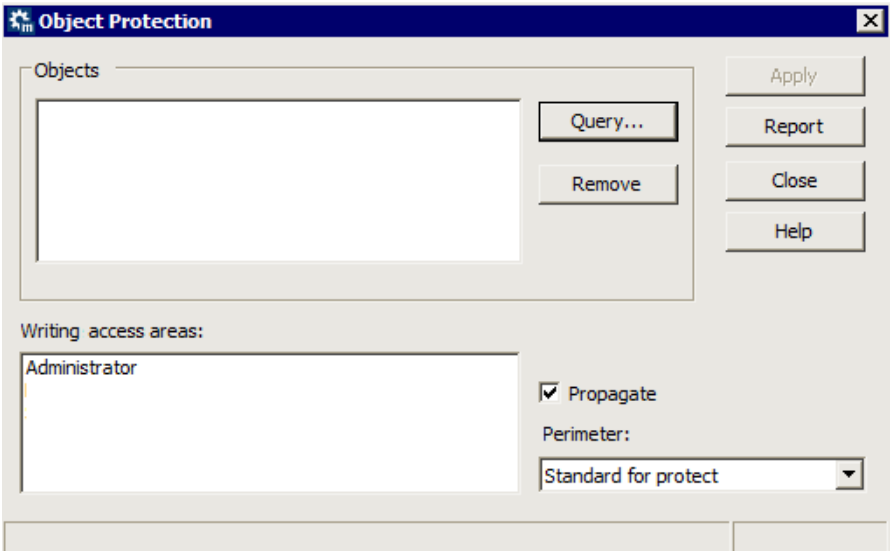
If you have a writing access area of level higher than or equal to that of an object group, you can modify the writing access area of this object group.

To modify the writing access area of an object group:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. In your **HOPEX** workspace, select **Tools > Manage > Protect Objects**. The **Protect Objects** dialog box opens.
3. In the **Objects** pane, click **Query** and select the object group.
4. In the **Writing Access Areas** pane, select the writing access area you want to assign to the object group.
5. (Optional) Select **Propagate** if you want the writing access area to be propagated to all dependent objects of the object group, as a function of the perimeter selected.



6. Click **Apply**.
   Object protection is applied.
7. Click **Report** to check if a conflict has been encountered at protection propagation in the repository.

Deleting Writing Access Areas

To delete a writing access area:

1. From **HOPEX Administration**, open the writing access diagram.

   ☛ *See "Opening the Writing Access Diagram", page 254.*

2. Open the properties dialog box of the writing access area concerned.
3. Select the **Users** tab and disconnect all **Access Area Members** of the writing access area, and connect these **Access Area Members** to another writing access area.
4. Delete the writing access area.

The writing access areas dependent on the deleted writing access area are, after updating, no longer attached to the writing access areas tree. It is therefore

preferable to first delete their links with the obsolete writing access area and attach them to a writing access area that will be retained.

Objects which had this writing access area can be protected with another writing access area. Otherwise, they are considered as being protected at the highest level, with "Administrator" writing access area level.

☛ *For more details on protection of objects, see "Protecting Objects", page 216.*

Propagating Object Writing Access Areas to Child Objects

The **Administration** navigation window of the **HOPEX** workspace allows:

- access to writing access areas
- simple automation of writing access area propagation to connected and child objects.

  ☛ *See "Associating Objects with Writing Access Areas", page 260.*
- connecting an object to a writing access area

You can propagate a writing access area from all objects connected to dependent objects, for a repository of the environment.

☛ *This action can take some considerable time, depending on repository size.*

To propagate a writing access area:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*
2. In your **HOPEX** workspace menu bar, select **View > Navigation Windows > Administration**.
3. In the **Navigation** tab, expand the **Writing Access Area** folder.
4. Right-click the writing access area to be propagated and select **Propagation of writing access area to associated occurrences**.
5. Click **Yes** to confirm.

Associating Objects with Writing Access Areas

The **Administration** navigation window of the **HOPEX** workspace allows:

- access to writing access areas
- simple automation of writing access area propagation to connected and child objects.
- connecting an object to a writing access area

☛ *See "Propagating Object Writing Access Areas to Child Objects", page 260.*

To connect an object to a writing access area:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*
2. In your **HOPEX** workspace menu bar, select **View > Navigation Windows > Administration**.
3. In the **Navigation** tab, expand the **Writing Access Area** folder.

4. Right-click the required writing access area and select **Connect > Object**.
5. In the query dialog box, find the required object and click **OK**.

To display the list of objects associated with a writing access area:

- Right-click the writing access area and select **Objects associated with writing access area**.
  A dialog box displays a list of these objects.

Tips on Using Writing Access Areas

Common data

MEGA recommends that you manage data common to several projects in a specific project. This simplifies control of their evolutions.

Tips

MEGA recommends:

- definition of certain users:
  - with writing access area level higher than projects so as to manage conflicts between projects, for example "Project Management".
  - with writing access area at creation at the level of the project, to avoid creation of objects that cannot be modified by the project.

    ☛ *See "Defining a Writing Access Area at Creation", page 257.*
- that only the "Administrator" user should be connected to the "Administrator" writing access area.
- that if a person produces on several projects, the person should have one **HOPEX** user per project.
  Objects are therefore directly created in the correct owner project, greatly simplifying management.

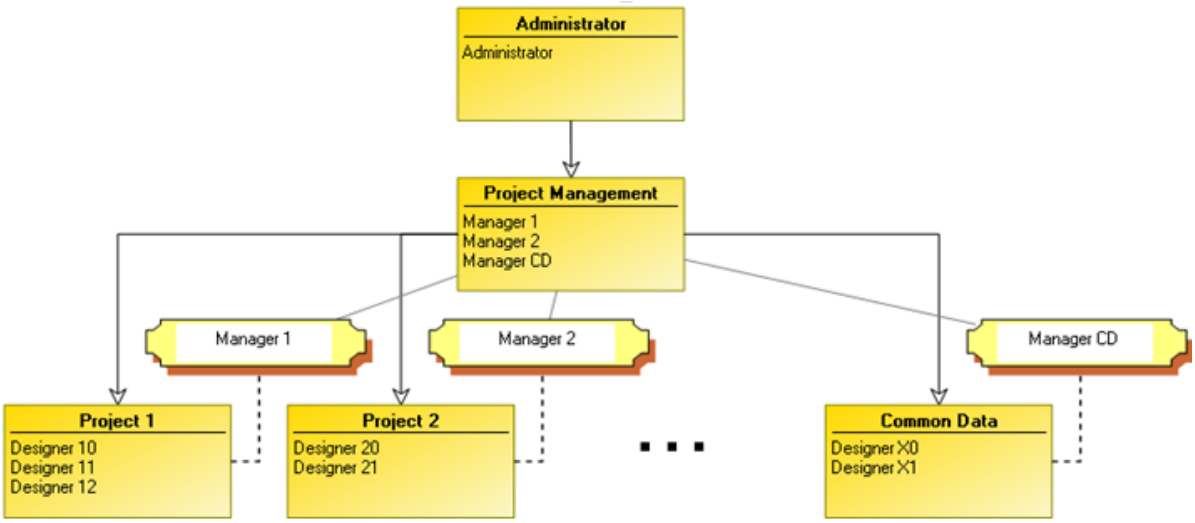
Typical example

The following example presents a typical case of writing access area use:

- Only the Administrator user has "Administrator" writing access area level.
- All managers can modify objects of all projects.
- Objects created by a manager are attached to a dedicated project.

  ☛ *Manager 1 can modify objects of all projects, by default the objects he/she creates are in project 1.*

- Data common to different projects ("Common Data") is managed in a dedicated project with a specific writing access area.



CUSTOMIZING WRITING ACCESS AREA MANAGEMENT

This section describes how to use and customize management of writing access areas:

- "Calculated Writing Access Area", page 263
- "Calculated MetaAttribute", page 263
- "Installing a Writing Access Diagram", page 264
- "Locking Validated Objects", page 265
- "Merging Two Projects", page 266
- "Splitting a Project", page 267

Calculated Writing Access Area

As standard, the writing access area of an object is stored in the "_Authorization" MetaAttribute and takes the value of the writing access area absolute identifier. It is assigned at creation and you can modify it.

You can install up calculated writing access area.

```
For example, you can deduce the writing access area of an
operation from that of the process on which it depends. You
need only change the writing access area of a process, and
those of the dependent operations will automatically adapt.
```

In this case, watch performance.

To customize the writing access area of an object:

- Replace the "_Authorization" MetaAttribute (which carries the object writing access area) by a calculated MetaAttribute.

Calculated MetaAttribute

A calculated MetaAttribute is a software device enabling deduction of the MetaAttribute value of an object as a function of data around the object or dependent on other sources (system, current user, etc.).

HOPEX uses a set of "conventional" MetaAttributes (including the writing access area) that do not require metamodel definition.

A substitution device is available in **HOPEX**; it enables replacement of an implicit MetaAttribute by another for a MetaClass.

This device is required when you need to alter behavior of an existing MetaAttribute by implementing a calculated MetaAttribute.

To customize writing access area of an object, you must:

1. Create a MetaAttribute with characteristics close to those of the "_Authorization" MetaAttribute.

2. Substitute the "_HexaIdAbs" value of the new MetaAttribute by the "_HexaIdAbs" value of the "_Authorization" MetaAttribute.
3. Calculate the writing access area.

Installing a Writing Access Diagram

To install a writing access area diagram in an environment already in production:

1. From **HOPEX Administration**, open the writing access diagram.

   ☛ *See "Opening the Writing Access Diagram", page 254.*

2. Ask users who have a private workspace in progress to close this.
3. Create an "Old Data" writing access area

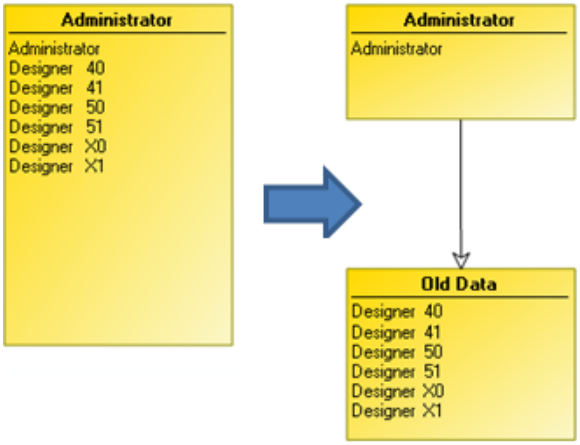
   ☛ *See "Creating a Writing Access Area", page 256.*

4. Attach all users (except the "Administrator" user) and all objects of all repositories to the "Old Data" writing access area.

   ☛ *See "Defining Writing Access Area Persons or Person Groups", page 256.*

   ☛ *See "Associating Objects with Writing Access Areas", page 260.*

   Users may resume working.

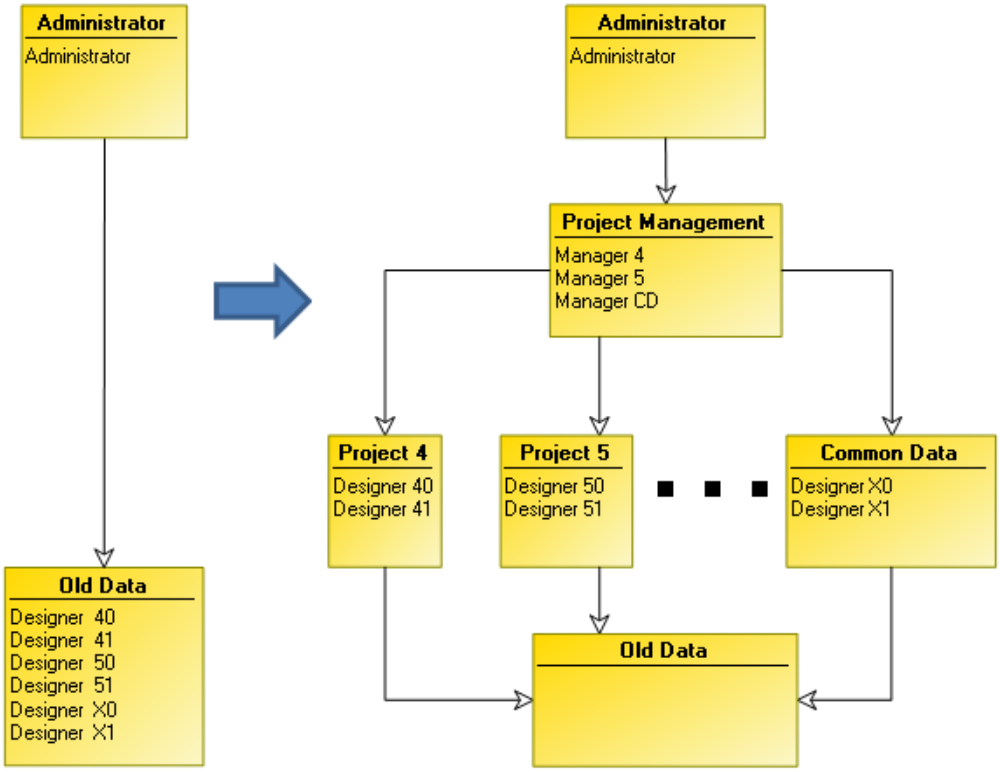


5. Create new writing access areas according to projects.
6. Distribute users between these writing access areas.
7. Distribute objects between these projects/writing access areas for all repositories of the environment.

   ☛ *See "Associating Objects with Writing Access Areas", page 260.*

   Until this distribution is completed, projects can interfere with each other, since they have rights to modify objects created before distribution.

8. (Optional) When all objects of all repositories have been distributed, you can delete the "Old Data" writing access area.



☛ *If the environment is new and there is no data to be distributed between the new writing access areas, you do not need to draw the diagram with the "Old Data" writing access area.*

Locking Validated Objects

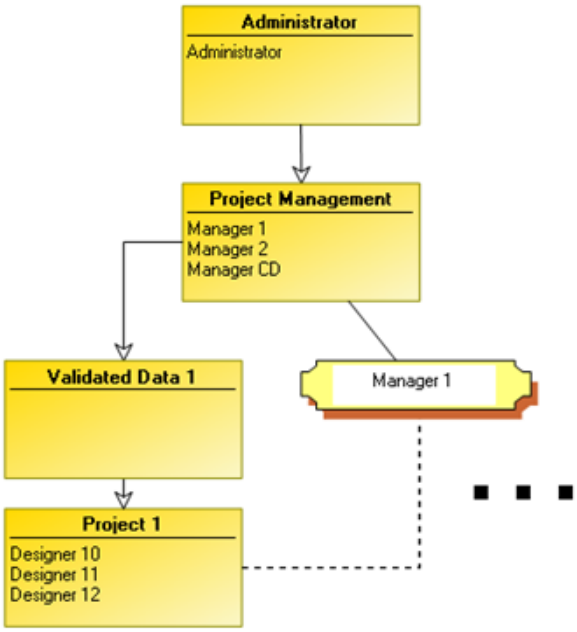
When objects have been validated, you can configure the writing access diagram so that these objects cannot be modified.

To lock objects:

1. From **HOPEX Administration**, open the writing access diagram.

   ☛ *See "Opening the Writing Access Diagram", page 254.*

2. Between the writing access area "Project Management" and that of the project, insert a writing access area dedicated to validated data of the project.



3. When data is validated, modify its writing access area from "Project" level to the higher level "Validated Data" writing access area.

   ☛ *See "Modifying Writing Access Areas of Objects", page 258.*

4. (Optional) If validated data must be modified:
   - it is modified by a user of "Project Management" writing access area level.
   - it is lowered to the "Project" writing access area level.

So the project perimeter is distributed between two writing access areas, but remains perfectly determined.

Merging Two Projects

To merge two projects:

1. From **HOPEX Administration**, open the writing access diagram.

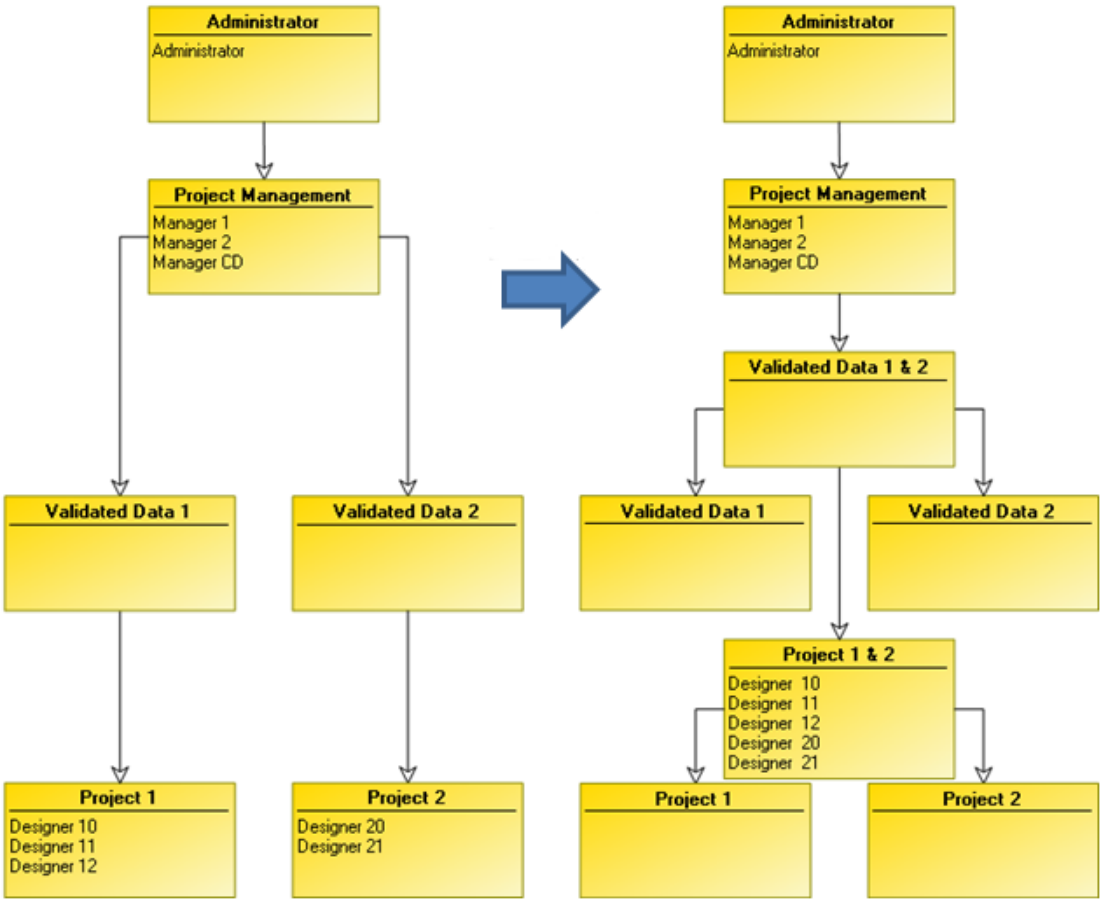
   ☛ *See "Opening the Writing Access Diagram", page 254.*

2. Create a new writing access area for the new project.

   ☛ *This new writing access area must be of higher level than the writing access area it will replace.*

3. Connect users of merged projects to this new writing access area.

   ☛ *See "Defining Writing Access Area Persons or Person Groups", page 256.*



Splitting a Project

To split a project:

1. From **HOPEX Administration**, open the writing access diagram.

   ☛ *See "Opening the Writing Access Diagram", page 254.*

2. Create writing access areas connected to new projects.
   These writing access areas must be of must be of higher level than the writing access areas they will replace.

   ☛ *See "Creating a Writing Access Area", page 256.*

3. Ask users who have a private workspace in progress to close this.

4. Distribute users between these writing access areas.

   ☛ *See "Defining Writing Access Area Persons or Person Groups", page 256.*

   ☛ *Users may resume working.*

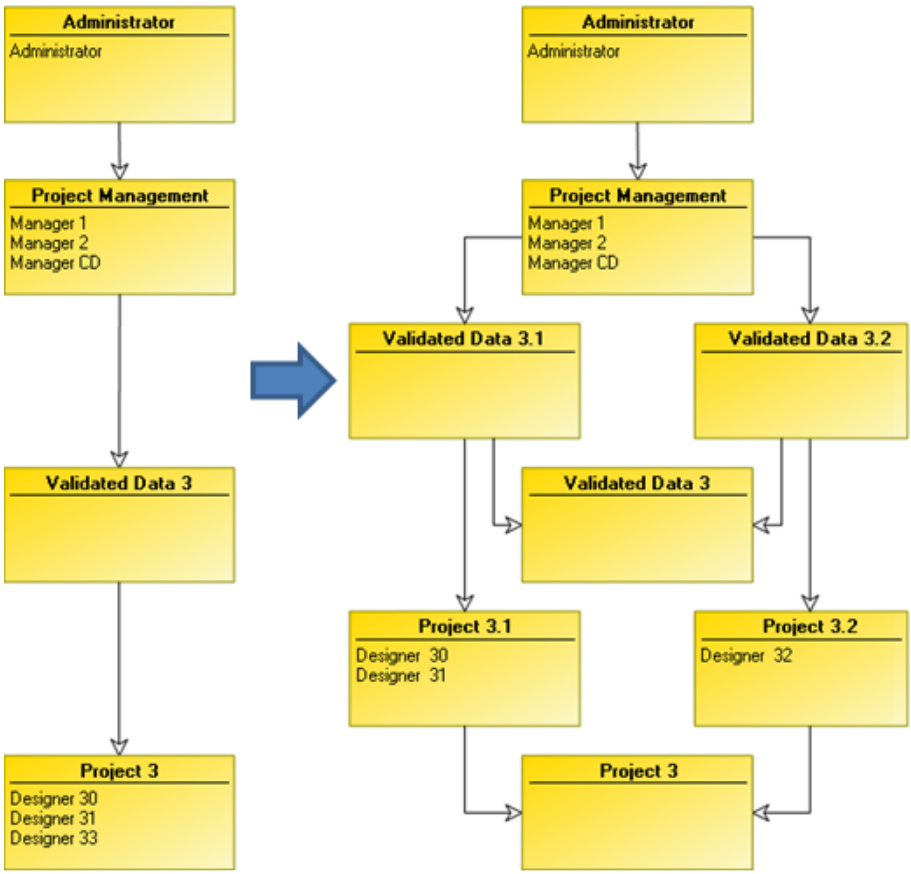
5. Distribute the objects between the new projects.

   ☛ *See "Associating Objects with Writing Access Areas", page 260.*

   Until this distribution is completed, projects can interfere with each other, since they do not yet have rights to modify objects created before distribution.

6. (Optional) Delete the old writing access areas.

   ☛ *(Optional) When all objects of all repositories have been distributed, you can delete old writing access areas.*



MANAGING USERS FROM THE WRITING ACCESS DIAGRAM

This section presents how to:

✓ "Creating Persons with Writing Access Areas", page 269
✓ "Creating Person Group with Writing Access Areas", page 269
✓ "Managing Users from the Writing Access Diagram", page 270
✓ "Compiling the Writing Access Diagram", page 270
✓ "Transferring the Writing Access Diagram", page 270

Creating Persons with Writing Access Areas

At creation, the user is not connected to any writing access area. To implement protection, the person should be connected to a writing access area by creation of a link between person and writing access area.

To create a person with a writing access area, from the writing access diagram:

1. From **HOPEX Administration**, open the writing access diagram.
   ☛ *See "Opening the Writing Access Diagram", page 254.*
2. In the writing access diagram, right-click a writing access area and select **Properties**.
3. In the **Users** tab, click the **New** button .
   A selection dialog box appears.
4. In the **MetaClass** field, select **Person**.
   The **Creation of Person** dialog box opens.
5. Follow the procedure "Defining a Person", page 53.
   The person is created and connected to the selected writing access area.
   ☛ *A person depends on a single writing access area.*

Creating Person Group with Writing Access Areas

To create a person group with a writing access area, from the writing access diagram:

1. From **HOPEX Administration**, open the writing access diagram.
   ☛ *See "Opening the Writing Access Diagram", page 254.*
2. In the writing access diagram, right-click a writing access area and select **Properties**.
3. In the **Users** tab, click the **New** button .
   A selection dialog box appears.
4. In the **MetaClass** field, select **Person Group**.
   A new person group appears in the list of access area members. The new group is connected to the selected writing access area.

5. (optional) Modify the **Short Name** of the new person group.

Managing Users from the Writing Access Diagram

User access rights to repositories and functions can be restricted by the administrator. You can carry out this modification user by user, or on all users simultaneously.

☛ *To manage user access rights, the **HOPEX Power Supervisor** technical module is required.*

To manage users from the writing access diagram:

1. From **HOPEX Administration**, open the writing access diagram.

   ☛ *See "Opening the Writing Access Diagram", page 254.*

2. From the writing access diagram, select **Diagram > Described Object > Manage Users**.
   The **Users** administration dialog box opens.

   ☛ *See "Introduction to User Management", page 26.*

Compiling the Writing Access Diagram

Changes to the writing access diagram only take effect after it has been compiled.

To compile the diagram:

- Select **Diagram > Described Object > Compile Writing Access Diagram**.

If the diagram has been modified, it is automatically compiled at closing. This allows you to check the validity of the user diagram.

When modifying the writing access diagram, so as to warn of rejects due, for example, to authorization restrictions or deletions before compilation it is recommended that:

- all changes made on the user workstations should be uploaded to the administrator workstation, or
- all private workspaces dispatched.

When compilation is complete, a message indicates whether the operation was successful or whether the diagram contains errors. The most frequent errors are:

- A writing access area (other than '"Administrator") is not attached to any other.
- A person or person group is not attached to any authorization.

Transferring the Writing Access Diagram

On workstations where the network is not available, when the user diagram has been updated and compiled it must be exported to the administrator workstation and imported on user workstations. You must also do this if the same diagram is to be used in several different environments.

To run export:

- Select **Diagram > Described Object > Export Writing Access Diagram**.

Import on user workstations is carried out in the normal way.

Customizing Writing Access Diagram Display

You can customize writing access diagram display:

- diagram structure representation
- display of persons connected to a writing access area

Customizing Diagram Structure Representation

You can customize representation of the structure of the writing access diagram using the drawing reorganization function.

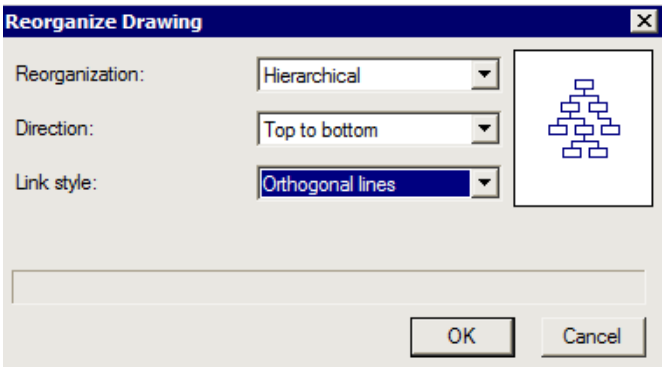
☺ *The automatic drawing reorganization functionality is automatically activated on loading a diagram that does not yet include a drawing.*

To modify organization of an existing drawing:

1. From **HOPEX Administration**, open the writing access diagram.

   ☛ *See "Opening the Writing Access Diagram", page 254.*

2. Select **Drawing > Reorganize Drawing**.
3. Select the desired reorganization mode, the direction and the style of links in the diagram.



☺ *The miniature image alongside the reorganization options gives you a view of each type of reorganization.*

4. Click **OK** to apply the modifications.

Customizing Writing Access Area Display

You can show or hide persons, person groups, objects connected to a writing access area.

To define writing access area display:

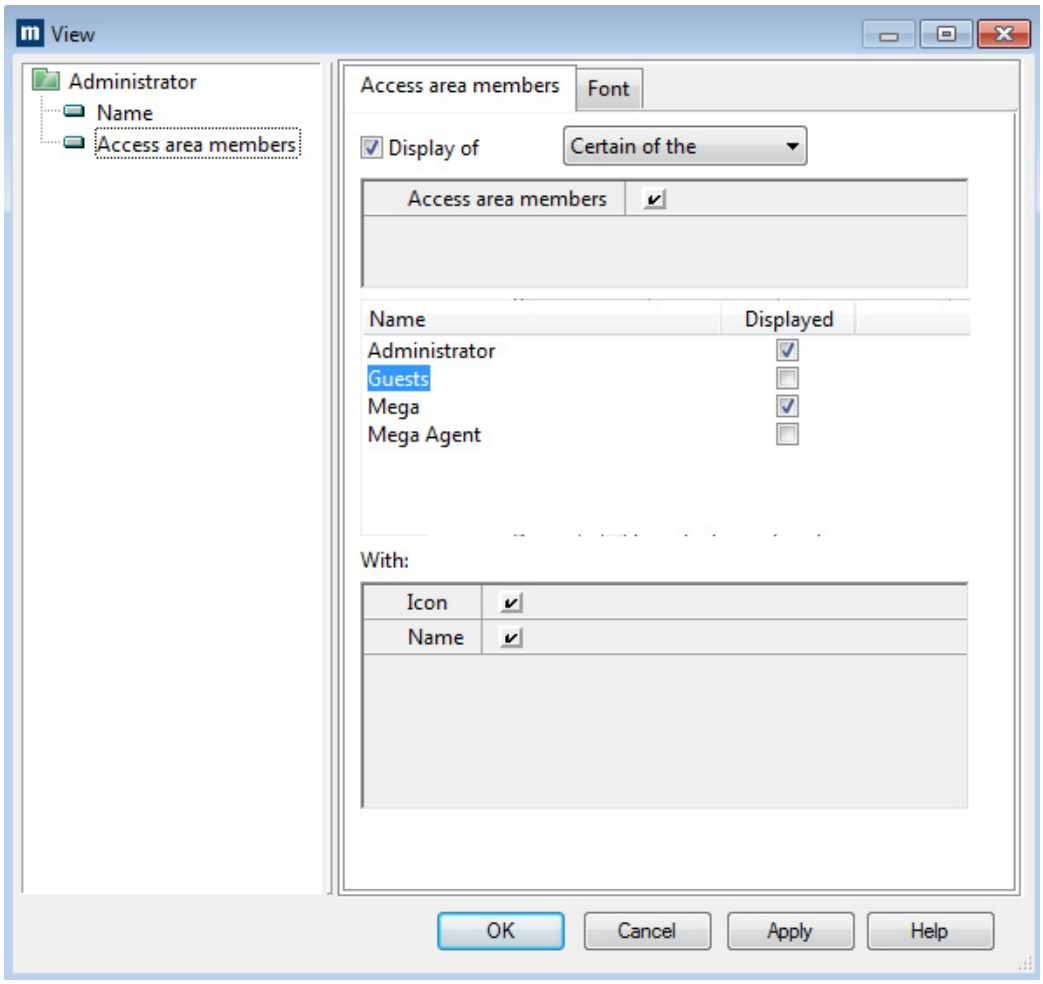
1. From **HOPEX Administration**, open the writing access diagram.

   ☛ *See "Opening the Writing Access Diagram", page 254.*

2. Right-click the writing access area concerned and select **Shapes and Details**.
   The **View** dialog box opens.
3. In the tree on the left, select **Access Area Members**.
4. In the pane on the right, select **Access Area Members**.
5. Select the **Display of** option, then click the arrow and select **Certain of the** access area members.

   ☛ *To display all or none of the access area members, select **All the** or **None of the**.*

6. Select the persons you want to see displayed.



7. Click **OK**.

MANAGING DATA READING ACCESS

The following points are covered here:

- ✓ "Introduction", page 276
- ✓ "Reading Access Area Matrix", page 280
- ✓ "Reading Access Diagram", page 284
- ✓ "Configuring Data Reading Access", page 295
- ✓ "MetaClass Confidentiality Exceptions", page 305

INTRODUCTION

☛ *Managing reading access areas is only available with the **HOPEX Power Supervisor** technical module.*

The following points are detailed here:

- "The Need to Manage Sensitive Data", page 276
- "General Concepts", page 276
- "Activating Data Reading Access Management", page 277
- "Consulting Environment Reading Access Information", page 277
- "Managing Reading Access in HOPEX", page 278
- "Compiling the Reading Access Diagram", page 278

The Need to Manage Sensitive Data

Certain modeling projects may be confidential or contain confidential or sensitive data: costs, risks, controls.

The **HOPEX** administrator may therefore need to mask objects corresponding to confidential or sensitive data.

These objects must be visible only to authorized users.

To meet these requirements concerning data confidentiality, **HOPEX** offers functionalities for implementing consistent and effective confidentiality policies.

General Concepts

To implement a data confidentiality policy, objects must be organized in distinct sets. Each set of objects is a ***reading access areas***.

🕮 *A user is a person with a login.*

🕮 *A person can belong to a group. A user group is a group of persons with a login.*

Each user or group of users is associated with a reading access area. It is the person or person group that carries the reading access area.

The reading access area to which the person or person group belongs determines the objects that the user or group of users can see. A user or user group can only see objects located in his/her own or lower confidentiality areas.

With definition of reading access areas, hidden objects are inaccessible. This concept differs from that of the filter, which hides occurrences of MetaClasses so as not to disturb the final view of the user, see "Managing UI Access (Permissions)", page 233.

Activating Data Reading Access Management

When you activate reading access management, confidential data is visible only to authorized users. Before activating reading access management, **MEGA** recommends that you familiarize yourself with reading access management using **MEGA**.

☛ *For more details on confidential data, see "Confidential or Sensitive Object Behavior", page 301.*

To manage confidential or sensitive data , you must first activate data reading access area management.

To activate data reading access management:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. In the **User Management** folder, right-click **Data Reading Access** and select **Activate Management**.

   A message warns you that activation of reading access management is irreversible.

3. Carefully read this warning message, then click **Yes** if you wish to activate data reading access management.

Consulting Environment Reading Access Information

When working in **HOPEX** you can check:

- if reading access management is activated in your environment or not.
- the reading access area to which the connected user belongs.

To consult reading access information of your current environment:

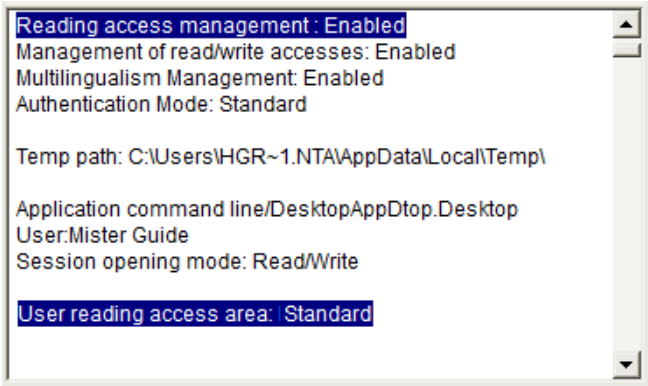
1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. In the **HOPEX** menu bar, select **Help > About HOPEX**.
   The **About HOPEX** dialog box appears.
3. Click **System Information**.
4. In the menu bar of the **System Properties** dialog box, select **System Info > Edit**.
   The Megasys.txt text file opens.

At the beginning of the file you can consult the properties of:

- ***Reading access management***
- ***Reading access area of the user***



Managing Reading Access in HOPEX

You can set up and manage data reading access in **HOPEX** in two ways:

Reading access area matrix method:

1. Create the different user reading access areas you require.
2. Distribute persons or person groups in user reading access areas.
3. Distribute objects in object reading access areas.
4. Associate user reading access areas with object reading access areas.

☛ *For more details, see "Reading Access Area Matrix", page 280.*

Reading access diagram method:

1. Define organization and hierarchy of the different reading access areas you require.
2. Create and organize these users in a reading access diagram.
3. Associate persons or person groups with different reading access areas. Objects created by users are then distributed in the user reading access area.

☛ *For more details, see "Reading Access Diagram", page 284.*

Compiling the Reading Access Diagram

Running the reading access diagram compilation ensures consistent **HOPEX** behavior with diagram declarations.

If the diagram is not compiled, there is a risk that certain users will be able to see objects that are normally hidden.

To compile the reading access diagram:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **User management** folder of the environment.
3. Right-click the user **Data Reading Access** folder and select **Compile**.
   On completion of compilation, a message indicates the result of the operation.

READING ACCESS AREA MATRIX

The reading access area matrix enables organization of user groups with object groups. Only a user with administrator profile connected to the maximum reading access area can configure reading access areas.

In the reading access area matrix, you can create two types of reading access areas:

- an **object reading access area** , grouping only **HOPEX** objects.
- a **user reading access area** , grouping only persons or person groups

  The user reading access area corresponds to the view the person or person group has of the repository: it defines objects that can be accessed by the person or person group.

  To ensure coherence of the reading access diagram, if you begin managing reading access of your data from the reading access area matrix, you must continue to manage reading access from this matrix.

You can create links between these two types of reading access area.

Accessing the Reading Access Area Matrix

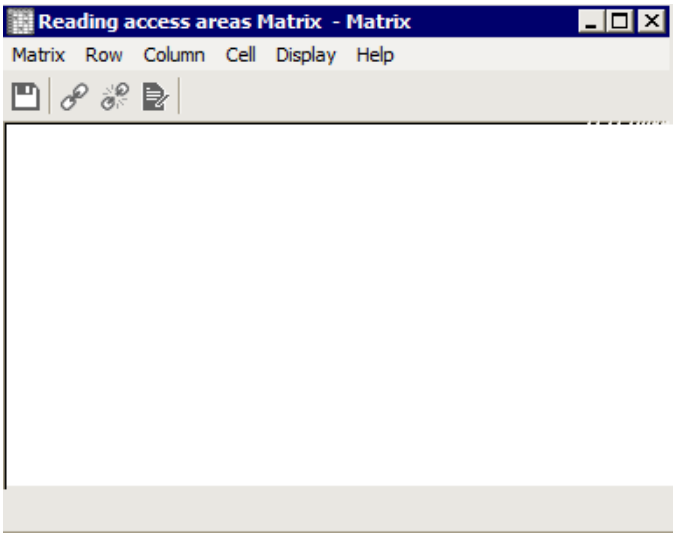
To access the reading access area matrix:

1. From **HOPEX Administration**, connect to the environment concerned.

   See "Connecting to an Environment", page 7.

2. Expand the **User management** folder of the environment.

3. Right-click the user **Data Reading Access** folder and select **Reading Access Matrix**.
   An empty reading access area matrix appears.



Adding an Object Reading Access Area

To add an object reading access area in the matrix:

1. Open the reading access area matrix.
   ☛ *See "Accessing the Reading Access Area Matrix", page 280.*
2. Select **Row > Create**.
3. In the creation window, enter the name of the object reading access area and click **Finish**.

Adding a User Reading Access Area

To add an user reading access area in the matrix:

1. Open the reading access area matrix.
   ☛ *See "Accessing the Reading Access Area Matrix", page 280.*
2. Select **Column > Create**.
3. In the creation window, enter the name of the user reading access area and click **Finish**.

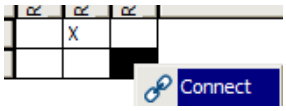
Associating User Reading Access Areas with Object Reading Access Areas

To associate a user reading access area with an object reading access area

1. Open the reading access area matrix.

   ☛ *See "Accessing the Reading Access Area Matrix", page 280.*

2. In the reading access area matrix, right-click the cell at the intersection of the user reading access area and the object reading access area and select **Connect**.



A cross represents the association between the two selected reading access areas. The corresponding links are automatically drawn in the reading access diagram.

If you begin management of reading access of your data from the reading access area matrix, you must continue management of reading access from this matrix. Do not manually modify links created automatically in the reading access diagram; you may invalidate the diagram.

☛ *For more details on the reading access diagram, see "Reading Access Diagram", page 284.*

Associating Users with User Reading Access Areas

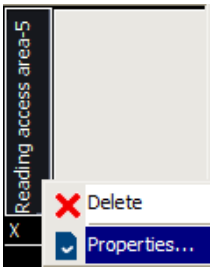
In the case of the reading access area matrix, to associate a user (or user group) with a reading access area:

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **User management** folder of the environment.
3. Right-click the user **Data Reading Access** folder and select **Reading Access Matrix**.
   The reading access area matrix appears.

4. Right-click the user reading access area concerned and select **Properties**.



The user reading access area properties window appears.

5. In the **Persons** tab, click **Connect**.

   ☛ *If you want to create a new Person (or new person group) and associate it with the reading access area, click **New**.*

6. In the query tool, click the arrow in the first field and select **Person** (or **Person Group**).

   ☛ *If you want to select persons and person groups, select Access Area Member.*

7. (optional) In the second field, enter the character string to be queried.
8. Click **Find** .
9. In the query result list, select the person (or person group) required and click **Connect**.

   ☛ *Press the [CTRL] key to select several persons and/or person groups simultaneously.*

The user (or user group) you have connected appears in the list of users in the selected reading access area.

READING ACCESS DIAGRAM

The *reading access diagram* enables organization of the repository by sets of objects, unlike the writing access diagram which enables organization by work groups.

> *The reading access diagram enables definition of reading access areas and their hierarchical organization. This diagram also enables creation of users and their association with reading access areas.*

The reading access diagram can be opened only by an Administrator profile user connected to the maximum reading access area.

> *The reading access diagram feature is accessible only if you have the **HOPEX Power Supervisor** technical module.*

> **If you begin management of reading access of your data from the reading access area matrix, you must continue to manage reading access from this matrix. Do not manually modify links created automatically in the reading access diagram, you might invalidate the diagram.**

> **Warning: on exiting the reading access diagram, if a message indicates that the diagram is incorrect, the diagram is not compiled and reading access management does not operate. MEGA recommends that you correct the error that prevents diagram compilation.**

This section covers the following points:

- "Reading Access Diagram Operating", page 284
- "Activating the reading access diagram", page 286
- "Prohibiting Reading Access Diagram Modification", page 286
- "Opening the reading access diagram (Windows Front-End)", page 287
- "Organizing Reading Access Areas", page 287
- "Adding a User in the Reading Access Diagram", page 289
- "Connecting Users to Reading Access Areas", page 289
- "Consulting Reading Access Diagram Information:", page 292
- "Customizing Reading Access Area Display", page 293

Reading Access Diagram Operating

The reading access diagram implements reading access areas of **General** type. These areas can group both objects and persons and/or person groups.

Reading access areas are organized hierarchically. **MEGA** provides two extreme reading access areas:

- **Maximum Reading Access**, the highest reading access level.
- **Standard**, the lowest reading access level.

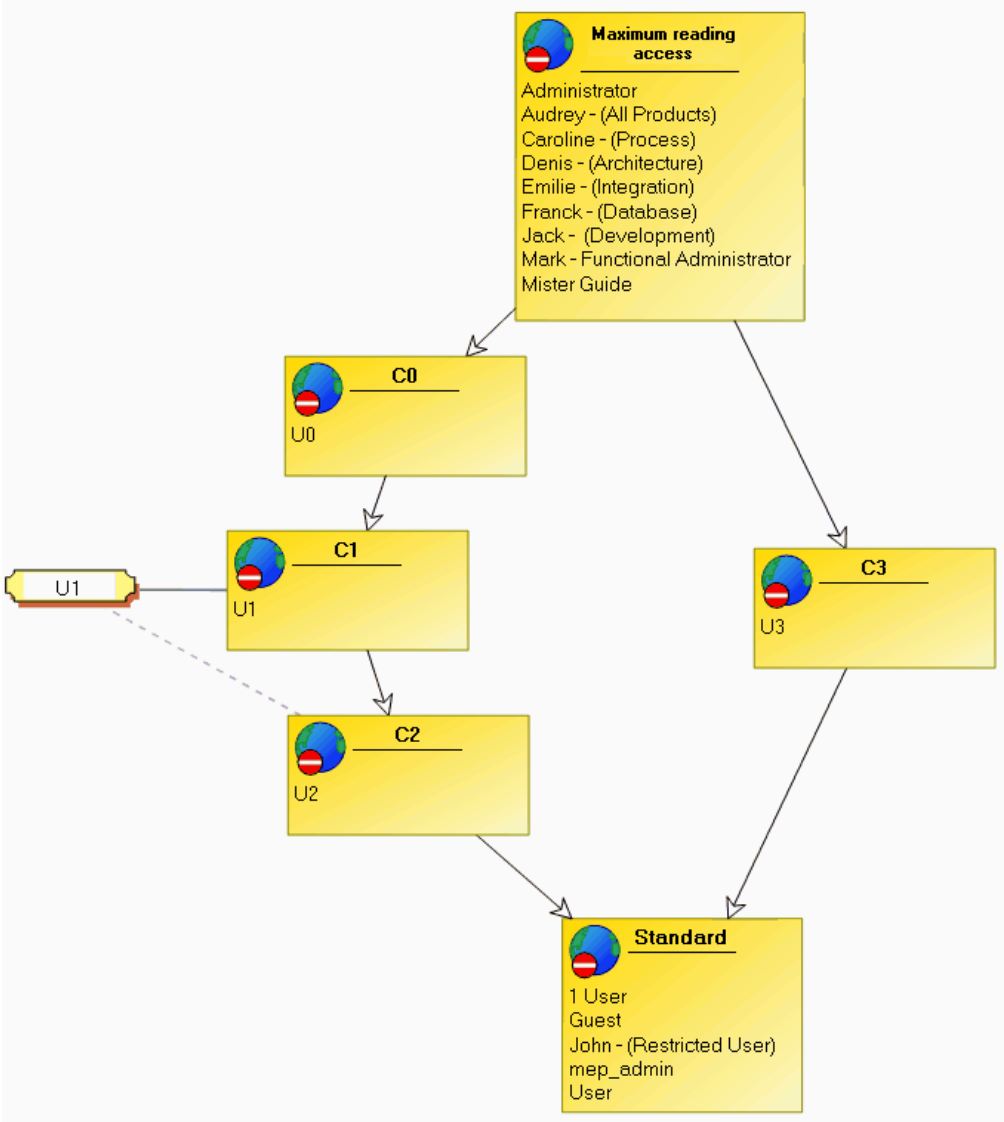
Each object belongs to a reading access area (**Standard** by default).

Each person or person group is connected to one of the reading access areas. The persons or person groups that are connected to:

- **Maximum reading access** sees all repository data.
- A created reading access area sees all data of this area, as well as that of lower level reading access areas.
- **Standard** sees only non-confidential data of the repository.

For example, a user U1 connected to a reading access area C1 sees all objects that belong to:

- his/her reading access area (C1)
- lower level reading access areas (C2 and Standard).



In the preceding reading access diagram, user U1:

- is connected to reading access area C1
- to a reading access area at creation C2.

When user U1 creates an occurrence of a MetaClass:

- if sensitive (high sensitivity), this belongs to reading access area C1.
- if non-sensitive (standard sensitivity), this belongs to reading access area C2.

☛ *For more details on MetaClass sensitivity, see "Managing MetaClass Sensitivity and Reading Access Areas", page 299.*

If a user does not have a reading access at creation, any occurrences of a non-sensitive MetaClass he/she creates belong to the standard reading access area.

However, Web sites and reports (MS Word) are always created at the reading access level of the user. This ensures confidentiality of the information they may contain.

Users connected to reading access area C3 cannot see objects belonging to reading access areas C0, C1, and C2, since area C3 does not belong to the same hierarchical branch as areas C0, C1, and C2.

Activating the reading access diagram

To be able to access the reading access diagram, you must first activate the reading access diagram option.

☛ *By default, only the reading access area matrix is accessible.*

To activate the reading access diagram option:

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Right-click the environment and select **Options > Modify**.
3. In the **Repository** group of options, select the **Activate the reading access diagram** option.
4. Click **OK**.
   The reading access diagram option is activated. The reading access diagram is accessible.

Prohibiting Reading Access Diagram Modification

By default, reading access diagram modification is authorized.

To prohibit reading access diagram modification:

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Right-click the environment and select **Options > Modify**.
3. In the **Repository** options group, for option **Authorize modification of writing access and reading access diagrams** select "Prohibit".
4. Click **OK**.
   Reading diagram modification is prohibited.

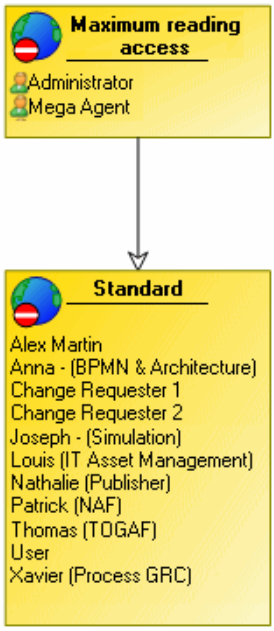
Opening the reading access diagram (Windows Front-End)

To access the reading access diagram:

1. From **HOPEX Administration**, connect to the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **User management** folder of the environment.
3. Right-click the **Data Reading Access** folder and select **Open Diagram**. The reading access diagram appears.



By default, the reading access diagram contains two reading access areas:

- **Maximum Reading Access** is the highest reading access area level. Users connected to this area can see all objects in the repository.
- **Standard** is the lowest reading access area level.

  ☛ *There can only be one maximum level and one minimum level (standard) reading access area in the diagram.*

Organizing Reading Access Areas

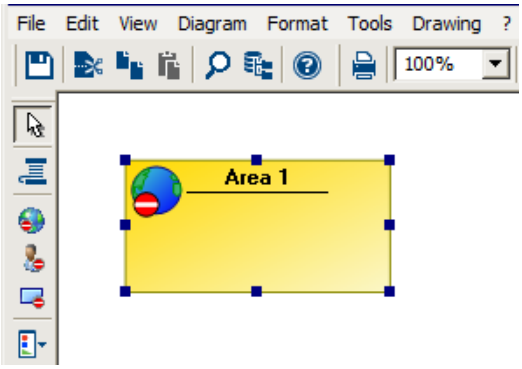
Creating reading access areas

To create a *reading access area* in the diagram:

1. Open the reading access diagram.

   ☛ *See "Opening the reading access diagram (Windows Front-End)", page 287.*

2. In the diagram insert toolbar, click **General Reading Access Area** , then click in the diagram.
3. In the creation wizard dialog box that appears, enter the name of the reading access area and click **Create**.
   The creation wizard allows you to modify the reading access area type if necessary.
4. Click **Finish**.
   The general reading access area appears in the diagram.



Connecting two reading access areas

Reading access areas must be hierarchically interlinked. Except for **Maximum Reading Access** and **Standard** reading access areas, each reading access area must be connected to a lower level area and a higher level area.

To connect two reading access areas:

1. Open the reading access diagram.

   ☛ *See "Opening the reading access diagram (Windows Front-End)", page 287.*

2. In the diagram insert toolbar, click **Link** and draw a link between the two reading access areas (from the higher level reading access area to the lower level reading access area).

Displaying reading access areas associated with a reading access area

To display object (or user) reading access areas associated with a reading access area:

1. Open the reading access diagram.

   ☛ *See "Opening the reading access diagram (Windows Front-End)", page 287.*

2. Open the properties dialog box of the reading access area in question.
3. Select the **Matching Object reading access areas** or **Matching User reading access areas**.
   The associated object reading access areas or user reading access areas are listed.

Adding a User in the Reading Access Diagram

You can add a person or person group in the reading access diagram.

Adding a person in the reading access diagram

To add a person in the reading access diagram:

1. Open the reading access diagram.

   ☛ *See "Opening the reading access diagram (Windows Front-End)", page 287.*

2. In the diagram insert toolbar, click **Person** , then click in the diagram.

   ☛ *If the **Person** icon is not present in the insert toolbar, add it from the **View > Views and Details** menu.*

   The **Add Person** dialog box appears.

3. In the **Name** field, click the arrow to find the person then click **Connect**.

   ☛ *To add a new person, in the **Name** field, enter the name of the person then click **Create**. Also create the login of the person.*

Adding a person group in the reading access diagram

To add a person group in the reading access diagram:

1. Open the reading access diagram.

   ☛ *See "Opening the reading access diagram (Windows Front-End)", page 287.*

2. In the diagram insert toolbar, click **Person Group** , then click in the diagram.

   ☛ *If the **Person Group** icon is not present in the insert toolbar, add it from the **View > Views and Details** menu.*

   The **Add Person Group** dialog box appears.

3. In the **Name** field, click the arrow to find the person group then click **Connect**.

   ☛ *To add a new person group, in the **Name** field, enter the name of the person group then click **Create**. Also create the login of the person group.*

Connecting Users to Reading Access Areas

A user can:

- be connected to a ***reading access area***
  This area defines the view the user has of the repository and the objects the user can access.
- have a ***reading access area at creation***
  Occurrences created by the user belong to this reading access area at creation.

  ☛ *If a user does not have a reading access area at creation, the occurrences he/she creates will belong to the standard reading access area.*

Reading access area of the user

To connect a user to a reading access area:

1. Open the reading access diagram.

   ☛ *See "Opening the reading access diagram (Windows Front-End)", page 287.*

2. Right-click the reading access area concerned and select **Properties**.
3. Select the **Persons** tab and click **Connect**.

   ☛ *If you want to create a new user and associate him/her to a reading access area, select **New > Person**.*

   The access area member search dialog box appears.

4. In the first query field, select the type of access area member you wish to connect: **Person** (or **Person Group**).

   ☛ *If you want to connect persons and person groups, select Access Area Member.*

5. (optional) In the second query field, enter the character string to be queried.
6. Click **Find**.
7. In the result list, select the person (press the [CTRL] key to select several) and click **Connect**.
   The user appears in the reading access area.

Reading access area at creation

You can assign a reading access area at creation to an existing user from:

- the reading access diagram
- the **Properties** dialog box of the person

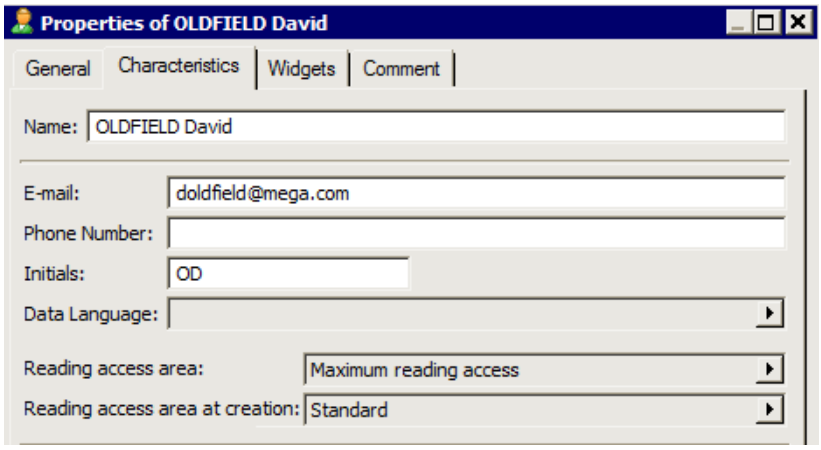
To assign a reading access area at creation to a user from the reading access diagram:

1. Place the user in the reading access diagram.
2. Connect the desired reading access area at creation to the person. This area must be at a level lower than or the same as the reading access area of the user.
   A dialog box asks you to select the type of link to be created: **Access area member** or **Access area member at creation**.
3. Select the link type **Access area at creation member**.
   This area is the reading access area at creation of the user.

To assign a reading access area at creation to a user from the user properties window:

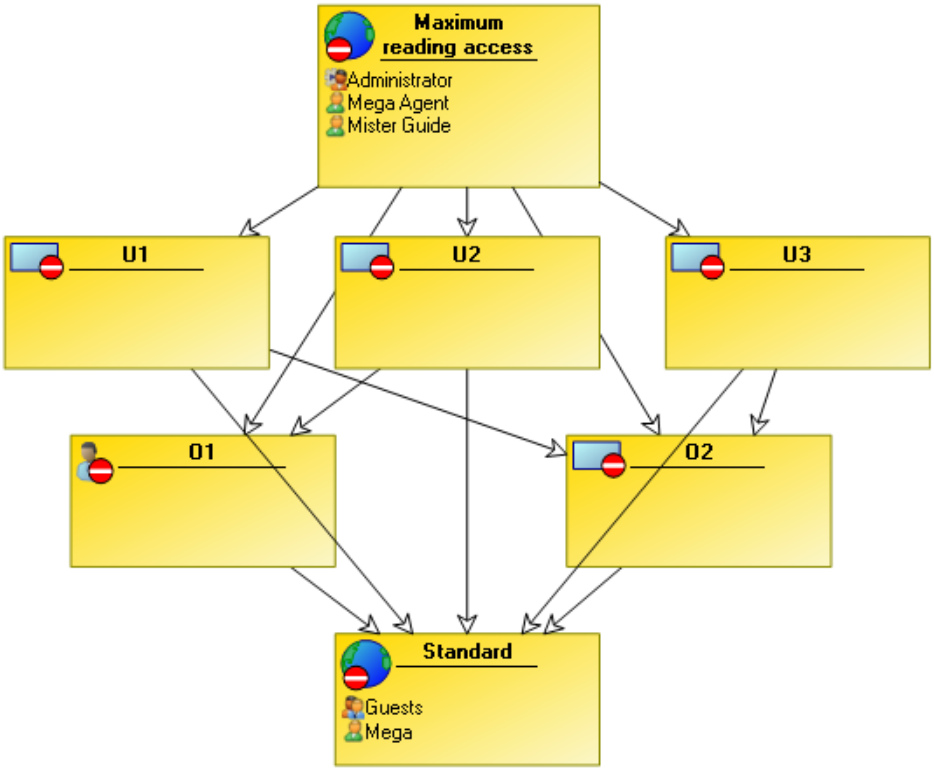
1. Open the **Properties** dialog box of the person and select the **Characteristics** tab, see *"Modifying User Properties", page 57*.

2. In the **Reading access area at creation** field, select the required value.



Consulting Reading Access Diagram Information:

At reading access diagram compilation, user and object are connected to the ***Maximum reading access*** reading access area and the ***Standard*** reading access area.



Open the user reading access area properties dialog box to consult:

- the list of persons connected to the area and to connect new users (**Persons** tab)
- the list of reading access areas for associated objects (**Matching Object Reading Access Areas** tab)

Open the object reading access area dialog box to consult:

- the list of user reading access areas associated with an object reading access area (**Matching User Reading Access Areas** tab)

Customizing Reading Access Area Display

By default, when you create a reading access area, all users belonging to this area are displayed in the diagram reading access area. You can decide to hide certain users in this reading access area.

To define users displayed in their reading access area:

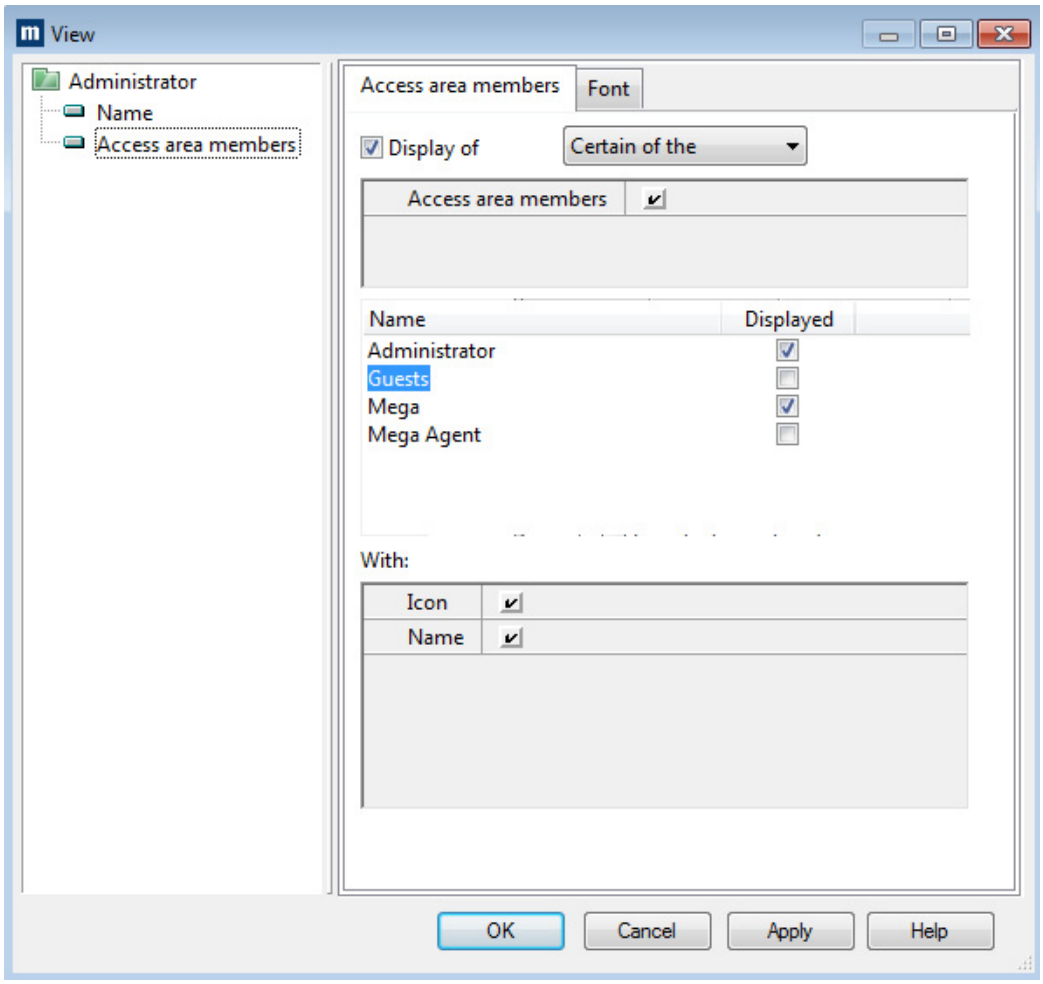
1. From **HOPEX Administration**, open the reading access diagram.

   ☛ *See "Opening the reading access diagram (Windows Front-End)", page 287.*

2. Right-click the reading access area concerned and select **Shapes and Details**.
   The **View** dialog box opens.
3. In the tree on the left, select **Access Area Members**.
4. In the pane on the right, select **Access Area Members**.
5. Select the **Display of** option, then in its drop-down menu select **Certain of the** *Access area members*.

   ☛ *To display all or none of the access area members, select **All the** or **None of the**.*

6. Select the persons you want to see displayed.



7. Click **OK**.

CONFIGURING DATA READING ACCESS

In the **HOPEX** desktop, the navigation window allows access to certain data reading access functions.

This section presents how to:

- "Associating Objects with Reading Access Areas", page 295
- "Associating user reading access areas with objects", page 296
- "Propagating Reading Access Areas", page 297
- "Managing MetaClass Sensitivity and Reading Access Areas", page 299
- "Confidential or Sensitive Object Behavior", page 301
- "Modifying Reading Access Areas", page 303

Associating Objects with Reading Access Areas

The **Administration** navigation window of the **HOPEX** desktop allows access to general and object reading access areas.

> ☛ *To access content of the **Administration** navigation window, you must have **Advanced** or **Expert** metamodel access (see "Configuring Metamodel Access", page 60).*

The reading access areas tree is used to:

- connect objects to a given reading access area
- quickly propagate the reading access area to child objects

This tree simplifies management of propagation of reading access areas (see "Propagating a reading access area from HOPEX", page 298). Its advantage compared with the classic propagation tool is that the propagation trace is kept thanks to the link.

Connecting objects to object reading access areas

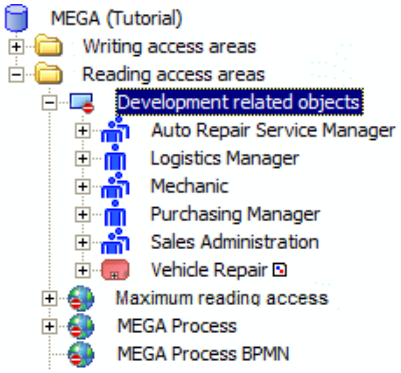
To connect an object to an object reading access area:

1. Connect to **HOPEX**.

   > ☛ *See "Connecting to HOPEX", page 136.*

2. Open the **Administration** navigation window.
3. Expand the **Data Reading Access** folder.
4. Right-click the object reading access area concerned and select **Connect Object**.

5. In the query dialog box, find and select the desired objects and click **OK**. The objects are connected to the object reading access area and appear in the tree.



Disconnecting objects from reading access areas

To disconnect an object connected to a reading access area:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. Open the **Administration** navigation window.
3. Expand the folder of the reading access area concerned.
4. Right-click the object you want to disconnect and select **Disconnect**. The object disappears from the tree.

Displaying the list of objects associated with a reading access area

To display the list of objects associated with a reading access area:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. Open the **Administration** navigation window.
3. Expand the **Data Reading Access** folder.
4. Right-click the reading access area concerned and select **Objects associated with reading access area**.
   A dialog box displays a list of these objects.

Associating user reading access areas with objects

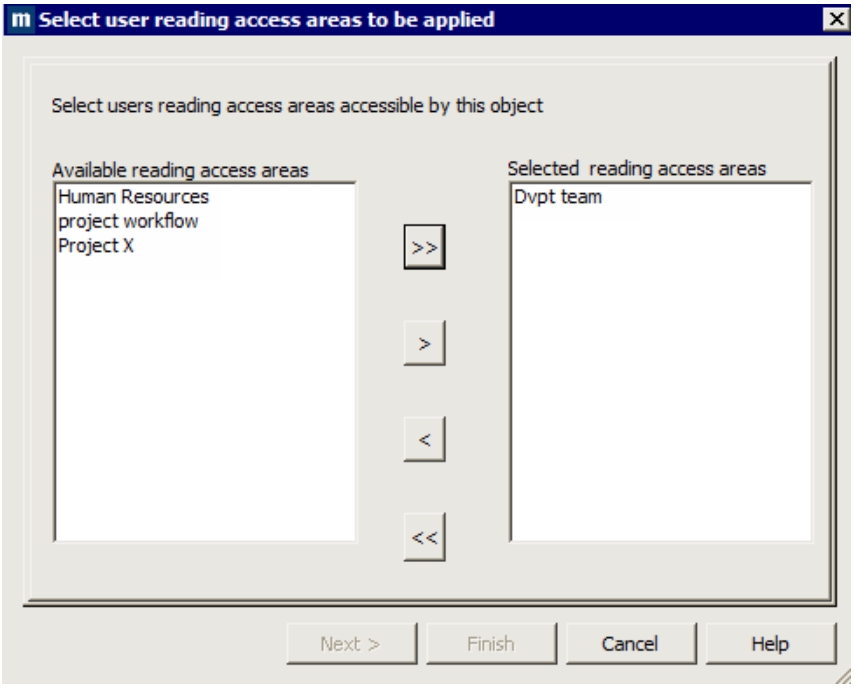
To associate an object with one or several groups of users, proceed as follows:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. Select an object.
3. In the object properties dialog box, select the **General** tab, then the **Administration** subtab.

4. In the **Reading Access Area** drop-down list, click the arrow and select **Associate User Reading Access Areas**.
5. Using the arrows in the selection dialog box, move the user user reading access areas from available to selected, then click **Next**.



6. In the next dialog box, if an object reading access area corresponds to the user reading access areas, you are invited to validate this area, otherwise you must enter the name of a new user reading access area, which will be created corresponding to the previously selected user reading access areas.
7. Click **Finish**.

If you had to create a new object reading access area, this is automatically added in the reading access diagram and is connected to the corresponding user reading access area or areas, as well as to the ***Standard*** reading access area.

Propagating Reading Access Areas

A reading access area can be propagated to objects connected to a given object. You can propagate reading access areas from:

- **HOPEX Administration**
- **HOPEX**.

☛ *The propagation trace is kept when you propagate from **HOPEX**.*

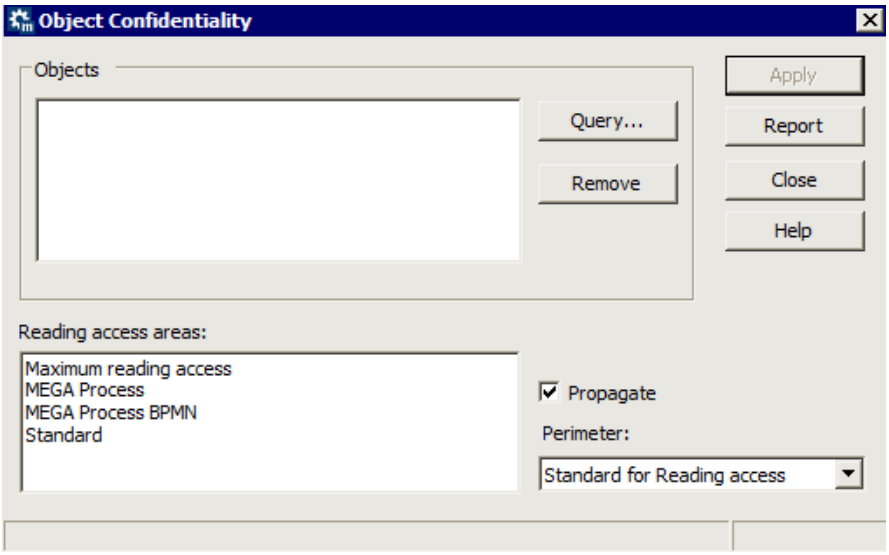
Propagating a reading access area from HOPEX Administration

To propagate a reading access area:

1. Connect to **HOPEX Administration** and select the repository concerned.

   ☛ *See "Accessing Repositories", page 74.*

2. Right-click the repository concerned and select **Object Management > Object Confidentiality Setting**.



3. In the **Object Confidentiality** dialog box that opens, click **Query**.
4. Select the desired objects using the query tool, then click **OK**.
5. In the **Object Confidentiality** dialog box, in the **Reading Access Areas** frame , select the reading access area you want to apply to the selected objects.
6. (Optional) Select **Propagate** to propagate the reading access area to sub-objects.
7. Click **Apply**.

   ☛ *The operator used to propagate reading access areas is the "Standard for reading access" operator.*

Propagating a reading access area from HOPEX

To propagate a reading access area:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. Open the **Administration** navigation window.
3. In the **Administration** navigation window, right-click the reading access area you wish to propagate and select **Propagation of reading access area to associated occurrences**.
   A dialog box may warn you of the presence of propagation conflicts.

   ☛ *Conflicts can arise if child objects are already connected to a different reading access area. You are informed that propagation stops*

at this child object (the physical link that connects the object to a reading access area is stronger than the reading access attribute value)

☛ *The operator used at reading access area propagation is the "Standard for reading access" operator.*

☛ *Alternatively, you can use the confidentiality area propagation tool via the menu **Tools > Manage > Object Confidentiality Setting**.*

Managing MetaClass Sensitivity and Reading Access Areas

☛ *The attribute enabling configuration of MetaClass sensitivity is accessible only if you have the **HOPEX Power Supervisor** technical module.*

In the reading access management frame:

- a user is connected to a reading access area that defines all the objects he/she can see.

  ☛ *See "Connecting Users to Reading Access Areas", page 289.*

- an object type (MetaClass) is characterized by its sensitivity.

A MetaClass can be of sensitivity:

- **standard** (default value)
  Occurrences of the MetaClass created by a user belong to the user reading access at creation area or the Standard reading access area if the user does not have a reading access at creation area.
- **High**
  Occurrences of the MetaClass created by a user belong to the reading access area of the user that creates them.

  ☛ *To modify the sensitivity of a MetaClass, you must have rights to modify **HOPEX** data. The option "Authorize HOPEX Data Modification" must be activated at environment level, see "Managing Options", page 323.*

You can modify sensitivity value of the MetaClass.

☛ *See "Modifying MetaClass sensitivity", page 300.*

☛ *By default, a MetaClass is **Standard** sensitivity.*

Opening the HOPEX MetaClasses reading access configuration dialog box

To open the **HOPEX** MetaClasses reading access configuration dialog box:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Expand the **User management** folder of the environment.

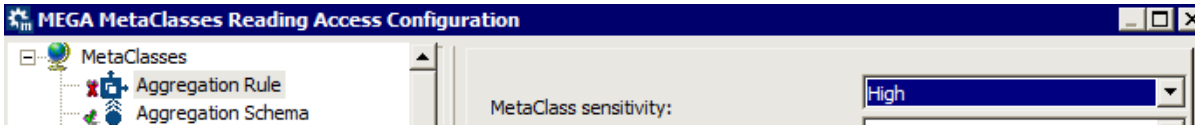
3. Right-click the **Data Reading Access** folder and select **Configure HOPEX MetaClasses for reading access**.
   The **HOPEX MetaClasses Reading Access Configuration** dialog box appears, listing the available **MetaClasses**.
   The icon alongside the name of each MetaClass indicates that default values:
   - have been modified
   - are retained.

Modifying MetaClass sensitivity

To modify MetaClass sensitivity:

1. Open the **HOPEX** MetaClasses reading access configuration dialog box.

   ☛ *See "Opening the HOPEX MetaClasses reading access configuration dialog box", page 299.*
2. In the list of **MetaClasses**, select the desired MetaClass.
3. In the right pane, select the MetaClass sensitivity value:



☛ *A red cross alongside the name of the MetaClass indicates that at least one attribute of the MetaClass has no longer its default value.*

4. Click **OK**.

   Modifications carried out may be canceled when your environment is updated. Remember to back up your extensions (metamodel and technical data).

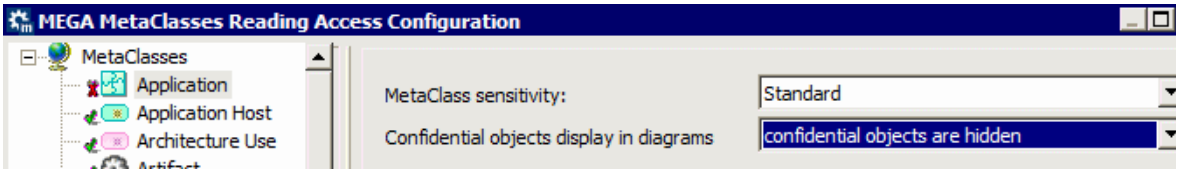
Hiding confidential or sensitive objects in a diagram

So as not to distort a diagram, confidential or sensitive objects are visible by default.

To hide confidential or sensitive objects in a diagram:

1. Open the **HOPEX** MetaClasses reading access configuration dialog box.

   ☛ *See "Opening the HOPEX MetaClasses reading access configuration dialog box", page 299.*
2. In the list of **MetaClasses**, select the desired object.
3. In the right pane, in the **Confidential objects display in diagrams** box, select "Confidential objects are hidden".



☛ *A red cross alongside the name of the MetaClass indicates that the MetaClass default value has been modified.*

4. Click **OK**.

> ☛ *These modifications are not taken into account in the metamodel until the metamodel is compiled. Compile the metamodel before closing the **HOPEX Administration** module, see "Compiling an Environment", page 177.*

Objects corresponding to this MetaClass will be hidden in the diagram.

> ☛ *When you select "Confidential objects are visible", objects corresponding to this MetaClass appear grayed in the diagram and you cannot access information relating to these objects.*

> **Modifications carried out may be canceled when your environment is updated. Remember to back up your extensions (metamodel and technical data).**

Confidential or Sensitive Object Behavior

A confidential object is inaccessible to a user that does not have access to the corresponding reading access area.

It is as if the object did not exist, the object:

- does not appear in lists.
- is ignored in query results.
- does not appear in reports (MS Word) or Web sites.
- is not exported, duplicated, deleted or backed up, and its possible "children" (operations of a process, for example) are considered as orphans.
- only appears when another object is created with the same name or when a higher level object with a lower reading access level is deleted.
- cannot be modified.

> ☛ *Reading access management is not supported in **HOPEX Database Builder**.*

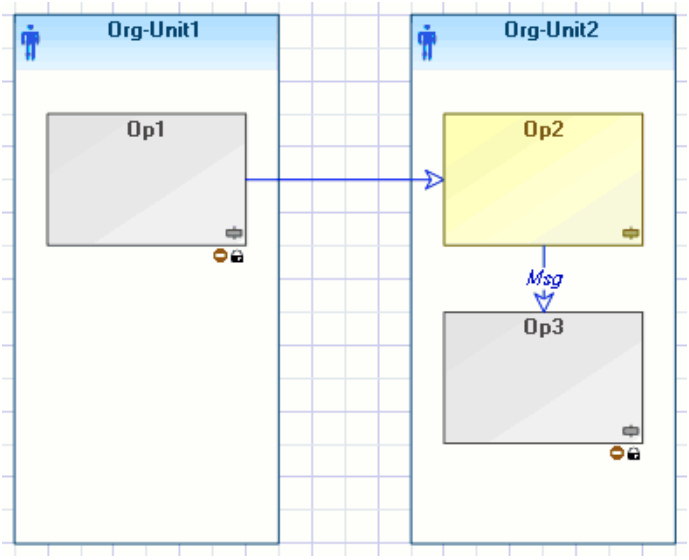
For more details on reading access areas, see "Managing Reading Access in HOPEX", page 278.

Displaying a confidential or sensitive object in a diagram

By default, confidential or sensitive objects are visible in diagrams.

> ☛ *To hide confidential or sensitive objects, see "Hiding confidential or sensitive objects in a diagram", page 300.*

The name of a confidential or sensitive object is visible in the diagram, but its properties are not accessible. It appears grayed and an icon at bottom right indicates that the object is confidential or sensitive.



A confidential or sensitive object can only be resized and moved.

To hide MetaClass occurrences in diagrams:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. In the **HOPEX** menu bar, select **View > Navigation Windows > MetaStudio**.
3. Expand the **MetaClass** folder and its sub-folders.
4. Right-click the desired MetaClass and select **Properties**.
5. In the **Characteristics** tab (**Standard** subtab) for the **Confidential objects display in diagrams** property, select **Confidential objects are hidden**.
6. Click **OK**.
   The occurrences of the corresponding MetaClass are now hidden in diagrams.

   ☛ *You can also configure display of objects in diagrams via the reading access diagram, see "Hiding confidential or sensitive objects in a diagram", page 300.*

Export and Duplication

In the case of object export and duplication, if these operations have an impact on confidential objects, a warning message will ask you if you want to continue. If you do execute export or duplication, the confidential objects concerned will be neither copied nor exported.

Invisible objects (confidential) are not duplicated during object duplication. Only a message in the status bar indicates that the duplication result is incomplete.

Generation of reports (MS Word) and Web sites

When created, reports (MS Word) and Web sites are created with the reading access level of their creator (and not with the reading access level at creation)..

They are then always generated at their their reading access level.

A **Reading access area path** attribute exists on all reading access areas. If this attribute is specified, Web sites and reports (MS Word) are generated in this folder. Reports (MS Word) and Web sites are generated in this folder to facilitate reading access management of generated files by defining access rights to this generation folder. This task should be handled by the system administrator.

Confidential report (MS Word) and Web site generation paths can be defined in the properties dialog box of a reading access area.

To define this path:

1. In the reading access area properties dialog box, select the **Characteristics** tab.
2. In the **Reading access area path** field, specify the required path.
3. Click **OK**.

Macros

The principle of reading access management in macros is to carry out all calculations in **HOPEX** and hide confidential or sensitive objects from users that do not have sufficient reading access area access rights to view them.

By default a macro is executed at user reading access level.

A macro can also be executed at its own reading access level.

If you execute a macro with a reading access level higher than the level of the current user, the methods.

- GetProp(xxx, "display") and GetFormatted return empty,
- GetProp("xxx") returns the value.

ExecuteGlobal and ***CreateObject*** ("Mega.Application") methods are prohibited in macros.

Other properties are accessible.

So that a macro can be executed with its reading access level:

1. In the macro properties dialog box, select the **Characteristics** tab.
2. In **_ExecutionOptions**, select the **Execution at Reading Access level** option.

Confidential or sensitive objects and namespaces

If an object with a namespace is not confidential or sensitive, but its parent is confidential or sensitive, the name of the latter will be masked in **HOPEX**. It appears in **HOPEX** as:

```
" ***::Operation 1 "
```

Modifying Reading Access Areas

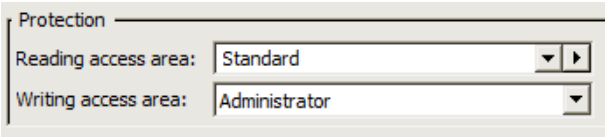
This section explains how to modify the reading access area area of an object or a user:

Modifying object reading access areas

From an object properties window, you can consult the reading access area to which it belongs.

To determine the reading access area of an object:

1. Right-click the object and select **Properties**.
2. In the object **Properties** dialog box, select the **General** tab, then the **Administration** subtab.
   In the **Protection** frame, you can consult and modify the **Reading access area**.



Modifying user reading access areas

You can modify a user reading access area from:

- the user management window (**Reading access area** column of the person)

  ☛ *See "Modifying User Properties", page 57.*

  ☛ *The reading access diagram is compiled to take account of modifications.*

- the person properties window

  ☛ *See "Defining a Person", page 53.*

  Warning: if you modify the reading access area in the user properties dialog box, you must recompile the reading access diagram so that the modification will be taken into account.

- the reading access diagram

  ☛ *See "Connecting Users to Reading Access Areas", page 289 or "Reading access area of the user", page 290.*

You can modify a user reading access area at creation from:

- the person properties window

  ☛ *See "Defining a Person", page 53.*

  Warning: if you modify the reading access area in the user properties dialog box, you must recompile the reading access diagram so that the modification will be taken into account.

- the reading access diagram

  ☛ *See "Reading access area at creation", page 290.*

MetaClass Confidentiality Exceptions

The following MetaClasses cannot be made confidential:

_Add-ins_data
_Add-ins_meta
_Brick
_Class
_ClassCommand
_Code Template
_Command
_Dispatch
_Executable
_MappingTypeItem
_MappingTypeItemProperty
_Object
_Operator
_Property
_Proposed_Table
_Resource
_StdFile
_Style
_TagAttributeDef
_TagAttributeDefValue
_TagDef
_Template
_Text
_Transaction
_TransactionData
_TransferredObject
_Type
_UML Reserved Word
Method author
MEGA Repository
ChangeItemData
ChangeItemDataTechnical
ChangeItemSystem
Generation kinematics
Component Template
DiagramTypeLink
DiagramTypeLinkStyle
DiagramTypeObject
DiagramTypeCollection
DiagramTypeField
DiagramTypeFormat
DiagramTypeParam
DiagramTypePath
DiagramTypePathPart
DiagramTypePopulating
DiagramTypeProperty
DiagramType
DiagramTypeView
Stem Codes Folder
Web Site Templates Folder
Analysis Templates Folder
Diagram Types Folder
HTML Formatter
Generality
Generator
Programming Language
Language
Animation Mask
Matrix Template

MetaAssociation	Associative Object
MetaAssociationEnd	Default Associative Object
MetaAssociationType	Generic Object
MetaAttribute	System Generic Object
MetaAttributeGroup	Analysis Parameter
MetaAttributeValue	Profile
Metaclass	Query Parameter
MetaClassDiagramType	Generation rule
MetaCommand	Modeling Rule
MetaField	Modeling Regulation
MetaList	Query
MetaListType	Web Site Template
MetaPattern	SQL Clause Type
MetaPicture	TaggedValue
MetaPropertyPage	Analysis Type
MetaTest	user
MetaTree	Descriptor Setting
MetaTreeBranch	DBMS Version
MetaTreeNode	Writing access area
Method	Reading access area

COMMAND FILE SYNTAX

The following points are covered here:

- ✓ "Command file extensions", page 308
- ✓ "Object Naming Rules", page 309
- ✓ "Commands", page 311
- ✓ "Basic Syntax", page 313

COMMAND FILE EXTENSIONS

A command file is a file containing repository update commands. It can be generated by backup or object export (.MGR extension) or by logfile export (.MGL extension).

Command files can be obtained in two ways:

- By logical backup or by object export (.MGR): the *absolute identifiers* (IdAbs) of the imported objects are used and the authorization levels are kept.

  An absolute identifier is a string of characters associated with each object in the repository. This string is computed using the date and time the session was opened, the number of objects created since the session started, and the object creation date (in milliseconds). An absolute identifier provides a unique way to identify an object in the repository, so that even if the object is renamed, all the links to it are retained.

- By logfile export (.MGL): the commands contain, in addition to the *absolute identifiers* (IdAbs) of objects, that of the user executing each command to check at import that the user had the necessary rights to execute this update.

Result of the import of these files is therefore different:

- ".MGR" corresponds to an image, complete or partial, of the repository at a given moment. It is therefore recommended that it be imported into an empty repository to rebuild this image.
- ".MGL" corresponds to commands to be applied to the repository to pass from initial state to final state.

At import, checks are performed automatically as a function of the file extension:

- For command files with the MGR extension, the absolute identifiers of the imported objects are used and the writing access levels are kept.
- For command files with the MGL extension (exported logfile or backup logfile), the absolute identifiers of the imported objects are used. Writing access levels are checked. The authorization levels are kept if the updates are consistent with the writing access diagram for the environment, otherwise they are rejected.

OBJECT NAMING RULES

Object naming will depend on the uniqueness rule applied to its name. This rule is important, since the name appears in command files.

An object has a unique name

An object must have a unique name throughout the repository.

```
For example, a report template (MS Word) has a name that
appears in command files.
```

A name is unique in a given context

Several objects can have the same name, but the name must be unique in a particular context: therefore it has a namespace.

```
For example, an operation of an organizational process: its
name must be unambiguous within the process, but several
different process operations can carry the same name.
```

For these objects, two names are presented to the user in the user interface:

Con-cept	Example	Comment
Name	Hire::Call candidate	Complete object name. Unique in the repository. Calculated from the local name and the name-space name (which can itself have a name-space). Here the operation "Call candidate" belongs to the "Hire" process.
Local Name	Call candidate	Name of the object in its namespace. Unique in the namespace.

Two names are used in **HOPEX** command language:

Concept	MetaAttribute	Example of value
Internal name of the object. It contains HexaIdabs of the object.	Name	14B8162B3F3A0347
Local name of the object and Hex-aIdabs of its namespace.	Generic Local Name	Call candidate [85ED06B63EC95B6F]

These build rules ensure respect of naming rules imposed by the repository:

- The name must be unique: the IdAbs is built to be so.
- The local name must be unique in its context: specify a uniqueness constraint on the GenericLocalName.

☛ *If the object is detached from its namespace, in the local name the indicated HexaIdAbs is then a string of 16 "0".*

Objects without name constraint

There is no name uniqueness constraint on certain objects such as messages: the same operation can send or receive several messages with the same name.

In this case, the object constitutes its own namespace.

Two names are used in **HOPEX** command language:

Concept	MetaAttribute	Example of value
Internal "Name" of the object. It contains HexaIdabs of the object.	Name	14B8162B3F3A0347
Local name of the object and HexaIdabs of its namespace (itself).	Message Local Name	Convocation [14B8162B3F3A0347]

COMMANDS

Commands on objects

- .Create (creation of an object)
- .Update (modification of an object MetaAttribute)
- .Delete (deletion of an object)

Commands on links

- .Connect (creation of a link between two objects)
- .Disconnect (deletion of a link between two objects)
- .Change (modification of a link MetaAttribute)

Other Commands

- .Validate (triggers intermediate save on import)
- .Description (produces display in import dialog box)

Rules to be respected

Command files must comply with the following rules:

- A command line cannot contain more than 5000 characters.
- Object names are limited to:
  - 63 characters for object types without namespace.
  - 255 characters for object types with namespace (name or local name).
- Commands begin with a verb infinitive prefixed by ".".
- The "." of the command must be in the first column.
- Use a hyphen (-) at the end of a line to indicate that it continues on the next line.
- Comment lines are indicated by a hyphen (-) at the beginning of the line.
- Use double-quotes (") around values that contain spaces or characters other than letters or digits.

Remarks

- Certain objects are functionally invalid if one of their MetaAttributes is not entered or a link is not defined. For example, a diagram type object must be connected to another object by a descriptor type link. We say that the diagram describes this object.
- To exchange data between two **HOPEX** environments, they must have identical metamodels and coherent user diagrams.

Commands as function of file type

Each command must consist of:

- a verb indicating the action to be carried out
- a list of parameters required to carry out this action (object types and names)
- a keyword ".CHK" followed by a list of the IdAbs of objects impacted by this command.

> ☛ *The fact of repeating the object name and IdAbs in the command enables its correct execution, even if the object has been renamed.*

The difference between the command of an ".MGR" file and the same command of an ".MGL" file is in the ".CHK":

- they have the same verb
- they have the same list of parameters
- the ".CHK" of MGL contains in addition in last position, the IdAbs of the user that issued the order.

> ☛ *A third file format (obsolete in this version) is ".MGE". In these files commands do not have a .CHK. The IdAbs are assigned as required. It is therefore not possible to process "namespaced" objects for which the namespace IdAbs cannot be assigned, since it forms part of their name.*

References to the metamodel

Each metamodel instance (MetaClass, MetaAttribute, …) can be prefixed by its IdAbs. This assures permanence of files despite renamings which may be carried out in the metamodel.

```
Example:

"~OsUiS9B5iiQ0[Operation]" is equivalent to "Operation".
```

BASIC SYNTAX

See:

✓ "Creating an Object", page 313
✓ "Deleting Objects", page 314
✓ "Modifying an Object", page 314
✓ "Modifying Texts", page 315
✓ "Modifying a Name", page 317
✓ "Creating and Modifying an Object with a Single Command", page 317
✓ "Creating a Link Between Two Objects", page 318
✓ "Modifying a Link", page 319
✓ "Deleting a Link", page 319
✓ "Managing Translations", page 320
✓ "Validating Import", page 320
✓ "Displaying a Comment in the Import Dialog Box", page 320
✓ "Transforming an MGL File to MGR", page 321
✓ "Transforming an MGR File to MGL", page 321

Creating an Object

Syntax	***.Create ."Object type""Object name" -*** ***.CHK "..."***
Example 1	.Create ."~IdAe93gyh020[Report template (MS Word)]" "Application documentation" - .CHK "w0e4VVXC)440e0SDsNpple00"
Example 2	.Create ."~OsUiS9B5iiQ0[Operation]" "14B8162B3F3A0347" - .CHK "GZB5hOXE)Sq0C30000mCpCpC"

In this command, the ".CHK" is the concatenation of the following IdAbs:

- IdAbs of the object
- IdAbs of object writing access
- In the case of the MGL, the IdAbs of the user that made this command.

Certain MetaAttributes, such as "Creation date" or "Creator" can only be updated at object creation. They must therefore be incorporated in this command.

Example:

```
Create ."~OsUiS9B5iiQ0[Operation]" "14B8162B3F3A0347" -
    .CHK "GZB5hOXE)Sq0C30000mCpCpC" -
    ."~510000000L00[Creation Date]"          "2003/08/13 10:42:51" -
    ."~(10000000v30[Creator]"              "OmNRasMwq400" -
    ."~520000000L40[Create Version]"          "25088"
```

The "Creator" and "Modifier" MetaAttributes contain the IdAbs of users that have created and modified the object. If they are not specified in the command, they automatically take the IdAbs of the user importing the file.

Similarly, "Link creation date" and Link modification date" are specified from the import date if they are absent.

Deleting Objects

Syntax	***.Delete ."Object type" "Object name" -*** ***.CHK "..."***
Example 1	.Delete ."~ldAe93gyh020[Report template (MS Word)]" "Application documentation" - .CHK "w0e4VVXC)440"
Example 2	.Delete ."~OsUiS9B5iiQ0[Operation]" "14B8162B3F3A0347" - .CHK "GZB5hOXE)Sq0"

In this command, the ".CHK" is the concatenation of the following IdAbs:

- IdAbs of the object
- In the case of the MGL, the IdAbs of the user that made this command.

Deletion of an object systematically results in:

- Loss of its attribute and text values.
- Deletion of all of the links around the object.

Modifying an Object

Syntax	***.Modify ."Object type" "Object name" -*** ***.CHK "..." -*** ***."metaattribute1" "Value1" -*** ***."metaattribute2" "Value2"***
Example 1	.Update ."~MrUiM9B5iyM0[Application]" "874B9C483D7828C6" - .CHK "PjqX8n9UzOCA" - ."~610000000P00[Modification Date]" "2010/09/07 10:26:30" - ."~b10000000L20[Modifier]" "xDqT)UdFwC10" - ."~2yUL4SsRp4B0[Application Code]" "GESTCAT11" - ."~ByUL4SsRpeB0[Operating Application Date]" "1995/10/04 23:00:00"
Example 2	.Update ."~gsUiU9B5iiR0[Organizational Process]" "0A496AAE407D1621" - .CHK "Vba2kgMV05Y5" - ."~pjRX1OOKne20[Process Frequency]" "Q""

In this command, the ".CHK" is the concatenation of the following IdAbs:

- IdAbs of the object
- In the case of the MGL, the IdAbs of the user that made this command.

The "Modification date" and "Modifier" MetaAttributes can be modified like standard MetaAttributes. If they are not specified in the command, they automatically take the file import date and the IdAbs of the user importing the file.

In the case of "Example 2" with a tabulated attribute, the value to be entered is the internal value (for process frequency this is "D" and not "Daily").

Modifying Texts

Syntax	***.Modify ."Object type" "Object name" -*** ***.CHK "..." -*** ***."Text name" "Text format"*** ***Text value*** ***.***
Example 1	.Update ."~MrUiM9B5iyM0[Application]" "DFE4E02F4D4D2BB3" - .CHK "AH(tl0UJDDxA" - ."~f10000000b20[Comment]" "g3TCfAJnyq00" 00680SbnxCMPqRc5SN6bpSsvXS6DfCZ5dN38rPcLaN31cPcLaRc5iC35dUpOpRsPST7HkUsnY 00680C6PSRcPSN6nfQ6DcSt9XC7HbKqqWQ5CWR6nbR4GWVJjd2WrzQNPSQtTbP6vfTLmqN35Z 00602Sc5mPbnaPbmmC39pRqCWPMrj87Hk86PlS71XOsbiQNHX86vlS5mn3N9X3NqA000A .

Text format	Value
ASCII text	0 000000000000
Binary text	1 000000000001
RTF text	"MRDYO5Oe(smC"
HTML text	"LQDYO58M6tmC"
ANSI text	"G3000000W10S"

In this command, the ".CHK" is the concatenation of the following IdAbs:

- IdAbs of the object
- In the case of the MGL, the IdAbs of the user that made this command.

In a command file, each line of text is limited to 74 characters.

End of the text marked by a line containing only one point (".") in the first column.

A semi-colon in the first column inserts a blank line (the rest of the line must be left blank).

☛ *When text is extracted, lines are divided after character 73 and a semicolon is inserted in position 74, indicating that the next line is to be concatenated with the previous one.*

Complementary indicators:

- Each text modification applies to its totality. It is not possible to add just one complement.
- The semicolon character (;) in first position inserts an blank line. To reinitialize a text, it is sufficient that the text value contains just a semicolon.
- The characters period and semicolon (.;) in first and second position create a line containing a period only.
- Two semicolon characters (;;) in first and second position create a line containing a semicolon only.
- Apostrophe (’) and quotation marks (") are authorized as text values.
- The semicolon character (;) enables cutting of text lines exceeding 74 characters. The semicolon is therefore the last significant character in the line.

To reinitialize a text, it is sufficient that the text value contains just a semicolon.

```
Example:
Update ."~MrUiM9B5iyM0[Application]" "DFE4E02F4D4D2BB3" -
        .CHK "AH(tl0UJDDxA" -
        ."~f10000000b20[Comment]" "000000000000"
;
.
```

To delete a text, the text value should be left empty.

```
Example
Update ."~MrUiM9B5iyM0[Application]" "DFE4E02F4D4D2BB3" -
        .CHK "AH(tl0UJDDxA" -
        ."~f10000000b20[Comment]" "000000000000"
.
```

A reinitialized text contains nothing but it exists, while a deleted text no longer exists. For example the query *"Select Application where Comment null"* returns applications that have no comment, but not those that have a reinitialized comment.

Modifying a Name

Syntax	***.Modify ."Object type" "Object name" -*** ***.CHK "..." -*** ***."Name or Local name" "Value "***
Example 1	.Update ."~ldAe93gyh020[Report template (MS Word)]" "Report template (MS Word)-1" - .CHK "RJ(tBUUJD5(AV(WEIeZIDT4B" - ."~210000000900[Name]" "Report template (MS Word)-New"
Example 2	.Update ."~MrUiM9B5iyM0[Application]" "DFE4E02F4D4D2BB3" - .CHK "AH(tl0UJDDxA" - ."~g20000000f60[Generic Local name]" "Application-1[0000000000000000]"

Creating and Modifying an Object with a Single Command

At object creation, creation of a "modification" command by MetaAttributes to be specified is of no interest: MetaAttributes (non-textual) can be directly assigned by the create command.

Syntax	***.Create ."Object type" "Object name" -*** ***.CHK "..." -*** ***."metaattribute1" "Value1" -*** ***."metaattribute2" "Value2"***
Example 1	.Create ."~MrUiM9B5iyM0[Application]" "DFE4E02F4D4D2BB3" - .CHK "AH(tl0UJDDxAC30000mCpCpC" - ."~510000000L00[Creation Date]" "2011/02/05 11:41:35 PM" - ."~610000000P00[Modification Date]" "2011/02/06 12:31:38 AM" - ."~(10000000v30[Creator]" "V(WEIeZIDT4B" - ."~b10000000L20[Modifier]" "V(WEIeZIDT4B" - ."~520000000L40[Create Version]" "29248" - ."~620000000P40[Update Version]" "29248" - ."~)20000000z70[Confidentiality area identifier]" "sTlVwxdH3100" - ."~2yUL4SsRp4B0[Application Code]" "AA" - ."~KyUL4SsRpCC0[Version Number]" "12" - ."~ByUL4SsRpeB0[Operating Application Date]" "2011/02/11 11:00:00" - ."~)Sgoy)ygu020[Application Type]" "M" - ."~nOU8g8IMCb30[Converted Name Version]" "0" - ."~PYq45X2wBP92[Date of C&A Completion]" "2011/02/06 11:00:00" - ."~Ol8pdE(l8r(0[Deployment Date]" "2011/02/27 11:00:00" - ."~g20000000f60[Generic Local name]" "Application-1[0000000000000000]" - ."~PZq41c2wBXP2[Security Planning]" "Operational" - ."~a20000000H60[LanguageUpdateDate]" "2011/02/05 23:41:57"

Creating a Link Between Two Objects

Syntax	***.Connect ."Object type" "Object 1 name" ."MetaAssociationEnd" "Object 2 name" -*** ***.CHK "..."***
Example 1	.Connect ."~MrUiM9B5iyM0[Application]" "DFE4E02F4D4D2BB3" ."~mi54NLn-HzCj0[Application within Internal Architecture]" "DFE4F2274D4D2C43" - .CHK "AH(tl0UJDDxAbJ(td8VJDD4B" - ."~710000000T00[Link creation date]" "2011/02/06 00:58:15" - ."~810000000X00[Link modification date]" "2011/02/06 00:58:15" - ."~720000000T40[Link Creator]" "V(WEIeZIDT4B" - ."~920000000b40[Link Modifier]" "V(WEIeZIDT4B" - ."~410000000H00[Order]" "9999"

In this command, the ".CHK" is the concatenation of the following IdAbs:

- IdAbs of object 1
- IdAbs of object 2
- In the case of the MGL, the IdAbs of the user that made this command.

The "Order" MetaAttribute is optional. If present, the order is numeric on maximum four positions, otherwise the order is set by default to 9999.

The "Link creator" and "Link modifier" MetaAttributes contain the IdAbs of users that have created and modified the link. If they are not specified in the command, they automatically take the IdAbs of the user importing the file.

Similarly, "Link creation date" and Link modification date" are specified from the import date if they are absent.

> ☛ *If this link is used to build a namespace, it must be completed by modification of the local name of the namespaced object to maintain repository consistency.*

Similar to object creation, it is possible to specify link MetaAttributes (except texts) directly in this command, without passing via a modification command.

Modifying a Link

With the exception of its header, this command has the same syntax as the object modification command.

☛ *See "Modifying an Object", page 314.*

Syntax	***.Change ."Object type" "Object 1 name" ."MetaAssociationEnd" "Object 2 name" - .CHK "..." - ."metaattribute 1" "Value 1" - ."metaattribute 2" "Value 2" - ."Text name" "Text format" - Text value .***
Example	.Change ."~MrUiM9B5iyM0[Application]" "DFE4E02F4D4D2BB3" ."~mi54NLnHzCj0[Application within Internal Architecture]" "DFE4F2274D4D2C43" - .CHK "AH(tl0UJDDxAbJ(td8VJDD4B" - ."~810000000X00[Link modification date]" "2011/02/06 1:22:26 AM" - ."~920000000b40[Link Modifier]" "V(WEIeZIDT4B" - ."~b20000000L60[LinkLanguageUpdateDate]" "2011/02/06 01:22:26" - ."~C3cm9FyluS20[Link Comment]" "g3TCfAJnyq00" 00680SbnxCMPqRc5SN6bpSsvXS6DfCZ5dN38rPcLaN31cPcLaRc5iC35dUpOpRsPST7HkUsnY 00680C6PSRcPSN6nfQ6DcSt9XC7HbKqqWQ5CWR6nbR4GWVJjd2WrzQNPSQtTbP6vfTLmqN35Z 00362Sc5mPbnaPbmmC39pR68WR69XS5nX3N9X3NqA000A .

In this command, the ".CHK" is the concatenation of the following IdAbs:

- IdAbs of object 1
- IdAbs of object 2
- In the case of the MGL, the IdAbs of the user that made this command.

The MetaAttributes “Link modification date”, and “Link modifier” can be modified just like standard MetaAttributes. If they are not specified in the command, they automatically take the file import date and the IdAbs of the user importing the file.

Deleting a Link

Syntax	***.Disconnect ."Object type" "Object name 1" ."MetaAssociationEnd 2" "Object name 2" - .CHK "..."***
Example	.Disconnect ."~MrUiM9B5iyM0[Application]" "DFE4E02F4D4D2BB3" ."~mi54NLnHzCj0[Application within Internal Architecture]" "DFE4F2274D4D2C43" - .CHK "AH(tl0UJDDxAbJ(td8VJDD4B"

In this command, the ".CHK" is the concatenation of the following IdAbs:

- IdAbs of object 1
- IdAbs of object 2
- In the case of the MGL, the IdAbs of the user that made this command.

Deletion of a link results in loss of link MetaAttributes values.

> ☛ *If this link is used to build a namespace, it must be completed by modification of the local name of the "namespaced" object to maintain repository consistency. Its namespace has become "[0000000000000000]".*

Managing Translations

For each language supported by **HOPEX**, two MetaAttributes indicate the last modification date of translations in a language:

- "[LanguageUpdateDate (Language)]" for an object
- "[LinkLanguageUpdateDate (Language)]" for a link

These MetaAttributes are managed following the same rules as the"Modification date" and "Link modification date" MetaAttributes:

- They can be modified in the same way as standard MetaAttributes.
- If they are not specified in the command modifying a translation, they automatically take the file import date.

Validating Import

Syntax	***.Validate***

This command does not contain a .CHK and produces an intermediate save operation at import.

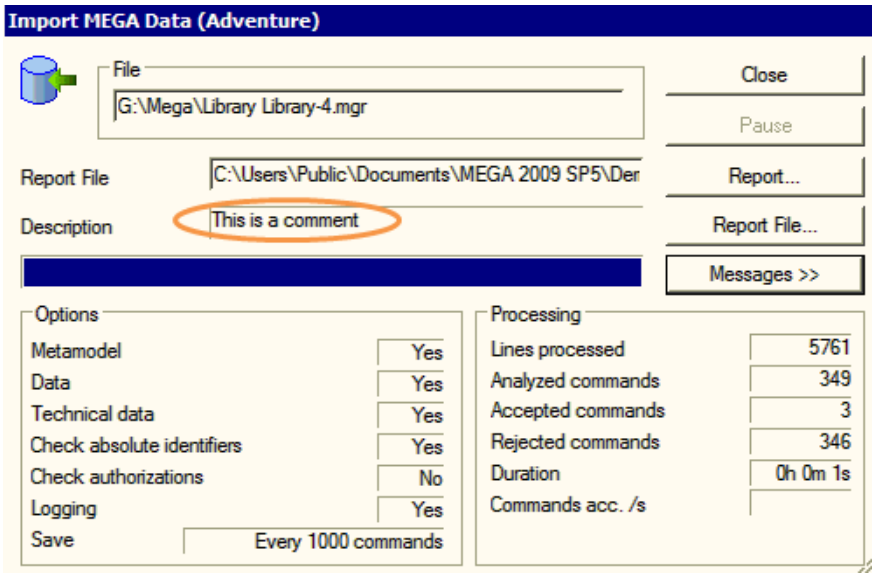
> ☛ *This command invalidates save operation selection made by the user interface. For example, to validate consistency of a command file, the user generally imports with "Save Never". Commands are then saved until the last ".Validate" of the file.*

Displaying a Comment in the Import Dialog Box

Syntax	***.Description 'Text'***
Example	.Description 'MetaClass: Acceptance Criteria'

This command does not contain a .CHK.

The text appears on the user interface at import.



Transforming an MGL File to MGR

☛ *See "Command file extensions", page 308.*

You do not need to transform an .mgl file to .mgr.

To obtain the same result, when importing an .mgl file:

- in the data import dialog box, in the **Checks** frame, clear the **Check Writing Accesses** check box.

Transforming an MGR File to MGL

☛ *See "Command file extensions", page 308.*

You do not need to transform an .mgr file to .mgl.

To obtain the same result, when importing an .mgr file:

- in the data import dialog box, in the **Filter** frame, select the **Reassign User** check box.
  Each command is then processed as if its CHK contained the IdAbs of the importing user. Writing access checks are carried out related to its rights.

  ☛ *At import in the CHK of an MGL command, the "Reassign User" check box also allows substitution of the IdAbs of the user by that of the person importing.*

MANAGING OPTIONS

This section presents the various tools and options used to configure and customize **HOPEX**.

The following points are covered here:

- ✓ "Options Overview", page 324
- ✓ "Option Window Presentation", page 325
- ✓ "Accessing Options", page 326
- ✓ "Generating the list of options", page 330
- ✓ "Available Option Groups", page 332
- ✓ "Managing Languages", page 334

OPTIONS OVERVIEW

HOPEX options concern:

- site technical configuration.
- values proposed by default for each function of **HOPEX**. These values can be modified by users on each workstation.
  This configuration is described in the guides covering each function.

HOPEX options are accessible at several levels. **HOPEX** functions can be configured at the following levels:

- site

  *The site is the location where **HOPEX** is installed; it is the root of the application.*
- environment
- profile (which groups a configuration common to several users)
- user
- workstation

There is by default an inheritance mechanism between these different levels (excepting workstation level):

- the environment inherits options define at site level.
- the profile inherits options defined at environment level.
- the user inherits options defined at connection profile level.

Customizations made at user level are of highest priority, followed in order of priority by those made at profile, environment and site levels.

Having modified option values, it is recommended that you dispatch or save your work, close HOPEX and then reopen it. Refresh issues can occur if these precautions are not taken.

For detailed information on these options, see the context-sensitive help in the lower part of the window.

Option Window Presentation

The left pane of the window contains the various option groups. It comprises two parts:

- options available for the site, environment, profile, and user

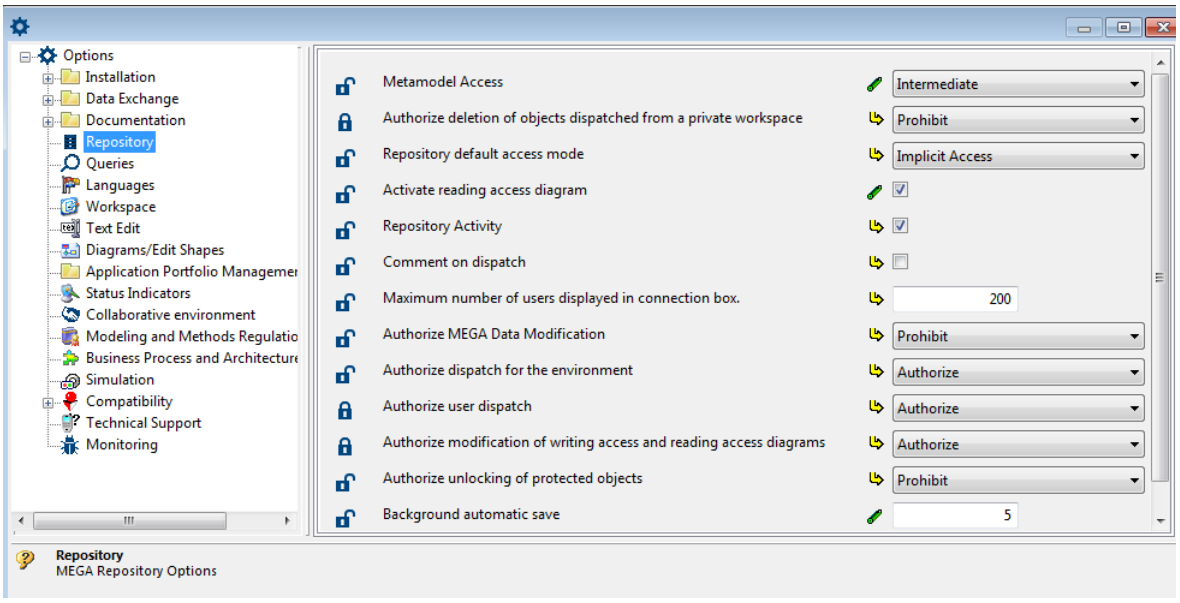
  ☛ *See "User options", page 332.*

- options specific to the workstation

  ☛ *See "Workstation Options", page 333.*

The right pane enables configuration of the various options corresponding to the group selected in the left pane.

☛ *Options vary depending on products you have available.*



For more details on an option:

❱ Click on the name of the option to display the context-sensitive help in the lower part of the window.

☛ *When the user has a private workspace in progress, you cannot modify its options from **HOPEX Administration**.*

ACCESSING OPTIONS

Options Level

You can modify options at the following levels:

- site
  - ☛ *See "Modifying options at site level", page 326.*
- environment
  - ☛ *See "Modifying options at environment level", page 326.*
- profile
  - ☛ *See "Modifying options at profile level", page 326.*
- user
  - ☛ *See "Modifying options at user level:", page 327.*
- workstation
  - ☛ *See "Modifying options at workstation level:", page 327.*

Modifying options at site level

To modify options at site level:

1. Start **HOPEX Administration**.
   ☛ *See "Connecting to HOPEX Administration", page 6.*
2. In the navigation tree, right-click the site name and select **Options > Modify**.
   The site options window opens.

Modifying options at environment level

To modify options at the environment level from **HOPEX Administration** :

1. Using **HOPEX Administration**, connect to the environment concerned.
   ☛ *See "Connecting to an Environment", page 7.*
2. Right-click the environment name and select **Options > Modify**.
   The environment options window opens.

Modifying options at profile level

To modify options at profile level from **HOPEX Administration**:

1. Using **HOPEX Administration**, connect to the environment concerned.
   ☛ *See "Connecting to an Environment", page 7.*
2. Open the manage profiles dialog box.
   ☛ *See "Opening the profile management window", page 47.*
3. In the **Profile** tab , right-click the profile and select **Options**.
   The profile options window opens.

Modifying options at user level:

You can modify user options from:

- **HOPEX Administration**
- **HOPEX**

To configure user options from **HOPEX Administration**:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*

2. Open the user management window.

   ☛ *See "Opening the User Management window", page 45.*

3. Select the **Persons** tab.
4. Right-click the desired person and select **Options**.
   The user options window opens.

To configure user options from **HOPEX**:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*

2. In the **HOPEX** menu bar, select **Tools > Options**.
   The user options window opens.

Modifying options at workstation level:

To modify options at workstation level:

1. Connect to **HOPEX Administration**.

   ☛ *See "Connecting to HOPEX Administration", page 6.*

2. Right-click **Workstation** and select **Options**.
   The workstation options window opens.

Each option can take several values.

Option Inheritance

An option inherits a value defined at a higher level:

- A user inherits options defined at the connection profile level.
- A profile inherits options defined at the environment level.
- An environment inherits options defined at the site level.

The icon located opposite the option indicates the inheritance, or not, from the higher level:

- **Default value** indicates the inheritance from the higher level.
- **Modified value** indicates that the inherited option value has been modified. The value is no longer inherited from the higher level.

To specify that an option does not inherit the value defined at higher level:

1. Open the options page.

   ☛ *See "Options Level", page 326.*

2. Click **Default value** .
   The icon changes in **Modified value** .

Checking Option Modifications

You can prohibit modification of any option at a level lower than your current level.

```
Example: if you open options of the environment, you can
prohibit modification of all options at user level.
```

Prohibiting modification of a lower level option

To prohibit modification of a lower level option:

1. Access the options.
   ☛ *See "Options Level", page 326.*
2. Click icon located opposite the option concerned.
   The padlock closes : option modification by a user is now prohibited from **HOPEX**.

Unlocking the modification of a lower level option

To unlock modification of a lower level option:

1. Access the options.
   ☛ *See "Options Level", page 326.*
2. Click the closed padlock icon.
   The padlock opens: modification of the option is again possible.

Reinitializing Option Values

You can reinitialize the values for:

- an option
- an option group

Reinitializing the values of an option

To reinitialize the value of an option:

1. Access the options.
   ☛ *See "Options Level", page 326.*

2. Click:
   - (Web Front-End) **Reinitialize**
   - (Windows Front-End) **Modified Value**, the icon changes to **Default Value**

   The value of the option is reinitialized.

Reinitializing the values of an option group (Windows Front-End)

To reinitialize values of an option group from **HOPEX Administration**:

1. Access the options.

   ☛ *See "Options Level", page 326.*

2. In the options tree, right-click the option group and select **Reinitialization**.
   All the options in the group selected are reset to their default values.

GENERATING THE LIST OF OPTIONS

You can generate a report that lists all options classified by group, with their comments.

To generate the list of options:

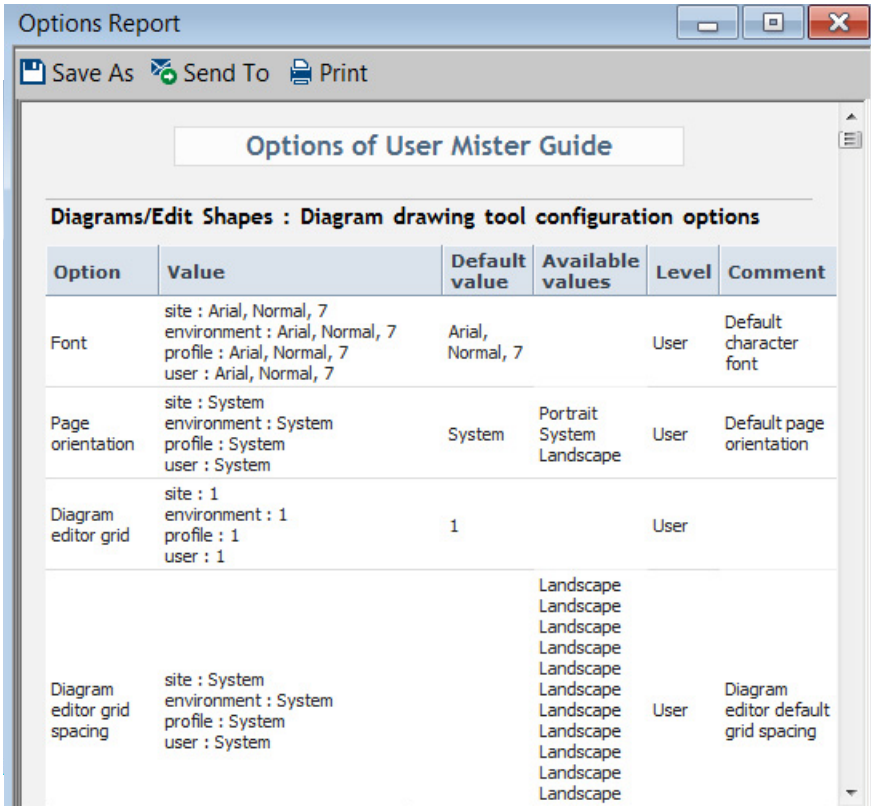
1. Open the options dialog box of the user.

   ☛ *See "Modifying options at user level:", page 327.*

2. Right-click **Options** and select **Report**.
   Report generation may take some time.

It contains:

- the names of available options
- the values available for each option
- the default value
- a comment explaining the option use context
- the option level:
  - user
  - environment
  - site



Option	Value	Default value	Available values	Level	Comment
Font	site : Arial, Normal, 7 environment : Arial, Normal, 7 profile : Arial, Normal, 7 user : Arial, Normal, 7	Arial, Normal, 7		User	Default character font
Page orientation	site : System environment : System profile : System user : System	System	Portrait System Landscape	User	Default page orientation
Diagram editor grid	site : 1 environment : 1 profile : 1 user : 1	1		User	
Diagram editor grid spacing	site : System environment : System profile : System user : System		Landscape Landscape Landscape Landscape Landscape Landscape Landscape Landscape Landscape Landscape Landscape	User	Diagram editor default grid spacing

To save this report in .html format:

- Click **Save As** and select *.htm format.

AVAILABLE OPTION GROUPS

User options

☛ *At user configuration level, certain options are grayed. They can be defined only for an environment or site and not for a user.*

Note that repository and modeling options contain important information for the functional administrator.

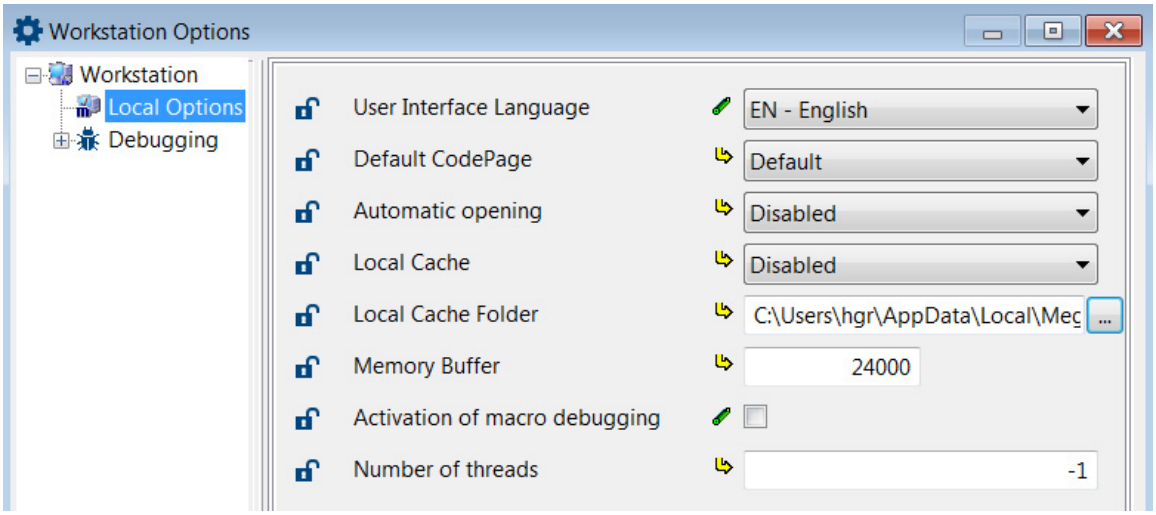
- **Installation**
  Options linked to installation: licenses, information on the company, Web application (options linked to the **HOPEX** user workspace (Web Front-End)
- **Data Exchange**
  Options linked to import/export, exchanges with third party tools.
- **Documentation**
  Options linked to documentation generated by **HOPEX** (reports (MS Word), reports (Open Office), Web sites, Description, reports, performance indicators)
- **Repository**
  Options authorizing or prohibiting access to certain repository functions.
- **Queries**
  Options linked to the query tool
- **Languages**
  Activated data languages
- **Workspace**
  Options linked to the user workspace of **HOPEX** (Windows Front-End). They enable display of a certain number of functions or not, as well as management of user inactivity or not.
- **Text Editing**
  Options concerning RTF format comment entry
- **Diagrams/Edit Shapes**
  Options of drawing tool configuration (diagrams and shapes editor)
- **Status indicators**
  Options concerning display of indicators available in workspace and diagrams
- **Collaborative Environment**
  Options available with **HOPEX Collaboration Manager** product
- **Mapping Editor**
  Options linked to the mapping editor, a tool enabling alignment of data models (essentially with **HOPEX Database Builder**)
- **Modeling and Methods Regulations**
  Options linked to modeling regulations and rules

- **Business Process and Architecture Modeling**
  Options linked to processes and architecture enabling display of certain functions
- **Simulation**
  Options enabling definition of **MEGA Simulation** use level
- **Compatibility**
  Options of compatibility concerning diagrams and obsolete functionalities
- **Technical Support**
  Options concerning access to Technical Support
- **Monitoring**
  Option used to supervise data access

Workstation Options

The local options group contains information specific to the workstation.

☛ *This information is stored in "MEGAWKS.INI" file (C:\ProgramData\MEGA\HOPEX <Version>\Cfg).*



MANAGING LANGUAGES

The following points are detailed here:

- "Changing User Interface (Windows Front-End) Language", page 334
- "Defining the Data Languages Available for a User", page 334
- "Changing User Data Language (Windows Front-End)", page 335
- "Installing Additional Languages", page 335
- "Defining the Language of e-mails in Workflows", page 335
- "Managing Languages in Web Applications", page 336

Changing User Interface (Windows Front-End) Language

You can change interface language of a **HOPEX** (Windows Front-End) user.

> ☛ *To modify the language of Web applications, see "Managing Languages in Web Applications", page 336*

To change the interface language of a **HOPEX** (Windows Front-End) user:

1. Connect to **HOPEX Administration**.

   > ☛ *See "Connecting to HOPEX Administration", page 6.*

2. In the **HOPEX** tree, right-click **Workstation** and select **Options**.
3. In the **Workstation** tree, select **Local Options**.
4. In the right pane, modify the value of the **User Interface Language** option.
5. Click **OK**.
   The change is effective at next restart of **HOPEX** (Windows Front-End).

Defining the Data Languages Available for a User

You can also select the languages available in **HOPEX** and in which you can enter data.

> ☛ *See "Installing Additional Languages", page 335.*

So that a **HOPEX** workstation can be used in multilingual mode, multilingual mode must be authorized for the site.

> ☛ *When you duplicate an object, it must be in the repository language so that translations in the various languages will be correctly transferred to the duplicates.*
>
> *To determine the language of your repository, consult repository properties.*
>
> ☛ *For more information on the use of languages, see the **HOPEX Common Features** guide, "**HOPEX** in a Multilingual Context" section.*

To define the data languages available for a user:

1. Connect to **HOPEX Administration**.

   > ☛ *See "Connecting to HOPEX Administration", page 6.*

2. Access the options management window.

   ☛ *See "Modifying options at environment level", page 326.*

   ☛ *See "Modifying options at profile level", page 326.*

   ☛ *See "Modifying options at user level:", page 327.*
3. In the options tree, select **Languages**.
4. In the right pane, select the languages available for the interface.
5. Click **OK**.

Changing User Data Language (Windows Front-End)

In **HOPEX** (Windows Front-End), the user can change his/her data language himself/herself.

☛ *To change user data language in Web applications, see "Managing Languages in Web Applications", page 336.*

To modify the data language from **HOPEX**:

1. Connect to **HOPEX**.

   ☛ *See "Connecting to HOPEX", page 136.*
2. In the menu bar, select **Tools > Languages**.
3. Select the data language.

   ☛ *Translated data appears in the selected language.*

   ☛ *To install additional languages, see "Installing Additional Languages", page 335:*

Installing Additional Languages

To install additional languages in **HOPEX**:

1. Connect to **HOPEX Administration** and select the environment concerned.

   ☛ *See "Connecting to an Environment", page 7.*
2. Right-click the environment (or site) and select **Metamodel > Install Additional Languages**.
3. The dialog box **Install Additional Languages** opens.
4. Select the languages you wish to have available in**HOPEX**, and click **OK**. A window indicates progress of import of the corresponding libraries.

   The additional languages are accessible from the **Tools > Language** menu of the **HOPEX** desktop.

Defining the Language of e-mails in Workflows

To define the language of e-mails in workflows:

1. Access the options management window.

   ☛ *See "Modifying options at environment level", page 326.*

2. In the options tree, expand the **Installation** folder and select **Workflows**.
3. In the right pane, in option **Language for sending mails**, select the language to be defined in e-mails.
4. Click **OK**.

Managing Languages in Web Applications

You can modify:

- the interface language in Web applications
- the data language in Web applications.

☛ *To manage languages in Web applications, see the HOPEX Administration - Supervisor Web guide.*

Frequently Asked Questions

The following points are covered here:

- ✓ "Common Operations", page 338
- ✓ "Recurrent Messages", page 339

COMMON OPERATIONS

How do I copy a repository from one environment to another?

Standard procedure: make a logical backup of the repository, then carry out a logical restore of the backup in an empty repository of the target environment.

For GBMS environments you can copy repository files (EMA, EMB, EMS, EMV) in a folder carrying the repository name, then create a reference for the repository in the second environment, but only if the metamodel is exactly the same in both environments.

Can I create a reference for an environment in another site?

No, the functional rule is that a reference for a HOPEX environment should only be referenced in an installation (site).

Can I delete a user?

Yes, you can delete a user. See "Deleting Users", page 58.

When you delete a user from the repository, all actions linked with this user are lost.

To delete a user but retain its actions, modify user repository access mode to **Inactive user** (see "Modifying User Properties", page 57). The user no longer appears in the connection dialog box, but its actions are kept.

Note that you cannot delete the "Administrator" user, or the "Administrator" writing access.

Can I delete a writing access area?

Yes, you can delete a writing access area.

When you delete a writing access area, the objects that were attached to it pass implicitly to "Administrator" writing access level.

MEGA recommends that before deletion, you modify the writing access of objects attached to the writing access area.

Recurrent Messages

Abnormal operation when refreshing a private workspace

Symptom: Message stating "Could not refresh your private workspace".

Reason: Rejects occurred when importing private workspace updates into the reference repository.

Solution:

1. Examine the reject file Rmmjj.MGL (eg: MGLR07150000.MGL) in the user work repository (<repository>\USER\<user code>).
2. Identify and process causes of rejects (see "Rejects When Dispatching", page 142).
3. Delete reject files that are no longer needed.

   ☛ *For as long as reject files are not deleted, a warning persists when connecting to HOPEX.*

Environment version

Symptom: Message "Your environment and site are not of the same version" when opening an environment from the HOPEX administration console" (or "Your environment and site are not of the same version. Your environment requires updating. Refer to documentation for how to carry out this action").

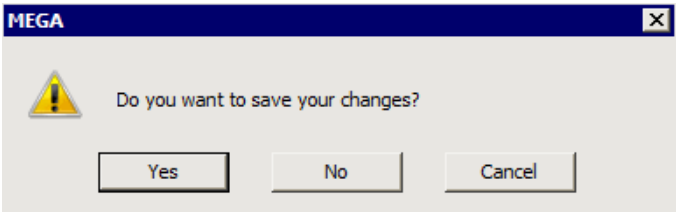
- Click **OK**.
  Another window appears displaying a second message: "Your environment requires an update for compatibility with your version of HOPEX". Do you wish to run this procedure now? ".

Reason: It is possible that the environment has not been created, or is not at the same version level as the site referencing it: an update is therefore proposed:

- if you have a physical backup of environment data, accept modification by clicking **Yes**.
- If this is not the case, refuse the modification by clicking **No** to exit the HOPEX data administration console. Then execute physical backup of data.

"Later" option not proposed at disconnect

Symptom: When you exit **HOPEX**, the dialog box that appears does not propose the "Later" option to save your modifications.



Reason:

- You are not connected to **HOPEX (Windows Front-End)** and the license used does not have technical module **MEGA Lan**.

  You prevent other users from dispatching their work performed in their private workspace.

- You are connected to **HOPEX (Web Front-End)**, in a public workspace.

PRODUCT CODES

Access your list of available products

To view the products to which you have access:

1. Start **HOPEX Administration**.

   ☛ *See "Accessing Repositories", page 74.*

2. In the menu bar, select **Help > About HOPEX**.
3. Click **System Information**.
4. With the drop-down menu, select **Available components**.
   All the products for which you have a license are listed as well as their associated code.

   ☛ *For information on product availability (Windows Front-End, Web Front-End) and storage, see online documentation (**Concepts > Products**).*

GLOSSARY

absolute identifier

An absolute identifier is a string of characters associated with each object in the repository. This string is computed using the date and time the session was opened, the number of objects created since the session started, and the object creation date (in milliseconds). An absolute identifier provides a unique way to identify an object in the repository, so that even if the object is renamed, all the links to it are retained.

access area member

Access area member groups all persons and person groups belonging to an access area. This area defines objects that can be accessed by the person or person group.

access path

An access path indicates which folder you can use when creating a reference for an environment database or a site environment. When all repositories are created in the same location as the environment, the path created at installation (the DB folder under the environment root) is sufficient and is given as the default. When you want to save a repository in a folder other than that of the environment, you must declare a new access path.

access rights

User access rights are the rules that manage access to software functions and databases. You can restrict the access rights of a user to the different repositories defined in his/her work environment. You can also restrict what software features a given user can run, such as the descriptor editor, modifying queries, designing report templates (MS Word), deleting objects, dispatching private workspaces, importing command files, managing environments, repositories, users, writing access, etc.

administration Administration consists of managing the work environment of repository users. This function is usually the responsibility of the administrator. Administration tasks include making backups of repositories, managing conflicts in data shared by several users and providing users with queries, descriptors, report templates (MS Word), etc. common to several projects.

Administration desktop The **HOPEX Administration** desktop (Web Front-End) is the Web version of the **Administration** (Windows Front-End) application accessible via an internet browser.

administrator The administrator is a person who has administration rights to manage sites, environments, repositories and users. In addition to Administrator (who cannot be deleted) and MEGA users, created at installation, you can grant administration rights to other users.

attribute See *Characteristic*.

backup A physical backup (GBMS only) consists of copying the files of a repository from their original location to another.

backup logfile The backup logfile is an additional file stored outside the repository or private workspace. It ensures that the changes made to the repository or private workspace can be recovered even if the repository files become corrupted. Creation of this file is requested in the configuration of each repository (including the system repository). It is created when the first update is made in a private workspace or repository.

business role A business role defines a function of a person in a business sense. A person can have several business roles. A business role is specific to a repository.

characteristic A characteristic is an attribute that describes an object or a link. Example: the Flow-Type characteristic of a message allows you to specify if this message is information, or a material or financial flow. A characteristic can also be called an Attribute.

command file A command file is a file containing repository update commands. It can be generated by backup or object export (.MGR extension) or by logfile export (.MGL extension).

comparison You can compare objects in two repositories, creating a file that will modify objects in one repository to make these equivalent to objects in the other repository (with the **HOPEX Power Supervisor** technical module or **HOPEX Collaboration Manager***)*. This comparison also allows you to list the differences between the contents of the two different objects.

compilation

Compilation is carried out after migration or customization. Compilation checks configuration of the environment concerned. When completed, processing for all users of this environment is speeded up. Metamodel compilation includes in parallel translation in the current language. You can also translate the metamodel into another language.

consolidation

Consolidation groups the updates from stand-alone workstations or remote sites (with Lan) and merges them in a reference site. After dispatch of the private workspaces of each of the users, the repository log is exported and reinitialized. The logfiles are imported into the reference repository, then this is recopied on each of the user sites.

description

Descriptors allow you to create reports (MS Word) containing part of the contents of the repository. Descriptor for an object includes the object characteristics, to which can be added the characteristics of objects directly or indirectly linked to it. The readable format for each of the objects encountered is entered as text in Word for Windows. You can insert descriptors into reports (MS Word) or report templates (MS Word) or use them to produce reports. Descriptors can be created or modified using the **HOPEX Power Studio** technical module.

desktop

The desktop specific to each user contains projects, diagrams, reports (MS Word), etc. handled by this user. The user has a different workspace in each of the repositories he/she accesses.

discard

Discarding the work performed in a private workspace cancels all modifications made since the last dispatch. All the work you have done since the beginning of the private workspace is lost. A warning message reminds the user of this. The user can request the discard of his/her private workspace from the **File > Discard** menu or at disconnection.

environment

An environment groups a set of *users*, the *repositories* on which they can work, and the *system repository*. It is where user private workspaces, users, system data, etc. are managed.

external reference

An external reference enables association of an object with a document from a source outside **HOPEX**. This can relate to regulations concerning safety or the environment, legal text, etc. Location of this document can be indicated as a file path or Web page address via its URL (Universal Resource Locator).

functionality A functionality is a means proposed by the software to execute certain actions (for example: the shapes editor and the descriptor editor are functionalities proposed as standard).

general UI access General UI access defines if tools are available or not. By default, general UI accesses have value *A (A: Available, *: default value)

group Descriptors, which are available with the **HOPEX Power Studio** technical module, comprise a tree of several successive groups. Each group concerns one object, and defines the query or link used to access this object from the preceding object. You can connect texts and other groups to a group. Users can define the order in which groups and texts are processed.

importing Importing a command file, a backup file, or a logfile consists of applying the commands in the file to this new repository.

LDAP parameter An LDAP parameter is a parameter that exists in the LDAP directory and is associated uniquely with a **HOPEX** attribute. For example, "mail" is a parameter of the "Active Directory" LDAP directory and can be associated with the e-mail of the person.

LDAP server The LDAP server is the server on which the LDAP directory is installed. The LDAP directory can be an Active Directory directory.

link A link is an association between two types of object. There can be several possible links between two object types, for example: Source and Target between Org-Unit and Message.

link orientation The two objects connected by a link do not normally have symmetrical roles. For example, to connect an operation to an organizational process with the **HOPEX Power Supervisor** technical module, you must be authorized to modify the organizational process, since this action will modify its behavior. This action will not however modify the operation. You do not need authorization to modify the operation to create this link. The organizational process is said to be major for this link, the operation is minor. This characteristic is used by administration tools for object export, protection, object comparison and querying isolated objects.

lock

A lock is a logical tag assigned to an object to indicate that it is currently being modified by a user. Simultaneous access to an object by several users can also be checked. Locks apply to all types of object. When a user accesses an object to modify it, a lock is placed on the object.

When a lock is placed on an object, another user is only able to view the object. This second user will not be able to modify the object until the first user dispatches his/her work, and the second user refreshes. This is done to avoid conflicts between the state of the object in the repository, and the obsolete view of the second user.

logfile

Logfiles contain all the actions performed by one or more users over a given period. The private workspace log contains all the changes made by a user in his/her private workspace. This logfile is used to update the repository when the user dispatches his work. For additional security, it is also possible to generate another log called the backup logfile.

logfile export

Export of a logfile creates a command file from the logfile of user actions in a repository. You can keep this file and import it later into a repository. You can selectively export modifications made in the work repository, or those made to technical data (descriptors, queries, etc.) in the system repository.

login

A Login uniquely defines a user or user group. It can be assigned to only one Person or Person Group.

major

The major object in the link is the one whose nature changes with the presence or absence of the link. For example a process, defined as a succession of operations, is modified if you remove an operation. The process is then major for the link. If the objects are protected, you must have the correct authorization for modifying the major object in order to create or delete the link.

matrix

A matrix is a table comprising rows and columns containing objects from the repository. Matrices show the relationships between two sets of objects and allow you to create or delete links without having to open the diagrams themselves. For example, you can build a matrix showing the messages sent by the different org-units in a project.

MetaAssociation

see "link".

Metaclass

see object type

Metamodel

The metamodel defines the language used for describing a model. It defines the structure used to store the data managed in a repository The metamodel contains all the MetaClasses used to model a system, as well as their MetaAttributes and the MetaAssociations available between these MetaClasses. The metamodel is saved in the system repository of the environment. You can extend the metamodel to manage new MetaClasses. Repositories that exchange data (export, import, etc.) must have the same metamodel, otherwise certain data will be rejected or inaccessible.

minor

The minor object in a link is the one whose nature is not modified or only slightly modified by presence or absence of this link. For example, removing an operation from a process does not change characteristics of this operation. Therefore the process is minor in the link.

model

A model is a formal structure which represents the organization of a company, or its information system. In another sense, a model can be a template for reproducing objects with similar characteristics. This is the case for report templates (MS Word) and matrix models.

object

An object is an entity with an identity and clearly defined boundaries, of which status and behavior are encapsulated. In a **HOPEX** repository, an object is often examined along with the elements composing it. For example, a Diagram contains Org-Units or Messages. A Project contains Diagrams, themselves containing Org-Units, Messages etc. Database administration frequently requires consideration of consistent sets of objects. This is the case for object export, protection and object comparison. Isolated objects are found by checking that each object is used in another object. For example, a Diagram containing Org-units and Messages is itself used by a Process, an Org-unit, a Project, etc. This functionality is available with the **HOPEX Power Supervisor** technical module.

object export

The export of one or several objects enables transfer of a consistent data set from a study to another repository. For example, export of objects from a project includes the project diagrams, together with the objects these diagrams contain, such as messages and org-units, as well as dependent objects. All the links between objects in this set are also exported.

Object type

An object type (or MetaClass) is that part of the database containing objects of a given type. The objects created are stored in the repository according to their type. Segments are used when searching for objects in the repository and when the metamodel is extended to include a new type of object. Example: message, org-unit, etc.

object UI access

Object UI access defines user rights on creation, reading, update, and deletion on these objects and their tools. By default, object UI accesses have value *CRUD (C: create, R: read, U: update, D: delete, *: default value).

person

A person is defined by his/her name and electronic mail address.

A person can access **HOPEX** once the administrator assigns him/her a login and a profile.

The list of persons can for example come from an LDAP server.

Person group

(Web Front-End specific) A person group groups persons in a group. These persons share the same connection characteristics.

private workspace

A private workspace is a temporary view of the repository in which the user is working before dispatching his/her work. This view of the repository is only impacted by the changes of the user, and does not include concurrent modifications made by other users. This private workspace exists until it is refreshed, dispatched or discarded. Note that a private workspace is kept when the user disconnects from the repository, unless the user indicates otherwise. A user can see modifications dispatched by other users of this repository without dispatching his/her own modifications. To do this, the user refreshes his/her private workspace. The system then creates a new private workspace, and imports the logfile of his/her previous modifications into it.

private workspace log

The private workspace log contains all modifications made by a user in his/her private workspace. It is applied to the repository at dispatch, then automatically reinitialized. This log is stored in the EMB private workspace file.

profile

A profile defines what a person can see or not see and do or not do in tools, and how he/she sees and can do it. The profile defines options, access rights to repositories and products, read/write and read-only rights on objects.

All users with the same profile share these same options and rights. A user can have several profiles. A profile is available for all repositories in a single environment.

profile assignment

The profile assignment defines the following for each person: the repository concerned by the assignment, access rights to the repository, the validity period of the assignment, (optional, with access to the repository in read only) connection repository snapshot

protection

When several users work on the same project, it is important to provide them with the means to work on a new part of the project without inadvertently interfering with previous work. To do this, you can protect the objects concerned by assigning to them a writing access area (**HOPEX Power Supervisor** technical module). It is then possible to connect these objects to others, if the link does not modify the nature of the object. The link orientation determines whether or not the nature of the object is affected.

publish

Dispatching your work allows the other users to see the changes you have made to the repository. They will see these when they open a new workspace, either by dispatching, refreshing or discarding their work in progress

query

A query allows you to select a set of objects of a given type using one or more query criteria. Most of the MEGA software functions can handle these sets. For example, you can use a query to find all enterprise org-units involved in a project.

reading access

see "reading access area".

reading access area

The user reading access area corresponds to the view the person or person group has of the repository: it defines objects that can be accessed by the person or person group.

reading access diagram

The reading access diagram enables definition of reading access areas and their hierarchical organization. This diagram also enables creation of users and their association with reading access areas.

reflexive link

A reflexive link is a link between two objects of the same type, for example: the link between projects that allows you to define sub-projects.

refresh

Refreshing his/her private workspace allows the user to benefit from changes dispatched by other users since creation of his/her workspace. In this case, it keeps repository modifications carried out by the user without making these available to other users. The system creates a new private workspace and the logfile of previous changes is imported into it.

reject file

When updating a repository (importing, restoring, dispatching), a reject file is created in order to store rejected commands. Rejected commands are stored with the reason for which they were rejected. These are found in the "MegaCrd.txt" file located in the environment folder.

reorganization

Reorganizing a repository consists of executing a logical backup of the repository, reinitializing it and reimporting the logical backup (without log).

report (MS Word)

Reports (MS Word) managed by **HOPEX** are objects allowing you to transfer written knowledge extracted from the data managed by the software.

report (MS Word) element

A report (MS Word) element is the instancing of a report template (MS Word) element. It is the result, formatted in the word processing software, of execution of the query defined in the report template (MS Word) element.

report file

The environment report file, MegaCrdYYYYmm.txt (where YYYY and mm represent year and month of creation) indicates all administration operations (backup, export, restore, controls, etc.) carried out in the environment. The report file is stored in the user work folder associated with the environment system repository: SYSDB\USER\XXX\XXX.WRI where XXX is the user code.

report template (MS Word)

A report template (MS Word) is a structure with characteristics that may be reproduced when producing reports (MS Word). A report (MS Word) can be created and modified as many times as necessary; however to produce several reports (MS Word) of the same type, use of a report template (MS Word) is recommended.

A report template (MS Word) provides the framework for the report (MS Word), which is completed with data from the repository when the report (MS Word) is created. A template contains the formatting, footers, headers and text entered in MS Word. It also contains report template (MS Word) elements that allow you to format data from the repository. It allows you to produce reports (MS Word) associated with main objects of the repository.

report template (MS Word) element

A report template (MS Word) element is the basic element of a report template (MS Word). It comprises a query which enables specification of objects to be described and a descriptor for their formatting. When creating a report (MS Word) from a report template (MS Word), each report template (MS Word) element is instanced by a report (MS Word) element.

repository

See *repository*.

repository

A repository is a storage location where HOPEX manages objects, links, and inter-repository links.

The main part is managed by a database system (GBMS, SQL Server, or Oracle). The remainder is in a directory tree (content of Business Document versions, backup logfiles, locks.

The different users in the environment can access the repositories connected to it.

repository log

The repository log stores all the updates of users working in a repository. It is reinitialized at repository reorganization , or by selecting **Repository Log > Manage Repository and Object Log** in the repository pop-up menu. This logfile is stored in the .EMB file of the repository.

repository snapshot

A repository snapshot identifies an archived state of the repository.

Creating a repository snapshot allows you to label important states in the repository life cycle.

The repository archived states for which a snapshot exists are not deleted by repository cleanup mechanisms (Repository history data deletion).

restore

A physical restore consists of copying previously saved repository files.

saving

The work done in a session is saved when you request it, or when you exit. By default, the software executes an automatic save (the time interval between two saves is specified in the options: **Options > Repository > Background Automatic Save**). Messages appear asking you to confirm saving the changes you made in each open report template (MS Word) or report (MS Word) (except during the automatic save). It is recommended that you regularly save your session to avoid losing your work if your computer locks up or loses power.

session

A session is the period during which a user is connected to a repository. A session begins when the user establishes connection and ends when he/she exits **HOPEX**. Sessions and private workspaces can overlap. When you dispatch, refresh or discard a private workspace, a new private workspace is created in the same session. Conversely, a user can keep his/her private workspace when exiting a session.

set

A set is a collection of objects with common characteristics. For example, you can build a set of messages sent or received by a certain org-unit in the enterprise. These sets are usually built using queries, and can be handled by most functions of the software.

setting

A setting is a parameter of which value is only determined when the function with which it is associated is executed. You can use variables to condition a query (**HOPEX Power Studio** technical module). When you execute this query, a dialog box asks you to enter these settings, with a box for each setting defined in the query.

site

A site groups together everything that is shared by all **HOPEX** users on the same local network: the programs, standard configuration files, online help files, standard shapes, workstation installation programs, and version upgrade programs. The site is installed on a local network resource or on each workstation if you are working without a network connection.

snapshot

See *repository snapshot*

style

A style is a particular format applied to a paragraph of text in a word processor. Styles allow you to systematically apply formats such as fonts, margins, indents, etc. Several styles are available for report (MS Word) configuration. These styles have the M- prefix and are based on the M-Normal style that is similar to the Word Normal style. Note that the M-Normal style text is in blue. All these styles are contained in the Megastyl.dot style sheet.

SystDb

SystDb is a particular repository containing the metamodel and technical data (descriptors, Web site templates, queries, etc.). The metamodel and technical data are common to all repositories in the same environment. Definition of users and their rights are stored in this repository, essential for operation of the software.

system repository

See *SystDb*.

text

You can associate text with each object found when browsing object descriptors (**HOPEX Power Studio** technical module). This text is formatted for MS Word. It presents what will be displayed for each of the objects in the generated report (MS Word). In the text you can insert the object name, its characteristics, and its comment. You can also insert the characteristics of other objects linked to it.

trace file

The trace file (Megaerr*.txt) is accessible via menu **Help** or from the **HOPEX Server Supervisor** tool (available in the **Utilities** folder). It traces all problems and errors encountered on the workstation. Technical support may ask you to check this file.

user

A user is a person with a login.

A user is authorized to access certain functions of the product and certain repositories. Each user has a specific desktop in each database, and can connect to this desktop from any workstation in a given environment.

The code associated with the user is used to generate file names as well as a specific work folder for the user.

By default at installation, Administrator persons (Login: System) and Mega (Login: Mega) enable administration of repositories and creation of new users.

work folder

Each user has a work directory in each repository that he/she uses. This directory is located in sub-directory User\XXX of the repository (XXX represents the user code).

workstation

A workstation is defined for each computer connected to the environment. A workstation contains programs and a configuration file that allow you to use **HOPEX** on that machine.

writing access

see "writing access area".

Writing access area

A writing access area is a tag attached to an object to protect it from unwanted modifications. Each user is assigned a writing access area. There is a hierarchical link between writing access areas. A user can therefore only modify objects with the same or lower authorization level. The structure of writing access areas is defined in the writing access diagram. By default there is only one writing access area, "Administrator", to which all objects and users are connected. Writing access management is available with the **HOPEX Power Supervisor** technical module.

writing access diagram

The writing access diagram is available if you have the **HOPEX Power Supervisor** technical module. With this diagram, you can create users and manage their writing access rights for repositories and product functions. By default only one writing access area is defined, named "Administrator". Attached to it are the "Administrator" and "Mega" persons. This is the highest writing access level and it should normally be reserved for repository management. You cannot delete this writing access area.

Technical Articles (EN)

Web connection overloading and configuration MEGA HOPEX 1.0 EN

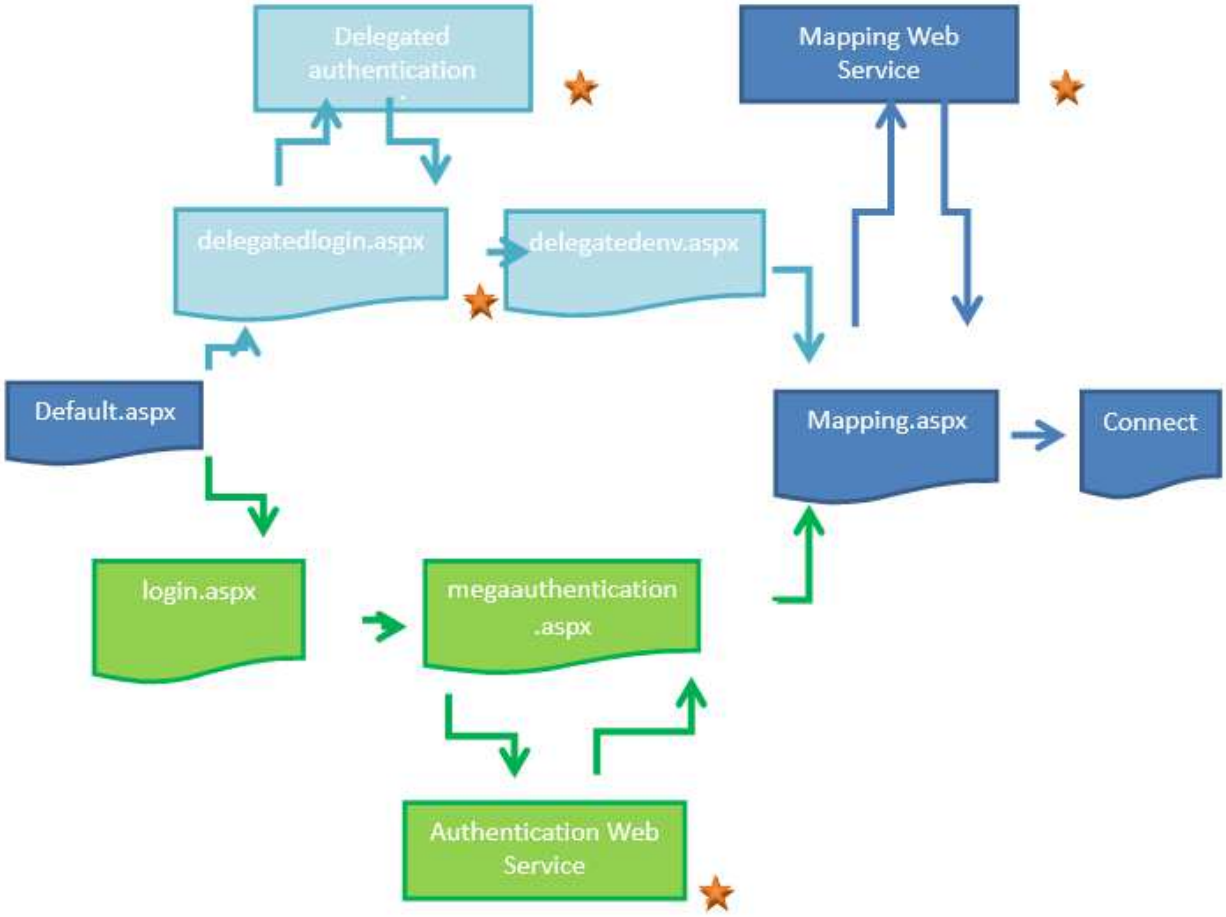
This technical article describes how to overload and configure web connection. This includes:

- the authentication and mapping services: this enables using non standard algorithms to authenticate or map users, or even to use centralized authentication, enabling Single Sign On scenarios.

- timeout and preload configuration: this enables improvements in first connection response times

AUTHENTICATION AND MAPPING PRINCIPLES

The authentication and mapping process is as follows:



The ★ sign denotes components whose overloading is explained in this document.

The administration of standard authentication and mapping (as provided by the standard web services) is explained in the product documentation.

> You should never modify any of the files provided except as noted in this document. Any other modification would be unsupported and could be erased by any patch or update.

OVERLOADING THE AUTHENTICATION WEB SERVICE

Defining the web service to call

By default, authentication is provided by a Mega Web service. The web service to call can be changed by editing the following entry in the Web.Config file:

```
<add key="AuthentificationWebService" value=""/>
```

If the entry is not present or its value is empty, the default service is used. If you wish to use your own service, you should provide its URL here. For example:

```
<add key="AuthentificationWebService"
value="https://myservices.example.com/mywebservice.myextension"/>
```

> As this service receives the password typed by the end user, use of the HTTPS (HTTP over SSL/TLS) protocol is highly recommended, especially if the service is not on the same machine or network as the Mega IIS Web server.

Web service interface to implement

The web service is called using POST with a JSON in the body of the request and should send a JSON back.

The calling JSON has the following format:

```
{"WebUserName":"sUser","WebUserPwd":"sPasswd","Environment":"sEnv","AuthToken":"sAuthToken",
"ContextHash":"sContextHash"}
    - WebUserName: the login that was entered by the user
    - WebUSerPwd: the password that was entered by the user
    - Environment: the environment path or Id
    - AuthToken: a token if the user was previously authenticated and did not re-enter its
      passoword and name
    - ContextHash: a context hash identifying the computer from which the connection has
      been received
```

The response JSON must have the following format:

```
{"Error":"sError","IsAuthenticated":"bIsAuthenticated","AuthenticatedWebUser":"sAuthenticate
dWebUser","AuthenticatedLogin":"sAuthenticatedLogin","PassdIsExpired":false,"AuthToken":"sAu
thToken"}
    - Error: an error message if IsAuthenticated is false
    - IsAuthenticated: is the person authenticated?
    - AuthenticatedWebUSer: the Person that was found
    - AuthenticatedLogin: the Login that was found
    - PasswdIsExpired: has the password expired? If yes, a password modification will be
      requested
    - AuthToken: a token that can be used to re-authenticate without login/password.
```

OVERLOADING THE MAPPING WEB SERVICE

Defining the web service to call

By default, mapping is provided by a Mega Web service. The web service to call can be changed by editing the following entry in the Web.Config file:

```
<add key="MappingWebService" value=""/>
```

If the entry is not present or its value is empty, the default service is used. If you wish to use your own service, you should provide its URL here. For example:

```
<add key="MappingWebService"
value="https://myservices.example.com/mywebservice.myextension"/>
```

Use of the HTTPS (HTTP over SSL/TLS) protocol is recommended, especially if the service is not on the same machine or network as the Mega IIS Web server.

Web service interface to implement

The web service is called using POST with a JSON in the body of the request and should send a JSON back.

The calling JSON has the following format:

```
{"WebUserName":"sMegaWebUser","Environment":"sEnv","isSSO":bIsSSO,"SSOPersonInformations":"s
SSOPersonInformations"}
    -   WebUserName: the login that was entered by the user
    -   Environment: the environment path, or Id.
    -   isSSO: do we come from a centralized auth service?
    -   SSOPersonInformations: if isSSO is true, describes extra informations on the person
```

The response JSON must have the following format:

```
{"Error":"sError","Person":"sPersonId","PersonGroup":"sPersonGroupId","Login":"sLogin"}
    -   Error: an error message, if no Person or PersonGroup was found
    -   Person: IdAbs of the mapped Person
    -   PersonGroup: IdAbs of the mapped PersonGroup
    -   Login: Login of the Person or PersonGroup
```

DELEGATING AUTHENTICATION TO A CENTRALIZED AUTHENTICATION SERVICE (SSO)

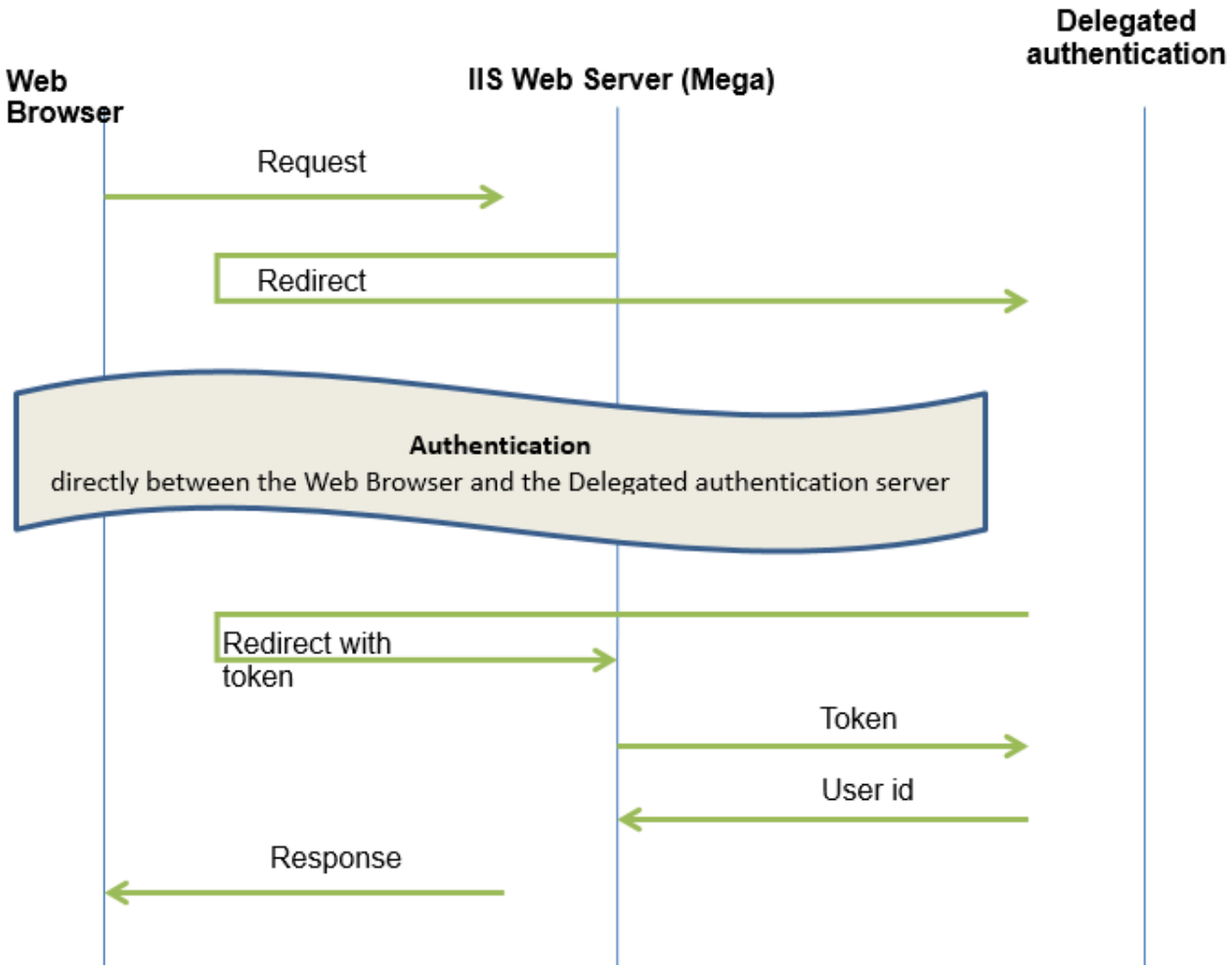
Delegated authentication principle

Delegated authentication means that the authentication of users is not handled by Mega but by a central authentication server shared between applications in your organization.

Many delegated authentication servers (e.g. ADFS, CAS, etc.) and protocols (e.g. SAML, WS-Federation, CAS, OpenID, etc.) exist and can be interfaced to the Mega web application by writing your own delegated authentication handler page.

Such a mechanism allows the setup of Single Sign-On (SSO) scenarios.

The general principle of delegated authentication is as follows:



Activating delegated authentication

Delegated authentication is activated in the Web.Config file of the web application. This file can also be modified through IIS Administration.

The following means that delegated authentication is disabled:

```
<add key="DelegatedLogin" value="0"/>
```

To enable it, replace this line with:

```
<add key="DelegatedLogin" value="1"/>
```

This configuration means that authentication will be handled by delegatedlogin.aspx rather than login.aspx.

The next step is to personalize delegatedlogin.aspx.cs to handle your delegated authentication protocol and server.

Delegated authentication samples

A few delegated authentication samples are delivered in the samples directory of the web application. These can serve as a starting point for writing your own delegatedlogin.aspx.cs file.

The following samples are available:

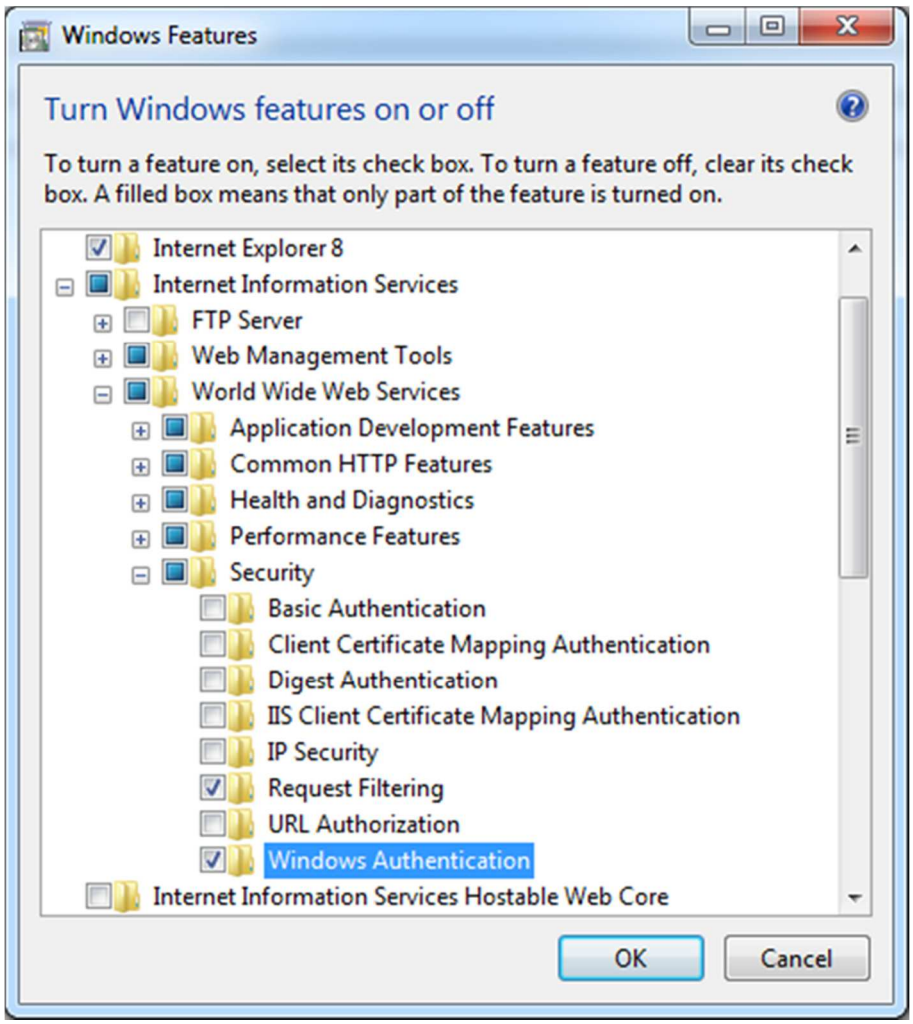
- delegatedlogin_cas.aspx.cs : delegated login using the CAS protocol
- delegatedlogin_passtrough.aspx.cs : pass-through login (no authentication)
- delegatedlogin_windows.aspx.cs : authentication using .Net Windows authentication

To use them, duplicate the chosen sample in the main directory of the web application and edit it. Do not forget to reference it in the delegatedlogin.aspx file.

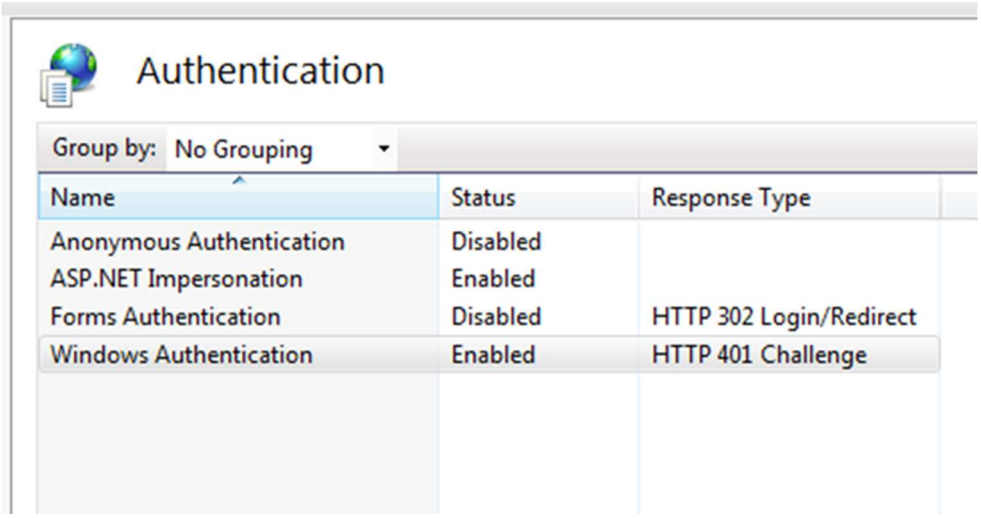
Windows .Net authentication

This authentication type requires extra configuration:

a) Add the “Windows Authentication” functionality to Windows Features (from Control Panel) :



b) Enable Windows Authentication for the site where Mega is installed in IIS, and disable anonymous authentication :



c) Add Windows authentication in the Web.config file, by replacing “None” by “Windows” :
<authentication mode="Windows" />

Writing your own delegated authentication service

Your delegatedlogin.aspx.cs must be written in C# .Net 4.0 and must provide at least the following services:

- When authentication is successful:
  - Set the session variable "AuthenticatedWebUser" with the name of the authenticated user
  - Redirect the browser to the delegatedenv.aspx page
- When delegated authentication fails:
  - Handle the error, for example by redirecting to the Mega login page, or by presenting the error directly to the end user

Here is a sample minimum implementation:

```
public partial class delegatedlogin : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
    …
    if (isAuthenticated)
    {
      Session["AuthenticatedWebUser"] = sMyAuthenticatedUser;
      Response.Redirect("delegatedenv.aspx");
    }
    else
    {
      Session["Error"] = "Delegated login failed";
      Response.Redirect("login.aspx");
    }
  }
}
```

You can also decide to provide more information to Mega about the user that is logging in. This information will be used to create the corresponding Person object in Mega. In order to do this, you can use the following code:

```
Dictionary<string, string> PersonInformations = new Dictionary<string, string>();
PersonInformations["idAbsOfTheAttribute1ToFillInOnPerson"] = "ValueOfTheAttribute1";
PersonInformations["idAbsOfTheAttribute2ToFillInOnPerson"] = "ValueOfTheAttribute2";
Session["PersonInformations"] = PersonInformations;
```

This allows you to give values to the attributes of your choice on the Person that will be created. It therefore only applies to the first connection of a given user. On subsequent connections, it will be ignored.

IMPROVING MEGA ADVISOR FIRST CONNECTION RESPONSE TIME

At the first Advisor connection, four processes are launched. By default, these processes are stopped, without requests, after 20 minutes of idle time.

The time spent to launch these processes slow down the connection to Advisor. To optimize the first connection time to the application you can configure IIS and the application to preload required processes and prevent them to stop.

To start with MEGA Advisor in optimized conditions, you must:

- Configure IIS and application timeout, the time with no activity before stopping mgwmwas.
- Configure Advisor to preload the Mega Sub Process needed for each mega profiles.
- Plan the preload using Windows Scheduler.

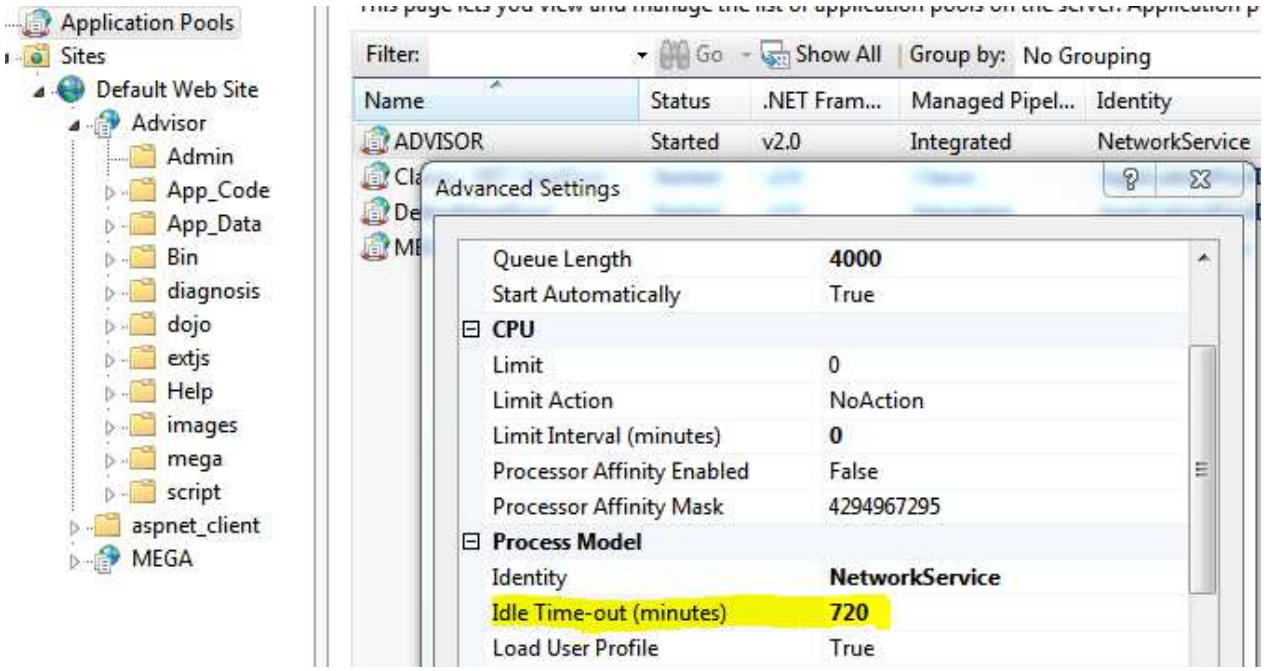
Timeout configuration

The default timeout configuration is 20 minutes. To prevent the mgwmwas process to shut down during a day, modify the default timeout to 12 or 24 hours.

Configuring IIS Application Pool timeout

From IIS:

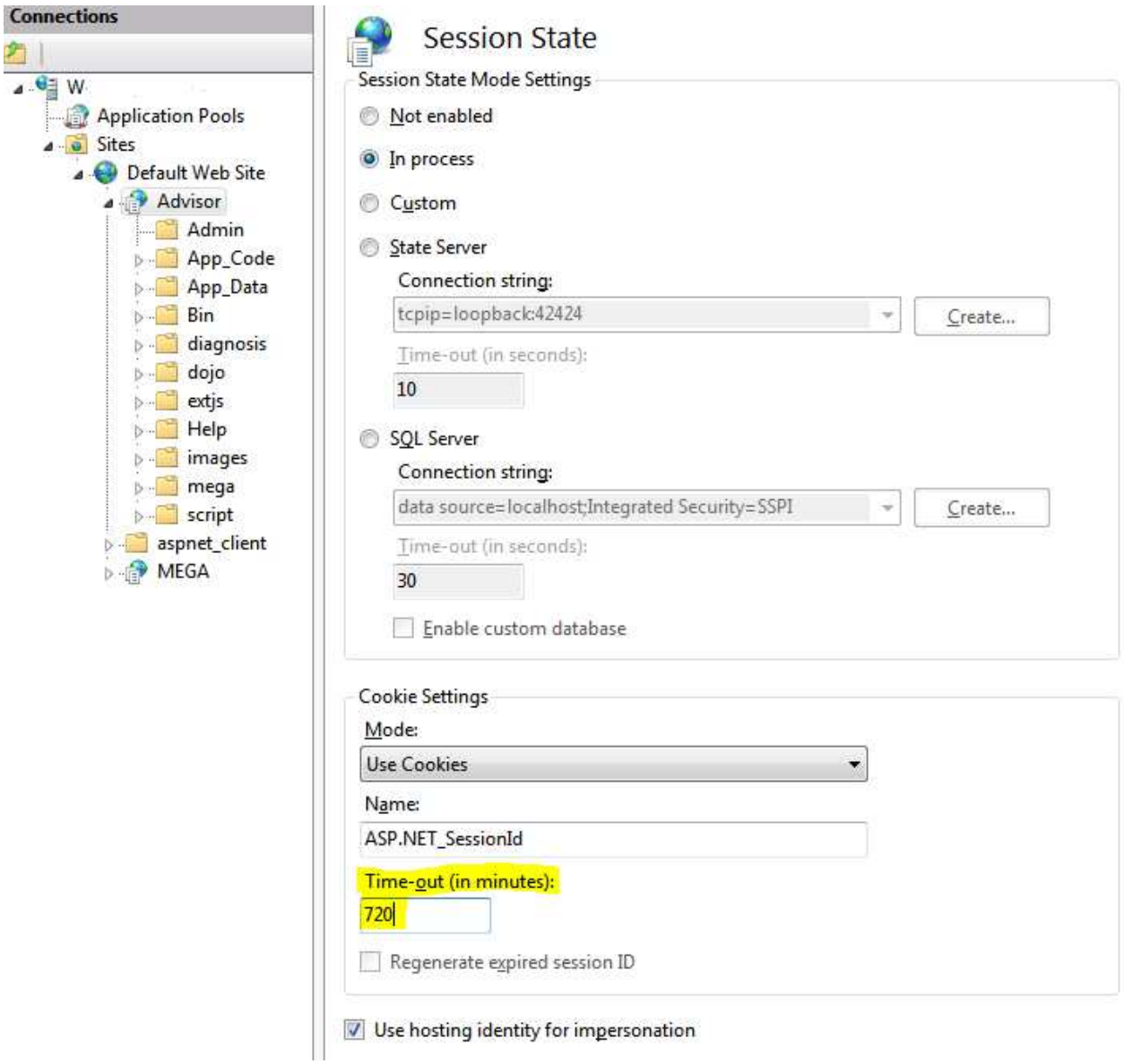
1. In the left pane, from the tree view select **Application Pools**.
2. In the right pane, right-click **ADVISOR** application and select **Advanced Settings**.
3. From the **Advanced Settings** window, in the **Process Model** part, modify the **Idle Time-out (minutes)** value.



Configuring Advisor ASP.NET Session State timeout

From IIS:

1. From the left pane, expend the **Sites** tree view and select the **Advisor** Application.
2. In the right pane, open the ASP.NET **Session State** feature.
3. Modify the **Time-out (in minutes)** value.
4. In the **Actions** pane, click **Apply** to validate your modifications



Configuring Advisor session timeout

From the root folder of the Advisor deployment path:

- In the Web.Config file, update the OpenSessionTimeout value.
  This value is given in milliseconds.

```
   <add key="OpenSessionTimeout" value="43200000"/>
   <!--      OpenSessionTimeout: Session Timeout in ms. Default value 1800000 (20 Mn)
-->
```

Values in the Web.config file can also be configured through IIS Application Settings

Preload profiles configuration

Configuring the KeepAlive value

By default a Mega Profile remains connected, and its associate Mega Sub Process alive, even when the last user using the profile has gone. The configuration parameter that controls this behavior is located in the Web.Config file.

- Make sure you have the following line in your Web.Config file:

```
<add key="KeepAlive" value="1"/>
<!--      KeepAlive:  0 Or 1 : 1 to keep mega sub process alive-->
```

Configuring the Preload profiles

Configuring the preload profiles consists of code instructions to open session for a user for each profile to preload and instructions to discard each preloaded environments.

1. From the **App_Code** folder, open the **preload.cs** file.
2. Modify the **Preload** method according to the following model:

<environment Path> must be replaced by the environment path with duplicate backslash.

```
  static public bool PreLoad(MEGAWebAccessLib.MegaWebAccess pMWA , settings
pSettings , int nLevel, bool bDiscarding)
    {
bool bPreloading = true;
if (bDiscarding && (nLevel == 1))
{
  pMWA.Discard(1, "<environment_1 path>");
  …
  pMWA.Discard(1, "<environment_n path>");
}
string[] sDiagramArray = new string[<nd - number of diagram type to
preload>];
----
sDiagramArray[0] = "<hexaidabs of a diagram for the first diagram type>";
…
sDiagramArray[<nd-1>] = "<hexaidabs of a diagram for the last diagram type>";

switch (nLevel)
{
  case 1:
```

```
                    PreLoadProfil(pMWA, pSettings,
"<LANGID>",
"<Environment Path with double backslash ie:
C:\\Users\\Public\\Documents\\MEGA 2009 SP4\\Demonstration>",
"<Base Name>",
"<APPID>",
"<DESKID>",
"<PERSONID>",
"<LOGINID>",
"<PROFILID>",
sDiagramArray);


…
  case <n>:
                    PreLoadProfil(pMWA, pSettings,
"<LANGID>",
"<Environment Path with double backslash ie:
C:\\Users\\Public\\Documents\\MEGA 2009 SP4\\Demonstration>",
"<Base Name>",
"<APPID>",
"<DESKID>",
"<PERSONID>",
"<LOGINID>",
"<PROFILID>",
sDiagramArray);
  break;
  default:
bPreloading = false;
  break;
}

return bPreloading;
}
```

Administration page

The administration page in the admin folder calls the Preload method when resetting cache with false for bDiscarding. Then only the selected environment will be discarded.

Otherwise to start the preload call the administration page with preload parameter set to 1 (see Preload planning p. 14).

Preload planning

You can plan the preload to have profiles loaded before the first user connection.

From the Windows Scheduler:

1. Create a new task.
2. Configure the trigger.

   Ex.: everyday at 6 AM.
3. Configure the action:

   a. Select **Start a program**.

   b. In the **Parameters** pane, **Browse** the mwsr.exe program (located in the Utilities\MEGA ldap folder).

   c. In the argument field set URL of the administration page with preload parameter set to 1, for instance :

```
"http://localhost/Advisor/admin/default.aspx?preload=1"
```

The preload start discarding environment which reset cache.

HOPEX – Securing the platform

INTRODUCTION

The goal of this document is to give detailed instructions as to how to secure the HOPEX platform. This document is valid for HOPEX V1R2, V1R3, and V2 versions.

Activating SSL on the website

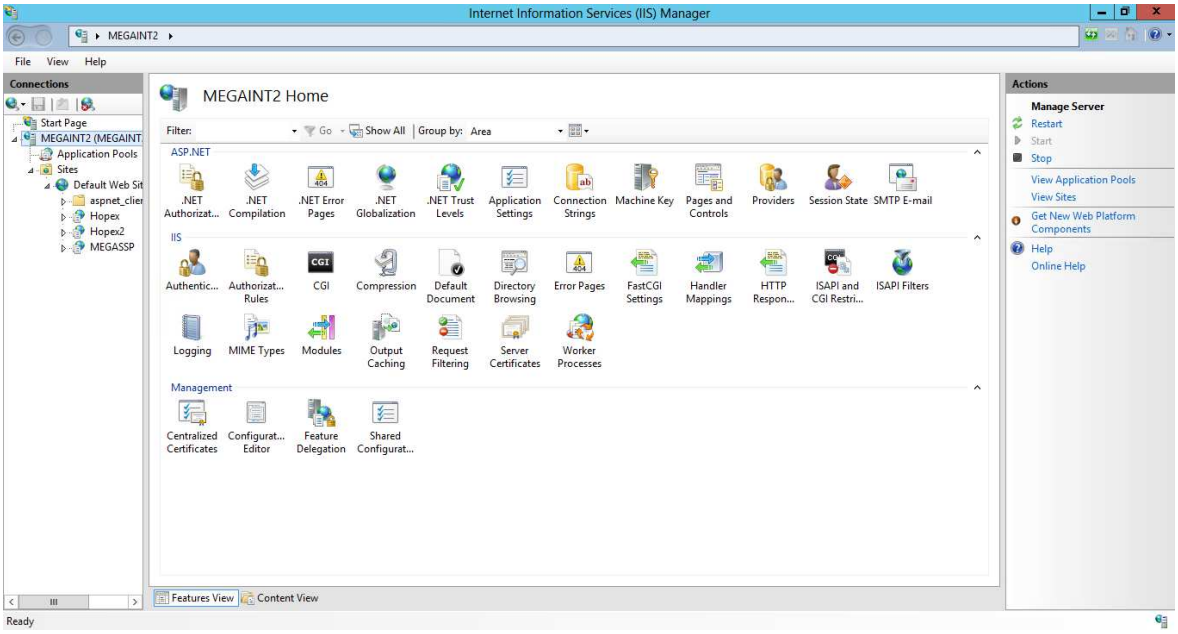
In order to encrypt the information sent between the users' workstation and the web server, the first step is to activate the SSL on the Hopex website.

The minimum level of certificate that is recommended is a certificate signed by a trusted third party. If internally, a root certificate exists and allows the applications to generate their own certificate, it is also a possibility.

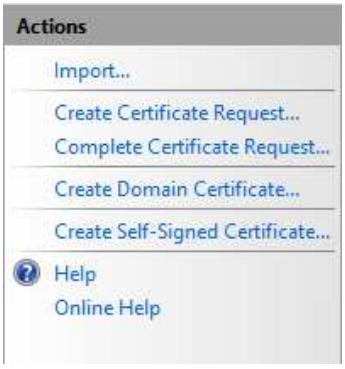
When you obtained that certificate, the steps to import it on your website are multiple. You will find hereafter one way to do it, and make sure that your website will be attainable only through HTTPS.

The following step-by-step procedure was done on a Windows Server 2012, with IIS 8.0. You need to adapt it based on your OS version.

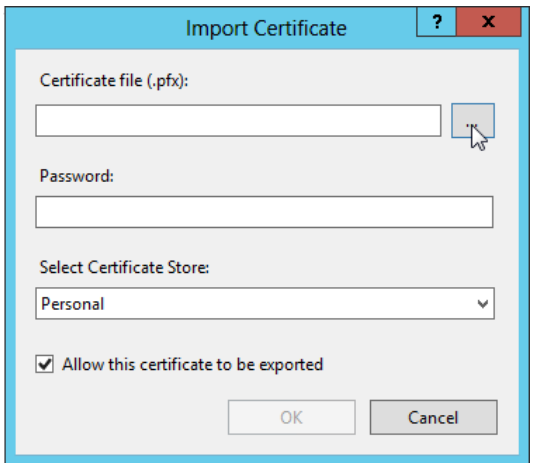
1. Copy the certificate on your server.
2. Open the "Internet Information Services (IIS) Manager" through the Administrative Tools of the server.
3. Go to the root of IIS :



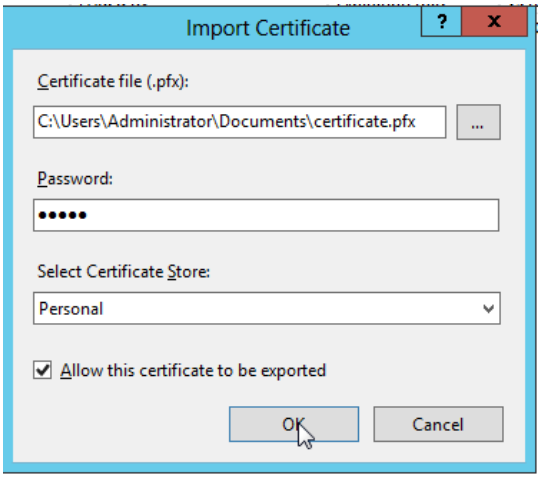
4. Double-click « Server Certificates ».
5. Once this feature is opened, on the right pane, click on the « Import » link :



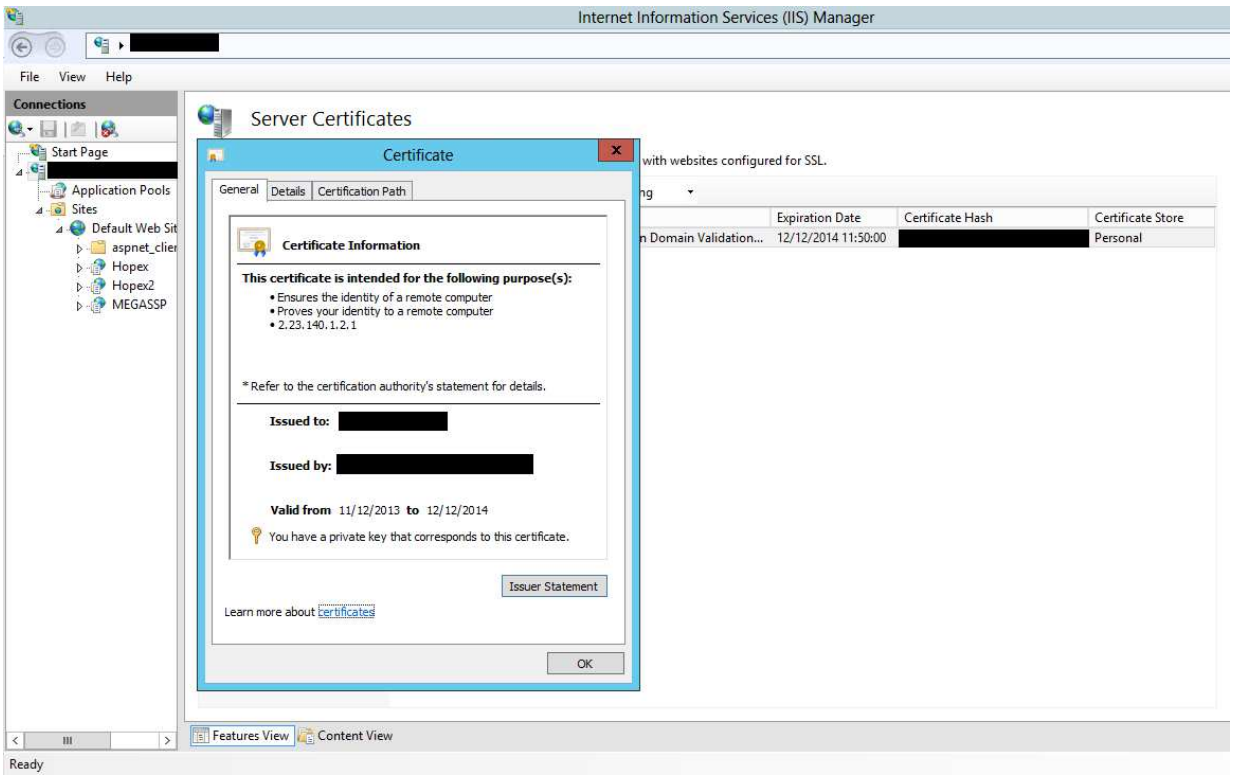
6. Click the button where the cursor is located, in order to look for the certificate :



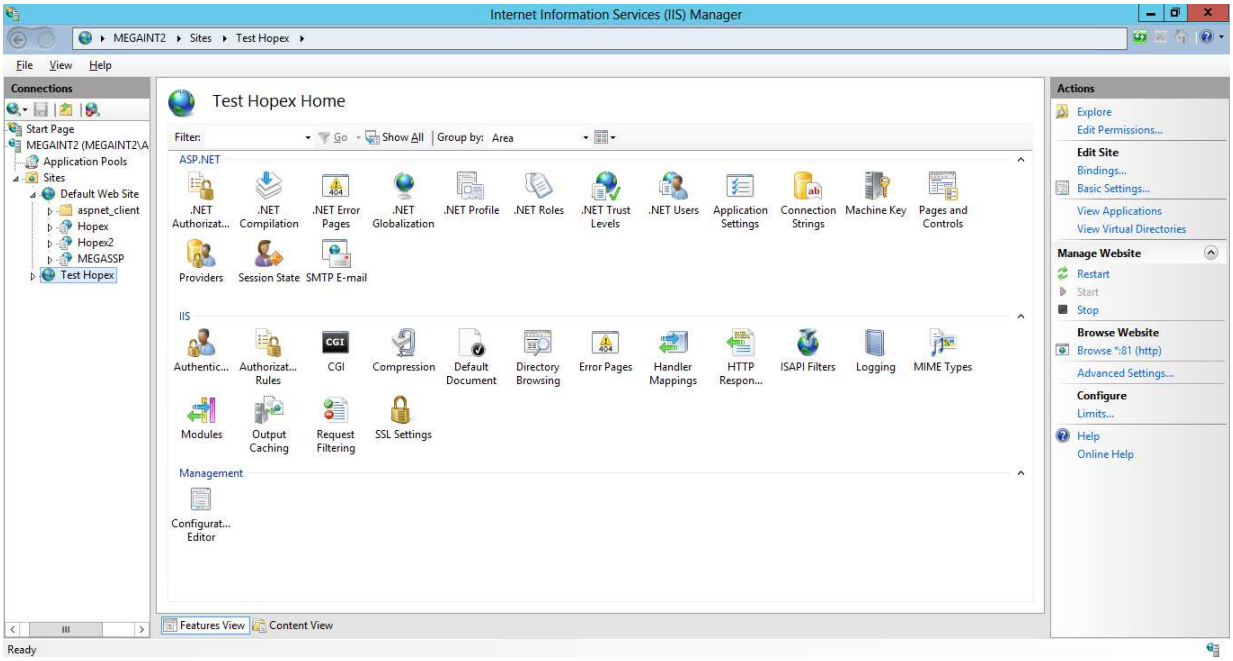
7. Select the certificate. Provide a password if necessary, and click « OK » to import the certificate :



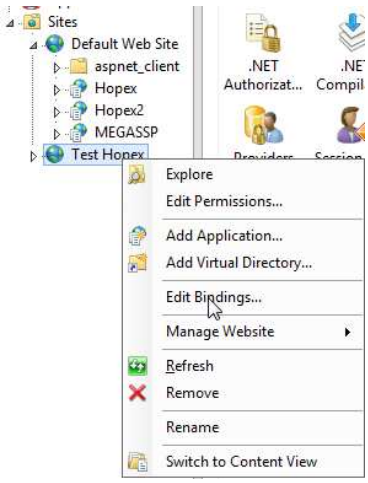
A new line will appear in the “Server Certificates” window. You can have a look at the imported certificate by double-clicking on the certificate :



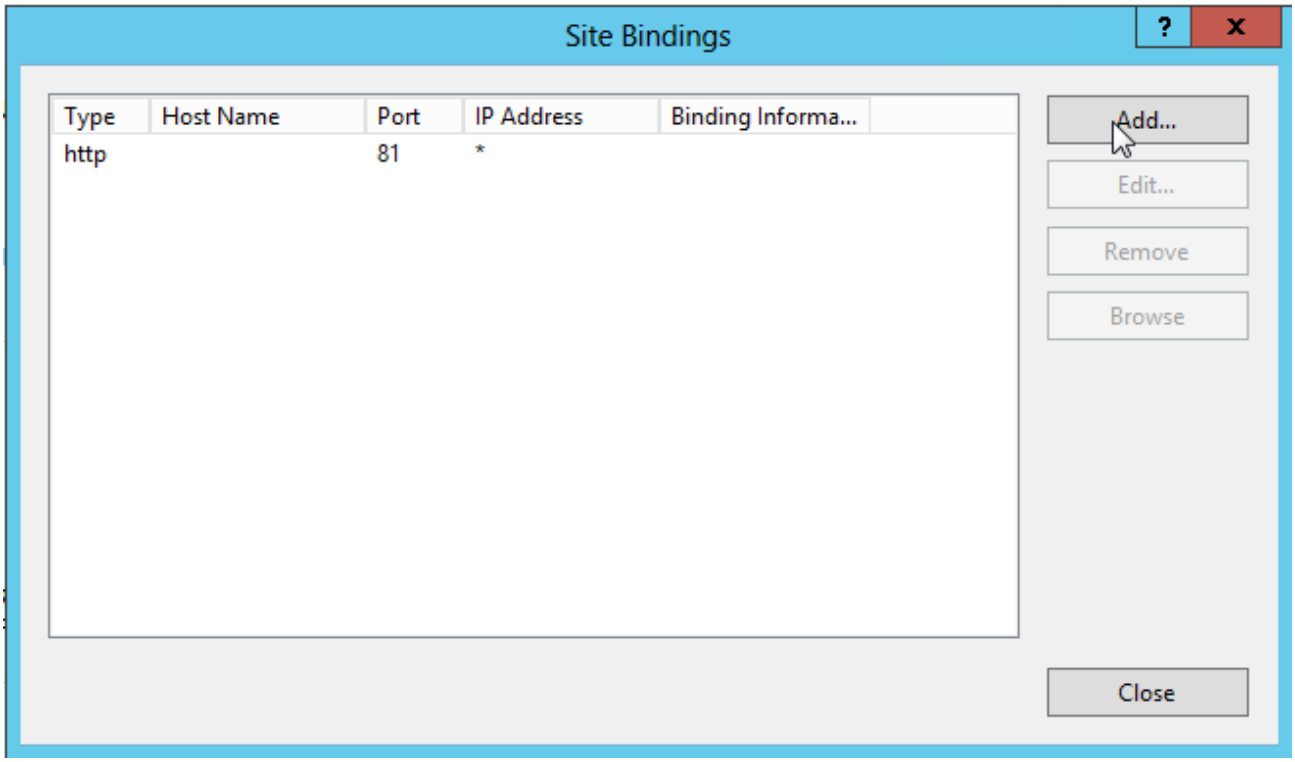
8. Locate the website where you want to create an HTTPS binding with the newly imported certificate. In this example, it is done on the "Test Hopex" website. Click the website to see its features.



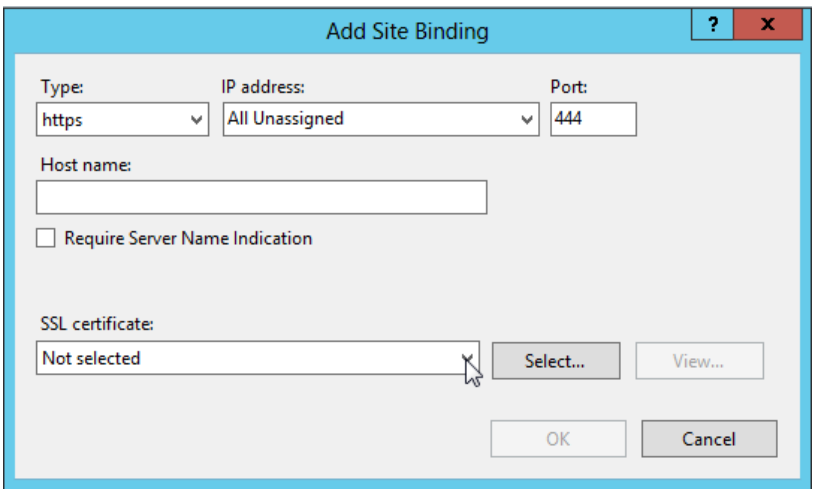
9. Right-click the website and select **Edit Bindings**.



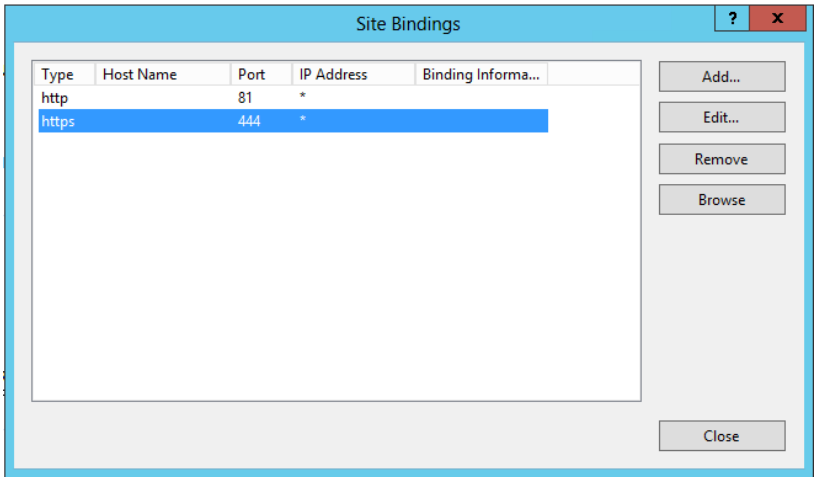
10. Add a new binding, for HTTPS, on a specific port (in this example, as the port 443 is already used, we will choose 444). Use the default port 443 whenever possible, as it permits you quite easily to make the URL of the website user-friendly :



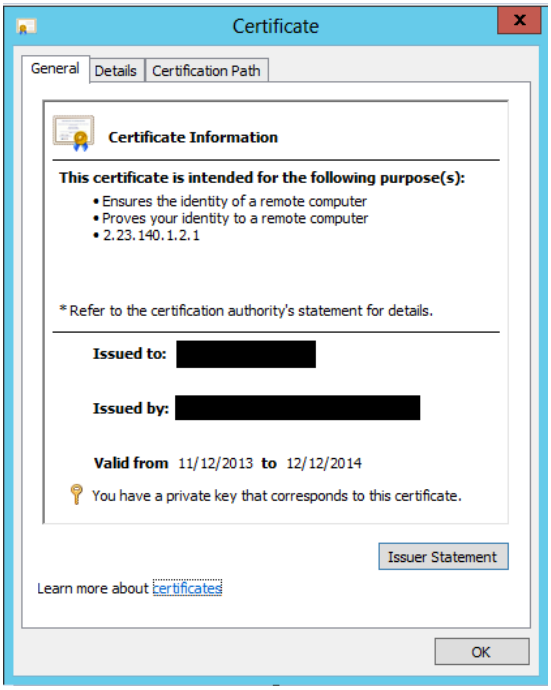
11. Use the dropdown list to select your certificate and click **OK**.



12. You see that a new binding exists. You can click **Close**.

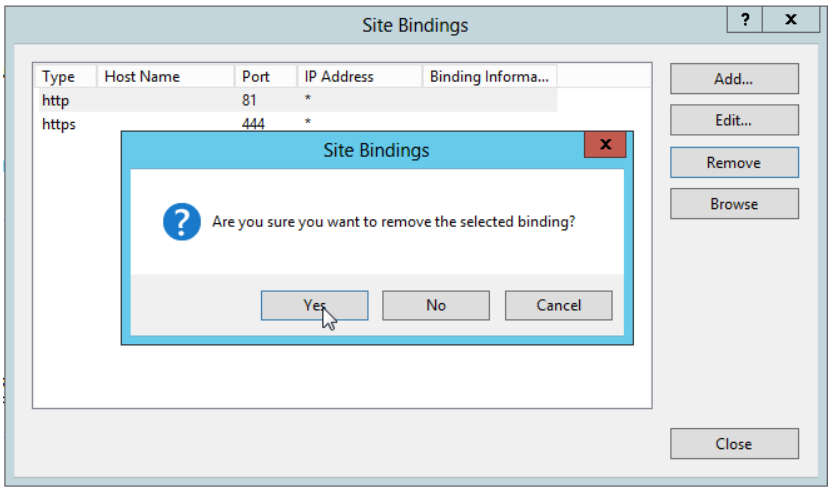


13. Install the application. Choose the appropriate website (here “test hopex”), and let the installation completes.
14. Whenever a URL is requested in the installation steps, make sure that you provide the HTTPS link, with possibly the port number, and that the address is relevant with the “Issued To” parameter when you open the certificate. Otherwise, you will receive some error messages in the application:

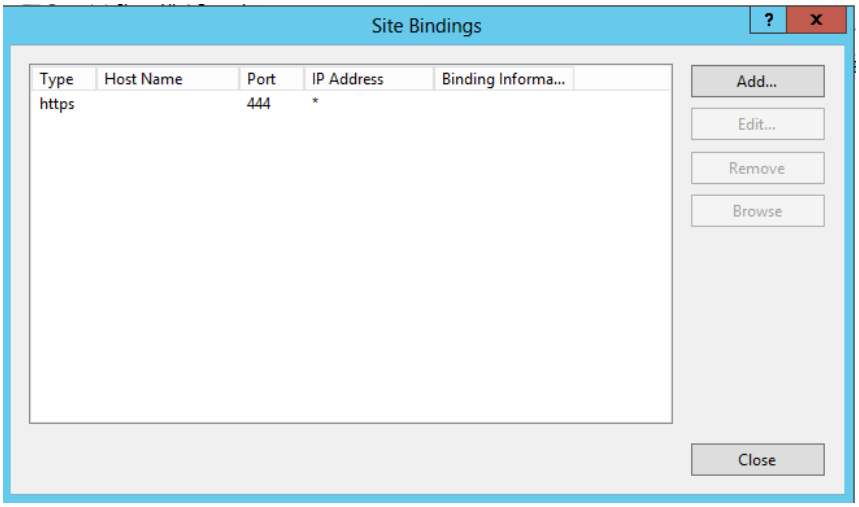


15. Once this is done, go back to IIS to force the use of SSL for your website. To do that, start by going back to the previous window to edit the bindings.

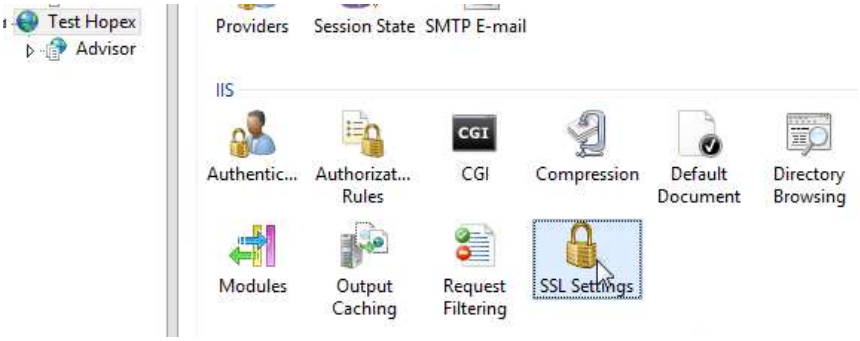
16. Select the "http" binding, and click **Remove**.



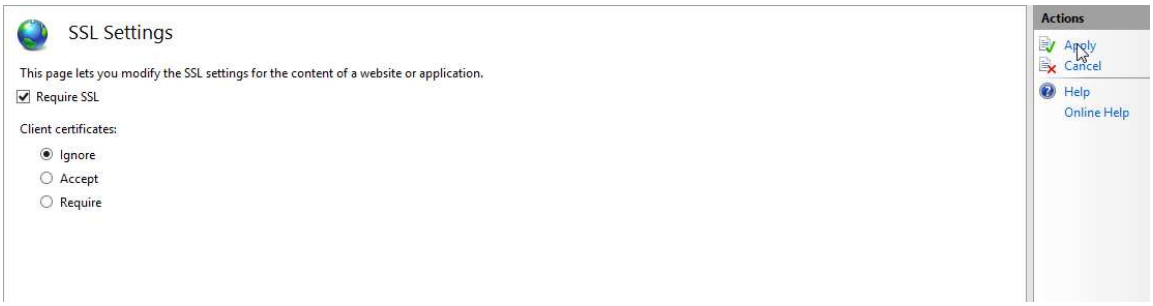
17. Check that only the https binding remains, and click **Close**.



18. Select your website, and open the « SSL Settings » feature :



19. Click « Require SSL » and apply :



20. Install the Hopex Web front-end application on your server.

21. Last step, after the installation, is to check that the configuration files of the application all contain the proper URL. You can also install the application without SSL, and then decide to activate it. In that second scenario, you need to update the following files. If you installed with SSL activated, you just need to check the configuration with the next steps. Two locations contain such strings: the *web.config* file of the "Hopex" web application, and the *MegaSite.ini* in the "Cfg" folder of the installation folder of the application:

a. In the MegaSite.ini file, the URL, WebApplicationPath, and WebApplicationPath2 parameters need to reflect that use :

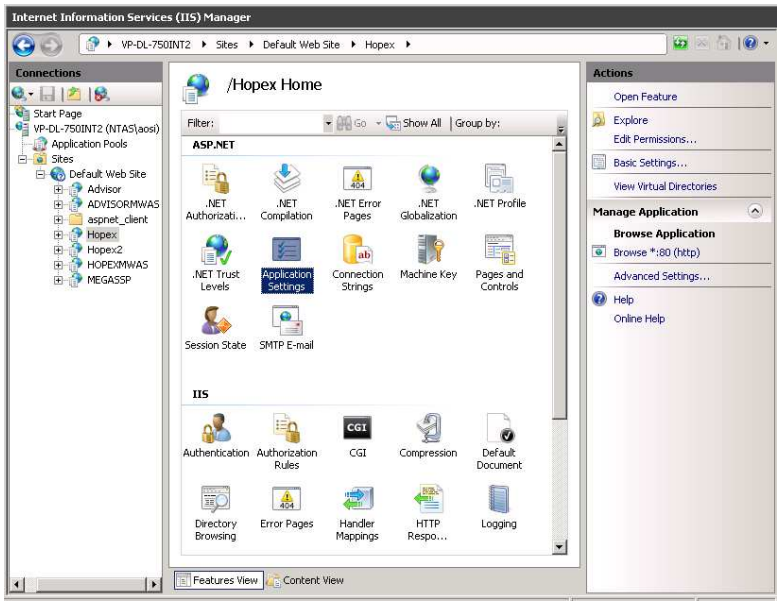
```
MegaSite.ini - Notepad
File  Edit  Format  View  Help

[SSP]
SecurityKey=¦¢+-20371ED720EC91F1471D1F4F22D02ABB
Url=https://[redacted]/MegaSSP

[Environment Shortcuts]
ECC7ECDC529E4C2F=D:\Environments\MEGA HOPEX V1R1\Demonstration

[System]
language=US

[General]
company=
sitename=Windows User

[LAN]
clusterroot=

[Must licence]
Path=\\[redacted]\Must\

[HOPEX]
WebApplicationPath=https://[redacted]/hopex
WebApplicationPath2=https://[redacted]/hopex
ErrorManagement=2

[SQL SERVER CONFIG]
ServerName=¦¢+-21A1B17F662AF9073E2DCE47AD9D40A781743BDB158A2E66C906A1E65395F19B
UserLogin=¦¢+-6D74F6C6E9B6E19F644368A100469619
UserPassword=¦¢+-F2978CCE890840D14F6470275C2B390F
Parameters=
```

b. Application settings of the Hopex web application (two techniques), the parameters MegaSiteProvider, and MegaWebAccessProvider, also need to be switched to HTTPS with the proper port :

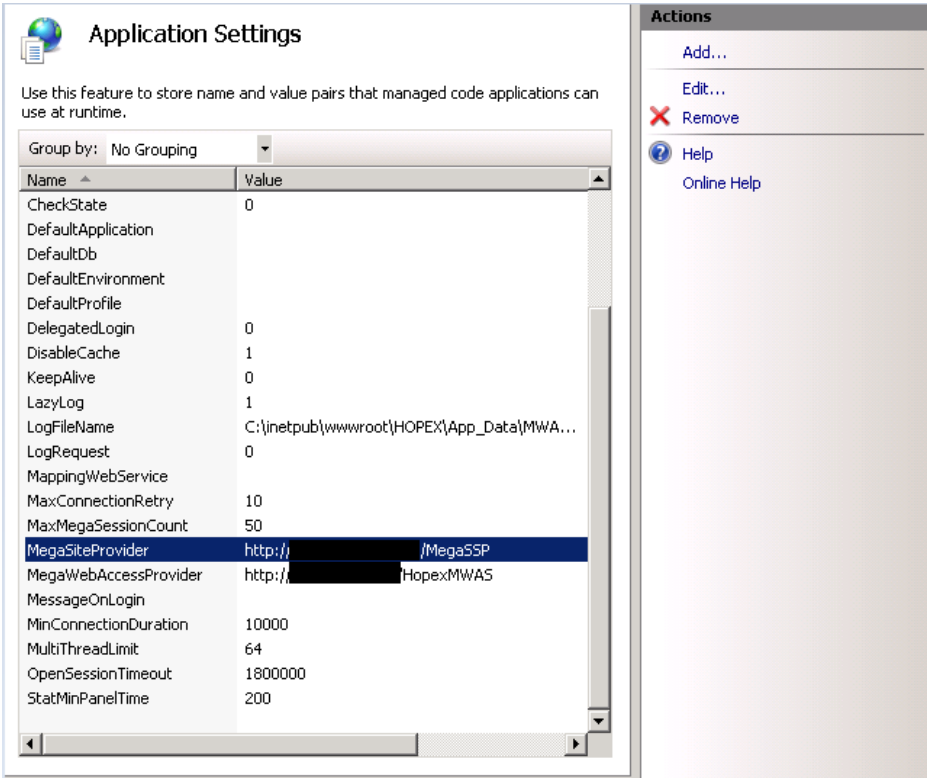
i. In the web.config file :

```
web.config - Notepad
File  Edit  Format  View  Help
    <add key="CheckState" value="0" />
    <!--      CheckState: 0 Or 1 : 0 deactivate the validity checking of the cache at a reconnection-->
    <add key="LazyLog" value="1" />
    <!--      LazyLog: 0 Or 1 : 0 deactivate package write in the log MWAS (attention,of perfomance probleme )-->
    <add key="DisableCache" value="1" />
    <!--      DisableCache:  0 Or 1 : 0 desactive cache-->
    <add key="AllowAnonymousConnection" value="0" />
    <!--      AllowAnonymousConnection: 0 Or 1 : 1 allows anonymous connection (without web user)-->
    <add key="LogRequest" value="0" />
    <!--      LogRequest: 0 Or 1 : 1 allows you to log all the requests made to MWAS-->
    <add key="MegaSiteProvider" value="https://[redacted]/MegaSSP" />
    <!--      MegaSiteProvider: URL of the Mega Site Provider, to be changed in cluster mode or in specific setups -->
    <add key="MegaWebAccessProvider" value="" />
    <!--      MegaWebAccessProvider: URL of the MegaWebAccessProvider -->
    <add key="DelegatedLogin" value="0" />
    <!--      DelegatedLogin: 0 (Mega Authentication) or 1 (Authentication delegated to another service through delegatedlogin.aspx) -->
    <add key="AuthentificationWebService" value="" />
    <!--      AuthentificationWebService: to define a non default authentification web service (e.g. "https://localhost/myservice?") -->
    <add key="MappingWebService" value="" />
    <!--      MappingWebService: to define a non default mapping web service (e.g. "https://localhost/myservice?") -->
    <add key="DefaultEnvironment" value="" />
    <!--      DefaultEnvironment: Environment used at connexion if none in URL -->
    <add key="DefaultDb" value="" />
    <!--      DefaultDb: Database used at connexion if none in URL -->
    <add key="DefaultProfile" value="" />
    <!--      DefaultProfile: Profile by default at connexion if none in URL  -->
    <add key="DefaultApplication" value="" />
    <!--      DefaultApplication: Application by default at connexion if none in URL -->
    <add key="KeepAlive" value="0" />
    <!--      KeepAlive:  0 Or 1 : 1 to keep mega sub process alive-->
    <add key="MessageOnLogin" value="" />
    <!--      MessageOnLogin: A specific message to show on the Login Page -->
    <add key="MegaWebAccessProvider" value="" />
    <!--      MegaWebAccessProvider: URL of the MegaWebAccessProvider, if accessed over HTTP(S). -->
    <add key="HideErrors" value="1"/>
  </appSettings>
```

ii. Through the « Application settings » feature in IIS :



You can select a line and click the **Edit** link on the right pane to update a URL :



Lastly, a you need to edit the « web.config » file of the HOPEX web application.

In the <system.web> element, add the following element:

```
<httpCookies requireSSL="true" />
```

An example of the file properly configured:

```
    </defaultDocument>
  </system.webServer>
  <system.web>
    <globalization fileEncoding="UTF-8" />
    <trace enabled="false" localOnly="true" pageOutput="true" traceMode="SortByTime" />
    <compilation debug="false" defaultLanguage="c#" targetFramework="4.0" />
    <authentication mode="None" />
    <!-- <authentication mode="Windows" /> -->
    <identity impersonate="true" userName="registry:HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\MEGA\HOPEX
    <httpRuntime requestValidationMode="2.0" requestValidationType="MegaRequestValidator" />
    <httpCookies requireSSL="true" />
  </system.web>
```

Note that in the latest CPs of the different versions of HOPEX, that line might already be present in the file, commented. Remove the comment brackets if it is:

```
<!-- <httpCookies requireSSL="true" /> -->
```

To

```
<httpCookies requireSSL="true" />
```

Save the file to apply that modification. IIS will automatically restart.

Disabling SSL v2, v3, TLS 1.0 and TLS 1.1

On Windows Server 2008 R2 and Windows Server 2012, the SSL V2 and V3 are activated by default. As there are known vulnerabilities with those, as well as with TLS 1.0, as well as TLS 1.1, we disable those, and then activate only TLS 1.2.

The steps are :

<u>For SSL V2:</u>

1. Through the Startup menu, go to "Run" and type :

   regedit.exe

2. Browse through the registry until you reach the following key :

   HKey_Local_Machine\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 2.0

3. In the "SSL 2.0" key, create a new one, of type DWORD (32 bit) with the following details :

   a. Name : Enabled

   b. Value : 0

4. If it does not exist, create a sub-key in "SSL 2.0" called "Client". Otherwise, go to 5. Directly.
5. In the key "CLIENT", create a DWORD (32 bit) with the following details:
    a. Name : DisabledByDefault
    b. Value : 1

<u>For SSL V3:</u>

1. Through the Startup menu, go to "Run" and type :

    regedit.exe

2. Browse through the registry until you reach the following key :

    HKey_Local_Machine\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0

3. In the "SSL 3.0" key, create a new one, of type DWORD (32 bit) with the following details :
    a. Name : Enabled
    b. Value : 0
4. If it does not exist, create a sub-key in "SSL 3.0" called "Client". Otherwise, go to 5. Directly.
5. In the key "Client", create a DWORD (32 bit) with the following details:
    a. Name : DisabledByDefault
    b. Value : 1
6. If it does not exist, create a sub-key in "SSL 2.0" called "Server". Otherwise, go to 7. Directly.
7. In the key "Server", create a DWORD (32 bit) with the following details:
    a. Name : DisabledByDefault
    b. Value : 1

<u>For TLS 1.0:</u>

1. Through the Startup menu, go to "Run" and type :

    regedit.exe

2. Browse through the registry until you reach the following key :

    HKey_Local_Machine\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols

3. Create the "TLS 1.0" key, and in that key, create a new entry of type DWORD (32 bit) with the following details :
    a. Name : Enabled

   b. Value : 0

4. Create a sub-key in "TLS 1.0" called "Server".
5. In the key "Server", create a DWORD (32 bit) with the following details:
   a. Name : DisabledByDefault
   b. Value : 1

<u>For TLS 1.1:</u>

1. Through the Startup menu, go to "Run" and type :

   regedit.exe

2. Browse through the registry until you reach the following key :

   HKey_Local_Machine\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols

3. Create the "TLS 1.1" key, and in that key, create a new entry of type DWORD (32 bit) with the following details :
   a. Name : Enabled
   b. Value : 0
4. Create a sub-key in "TLS 1.1" called "Server".
5. In the key "Server", create a DWORD (32 bit) with the following details:
   a. Name : DisabledByDefault
   b. Value : 1

<u>To **activate** TLS 1.2 :</u>

1. Through the Startup menu, go to "Run" and type :

   regedit.exe

2. Browse through the registry until you reach the following key :

   HKey_Local_Machine\System\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols

3. Create the "TLS 1.2" key, and in that key, create a new entry, of type DWORD (32 bit) with the following details :
   a. Name : Enabled
   b. Value : 1
4. Create a sub-key in "TLS 1.2" called "Client".

5. In the key "Client", create a DWORD (32 bit) with the following details:
    a. Name : DisabledByDefault
    b. Value : 0
6. Create a sub-key in "TLS 1.2" called "Server".
7. In the key "Server", create a DWORD (32 bit) with the following details:
    a. Name : DisabledByDefault
    b. Value : 0

Now we need to make sure that **the .Net layer will accept to use TLS 1.2**. To do that:

1. Through the Startup menu, go to "Run" and type :

    regedit.exe

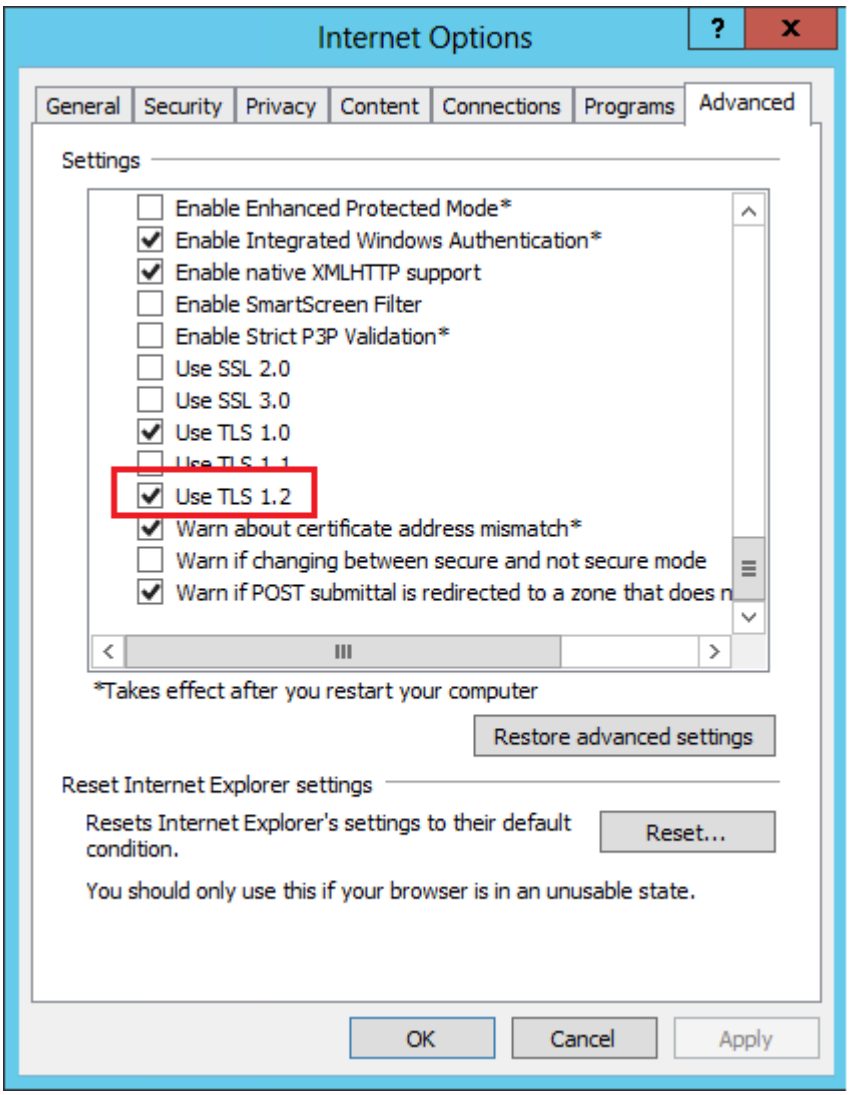
2. Browse through the registry until you reach the following key :

    HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Microsoft\.NETFramework\v4.0.30319

8. In that key, create a entry of type DWORD (32 bit) with the following details :
    a. Name : SchUseStrongCrypto
    b. Value : 1

Lastly, we need to make sure that Internet Explorer will allow the use of TLS 1.2. To do that, open the "Internet Options" of Internet Explorer, and go to the Advanced tab.

In the Security section, check the box "Use TLS 1.2":

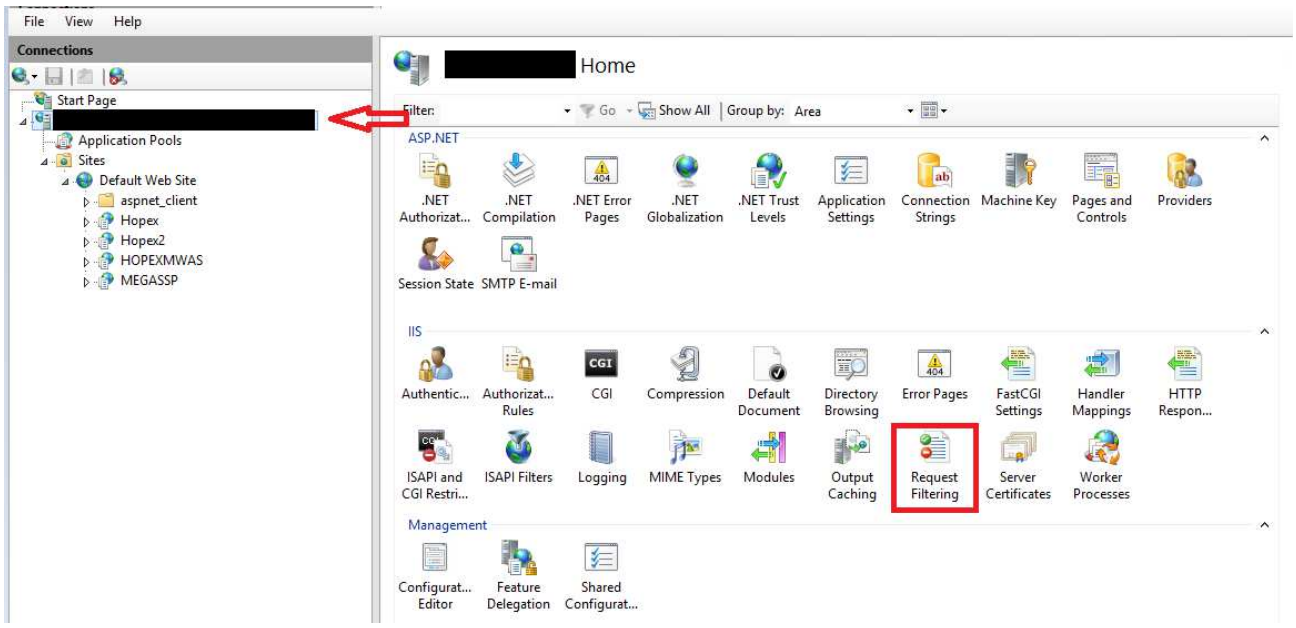


Close Internet Explorer and the registry, and reboot the server to take everything into account.

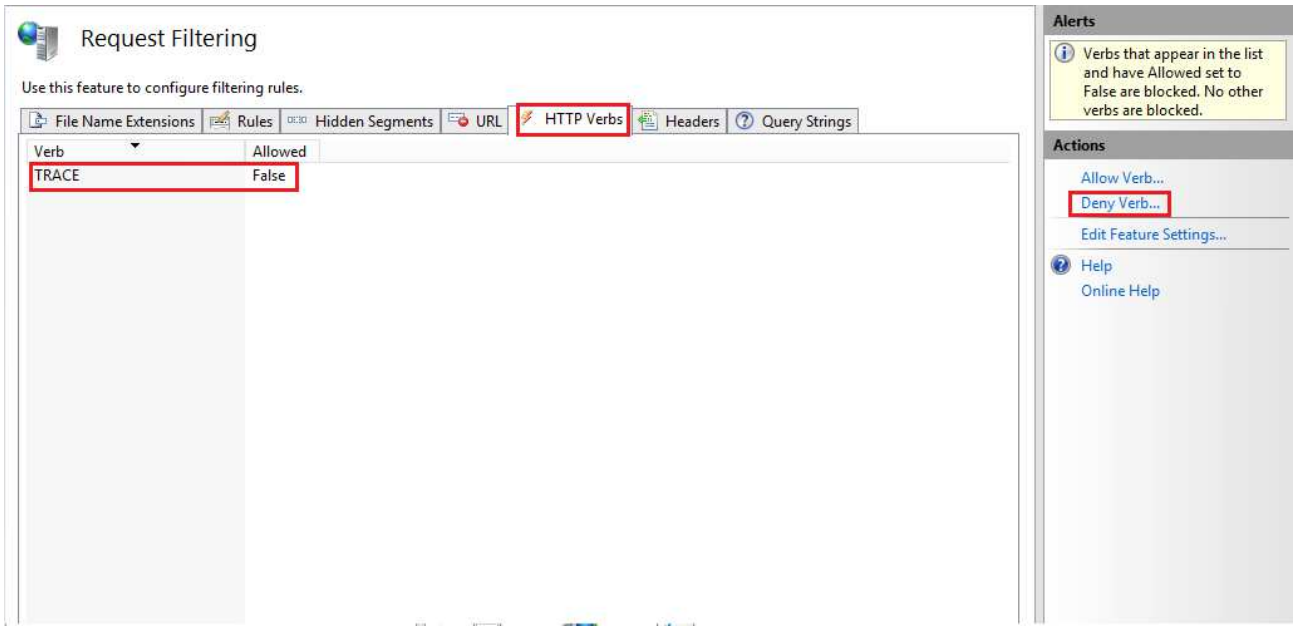
Block the TRACE HTTP request

To perform this action, three steps are needed:

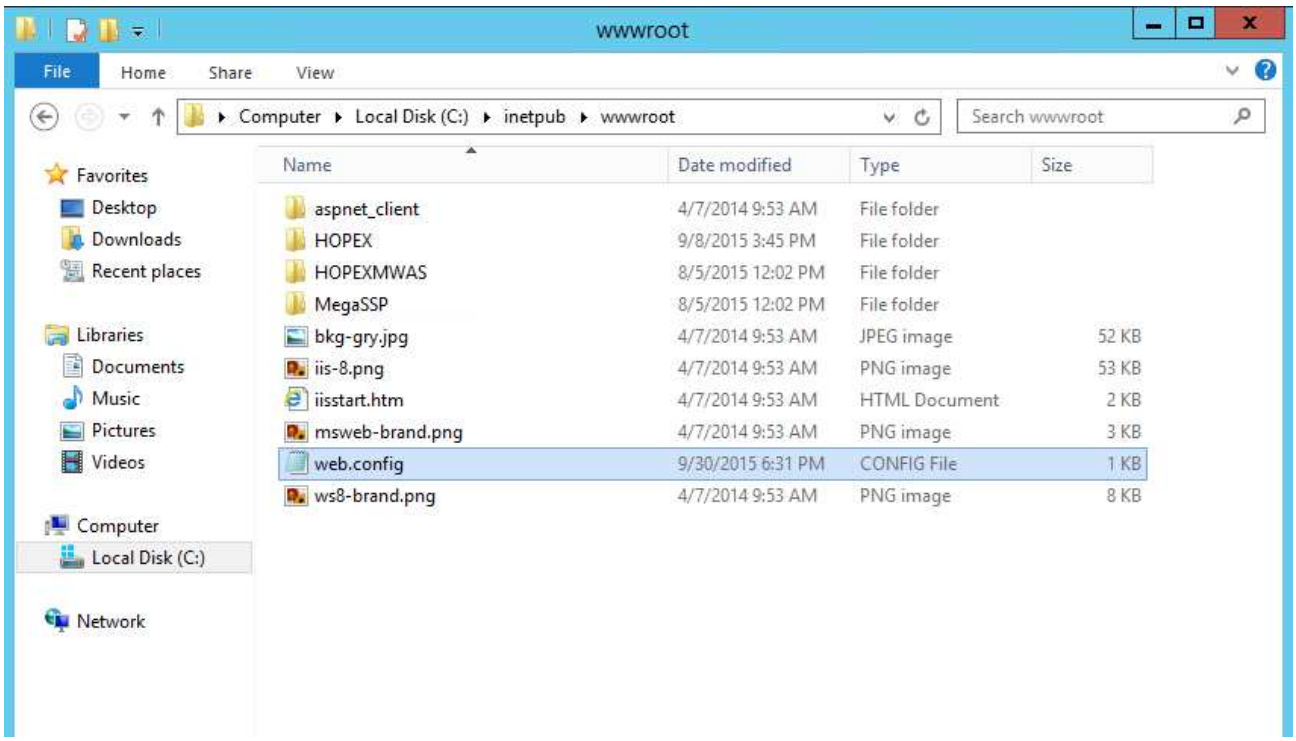
At the root of IIS, open the "Request Filtering" feature:



In the "HTTP Verbs" section, add a deny rule for the word "TRACE":



Then, open the "web.config" file at the root of your website:



And add the following lines :

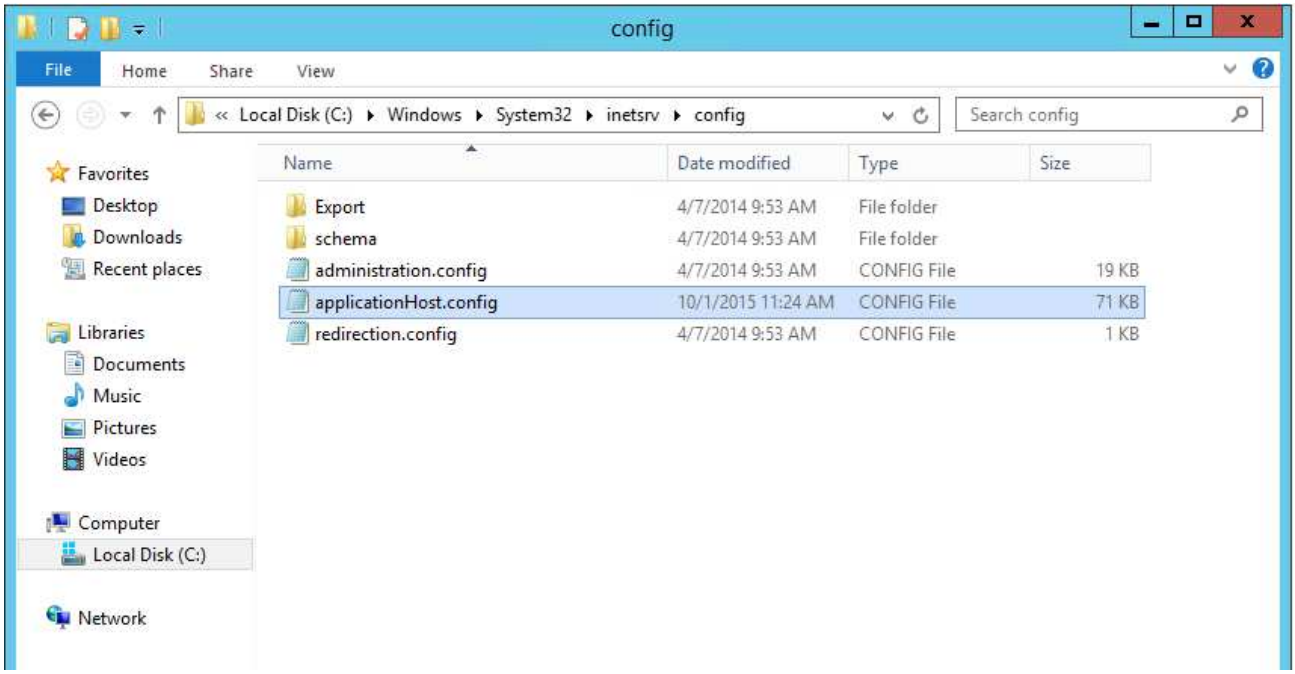
```
<authorization>
    <deny verbs="OPTIONS" users="*" />
    <deny verbs="TRACE" users="*" />
</authorization>
```

web.config - Notepad

File Edit Format View Help

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <system.webServer>
        <defaultDocument enabled="false" />
        <security>
            <requestFiltering>
                <verbs>
                    <remove verb="TRACE" />
                </verbs>
            </requestFiltering>
        </security>
    </system.webServer>
    <system.web>
 <authorization>
        <deny verbs="OPTIONS" users="*" />
        <deny verbs="TRACE" users="*" />
</authorization>
        <customErrors>
            <error redirect="https://megapoc4.com/hopex/customerrors.html" statusCode="404" />
        </customErrors>
    </system.web>
</configuration>
```

Lastly, open the "applicationHost.config" file located in the "%SystemRoot%\System32\inetsrv\config" folder:



In the "handlers" section, comment the lines for "TRACEVerbHandler" and "OPTIONSVerbHandler":

applicationHost.config - Notepad

File Edit Format View Help

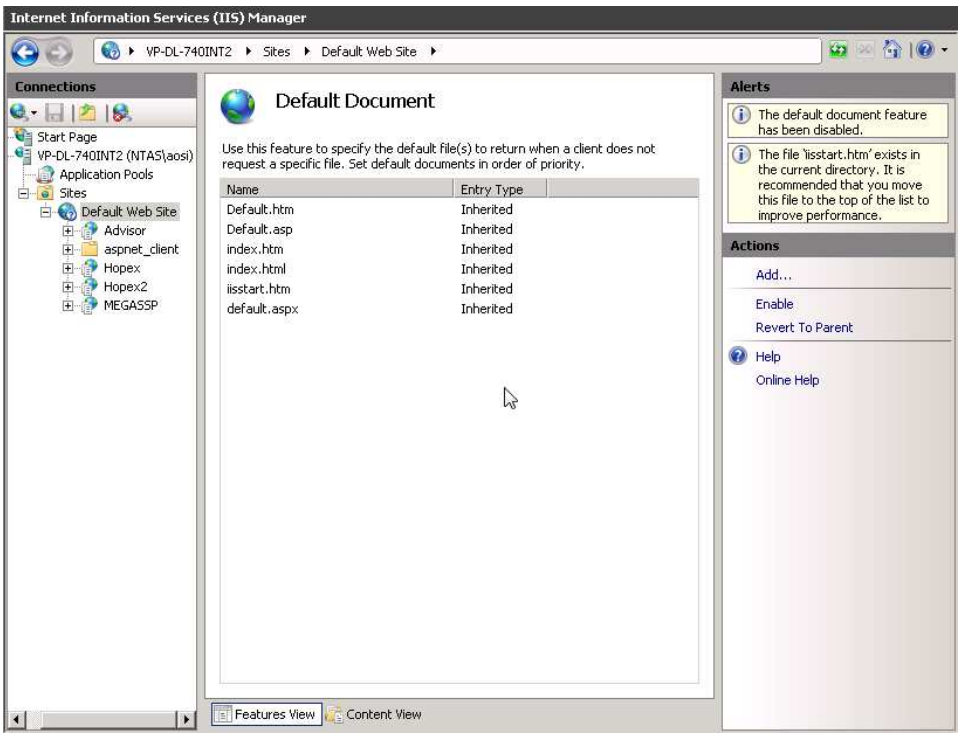
```
                <add name="SimpleHandlerFactory-Integrated-4.0" path="*.ashx" verb="GET,HEAD,POST,DEBUG" type="System.Web.UI.SimpleHandlerFactory"
                <add name="WebServiceHandlerFactory-Integrated-4.0" path="*.asmx" verb="GET,HEAD,POST,DEBUG" type="System.Web.Script.Services.Scrip
                <add name="HttpRemotingHandlerFactory-rem-Integrated-4.0" path="*.rem" verb="GET,HEAD,POST,DEBUG" type="System.Runtime.Remoting.Cha
                <add name="HttpRemotingHandlerFactory-soap-Integrated-4.0" path="*.soap" verb="GET,HEAD,POST,DEBUG" type="System.Runtime.Remoting.C
                <add name="aspq-Integrated-4.0" path="*.aspq" verb="GET,HEAD,POST,DEBUG" type="System.Web.HttpForbiddenHandler" preCondition="integ
                <add name="cshtm-Integrated-4.0" path="*.cshtm" verb="GET,HEAD,POST,DEBUG" type="System.Web.HttpForbiddenHandler" preCondition="int
                <add name="cshtml-Integrated-4.0" path="*.cshtml" verb="GET,HEAD,POST,DEBUG" type="System.Web.HttpForbiddenHandler" preCondition="i
                <add name="vbhtm-Integrated-4.0" path="*.vbhtm" verb="GET,HEAD,POST,DEBUG" type="System.Web.HttpForbiddenHandler" preCondition="int
                <add name="vbhtml-Integrated-4.0" path="*.vbhtml" verb="GET,HEAD,POST,DEBUG" type="System.Web.HttpForbiddenHandler" preCondition="i
                <add name="ScriptHandlerFactoryAppServices-Integrated-4.0" path="*_AppService.axd" verb="*" type="System.Web.Script.Services.Script
                <add name="ScriptResourceIntegrated-4.0" path="*ScriptResource.axd" verb="GET,HEAD" type="System.Web.Handlers.ScriptResourceHandler
            <add name="AXD-ISAPI-4.0_32bit" path="*.axd" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%windir%\Microsoft.NET\F
            <add name="PageHandlerFactory-ISAPI-4.0_32bit" path="*.aspx" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%windir%
            <add name="SimpleHandlerFactory-ISAPI-4.0_32bit" path="*.ashx" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%windi
            <add name="WebServiceHandlerFactory-ISAPI-4.0_32bit" path="*.asmx" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%w
            <add name="HttpRemotingHandlerFactory-rem-ISAPI-4.0_32bit" path="*.rem" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcesso
            <add name="HttpRemotingHandlerFactory-soap-ISAPI-4.0_32bit" path="*.soap" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProces
            <add name="aspq-ISAPI-4.0_32bit" path="*.aspq" verb="*" modules="IsapiModule" scriptProcessor="%windir%\Microsoft.NET\Framework\v4.0.30
            <add name="cshtm-ISAPI-4.0_32bit" path="*.cshtm" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%windir%\Microsoft.N
            <add name="cshtml-ISAPI-4.0_32bit" path="*.cshtml" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%windir%\Microsoft
            <add name="vbhtm-ISAPI-4.0_32bit" path="*.vbhtm" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%windir%\Microsoft.N
            <add name="vbhtml-ISAPI-4.0_32bit" path="*.vbhtml" verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%windir%\Microsoft
            <add name="CGI-exe" path="*.exe" verb="*" modules="CgiModule" resourceType="File" requireAccess="Execute" allowPathInfo="true" />
            <!--<add name="TRACEVerbHandler" path="*" verb="TRACE" modules="ProtocolSupportModule" requireAccess="None" />
            <add name="OPTIONSVerbHandler" path="*" verb="OPTIONS" modules="ProtocolSupportModule" requireAccess="None" />-->
            <add name="ExtensionlessUrlHandler-ISAPI-4.0_32bit" path="*." verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%windir
                <add name="ExtensionlessUrlHandler-ISAPI-4.0_64bit" path="*." verb="GET,HEAD,POST,DEBUG" modules="IsapiModule" scriptProcessor="%wi
                <add name="ExtensionlessUrlHandler-Integrated-4.0" path="*." verb="GET,HEAD,POST,DEBUG" type="System.Web.Handlers.TransferRequestHa
            <add name="StaticFile" path="*" verb="*" modules="StaticFileModule,DefaultDocumentModule,DirectoryListingModule" resourceType="Either"
        </handlers>

        <modules>
            <add name="HttpCacheModule" lockItem="true" />
            <add name="DynamicCompressionModule" lockItem="true" />
            <add name="StaticCompressionModule" lockItem="true" />
```

Remove the default install page in IIS

After the application is installed, you can disable the "Default Document" feature for the website on which the application is deployed.

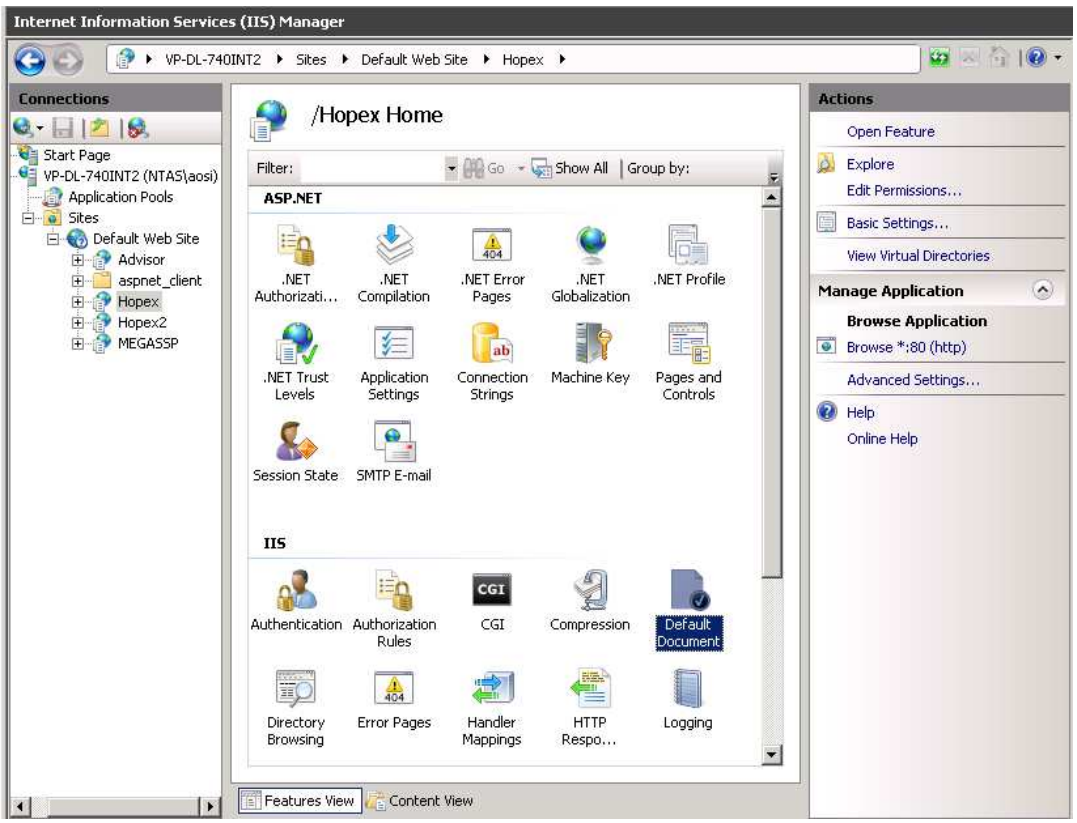
To do that :

1. Open the IIS Manager console.
2. Locate the website where the Hopex and/or Advisor web application is installed.
3. On the website, double-click **Default Document** to open this feature.
4. In the "Actions" pane on the right, click **Disable**. You should have this kind of screen after it is done :

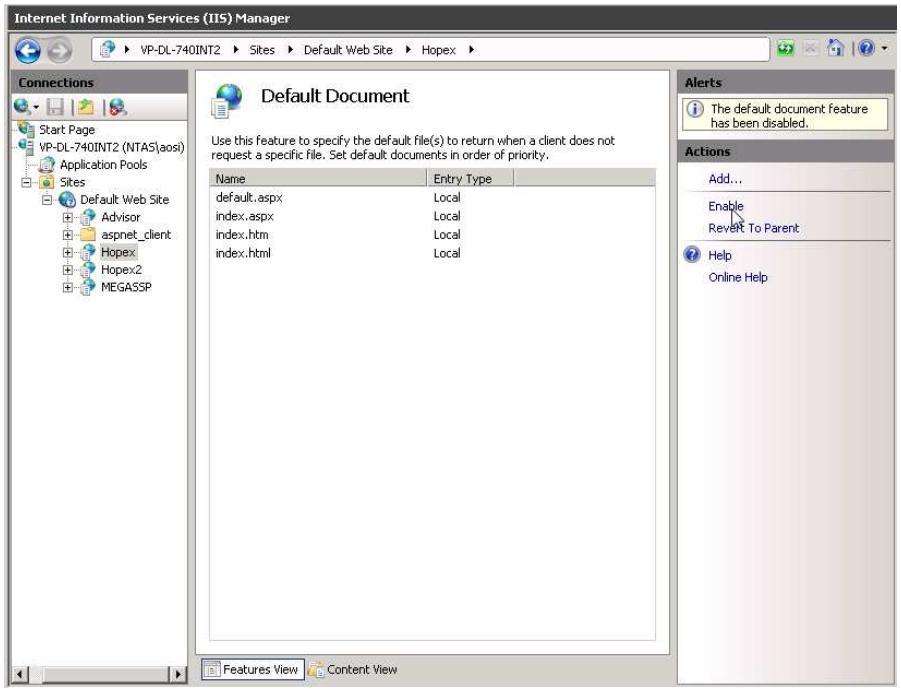


5. Make sure that the « Default Document » feature is enabled for the Hopex/Hopex2/Advisor web applications. Since it was disabled at the root level, you will have to :

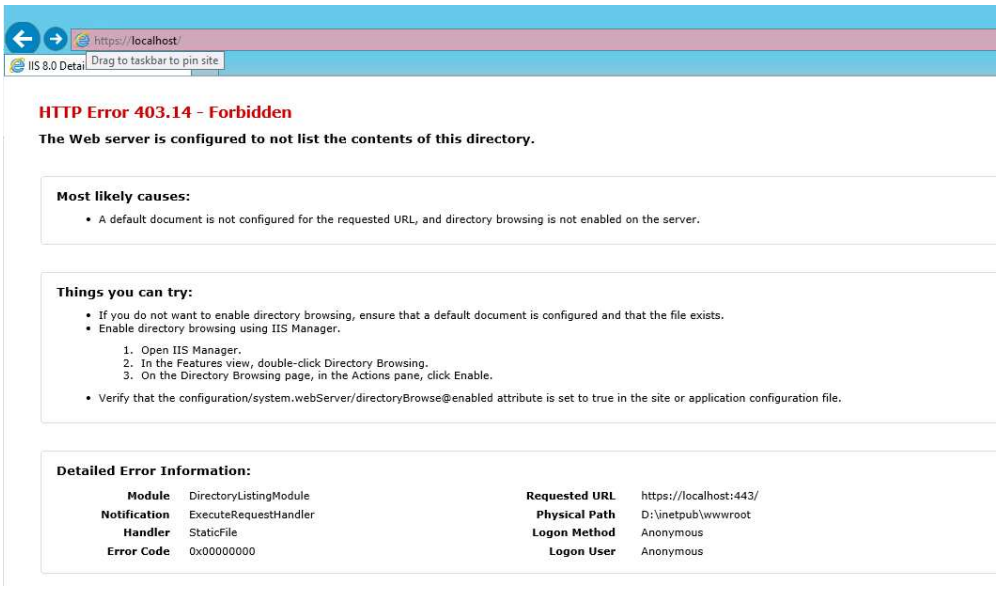
    a. Select each web application (here Hopex), and open the "Default Document" feature :

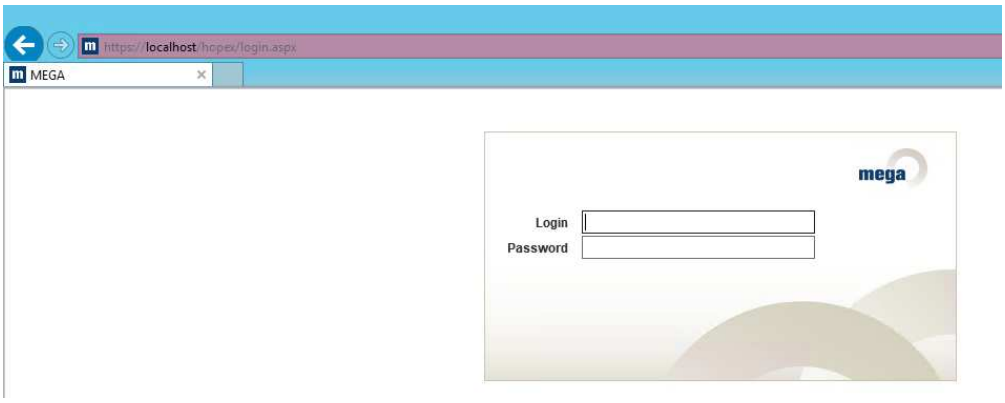


b. Click **Enable**.



c. This way, the URL https://servername won't reply, whereas the URL https://servername/hopex will redirect the users to the login page of Mega :





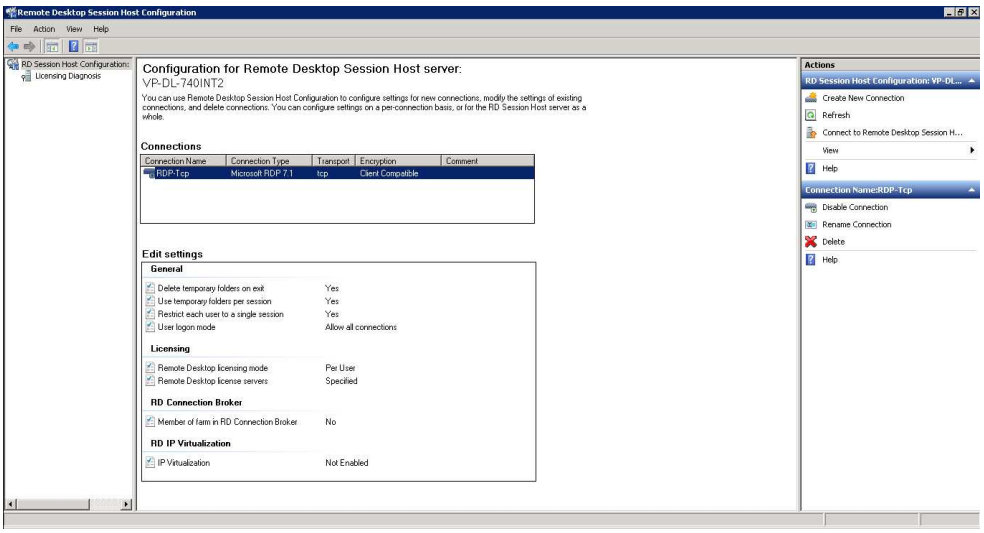
Securing the RDP access (Terminal Services)

Only valid for Windows Server 2008 R2. For later versions, we restrict the RDP access to the IPs of valid people or teams.

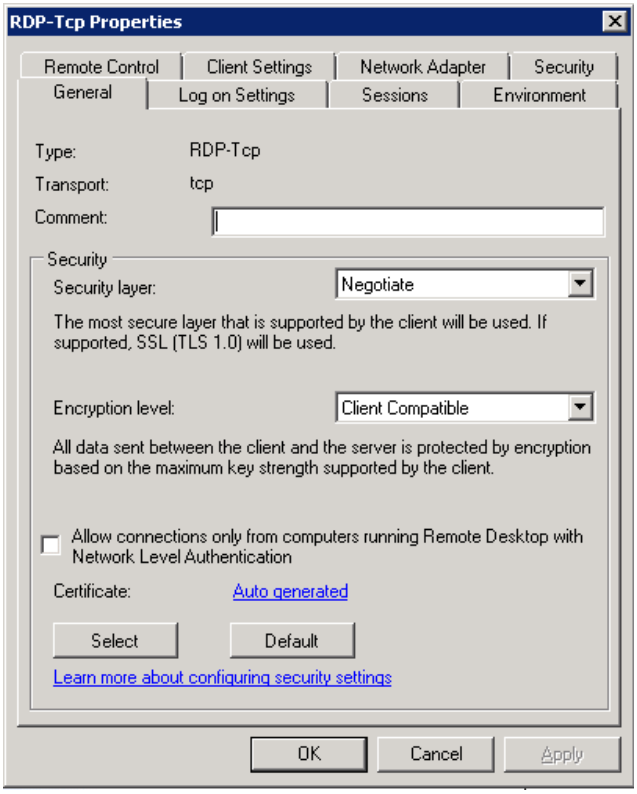
On a server hosted on a Windows 2008 R2 Operating System, you can secure the RDP layer by changing the security layer and the encryption layer of the Remote Desktop Protocol.

To do that:

1. Firstly, you have to open **Remote Desktop Session Host Configuration** the by clicking on "Start", point to "Administrative Tools", point to "Remote Desktop Services", and then click "Remote Desktop Session Host Configuration".

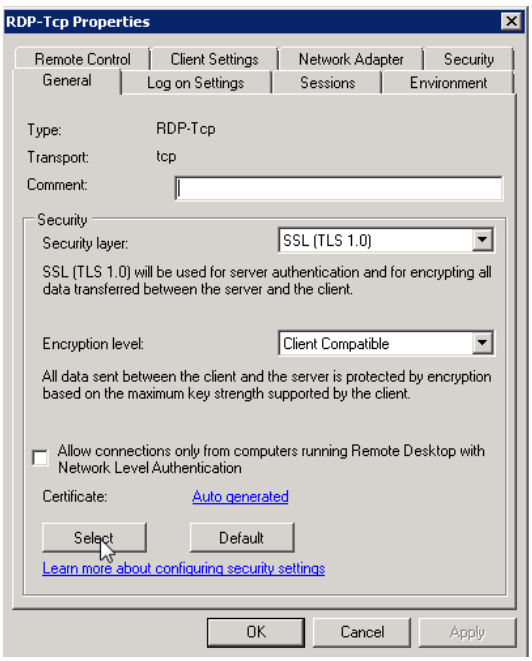


2. In the "Connections" section, do a right-lick on the "RDP-tcp" entry, and select "Properties" :

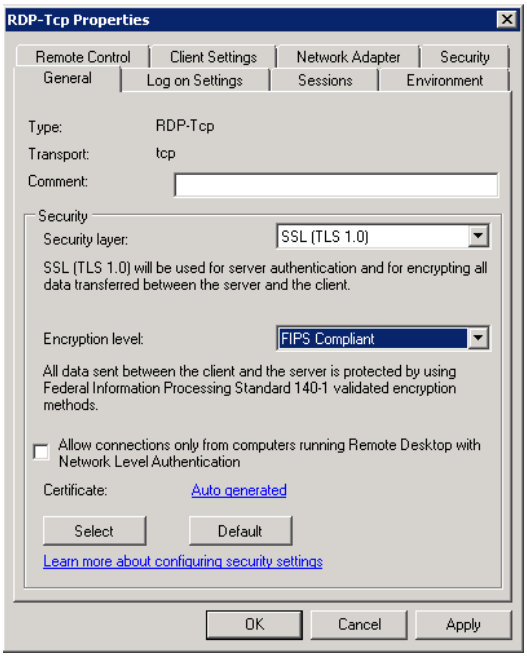


3. In the "General tab", as shown above, you can see the two security parameters that can be modified.

4. Change the "Security Layer" option to "SSL (TLS 1.0)". Be aware that you will need to have a specific certificate installed on your server to switch to that level. The "Select" button allows you to browse your installe certificates and choose the proper one:



5. Then, modify the « Encryption level » option and upgrade it to at least "High", and possibly "FIPS Compliant" :



6. Click **OK** to validate this modification and exit the configuration tool.

There are also numerous ways of securing the RDP protocol, whether it is using the Remote Desktop Gateway, filtering by IP address and user, etc.

Configuring the firewall

To avoid letting some unwanted users connect to the server or retrieve information from the server, some actions can/need to be taken on the firewall:

- If possible, restrict the access to port 3389 (RDP) to only the valid IP addresses of the System Administration, Application Administrators, and maybe the users that need to launch the Desktop client of Mega through Terminal Services.
- Restrict external access to all SMB services and ports, including TCP and UDP 135, TCP and UDP 139, and TCP and UDP 445.
- Regarding the ICMP protocol, block the following type of requests: ICMP timestamp requests (13), and ICMP timestamp replies (14).
- Limit the number of opened ports on the server to the least amount possible. Globally the server needs to be accessed through RDP, the SSL port needs to be opened, the SMB port also, and the communication port to the database server. All other ports need to be assessed in order to check that those are relevant.

Configuring the default error page of the application to hide application errors

Note that this section is necessary in the case of a remote access. While the application is only reachable through a secure network, this is not mandatory, and can make things more complex for administrators to understand what is happening on the application while errors occur on the system.

The configuration is as follows.

First, we put in the default folder of the web application a custom error file that will redirect, whenever a page does not respond, or a wrong page is accessed, to the default page of the application.

An example can be find below:

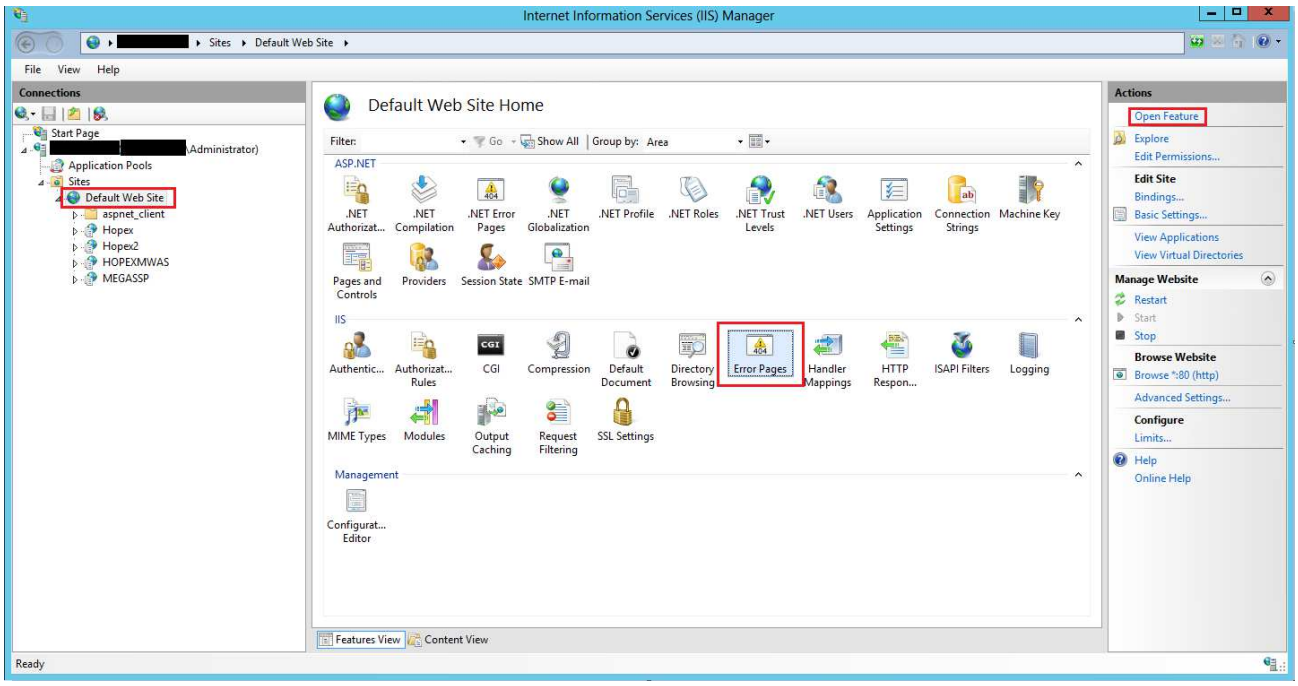
customerrors.html

Note that with the latest CPs of the HOPEX different versions, this file is most likely already in the below folder.

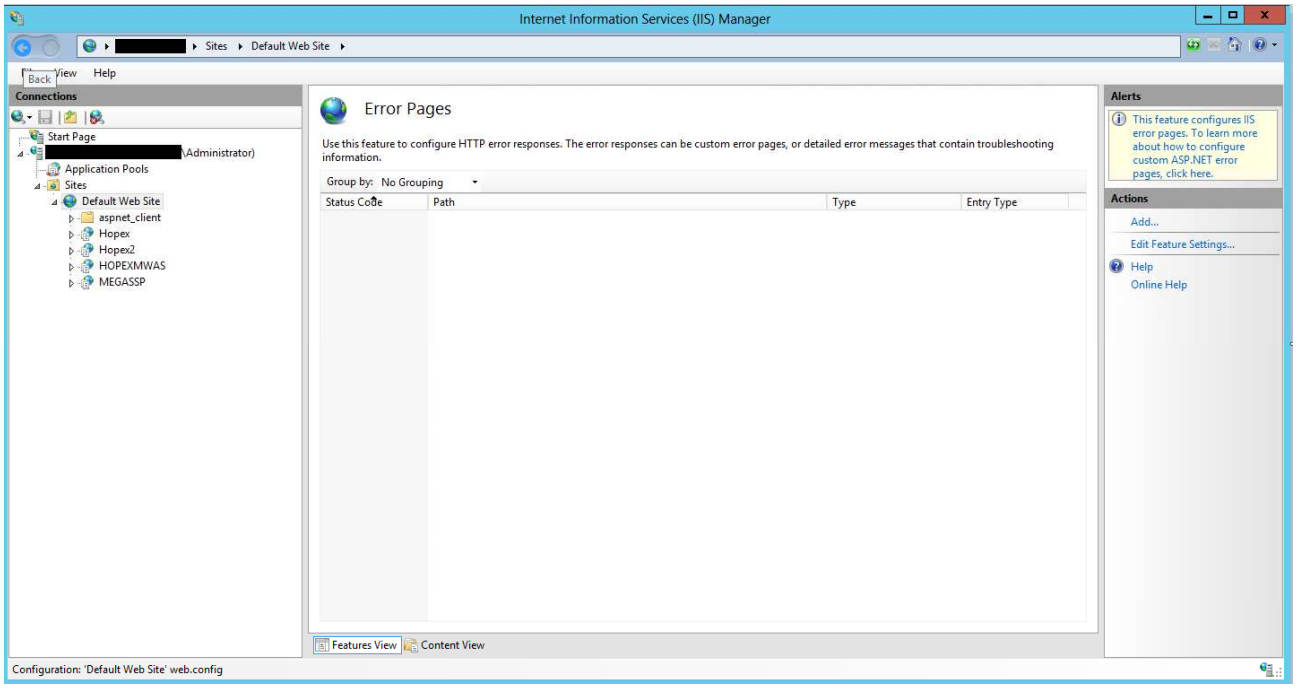
By default, it is located on the C:\inetput\wwwroot\HOPEX folder. In case you installed the "HOPEX (IIS)" feature and web application in a different location, target this path and copy the file there.

Next, remove the default error pages of the website.

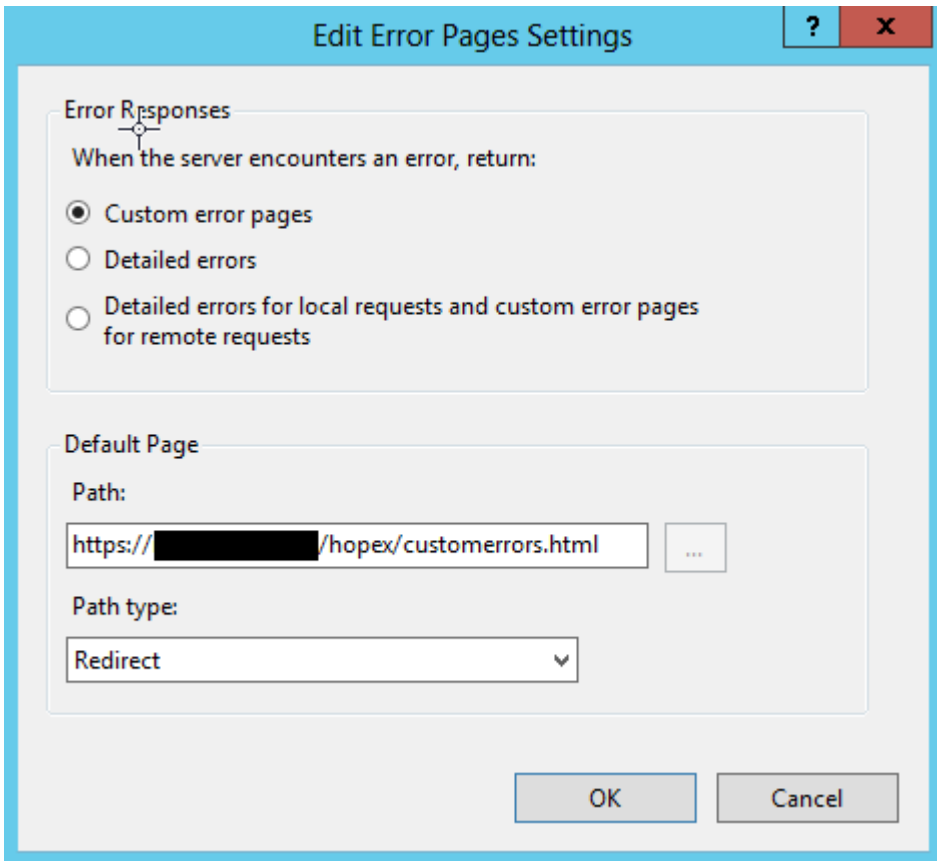
Go to IIS Manager, locate your website, and open the "Error Pages" feature:



Remove all lines to have this in the end:



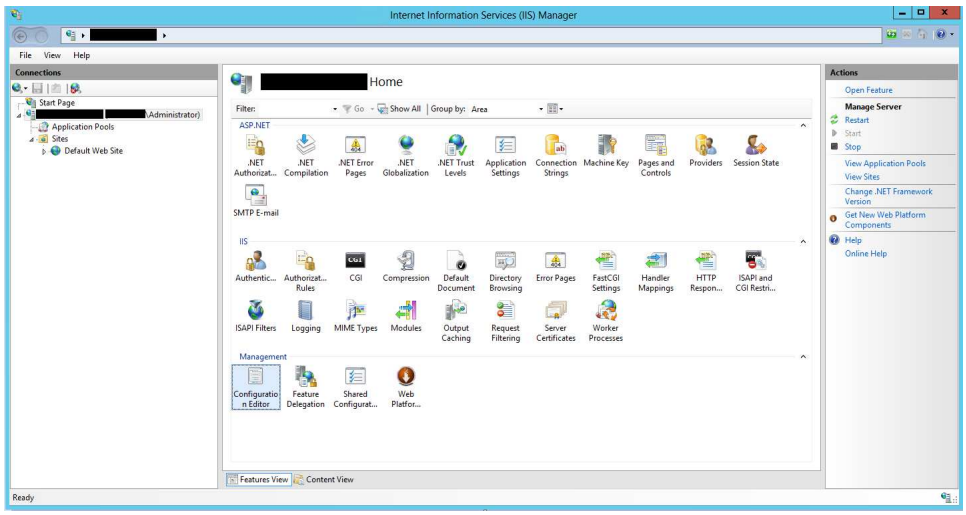
Lastly, in the same location in IIS, click on "Edit Feature Settings..." (for the "Error Pages" feature), and specify the following parameters, using the full URL of the custom error page you want to redirect to in case of error:



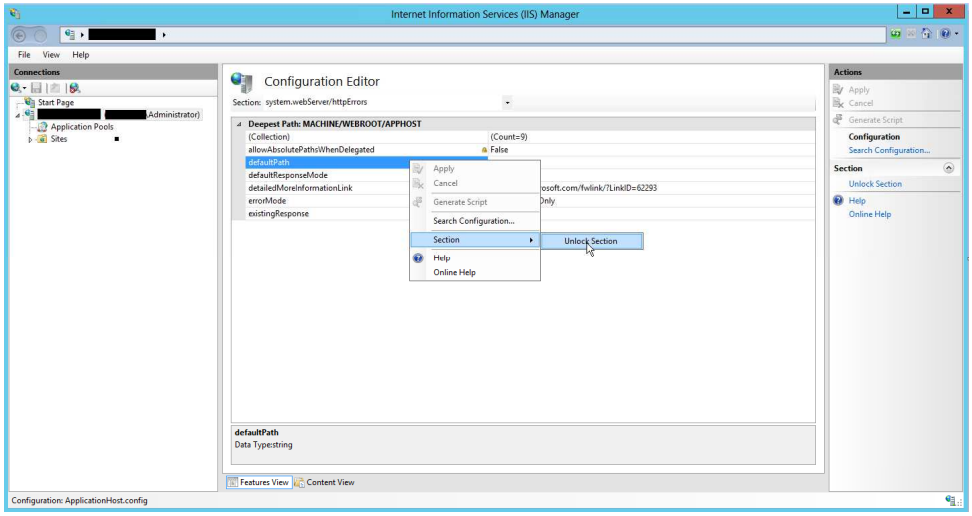
This will be reflected in the "web.config" of the website, where you will find this section ("your_url" being either the alias targeting the web server, of the name of the web server itself, depending on how the certificate was implemented) :

```
<httpErrors errorMode="Custom"
defaultPath="https://your_url/hopex/customerrors.html"
defaultResponseMode="Redirect">
            <remove statusCode="502" subStatusCode="-1" />
            <remove statusCode="501" subStatusCode="-1" />
            <remove statusCode="500" subStatusCode="-1" />
            <remove statusCode="412" subStatusCode="-1" />
            <remove statusCode="406" subStatusCode="-1" />
            <remove statusCode="405" subStatusCode="-1" />
            <remove statusCode="404" subStatusCode="-1" />
            <remove statusCode="403" subStatusCode="-1" />
            <remove statusCode="401" subStatusCode="-1" />
        </httpErrors>
```

Depending on the version of IIS, you may need to first deactivate the locking of that default page. You will have to do this if you get a "lock violation" error when validating the previous setting popup.
To do this, open the "Configuration Editor" at the root level of IIS :



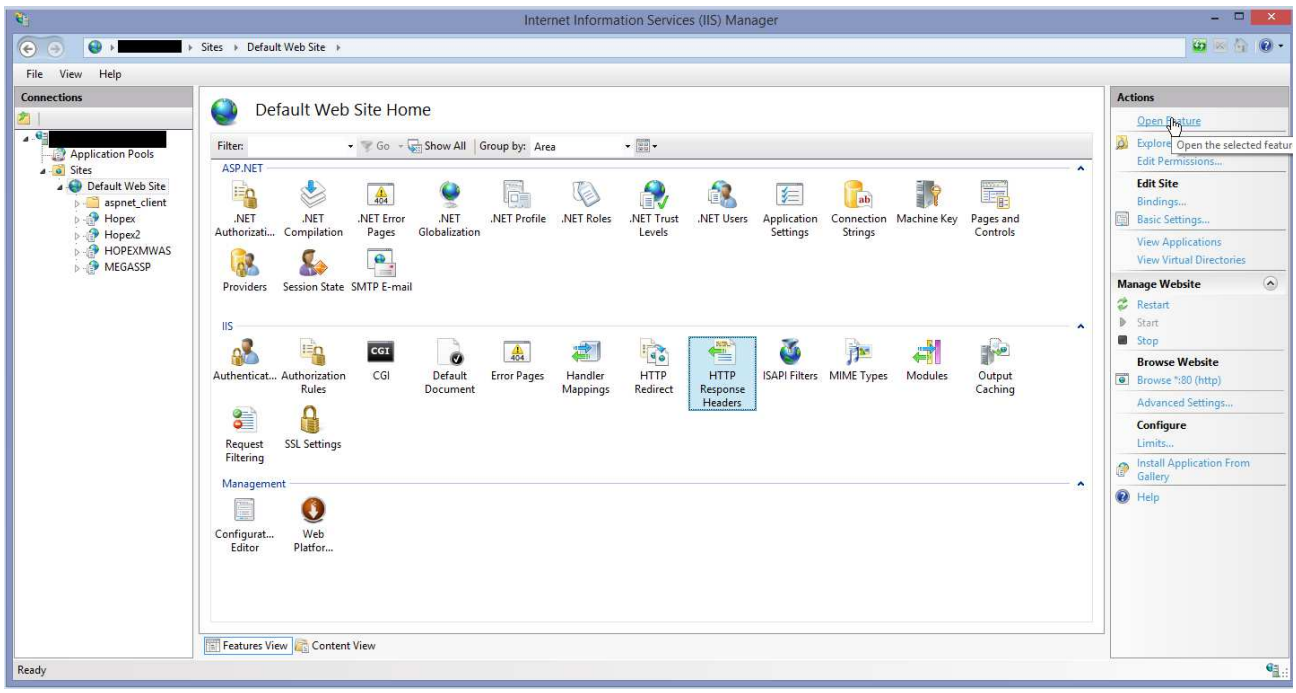
In “Section”, browse to “system.webServer/httpErrors”, and unlock the “defaultPath” attribute:



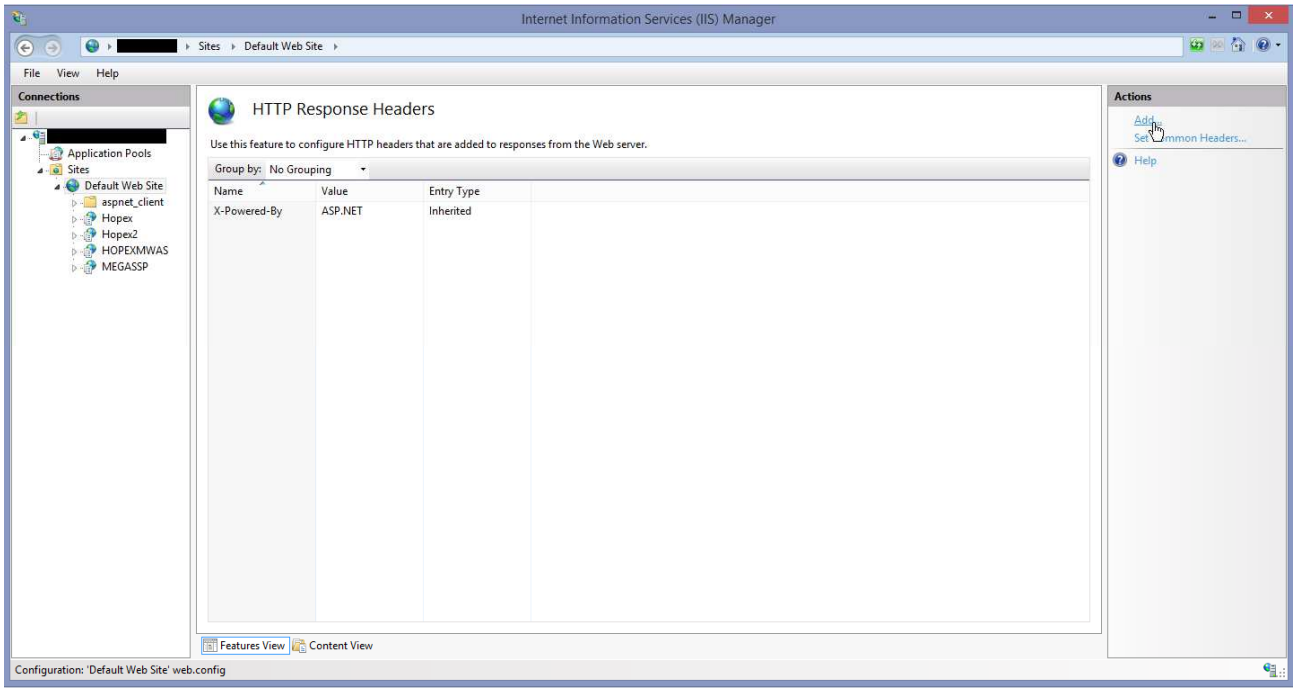
Protection against ClickJacking

To prevent the use of ClickJacking, through Frame-Sniffing, you can configure the website as follow:

1. From IIS manager, access the website hosting the application.
2. Select **HTTP Response Headers** and click **Open feature**.

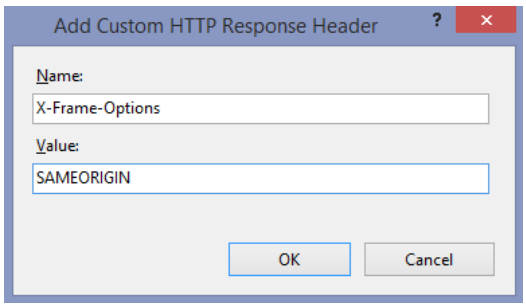


3. In the **Actions** pane, click **Add**.

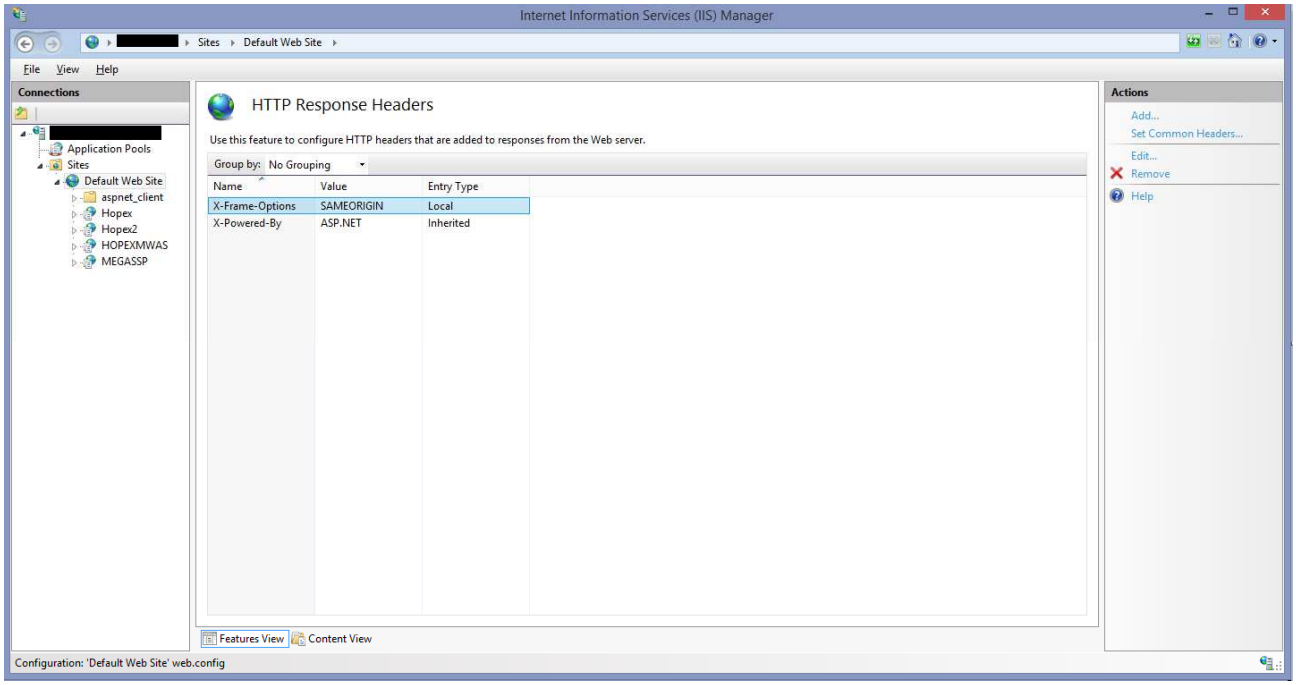


4. In the dialog box, enter:

   - In the **Name** field: "X-Frame-Options"
   - In the **Value** field: "SAMEORIGIN"



5. Click **OK**.



Securing the ASP.NET session cookie

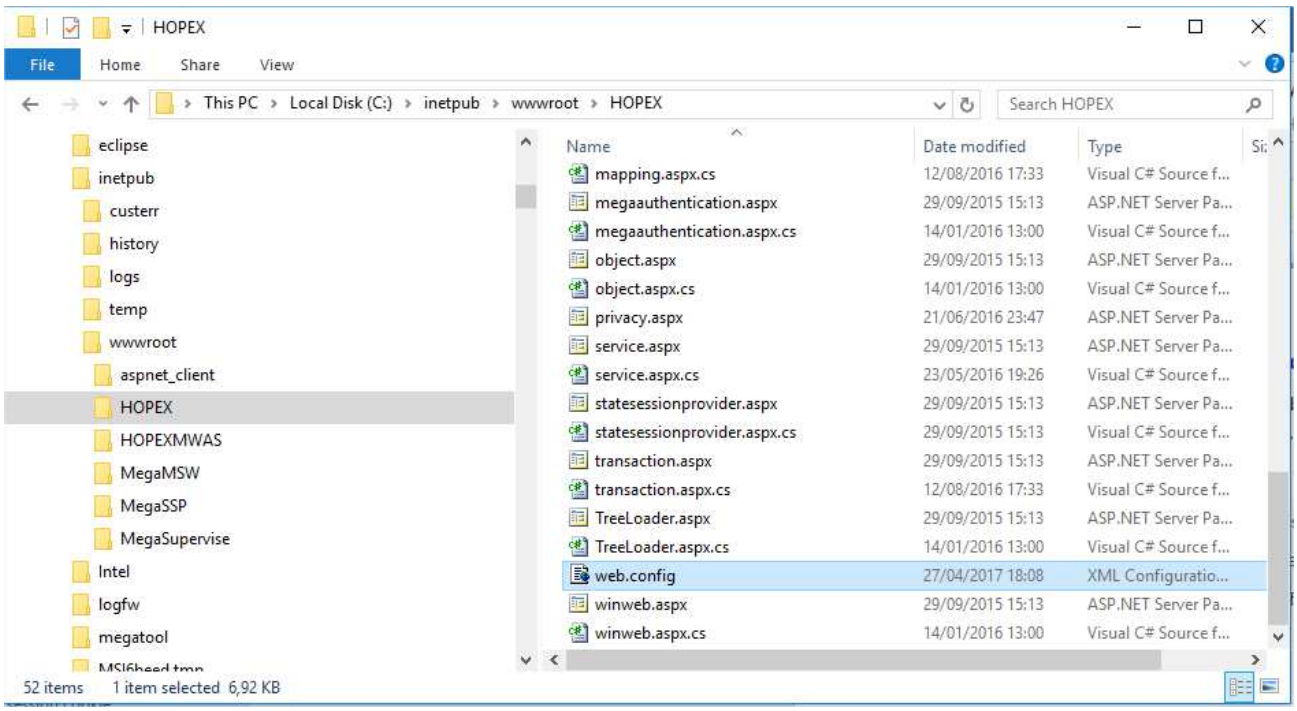
Prerequisite: requires the use of SSL on your website (see previous section in this document).

This feature is delivered by default with Hopex V1R2/R3 CP15, or Hopex V2 CP03.

For previous versions, a specific line will need to be added manually.

The goal is to secure the cookie of the users sessions.

1. To do that, your website requires to be configured in HTTPS.
2. Then, you need to update your web.config file of the "HOPEX" web application:



1. Browse through the file and go to the <system.web> section.
2. If you are using a version as shown at the beginning of this section, you will see the commented line hightlighted below:

```
  <system.web>
    <!--<httpCookies requireSSL="true" /> -->
    <globalization fileEncoding="UTF-8" />
    <trace enabled="false" localOnly="true" pageOutp
    <compilation debug="false" defaultLanguage="c#"
    <authentication mode="None" />
    <!-- <authentication mode="Windows" /> -->
    <identity impersonate="true" userName="registry:
    <httpRuntime requestValidationMode="2.0" request
  </system.web>
</configuration>
```

   a. Remove the comments, and save the file.

      It will automatically restart the application pool.

   b. Check if your website is still working:

```
  <system.web>
    <httpCookies requireSSL="true" />
    <globalization fileEncoding="UTF-8" />
    <trace enabled="false" localOnly="true" pageOutput="true" tr
    <compilation debug="false" defaultLanguage="c#" targetFramew
    <authentication mode="None" />
    <!-- <authentication mode="Windows" /> -->
    <identity impersonate="true" userName="registry:HKEY_LOCAL_M
    <httpRuntime requestValidationMode="2.0" requestValidationTy
  </system.web>
</configuration>
```

If you have previous version of Hopex, you will need to insert this line, as shown above, and save your file :

<httpCookies requireSSL="true" />

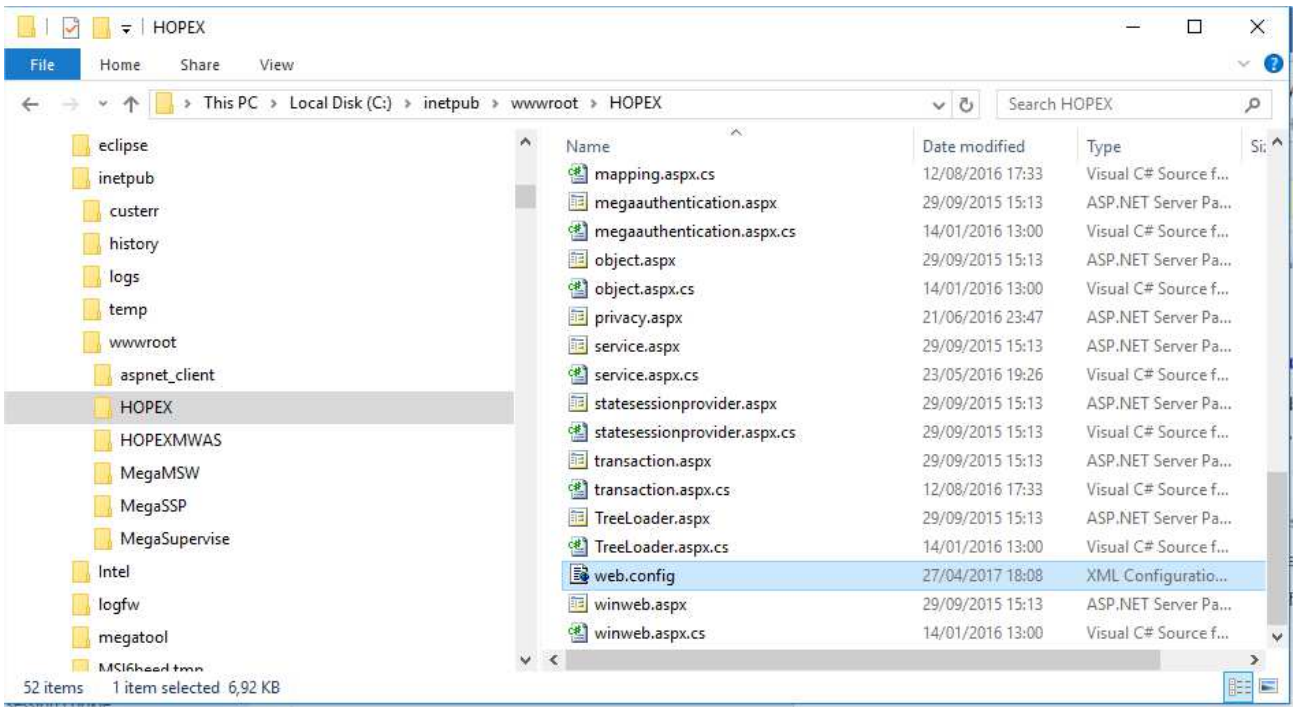
Hide ASP.Net version header

To prevent people from discovering some information about the platform and the application versions, a specific setting can be put in place to hide the ASP.Net headers.

This way, when checking the headers, people won't be able to detect that ASP.net is used, in which version. It also hides the build number of the application.

The steps to perform this setting are:

1. Update your "web.config" file of the "HOPEX" web application:



2. Browse through the file and go to the "<system.web>" section and locate the line where the "*httpRuntime*" is setup.

3. In that line, add this value:

   enableVersionHeader="false"

```
    <!-- <authentication mode="Windows" /> -->
    <identity impersonate="true" userName="registry:HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\MEGA\HOPEX\identity\ASPNET_SETREG,u
    <httpRuntime requestValidationMode="2.0" requestValidationType="MegaRequestValidator" enableVersionHeader="false"/>
  </system.web>
</configuration>
```

4. Save the file (this will restart the application pool).

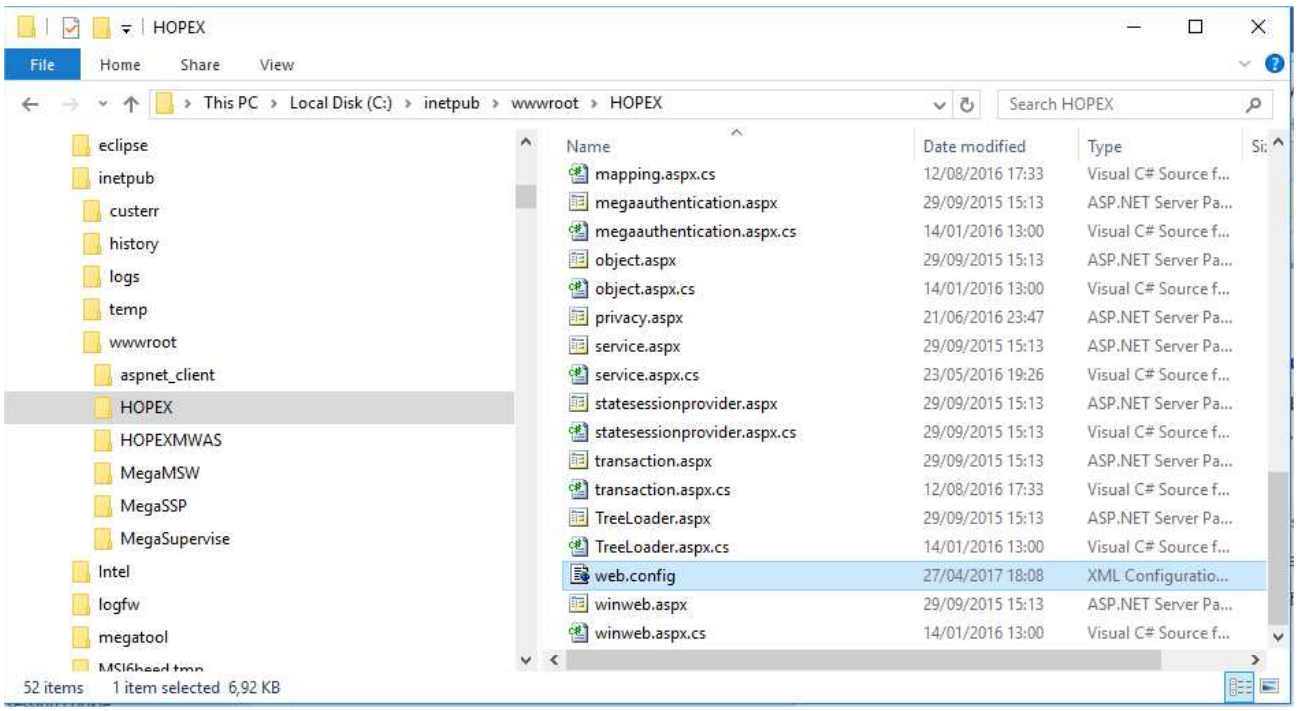
Strict-Transport-Security HTTP header

Prerequisite: only if SSL is activated on the web site.

If you are accessing your website through HTTPS, you may want to properly define the above header.

The steps to perform this setting are:

1. Update your "web.config" file of the "HOPEX" web application:



2. Browse through the file and go to the "<system.webServer>" section.
3. Add the following line in that section:

```
<httpProtocol>
    <customHeaders>
        <add name="Strict-Transport-Security" value="max-age=31536000"/>
    </customHeaders>
</httpProtocol>
```

The "max-age" value, in this example set up to **one year**, can be adapted depending on the internal policies.

```
<system.webServer>
  <httpProtocol>
      <customHeaders>
          <add name="Strict-Transport-Security" value="max-age=31536000"/>
      </customHeaders>
  </httpProtocol>
  <staticContent>
    <remove fileExtension=".woff" />
    <mimeMap fileExtension=".woff" mimeType="application/font-woff" />
```

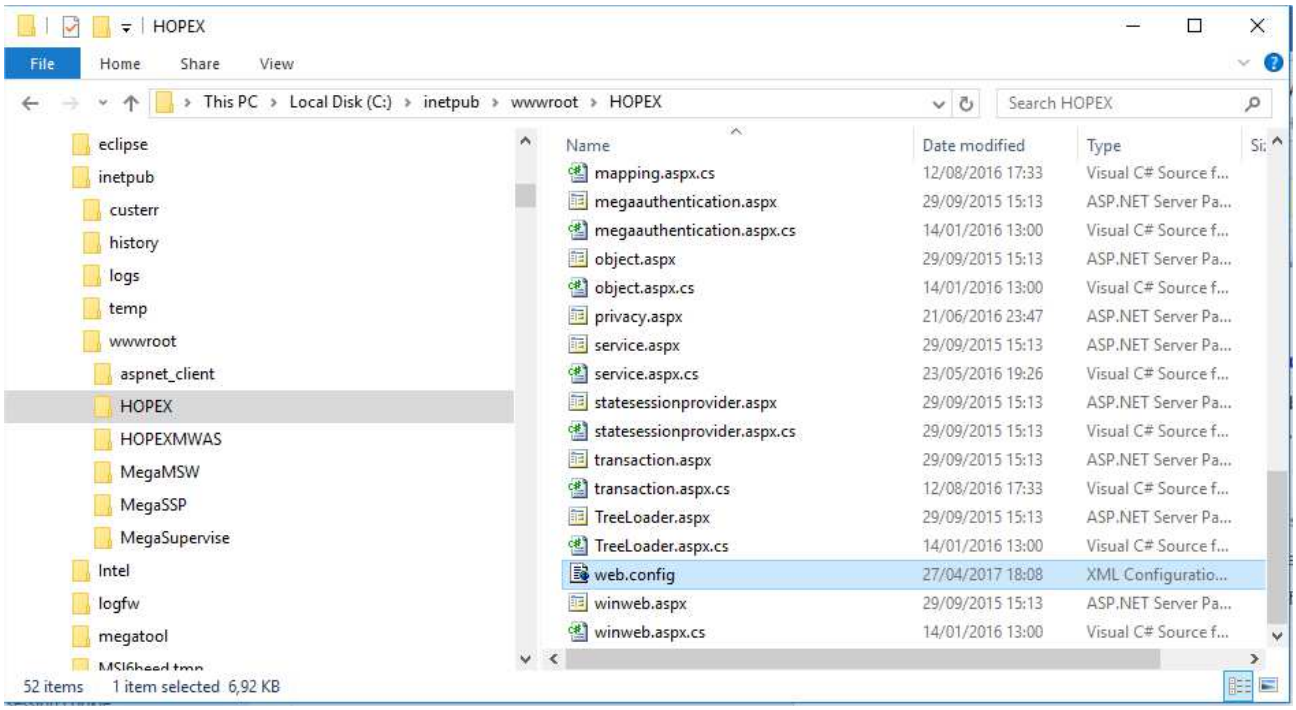
4. Save the file (this will restart the application pool).

Manage content type options

Depending on your version of the application, this section might already be activated by default. Review and check.

To prevent the user agent to render the site content in a different fashion to the MIME type:

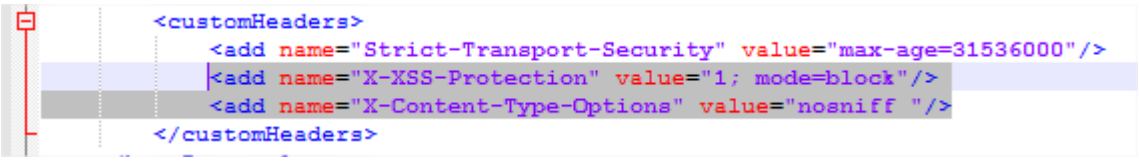
1. Update your "web.config" file of the "HOPEX" web application:



2. Browse through the file and go to the "<httpprotocol>" section and its "<customheaders>" subsection.

3. Add the following line in that section:

```
<add name="X-XSS-Protection" value="1; mode=block"/>
<add name="X-Content-Type-Options" value="nosniff "/>
```



4. Save the file (this action will restart the application pool).

SECURING THE APPLICATION

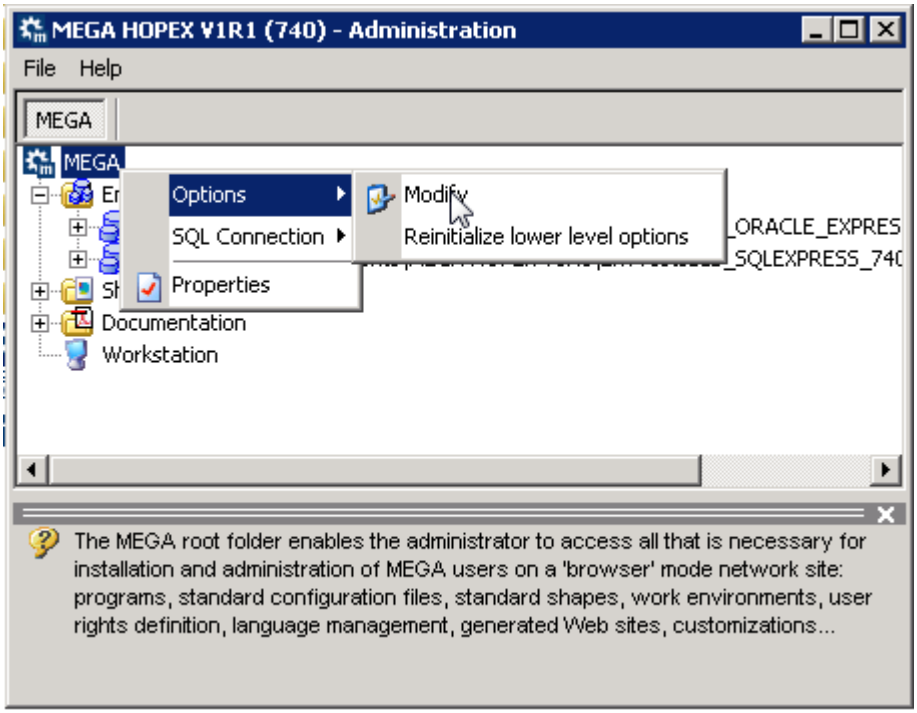
Note that the following sections are normally already configured by default with HOPEX V1R2-V1R3 CP8, as well as with HOPEX V2 regardless the patch level. You can check those, and tune up the second section depending on the timeout you want to put in place.

Hiding the error details

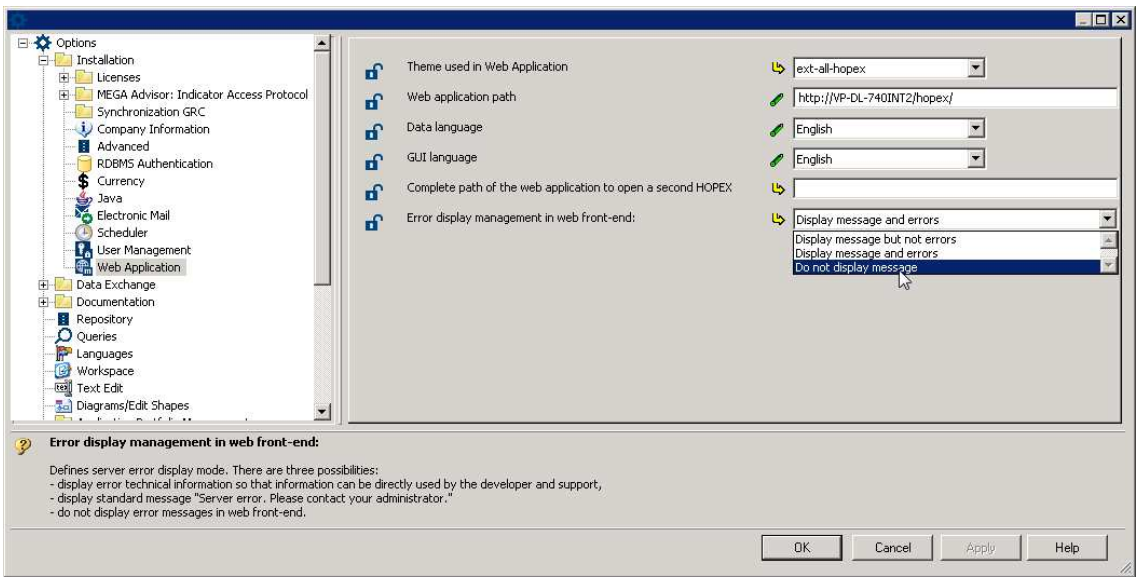
To prevent the end users from seeing the error details and get knowledge about how the application is written, some actions can be taken to hide those:

1. Open the Administration module of Mega on the web server (Administration.exe, in the installation module of Mega).

2. Open the options at the root level:



3. Go to “Installation”, and then “Web Application”, and change the option “Error display management in web front-end” to “Do not display message”:



4. Close the options and the Administration module.

5. Locate the web.config file of the “Hopex” web application (by default in “C:\inetpub\wwwroot\HOPEX”), and edit it.

6. Add the following key in the file :

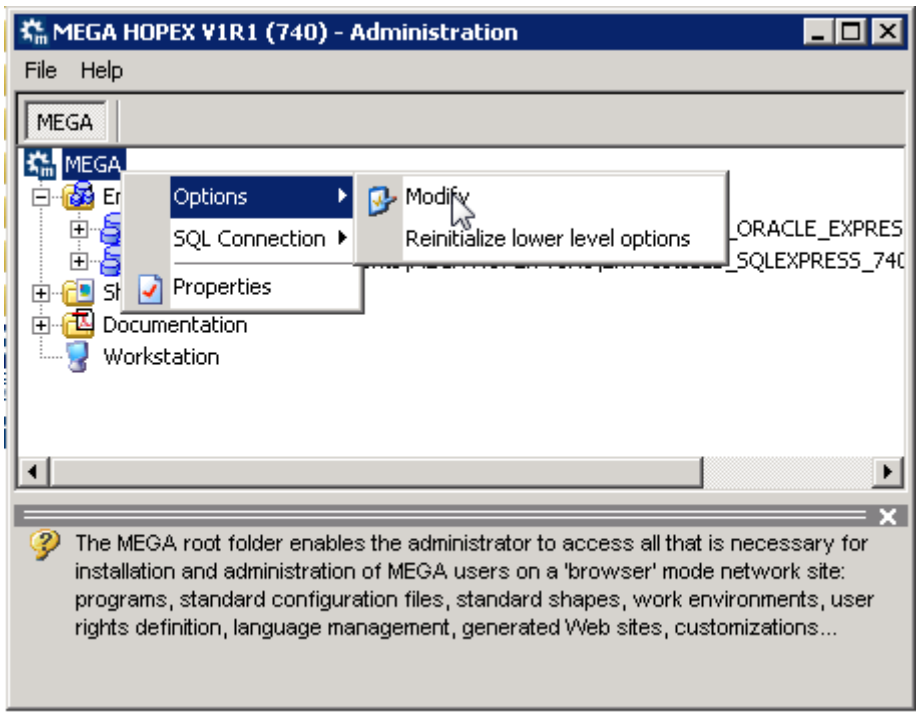
```
<add key="HideErrors" value="1"/>
```

Activating the automatic logoff

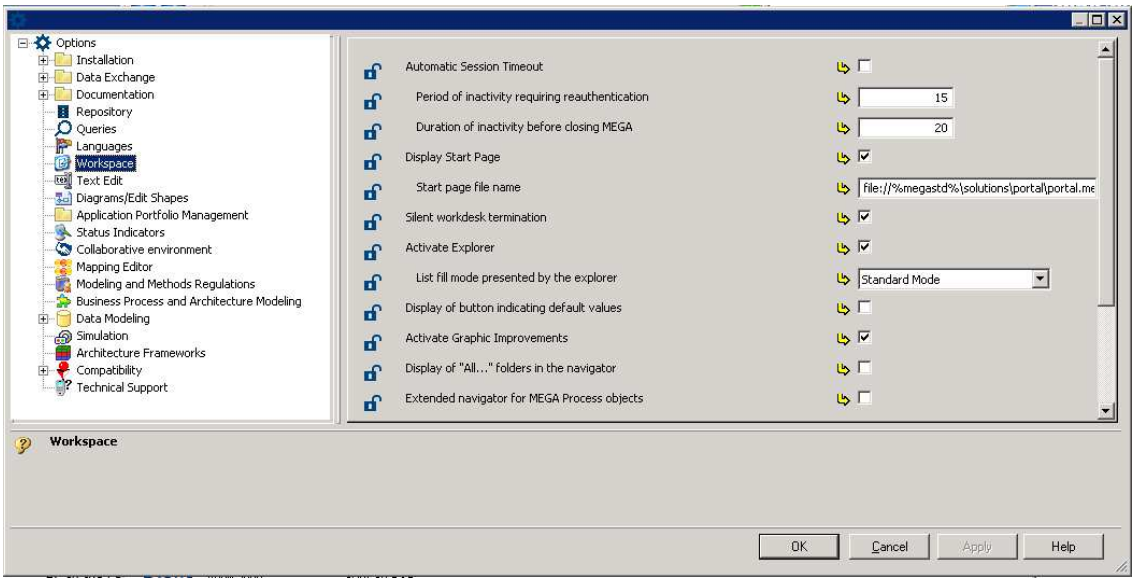
You can activate an automatic logoff of the users after a certain time of inactivity. To do so:

1. Open the Administration module of Mega on the web server (Administration.exe, in the installation module of Mega).

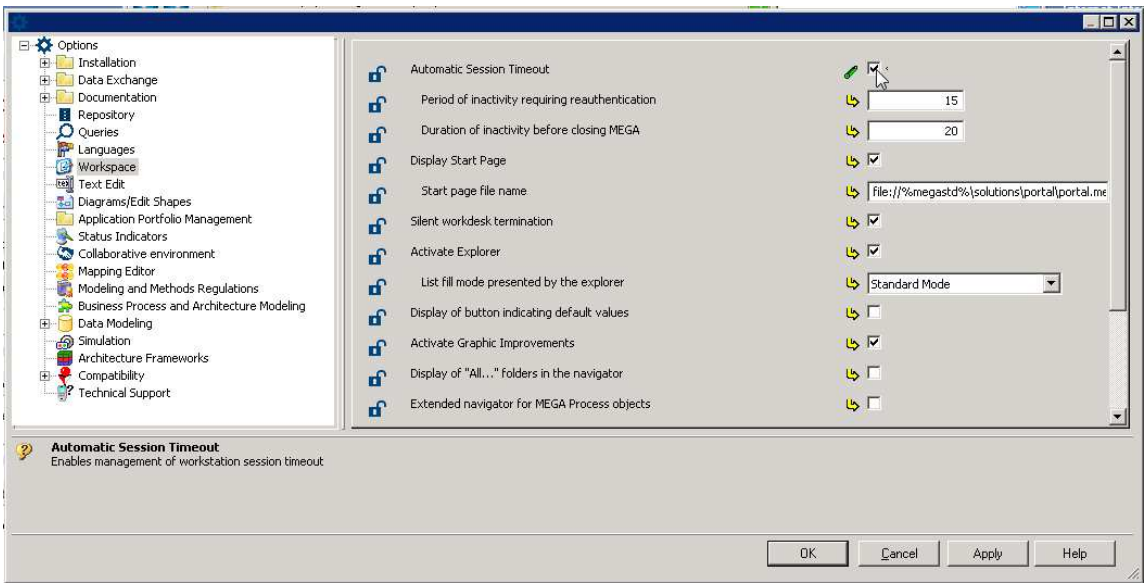
2. Open the options at the root level:



3. Go to "Workspace":



4. Select "Automatic Session Timeout" option, and tune up the parameters "Period of inactivity requiring authentication"and "Duration of inactivity before closing MEGA" to the wanted values (in minutes, by default they are set to 15 and 20, respectively) :



5. Click **OK**, and close the Administration module.
6. Restart the application to validate this whole configuration.

Hiding the information when entering the wrong credentials

For versions, before V1R2 **CP15**, or V1R3 CP15, or V2 **CP03**, when someone enters a wrong password of an existing user, or tries to authenticate with a user that doesn't exist, he will get a clear message telling him what is the case.

If you want to have a generic message preventing someone from discovering the users declared in an environment, you will need to upgrade to the above-mentioned versions of the application.

The message will then be this one:

"No such login, or no security question defined for this login, or the configuration of your login does not allow you to reinitialize your password. Check the login you entered or contact your HOPEX Administrator."

Description of MEGA Data Exchange XML Format

This technical article presents detailed explanation on various tags used in MEGA XML data exchange files.

INTRODUCTION TO MEGA XML DATA EXCHANGE FORMAT

MEGA allows you to import and export data in a standard XML exchange format.

Data contained in MEGA data XML documents can be described in the form of commands as for MGR, MGL or MGE documents. It can also be described in Content mode. Content mode now allows processing of sets of objects independently, free of any context (see Content exchange: <Content> tag, Content Mode, Hierarchical link in content mode).

Importing an XML document in a MEGA repository means executing or creating the commands it contains in this repository.

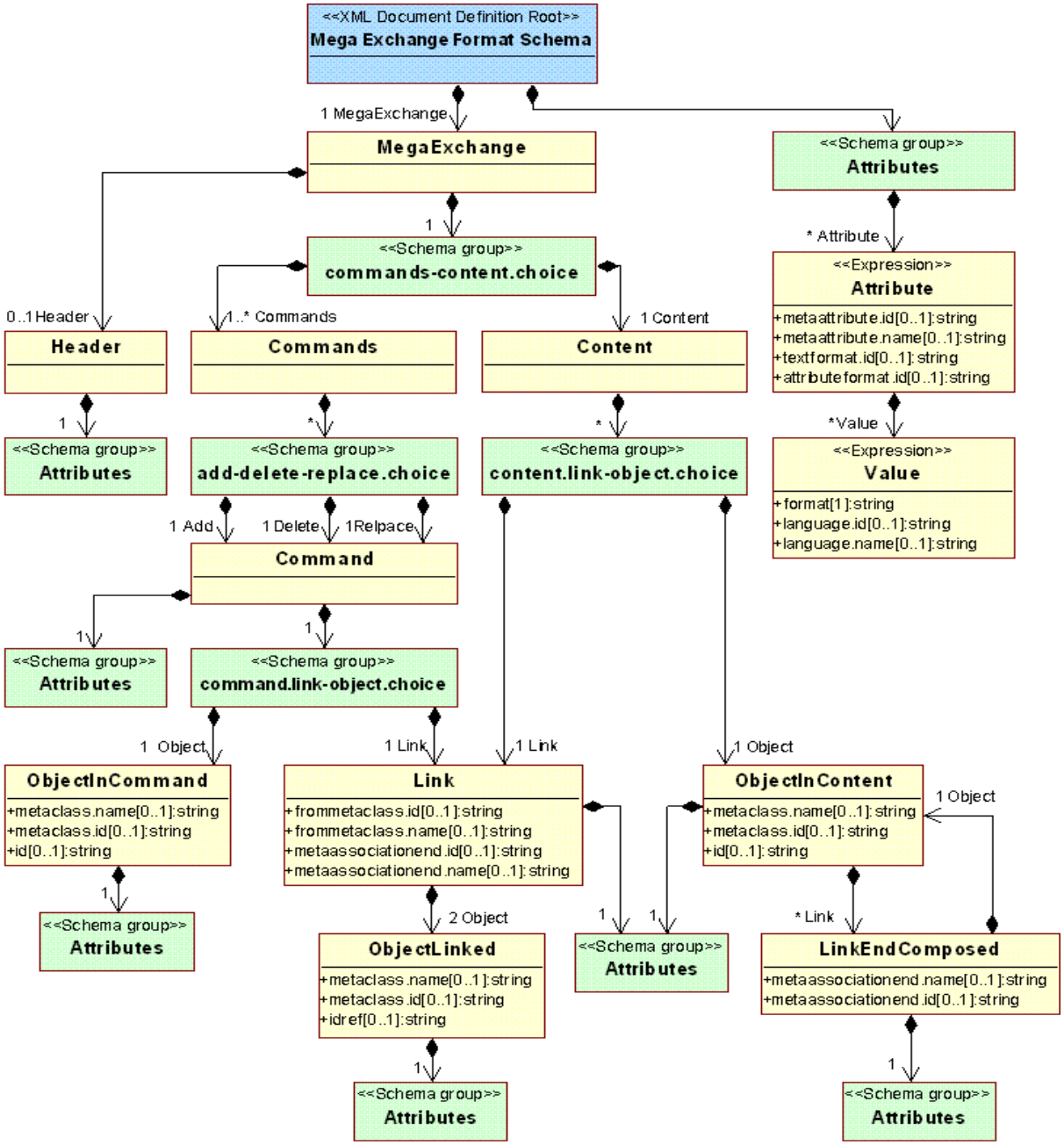
Exporting MEGA (objects, command) consists of dispatching this data in XML standard format. MEGA data exchange XML documents can also be created by software external to MEGA with a view to integrating the data they contain in a MEGA repository. Finally, they can be created manually.

You will find the XSD schema (“xmlmega.xsd”) of MEGA data exchange XML files in the MEGA installation "system" directory.

An XSLT style sheet is also provided in the MEGA installation "system" directory, which presents MEGA data exchange XML documents. This style sheet is provided only as an example.

Note: References to documentation concerning the following are given in the glossary: MEGA metamodel, XML language, importing a data file, exporting a data file.

FORMAT XML SCHEMA MODEL



Description

TAGS IN BRIEF

<MegaExchange>

This is the root element of the document. It contains all other tags.

(See Logical structure)

<Header>

This tag describes the characteristics of the document itself. We find here for example the exchange format version, the document creation date, the default exchange language, etc.

(See Logical structure)

<Attribute>

An <Attribute> tag enables expression of a command characteristic in an Add>, <Delete> or <Replace> tag, of an object in an <Object> tag, of a link in a <Link> tag, or of the exchange document itself in the <Header> tag.

(See Description of attribute values)

<Value>

This tag contains an attribute value. It appears only in <Attribute> tags. It serves to express a value in a particular format when this value can be expressed in several possible formats in an <Attribute> tag (for example, "internal" or "display" format).

(See Attribute value format)

<Commands>

This tag appears at root element level and enables expression of a set of commands.

(See Command exchange: <Commands> tag)

<Content>

This tag appears at root element level and enables inventory of a set of objects and links.

(See Content exchange: <Content> tag)

<Object>

An <Object> tag contains the description of an object: it is characterized by its type (MetaClass) and its attribute values. It can serve to describe or identify an object, for example an object to be connected at creation of a link.

(See Object Descriptions)

<Link>

A <Link> tag contains the description of a link: its type (MetaClass of the object to be connected and MetaAssociationEnd by which the second object is connected), identifications of the two objects to be connected and the link attribute values.

(See Link description)

<Add>

The <Add> tag is used in the <Commands> tag. It enables representation of an object or link creation command. Content is either an <Object> tag describing the object to be created, or a <Link> tag describing the link to be created.

(See Command exchange: <Commands> tag, Command Mode)

<Delete>

The <Delete> tag is used in the <Commands> tag. It enables representation of an object or link deletion command. Content is either an <Object> tag describing the object to be deleted, or a <Link> tag describing the link to be deleted.

(See Command exchange: <Commands> tag, Command Mode)

<Replace>

The <Replace> tag is used in the <Commands> tag. It enables representation of an object or link update command. Content is either an <Object> tag describing the object to be modified, or a <Link> tag describing the link to be modified.

(See Command exchange: <Commands> tag, Command Mode)

<Extension>

Various tags can contain the <Extension> tag.. When it is present in an element, this tag enables addition of supplementary information to the element (for example, addition of information concerning reasons for command reject). This information can be taken into account in a particular way according to tools used.

FORMAT IN DETAIL

In the remainder of this document, MEGA data exchange XML format will be referred to as "MEGA XML".

MEGA data exchange XML document structure

Physical structure

Like all XML documents, MEGA XML documents must start with XML declaration:

<?xml version="1.0"?>.

Encoding can be specified using the "encoding" attribute.

<?xml version="1.0" encoding="ISO-8859-1"?>

MEGA XML documents should be expressed in a code supported by the applications used to process them. For example, before importing a MEGA XML document it should be confirmed that MEGA can handle documents in the encoding proposed. MEGA import, like any XML analyzer, can basically handle input of Unicode encodings: UTF-8, UTF-16 little endian and big endian and ASCII. Other encodings such as ISO Latin1 FR can be used.

Note: see XML specifications for values to be specified for the "encoding" attribute (ref. Extensible Markup Language (XML) 1.0: http://www.w3.org/TR/2004/REC-xml-20040204/), and MEGA documentation to determine supported encodings.

Note: values that can be specified for the "encoding" attribute to identify the different encodings:

- "ISO-8859-1" corresponds to ISO Latin-1 FR encoding
- "UTF-8" corresponds to Unicode encoding on one or several bytes per character
- "UTF-16" corresponds to Unicode encoding on a multiple of two bytes per character

Logical structure

Like all XML documents, MEGA XML documents can have only one root tag, its name being *<MegaExchange>*.

A MEGA XML document firstly contains information relating to the document itself, such as the document creation date, the format version used or the attribute value expression language. This information is described in a *<Header>* tag.

In addition, information exchanged must be expressed either in a *<Commands>* tag, or in a *<Content>* tag

Example : MEGA data exchange XML document structure

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="format_version">Mega Xml Format Version
0.1</Attribute>
  </Header>
  <Commands>
    <!-- exchanged data -->
    <Add>
      <Object metaclass.name="Procedure" id="1">
        <Attribute metaattribute.name="Name">Procedure-1</Attribute>
      </Object>
    </Add>
  </Commands>
</MegaExchange>
```

Exchanged data description modes

Data exchanged via MEGA XML documents can be described in two modes:

- Either as a series of commands to be processed one after the other.
- Or as a repository content or sub-content, in other words as a set of objects.

Command exchange: <Commands> tag

Command series data expression mode is by use of the *<Commands>* tag.

It is this tag that contains command description tags in XML. The order of command description tags within the *<Commands>* tag is significant. It corresponds to the order in which commands must be processed by the

MEGA XML document analysis tools. In fact, a command may not be validated unless preceding commands have been processed.

For example, if a command for creation of a link between two objects appears before the command for creation of one (or both) of the objects themselves, the link creation command is not valid from a logical viewpoint.

Example : Command exchange

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Procedure" id="1">
        <Attribute metaattribute.name="Name">Procedure-1</Attribute>
      </Object>
    </Add>
  </Commands>
</MegaExchange>
```

Content exchange: <Content> tag

Content description mode uses the *<Content>* tag. This tag contains a collection of objects and links.

These objects and links should be interpreted as independent data free of any context: they are not connected to a particular repository and do not require any other data in order to be significant (except for the metamodel describing them).

Data of an XML exchange document using description mode produces creation commands (creation of described objects and links) when the document is imported.

Order of appearance of tags describing data contained in the *<Content>* tag is significant. It corresponds to the order in which data must be processed by the MEGA XML document analysis tools.

If a link between two objects appears before one (or both) of the objects themselves, the link description is not valid from a logical viewpoint.

Example : Content description mode data exchange

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Content>
    <!-- exchanged data -->
    <Object metaclass.name="Procedure" id="1">
      <Attribute metaattribute.name="Name">Procedure-1</Attribute>
    </Object>
  </Content>
</MegaExchange>
```

In content description mode, we can explicitly show the hierarchical view of exchanged data structure. In fact, tags describing objects can themselves contain other tags describing contained objects from a logical viewpoint (for example a procedure containing operations).

This hierarchical description is valid only in content description mode.

Example : <Object> tag containing an <Object> tag

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Content>
    <!-- exchanged data -->
    <Object metaclass.name="Procedure" id="1">
      <Attribute metaattribute.name="Name">Procedure-1</Attribute>
      <!-- hierarchical link use  -->
      <Link metaassociationend.name="Contained operation">
        <!-- contained object -->
```

```
        <Object metaclass.name="Operation" id="2">
          <Attribute metaattribute.name="Name">Procedure-1</Attribute>
        </Object>
      </Link>
    </Object>
  </Content>
</MegaExchange>
```

Commands description: <Add>, <Delete>, <Replace> tags

Command tags can be used only in command mode. These tags are contained in the *<Commands>* tag explained in the chapter Command exchange: <Commands> tag.

The three available commands are:

- Creation command represented by an *<Add> tag.*
- Deletion command represented by a *<Delete> tag.*
- Modification command represented by a *<Replace> tag.*

Each of these three commands can be applied to an object or to a link: a tag representing a command can contain one tag only: *<Object>* or *<Link>*.

In command mode, *<Object>* tags cannot directly or indirectly contain other *<Object> tags.* : The hierarchical aspect of exchanged data logical structure cannot be represented by the hierarchical aspect of XML when data is exchanged in command mode.

Example : Object and link creation commands

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Class">
        <Attribute metaattribute.name="UML ClassName (Français)">Ma classe</Attribute>
```

```
      </Object>
    </Add>
    <Add>
      <Link frommetaclass.name="Package" metaassociationend.name="Contained class ">
        <Object metaclass.name="Package">
          <Attribute metaattribute.name="Name">My package</Attribute>
        </Object>
        <Object metaclass.name="Class">
          <Attribute metaattribute.name="Name">My org-unit</Attribute>
        </Object>
        <Attribute metaattribute.name="Order">9999</Attribute>
      </Link>
    </Add>
  </Commands>
</MegaExchange>
```

Object Descriptions

Objects are described by the *<Object> tag.*.

The MetaClass of the object is identified by the name or MEGA absolute identifier (idabs in hexadecimal) of the MetaClass. The name of the MetaClass is specified by the value of the *"metaclass.name"* attribute of the *<Object> tag*, the idabs of the MetaClass is specified by the value of the *"metaclass.id"* attribute of the *<Object> tag*.

In addition, the objects themselves must be identified when we wish to make reference to them in the exchange document (see Object identification mechanisms).

Example : Object metaclass identification by metaclass name

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Content>
    <!-- exchanged data -->
    <Object metaclass.name="Procedure">
        <Attribute metaattribute.name="Name">Procedure-1</Attribute>
```

```
    </Object>
  </Content>
</MegaExchange>
```

Example : Object metaclass identification by metaclass idabs

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Content>
    <!-- exchanged data -->
    <Object metaclass.id="B1EDB25E2C1401BB">
      <Attribute metaattribute.name="Name">Procedure-1</Attribute>
    </Object>
  </Content>
</MegaExchange>
```

Command Mode

In commands description mode, an object is described in an *<Add>*, *<Delete>* or *<Replace>* tag, depending on whether we wish to create, delete or modify the object.

In this case, the *<Object>* tag describing the object can if necessary contain an *<Extension>* tag containing information not allowed for by MEGA XML format. It also contains *<Attribute>* tags specifying attribute values characterizing this object, these tags following the *<Extension>* tag if this is present.

Example : Object creation command by attribute specification

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
```

```
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Procedure" id="1">
        <Attribute metaattribute.name="Name">Procedure-1</Attribute>
        <Attribute metaattribute.name="Type-Procedure">General</Attribute>
      </Object>
    </Add>
  </Commands>
</MegaExchange>
```

An object described in a deletion command must make reference to an existing object recognized by the tool analyzing the MEGA XML document. The object description must therefore specify the value of its MEGA absolute identifier (object "_idabs" attribute) in an *<Attribute> tag*.

Example : Object deletion command

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Delete>
      <Object metaclass.name="Procedure">
        <!-- object identifier attribute -->
        <Attribute metaattribute.name="_IdAbs">MnJgyaAJ0100</Attribute>
      </Object>
    </Delete>
  </Commands>
</MegaExchange>
```

An object described in a modification command must make reference to an existing object recognized by the tool analyzing the MEGA XML document. The object description must therefore specify the value of its MEGA absolute identifier (object "_idabs" attribute) in an *<Attribute> tag*. In addition to the *<Attribute>* tag specifying

the MEGA absolute identifier value, we find *<Attribute>* tags giving the values of attributes to be changed for the object concerned.

Example : Object modification command

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Replace>
      <Object metaclass.name="Procedure">
        <!-- object identifier attribute -->
        <Attribute metaattribute.name="_IdAbs">MnJgyaAJ0100</Attribute>
        <!-- modified attribute -->
        <Attribute metaattribute.name="Code-Procedure">XYZ</Attribute>
      </Object>
    </Delete>
  </Commands>
</MegaExchange>
```

Content Mode

In content description mode, *<Object>* tags describing objects are contained in the *<Content> tag*.

The *<Object>* tag describing the object can if necessary contain an *<Extension>* tag containing information not allowed for by MEGA XML format. It also contains *<Attribute>* tags specifying attribute values characterizing this object. These tags follow the *<Extension>* tag if this is present.

In addition, in content description mode, objects can be described as containing other objects. The *<Object>* tag can therefore indirectly contain other *<Object>* tags describing contained objects (see Hierarchical link in content mode).

The same considerations apply to *<Object>* tags representing contained objects, which can themselves contain *<Object> tags*.

Example: Object description in content mode

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Content>
    <!-- exchanged data -->
    <Object metaclass.name="Procedure" id="1">
      <Attribute metaattribute.name="Name">Procedure-1</Attribute>
    </Object>
  </Content>
</MegaExchange>
```

Object identification mechanisms

We need to make reference to objects internal or external to the document within the framework of:

- An object modification command
- An object modification command
- The link between one object and another

We distinguish objects described in the document from those not described in the document but which are the object of a command:

- Objects internal to the document: these are described in the document and can be referenced in the document via use of *"id"* and *"idref"* attributes of the *<Object> tag*.
- Objects external to the document: these are not described in the document but can be the target of a command. These should be recognized by the tool processing the document and are identified by the "_idabs" attribute of the object.

Object identification therefore uses either an identifier internal to the document or a MEGA absolute identifier.

Identification internal to the document is by use of the *"id"* attribute of the *<Object>* tag. The value of the *"id"* attribute must be unique throughout the document with no other constraint on form; it is an ID type attribute (ID type is defined in XML1.0 specifications). Reference to a document object is by using the *"idref"* attribute of the *<Object>* tag, which must therefore have the same value as the *"id"* attribute of the *<Object>* tag representing the referenced object. For example, within the framework of a link, the *<Object>* tags referencing linked objects can each have an *"idref"* attribute.

External identification is by definition of an *<Attribute>* tag defining the value of its MEGA absolute identifier (object "_idabs" attribute). Objects can be created with an "_idabs" attribute by use of the *<Attribute>* tag (see Description of attribute values). Similarly, *<Object>* tags referencing objects, as in the *<Delete>*, *<Replace>* or *<Link>* tags, can identify objects by their MEGA absolute identifier, and in this case they contain an *<Attribute>* tag specifying the value of the object "_idabs" attribute.

Example : Identification by <Object> tag id attribute

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Content>
    <!-- exchanged data -->
    <Object metaclass.name="Procedure" id="1">
      <Attribute metaattribute.name="Name">Procedure-1</Attribute>
    </Object>
    <Object metaclass.name="Procedure" id="2">
      <Attribute metaattribute.name="Name">Procedure-2</Attribute>
    </Object>
    <Link frommetaclass.name="Procedure" metassociationend.name="Next procedure">
      <Object metaclass.name="Procedure" idref="1"/>
      <Object metaclass.name="Procedure" idref="2"/>
    </Link>
  </Content>
</MegaExchange>
```

Example : Identification by idabs

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Procedure">
        <Attribute metaattribute.name="Name">Procedure-1</Attribute>
        <!-- object identifier attribute -->
        <Attribute metaattribute.name="_IdAbs">MnJgyaAJ0100</Attribute>
      </Object>
    </Add>
    <Delete>
      <Object metaclass.name="Procedure">
        <!-- object identifier attribute -->
        <Attribute metaattribute.name="_IdAbs">MnJgyaAJ0100</Attribute>
      </Object>
    </Delete>
  </Commands>
</MegaExchange>
```

Link description

Links are described by the *<Link> tag.*.

The MetaAssociation of the link is identified by the name or MEGA absolute identifier (idabs in hexadecimal form) of the MetaAssociationEnd by which the objects are connected.

The name of the MetaAssociationEnd is specified by the value of the *"metaassociationend.name"* attribute of the *<Link>* tag.

The idabs of the MetaAssociationEnd is specified by the value of the *"metaassociationend.id"* attribute of the *<Link>* tag.

One of the attributes *"metaassociationend.name"* or *"metaassociationend.id"* must be present to identify the link. Both can be present simultaneously.

Hierarchical link in content mode

In content mode, a link can be represented hierarchically. The *<Object>* tag representing the first object therefore contains a *<Link>* tag which itself contains the *<Object>* tag describing the connected object.

The *<Link>* tag serves to specify the MetaAssociationEnd by which the second object is connected. To do this, we must therefore specify the *"metaassociationend.name"* attribute or the *"metaassociationend.id"* attribute.

In addition, the *<Link>* tag can contain *<Attribute>* tags describing link attribute values. These *<Attribute>* tags are placed before the *<Object>* tag.

Example: Hierarchical ink description in content mode

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Content>
    <!-- exchanged data -->
    <Object metaclass.name="Procedure" id="1">
      <Attribute metaattribute.name="Name">Procedure-1</Attribute>
      <!-- hierarchical link use  -->
      <Link metaassociationend.name="Contained operation">
        <!-- link attributes -->
        <Attribute metaattribute.name="Order">1</Attribute>
        <!-- contained object -->
        <Object metaclass.name="Operation" id="2">
          <Attribute metaattribute.name="Name">Procedure-1</Attribute>
        </Object>
      </Link>
    </Object>
  </Content>
</MegaExchange>
```

Other links

Links that are not hierarchical links can be used in both command mode command tags (*<Add>*, *<Delete>*, *<Replace>*) and in the content mode *<Content>* tag. These links make reference to two connected objects describing but not referencing these objects. The connected objects can be external to the document.

A link between two objects is described by the MetaAssociationEnd by which the second object is connected to the first. To do this, the *<Link>* tag representing the link contains either the MetaAssociationEnd idabs value in hexadecimal form in the *"metaassociationend.id"* attribute, or the MetaAssociationEnd name value in the *"metaassociationend.name"* attribute. In the latter case, the MetaClass of origin of the MetaAssociationEnd must be specified, ie. the MetaClass of the first object. This is done via the *"frommetaclass.id"* or *"frommetaclass.name"* attributes of the *<Link>* tag, specifying either the MetaClass idabs in hexadecimal form or the MetaClass name.

In addition, the *<Link>* tag contains two *<Object>* tags that reference the two connected objects. Each of these has a *"metaclass.id"* attribute specifying either the MetaClass idabs in hexadecimal form, or a *"metaclass.name"* attribute specifying its name. The object is referenced either by specifying the *"idref"* attribute value of the *<Object>* tag in the *<Link>* tag, which should be equal to the *"id"* attribute value of the *<Object>* tag describing the object referenced in the document, or by adding an *<Attribute>* tag specifying the "_idabs" attribute value of the referenced object, which in this case can be an object not described in the document.

Finally, the *<Link>* tag can contain *<Attribute>* tags describing link attribute values. These *<Attribute>* tags are placed after the two *<Object>* tags.

<u>Example : Description of a link between two objects of the document: use of “idref” attribute</u>

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Content>
    <!-- exchanged data -->
    <Object metaclass.name="Package" id="1">
      <Attribute metaattribute.name="Name">Package-1</Attribute>
    </Object>
    <Object metaclass.name="Package" id="2">
      <Attribute metaattribute.name="Name">Package-2</Attribute>
    </Object>
```

```
    <Link frommetaclass.name="Package" metaassociationend.name="Referenced package">
      <!-- link source object -->
      <Object metaclass.name="Package" idref="1"/>
      <!-- link destination object -->
      <Object metaclass.name="Package" idref="2"/>
      <!-- link attributes -->
      <Attribute metaattribute.name="Order">9999</Attribute>
    </Link>
  </Content>
</MegaExchange>
```

Example : Description of a link between two objects by idabs

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Link frommetaclass.name="Package" metaassociationend.name="Referenced package">
        <!-- link source object -->
        <Object metaclass.name="Package">
          <Attribute metaattribute.name="_idabs">ODiTpSNApa10</Attribute>
        </Object>
        <!-- link destination object -->
        <Object metaclass.name="Package">
          <Attribute metaattribute.name="_idabs">dCyP0ZtmxSA8</Attribute>
        </Object>
        <!-- link attributes -->
        <Attribute metaattribute.name="Order">9999</Attribute>
      </Link>
    </Add>
  </Commands>
</MegaExchange>
```

Description of attribute values

Attribute values are specified in *<Attribute>* tags. The following tags can contain attribute values: *<Header>*, *<Link>*, *<Object>*, *<Add>*, *<Delete>*, *<Replace>*.

In an *<Attribute>* tag, the valuated characteristic is identified by the *"metaattribute.name"* attribute, which determines its name, or by the *"metaattribute.id"* attribute, which determines its idabs in hexadecimal form.

The value is directly contained in the *<Attribute>* tag..

Example : Description of an attribute value

```
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Package">
        <!-- object attributes -->
        <Attribute metaattribute.name="_idabs">ODiTpSNApa10</Attribute>
        <Attribute metaattribute.name="nom">Packagee-1</Attribute>
      </Object>
    </Add>
  </Commands>
</MegaExchange>
```

Attribute value format

Direct values in *<Attribute>* tags are expressed in the format specified by the *<Attribute>* tag representing the *"default_format"* attribute of the *<Header>* tag.

If this attribute is not specified, default format is left to the tool analyzing the document.

Example : Attribute value default format

```
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
    <Attribute metaattribute.name="default_format">internal</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Package">
        <!-- object attributes -->
        <Attribute metaattribute.name="_idabs">ODiTpSNApa10</Attribute>
        <Attribute metaattribute.name="nom">Packagee-1</Attribute>
      </Object>
    </Add>
  </Commands>
</MegaExchange>
```

The *<Attribute>* tag enables attribute value specification in several formats. The different forms that the attribute value can take are each contained in a *<Value>* tag, the format being specified by the *<Value>* tag *"format"* attribute.

<u>Example : Attribute value expressed in several formats</u>

```
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="xmg_version">XMG version 0.1</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Application">
        <!-- object attributes -->
        <Attribute metaattribute.name="_idabs">ODiTpSNApa10</Attribute>
        <Attribute metaattribute.name="name">Application-1</Attribute>
        <Attribute metaattribute.name="MMI">
          <Value format="internal">W</Value>
          <Value format="display">Windowed</Value>
        </Attribute>
```

```
      </Object>
    </Add>
  </Commands>
</MegaExchange>
```

Translatable attributes and attribute value expression language

In the MEGA repository, for a given translatable attribute, there are as many attributes as there are languages into which the attribute is translatable. For example, for the "Name" attribute there are corresponding "Name (English)","Name (French)", "Name (Dutch)", etc. attributes.

In a MEGA XML document, the *"language.id"* and "language.name" attributes of the *<Value>* tag enable specification of the language in which the attribute value is expressed.

Translatable attributes are therefore identified by the name or absolute identifier of the root attribute using the *"metaattribute.name"* and *"metaattribute.id"* attributes of the *<Attribute>* tag, and the language in which the value is expressed using the *"language.name"* and *"language.id"* attributes.

For value expression language to be specified, at least one of the two attributes *"language.name"* or *"language.id"* must be present in the *<Value>* tag.

When neither the *"language.name"* nor the *"language.id"* attribute is present, the value is expressed in the language specified by the *<Attribute>* tag representing the *"model_default_language"* attribute of the *<Header>* tag.

<u>Example : Attribute value expressed in default language</u>

```
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="model_default_language"> Français</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Application">
        <!-- object attributes -->
        <Attribute metaattribute.name="_idabs">ODiTpSNApa10</Attribute>
        <Attribute metaattribute.name="name">Application-1</Attribute>
      </Object>
    </Add>
```

```
  </Commands>
</MegaExchange>
```

Example : Attribute value expressed in several languages

```
<MegaExchange>
  <Header>
    <!-- document information -->
    <Attribute metaattribute.name="model_default_language"> Français</Attribute>
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Application">
        <!-- object attributes -->
        <Attribute metaattribute.name="_idabs">ODiTpSNApa10</Attribute>
        <Attribute metaattribute.name="name">
          <Value>Invoicing</Value>
          <Value language.name="English">Invoicing</Value>
        </Attribute>
      </Object>
    </Add>
  </Commands>
</MegaExchange>
```

MEGA XML format extensions

MEGA XML format includes an extension mechanism enabling addition of information to different format tags.

A format extension is created by use of the *<Extension>* tag. The extension tag can contain any information expressed in XML.

Information contained in the *<Extension>* tag is not part of the format. It cannot be taken into account by a tool using MEGA XML format.

The following tags can contain an extension tag: *<Header>*, *<Link>*, *<Object>*, *<Add>*, *<Delete>*, *<Replace>*.

Example of use of the *<Extension>* tag by MEGA

The *<Extension>* tag is used by the MEGA import function which creates a report of commands analyzed at import. The report contains commands analyzed in MEGA XML format indicating if they have been accepted, rejected or contain warnings. Therefore each MEGA XML format command in the report can contain an *<Error>* tag or *<Warning>* tags in the *<Extension>* tag of the command concerned. These tags enable appreciation of the import result, but are not to be taken into account in another context.

Example 1 : Use of the *<Extension>* tag by MEGA

```
<MegaExchange>
  <Header>
    <!-- document information -->
  </Header>
  <Commands>
    <!-- exchanged commands -->
    <Add>
      <Object metaclass.name="Application">
        <Extension>
          <Error code="XXXXXXXX">There is no « metaattribute » with « color » value as
« name »<Error>
        </Extension>
        <!-- object attributes -->
        <Attribute metaattribute.name="_idabs">ODiTpSNApa10</Attribute>
        <Attribute metaattribute.name="name">Invoicing</Attribute>
        <Attribute metaattribute.name="color">blue</Attribute>
      </Object>
    </Add>
  </Commands>
</MegaExchange>
```

Attributes of MEGA XML documents used by MEGA

A MEGA XML document exported from a MEGA repository contains a certain number of attributes that describe general elements relating to the document itself. These are described in this section.

Document attributes are located in the *<Header>* tag of the document.

Example: Document attributes

```
<?xml version="1.0" encoding="UTF-8"?>
<MegaExchange>
  <Header>
```

```
    <Attribute metaattribute.id="27C729AE3F4E004A"
metaattribute.name="exchange_format_version">
      <Value format="internal">000100</Value>
      <Value format="display">MEGA XML version 0.1</Value>
    </Attribute>
    <Attribute metaattribute.id="27C729AE3F4E0050" metaattribute.name="mega_version">
      <Value format="internal">25856</Value>
      <Value format="display">MEGA Version 2005 Pre release</Value>
    </Attribute>
    <!-- ... -->
  </Header>
  <Commands>
    <!-- ... -->
  </Commands>
</MegaExchange>
```

Document attributes list

- exchange_format_version : this attribute indicates the version of MEGA XML exchange format.
- mega_version : this attribute indicates the version of MEGA used at export.
- metamodel_language : this attribute specifies the language used to identify types in the attributes of *"metaclass.name"*, *"metaattribute.name"*, *"metaassociationend.name"* tags.
- model_default_language : this attribute specifies the default language used for translatable attribute values of objects and links when the language has not been specified by *"language.name"* or *"language.id"* attributes of the *<Value>* tag.
- author : the value of the *author* attribute identifies the user that executed export.
- source_database : the value of the *source_database* attribute identifies the repository from which export was executed.
- creation_date : specifies MEGA XML document creation date.

GLOSSARY

XML attribute:

An XML attribute is unstructured text data contained in an XML tag.

Example: <attribute tag="value"/>

MEGA object attribute:

Object characteristics are called attributes. For a given object, attributes can take a value of which the type is defined for the attribute.

XML tag:

XML tags are syntax elements that structure XML documents. They mark opening and closing of XML document data definition.

Example: <tag>…</tag>

MEGA command:

The various actions possible that modify a MEGA repository are called commands. In particular these include creation, deletion and modification of objects and links.

XML element :

An XML element is structured data comprising an opening tag, attributes of this tag, a closing tag and text data and tags contained between opening and closing tags.

Example: <attribute tag="value">text<sub-tag/></tag>

File export

MEGA repository data can be exported in a file. From a MEGA repository we can export : objects (exported data describing exported objects), transactions (exported data describing transaction commands) or the complete repository (exported data describing all repository objects, this procedure also being known as logical backup). The various export functions are described in the "MEGA Help" CHM file and in the "MEGA Administration" PDF user guide in the following sections: "Administration > Managing Transactions > Managing Updates > Exporting the Transaction Logfile", "Administration > Managing Objects > Exporting MEGA objects", "Administration > Managing Repositories > Reorganizing a Repository > Logical Backup of a Repository".

MEGA APIs enable file export using Automation interfaces. Data export and logical backup functions using API are covered in the "MEGA Help" CHM file in the section "API > API - Reference Guide > Administration > MEGA Database" and in the "MEGA API" PDF user guide in the similarly named section.

IdAbs

Absolute identifier of data stored in the MEGA repository, idabs is generally represented in hexadecimal or base 64 form. The idabs is for example an object attribute (attribute "_idabs").

File import

Importing an XML document in a MEGA repository means executing or creating the commands it contains in this repository. MEGA can import files of type MGR, MGL, MGE and MEGA XML. The import function is covered in the "MEGA Help" CHM file in the section "Administration > Managing Repositories > Reorganizing a Repository > Updating a Repository (Importing)" and in the "MEGA Administration" PDF user guide in the similarly named section.

MEGA APIs enable file import using Automation interfaces. The API import function is covered in the "MEGA Help" CHM file in the section "API > API - Reference Guide > Administration > MEGA Database" and in the "MEGA API" PDF user guide in the similarly named section.

MEGA links :

For a given repository object, links enable definition of connected repository objects.

MEGA MetaAssociation :

This term indicates relationships existing between repository object types. MetaAssociations define repository links.

MEGA MetaClass :

This term indicates object types stored in the repository. MetaClasses define repository objects.

MEGA metamodel :

This comprises all MetaClasses and MetaAssociations defining objects that can be created and handled in the MEGA repository. The MEGA metamodel is covered in the "MEGA Help" CHM file in section "Metamodel and Glossaries > Metamodel".

MEGA object

Data stored in a MEGA repository are called repository objects. An object comprises attributes and can be connected to other repository objects. In addition, every object is identified uniquely by an attribute called the MEGA absolute identifier or idabs.

MEGA XML

Common name for MEGA data exchange XML format.

XML

Extendable tag language (eXtensible Markup Language), XML is a metalanguage describing languages of tagged hierarchical structure. XML specifications provide syntax basics of XML languages (ref. Extensible Markup Language (XML) 1.0: http://www.w3.org/TR/2004/REC-xml-20040204/).

REPORTING DATAMART

MEGA International

9 avenue René Coty - 75014 Paris, France
tél +33 (0)1 42 75 40 00 - fax +33 (0)1 42 75 40 95 - info@mega.com - www.mega.com
S.A. au capital de 1 106 698 € - RC Paris B 385 185 806 000 28 / NAF 741 G

1. REPORTING DATAMART OVERVIEW 3
2. REPORTING DATAMART DESCRIPTION 4
2.1. Structure 4
2.2. Data 5
2.3. Content 5
2.4. Excluding MetaAttribute values from the Reporting Datamart 5
3. CREATING AND INITIALIZING A REPORTING DATAMART 6
3.1. Creating a Reporting Datamart 6
3.2. Initializing the Reporting Datamart structure 9
3.3. Initializing the Reporting Datamart data 9
4. SYNCHRONIZING THE REPORTING DATAMART WITH THE HOPEX REPOSITORY CONTENT 10
4.1. Synchronization frequency 10
4.2. Launching an incremental synchronization 10
4.3. Launching a calculated MetaAttribute synchronization 11
4.4. Launching a diagram synchronization 12
5. DELETING A REPORTING DATAMART 14
6. USING THE REPORTING DATAMART 15
6.1. Usage 15
6.2. Advantages 15
7. REPORTING DATAMART DETAILED DESCRIPTION 16
7.1. Reporting Datamart table classification 16
7.2. Reporting Datamart table name format 16
7.2.1. Technical tables 17
7.2.2. MetaClass occurrence tables 18
7.2.3. MetaAssociation tables 18
7.3. Reporting Datamart columns 20
7.3.1. MetaClass occurrence table columns 20
7.3.2. Link table columns 20
7.4. Use case: reading the Reporting Datamart through a link 21
7.4.1. Accessing your Reporting Datamart tables 21
7.4.2. Understanding a link table 22
7.4.3. Finding the MetaClass corresponding to an IdAbs 22
7.4.4. Finding attributes or tagged values belonging to a MetaClass occurrence 23
7.4.5. Finding information on an attribute 24
7.5. Use case: saving the diagram drawings 26

1. REPORTING DATAMART OVERVIEW

With the **Reporting Datamart** feature you can create a **Reporting Datamart**, which is a replicated RDBMS Database from an HOPEX repository content.

The **Reporting Datamart** is made up of data selected at creation and synchronized on regular basis, to keep the **Reporting Datamart** updated according to the HOPEX repository content.

The purpose of the **Reporting Datamart** feature is to be used as a source for any usage that needs HOPEX data (for example: reporting).

The **Reporting Datamart** feature is available with **HOPEX Datamart** license.

2. REPORTING DATAMART DESCRIPTION

2.1. Structure

The **Reporting Datamart** structure is initialized with the RDBMS database standard structure.

Tables are created and fed with the HOPEX repository and SystemDb repository data. If there is no data to be exported to a table, this table remains empty, so that the **Reporting Datamart** structure is always consistent. This ensure the user that any of his queries run properly on the **Reporting Datamart**.

Reporting Datamart table and column names are in English language.

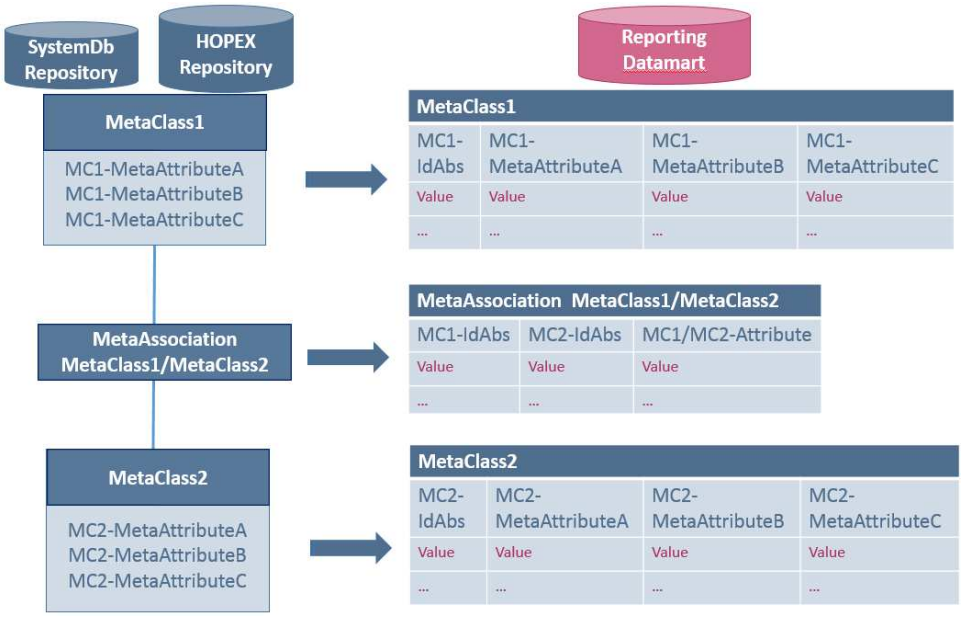
Reporting Datamart data is stored in the language selected at **Reporting Datamart** creation.

SQL database-based Reporting Datamart

For an SQL database-based **Reporting Datamart** the rule is as follows:

- an SQL Database Table is created for each HOPEX MetaClass
- an SQL Database Table is created for each HOPEX MetaAssociation
- an SQL Database Column is added in the SQL database Table for each HOPEX MetaAttribute

See Reporting Datamart Detailed description for a detailed description.



2.2. Data

HOPEX repository

Most of the HOPEX repository data is exported to the **Reporting Datamart**, including:

- MetaClasses, MetaAssociations, and MetaAttributes
- diagrams exported in BLOB column of their object
- calculated MetaAttributes
- comments, in html format (instead of RTF format)

Logs and object history are not exported.

SystemDb repository

Few data is exported from the SystemDb repository:

- Logins
- Persons (system)
- Person groups

2.3. Content

With the **Reporting Datamart,** data is filtered so that the data amount is reduced to the data needed only:

- Only the last dispatched version of each object is kept in the Reporting Datamart.
- Data is filtered at Reporting Datamart creation according to a user rights (confidentiality) and his profile permissions.
- Data name is displayed (not Idabs) for an easy reading.
- Data is in the **Reporting Datamart** language.

  Note: if you need the **Reporting Datamart** data in multiple languages, you need to create as many **Reporting Datamart** as languages needed.

2.4. Excluding MetaAttribute values from the Reporting Datamart

In some cases, you might not need to export some specific MetaAttribute values in your **Reporting Datamart**.

To exclude MetaAttribute values from the Reporting Datamart:

1) In HOPEX, access the MetaAttribute properties.
2) Click the **Characteristics** tab, then **Advanced** subtab.
3) Click the **Extended Properties** field arrow and select **Exclude from Reporting Datamart**.

3. CREATING AND INITIALIZING A REPORTING DATAMART

You can create and initialize a **Reporting Datamart** from an RDBMS HOPEX repository of your HOPEX Environment as follows:

Step	Reporting Datamart		Description	See
1	Creation		- Reporting Datamart data language definition - User and profile selection	Creating a Reporting Datamart
2	Initialization	Structure (optional)	Creation of all the tables and columns according to the HOPEX repository metamodel	Initializing the Reporting Datamart structure
		Data	Creation and feeding of all the Reporting Datamart tables, columns and rows according to the HOPEX repository content	Initializing the Reporting Datamart data

Note: you can also initialize the Reporting Datamart using APIs, see HOPEX Power Studio > Initializing and synchronizing a Reporting Datamart.

3.1. Creating a Reporting Datamart

At **Reporting Datamart** creation you define:

- the data language of the **Reporting Datamart**

  Note: not translated data appears in its Idabs format.
- the user and its profile

  They both define and filter the HOPEX repository data according to their access rights (confidentiality: reading access areas of object occurrences and of linked objects for link occurrences) and permissions.

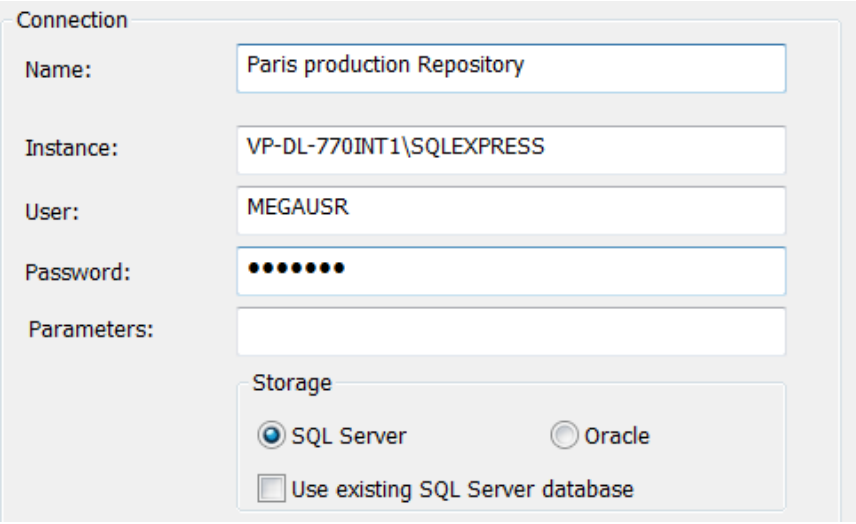
To create a Reporting Datamart from a HOPEX repository:

1) Launch **MEGA Administration** application.
2) Connect to the environment from which you want to replicate the data.
3) In the **Repositories** folder, expand the repository folder concerned.
4) Right-click **Reporting Datamart** and select **New**.

   The Creating a Reporting Datamart window opens.
5) In the **Connection** pane:
   a) In the **Name** field, enter the name of the **Reporting Datamart**.

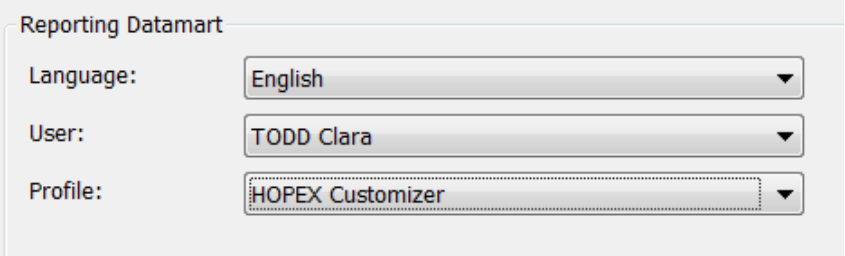
      (Optional) If needed you can modify the HOPEX repository **Instance** set by default as well as its connection parameters (**User**, **Password**, **Parameters)**, see *RDBMS Repository Installation Guide* for detailed information.
   b) In the **Storage** pane, select the **Reporting Datamart** storage type.

Note: if you do not have the right to create a database, select **Use existing SQL Server database** (in the **Name** field you must have entered the name of the existing database you want to use).

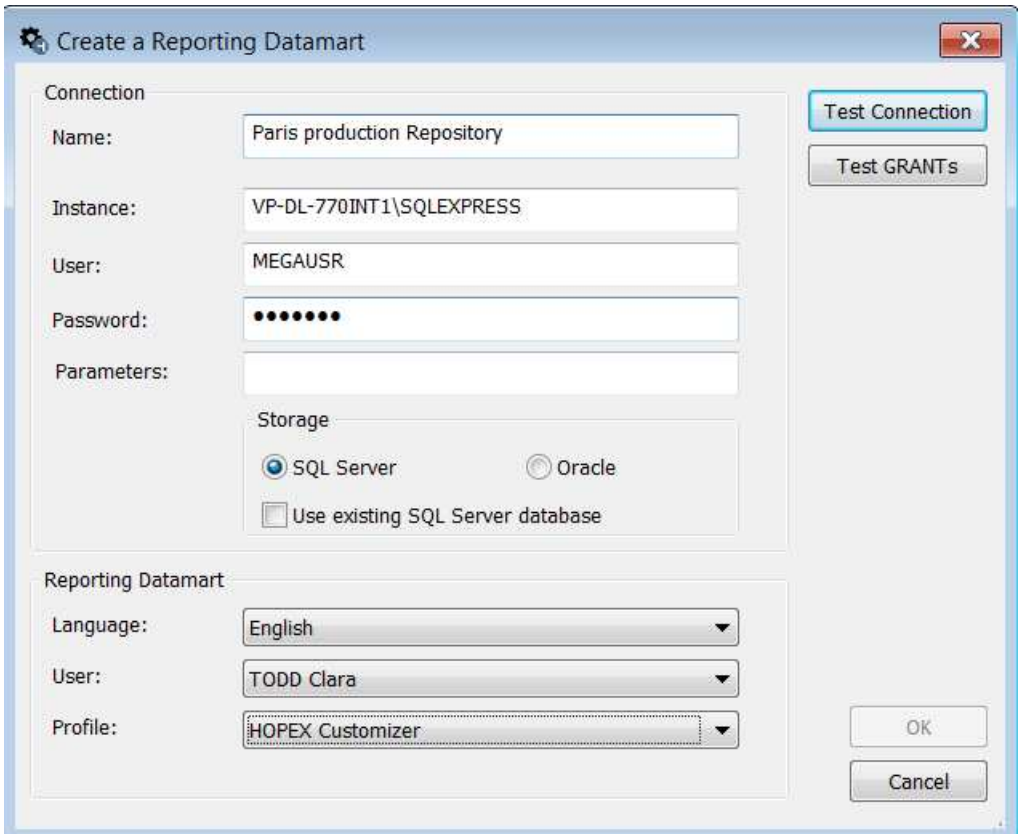


6) In the Reporting Datamart pane:
   a) In the **Language** field, select the data language of the **Reporting Datamart**.

      Note: select a language in which the data is available, so that you can read it.
   b) In the **User** field, select the user that defines the access to HOPEX repository.
   c) In the **Profile** field, select the profile that defines the access to HOPEX repository.



7) Click **Test Connection**.
8) Click **Test GRANT's**.



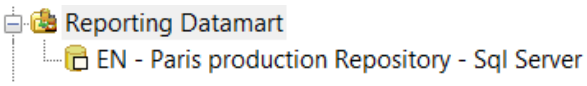
9) Click **OK**.

The Reporting Datamart is created.

The **Reporting Datamart** is added in the Reporting Datamart folder with the following format:

<Language> - <Reporting Datamart name> - <Storage type>

`Example: EN – Paris production Repository – Sql server`



3.2. Initializing the Reporting Datamart structure

Once the **Reporting Datamart** is created you can initialize the structure with the HOPEX repository metamodel.

All the tables and columns are created even if there is no data to feed them. The table remains empty, so that the **Reporting Datamart** structure is always consistent. This ensures the user that any of his queries run properly on the **Reporting Datamart**.

To initialize the Reporting Datamart structure from the HOPEX repository:

1) Launch **MEGA Administration** application.
2) Connect to the environment concerned.
3) In the **Repositories** folder, expand the repository folder concerned.
4) In the **Reporting Datamart** folder, right-click the **Reporting Datamart** concerned and select **Initialize > Structure**.

   The **Reporting Datamart** structure creation starts.

   The **Reporting Datamart** is initialized according to the HOPEX repository metamodel: all the tables (columns and rows) are created.

   If you have already initialized the **Reporting Datamart** data, only the empty tables and columns are added.

   Else initialize the **Reporting Datamart** data to feed the tables, rows and columns (see Initializing the Reporting Datamart data section).

3.3. Initializing the Reporting Datamart data

Once the **Reporting Datamart** is created you need to initialize the **Reporting Datamart** data with the HOPEX repository content according to the user/profile permissions and the object type selected.

To initialize the Reporting Datamart data from the HOPEX repository:

1) Launch **MEGA Administration** application.
2) Connect to the environment concerned.
3) In the **Repositories** folder, expand the repository folder concerned.
4) In the **Reporting Datamart** folder, right-click the **Reporting Datamart** concerned and select **Initialize > Data**.
5) Click **Start**.

   The **Reporting Datamart** data initialization starts.

   The **Reporting Datamart** Data is initialized according to the HOPEX repository content.

4. SYNCHRONIZING THE REPORTING DATAMART WITH THE HOPEX REPOSITORY CONTENT

To keep your **Reporting Datamart** up-to-date with HOPEX repository updates, you can synchronize it with HOPEX Repository content thanks to the following provided triggers:

- Incremental synchronization
- Calculated MetaAttribute synchronization
- Diagram synchronization

Although, if needed, you can modify the **Reporting Datamart** data, these modifications are never synchronized with the HOPEX repository. Synchronization is only one way: from HOPEX repository to **Reporting Datamart**.

4.1. Synchronization frequency

The schedule policy of these synchronization triggers depends on the object volume in the HOPEX Repository and on the up-to-date data level you want to deliver to the Reporting Datamart users.

If you need a perfect synchronization between data, calculated MetaAttributes and Diagrams you can daily run the corresponding triggers at the same time.

Synchronization scheduling use: once scheduled the trigger launch the synchronization on all the repositories and for all their Reporting Datamarts.

To trigger the action on a specific repository for a specific Reporting Datamart, see HOPEX Power Studio > All about starting with APIs > Initializing and synchronizing a Reporting Datamart.

4.2. Launching an incremental synchronization

The HOPEX repository log must be activated (default value for an RDMS repository).

Launch an incremental synchronization to update the **Reporting Datamart** with all the dispatches performed after the last synchronization.

You can:

- at any time, manually launch an incremental synchronization
- define a scheduled trigger for the incremental synchronization

  Trigger: Reporting Datamart Synchronization

  Frequency: you can schedule the incremental synchronization with a high frequency (i.e.: every 10').

  For information regarding the scheduling, see *HOPEX Power Studio > Using the Scheduler > Scheduler*.

To manually launch an incremental synchronization:

1) Launch **MEGA Administration** application.
2) Connect to the environment concerned.
3) In the **Repositories** folder, expand the repository folder concerned.
4) In the **Reporting Datamart** folder, right-click the Reporting Datamart concerned and select **Synchronize > Incremental update**.

To define the scheduled trigger for an incremental synchronization:

1) Launch **MEGA Administration** application.
2) Connect to the environment concerned.
3) In the **Repositories** folder, expand the repository folder concerned.
4) Right-click **Scheduler** and select **Manage triggers.**
5) (If needed) In the **System Triggers** tab, right-click **Reporting Datamart Synchronization** and select **Update Scheduling** to modify the scheduling according to your needs.
6) In the **System Triggers** tab, right-click **Reporting Datamart Synchronization** and select **Activate**.

   The **Reporting Datamart Synchronization** is scheduled as defined (by default every 10').

4.3. Launching a calculated MetaAttribute synchronization

Launch a calculated MetaAttribute synchronization to scan all the objects and links of the HOPEX repository that can have calculated MetaAttribute values and put their values in the **Reporting Datamart**.

Do not launch or schedule a calculated MetaAttribute synchronization if you do not use calculated MetaAttributes.

To exclude specific MetaAtribute values from this synchronization, see **Error! Reference source not found.**.

You can:

- manually launch a calculated MetaAttribute synchronization
- define a scheduled trigger for the diagram synchronization

  Trigger: Reporting Datamart Synchronization (Calculated MetaAttribute)

  Frequency: The synchronization may take time according to the HOPEX Repository calculated MetaAttribute volume. You should first manually launch a calculated MetaAttribute synchronization once to check the synchronization duration, so as to determine the synchronization scheduling frequency.

When you schedule a "Reporting Datamart Synchronization (Calculated MetaAttribute)" the trigger is launched on all the repositories and for all their Reporting Datamarts.

To trigger the action on a specific repository for a specific Reporting Datamart, see HOPEX Power Studio > All about starting with APIs > Initializing and synchronizing a Reporting Datamart.

For information regarding the scheduling, see *HOPEX Power Studio > Using the Scheduler > Scheduler*.

To manually launch a calculated MetaAttribute synchronization:

1) Launch **MEGA Administration** application.
2) Connect to the environment concerned.
3) In the **Repositories** folder, expand the repository folder concerned.
4) In the **Reporting Datamart** folder, right-click the Reporting Datamart concerned and select **Synchronize > Attributes calculated**.

To define the scheduled trigger for a calculated MetaAttribute synchronization:

1) Launch **MEGA Administration** application.
2) Connect to the environment concerned.
3) In the **Repositories** folder, right-click **Scheduler** and select **Manage triggers.**
4) In the **System Triggers** tab, right-click **Reporting Datamart Synchronization (Calculated MetaAttribute)** and select **Execute**

   Check the synchronization duration.
5) In the **System Triggers** tab, right-click **Reporting Datamart Synchronization (Calculated MetaAttribute)** and select **Update Scheduling**.
6) Modify the scheduling according to your needs.
7) Click **OK**.
8) In the **System Triggers** tab, right-click **Reporting Datamart Synchronization (Calculated MetaAttribute)** and select **Activate**.

   The **Reporting Datamart Synchronization (Calculated MetaAttribute)** is scheduled as defined.

4.4. Launching a diagram synchronization

Launch a diagram synchronization to scan all the HOPEX repository diagrams and update the **Reporting Datamart** with their drawing representation in the SVG format.

Do not launch or schedule a diagram synchronization if you do not use diagrams.

You can:

- manually launch a diagram synchronization
- define a scheduled trigger for the diagram synchronization

  Trigger: Reporting Datamart Synchronization (Diagrams)

  Frequency: The synchronization may take time according to the HOPEX Repository diagram volume. You should first manually launch a diagram synchronization once to check the synchronization duration, so as to determine the synchronization scheduling frequency.

When you schedule a "Reporting Datamart Synchronization (Diagrams)" the trigger is launched on all the repositories and for all their Reporting Datamarts.

To trigger the action on a specific repository for a specific Reporting Datamart, see HOPEX Power Studio > All about starting with APIs > Initializing and synchronizing a Reporting Datamart.

For information regarding the scheduling, see *HOPEX Power Studio > Using the Scheduler > Scheduler*.

To manually launch a diagram synchronization:

1) Launch **MEGA Administration** application.
2) Connect to the environment concerned.
3) In the **Repositories** folder, expand the repository folder concerned.
4) In the **Reporting Datamart** folder, right-click the Reporting Datamart concerned and select **Synchronize > Diagrams**.

To define the scheduled trigger for a diagram synchronization:

1) Launch **MEGA Administration** application.
2) Connect to the environment concerned.
3) In the **Repositories** folder, right-click **Scheduler** and select **Manage triggers.**
4) In the **System Triggers** tab, right-click **Reporting Datamart Synchronization (Diagrams)** and select **Execute.**

   Check the synchronization duration.
5) In the **System Triggers** tab, right-click **Reporting Datamart Synchronization (Diagrams)** and select **Update Scheduling**.
6) Modify the scheduling according to your needs.
7) Click **OK**.
8) In the **System Triggers** tab, right-click **Reporting Datamart Synchronization (Diagrams)** and select **Activate**.

   The **Reporting Datamart Synchronization (Diagrams)** is scheduled as defined.

5. DELETING A REPORTING DATAMART

You can delete a **Reporting Datamart**.

To delete a Reporting Datamart:

1) Launch **MEGA Administration** application.
2) Connect to the environment concerned.
3) In the **Repositories** folder, expand the repository folder concerned.
4) In the **Reporting Datamart** folder, right-click the reporting Datamart concerned and select **Delete**.

6. USING THE REPORTING DATAMART

6.1. Usage

The **Reporting Datamart** can be used as:

- a data source to generate reports, through queries on a person data according to a specific profile.
- a repository to build custom static web sites with external CMS tools
- a repository to build a custom application, or an outbound interface to another tool
- a repository for mobile applications.

6.2. Advantages

Advantages of using the **Reporting Datamart** are that:

- data amount is reduced to comply with the creation parameters (user and profile rights) (see Content section)
- data access is simplified
- it contains up-to-date information (see Synchronizing the Reporting Datamart with the HOPEX Repository content section)
- it may run on a separate server
- it does not affect HOPEX repository performance while using it

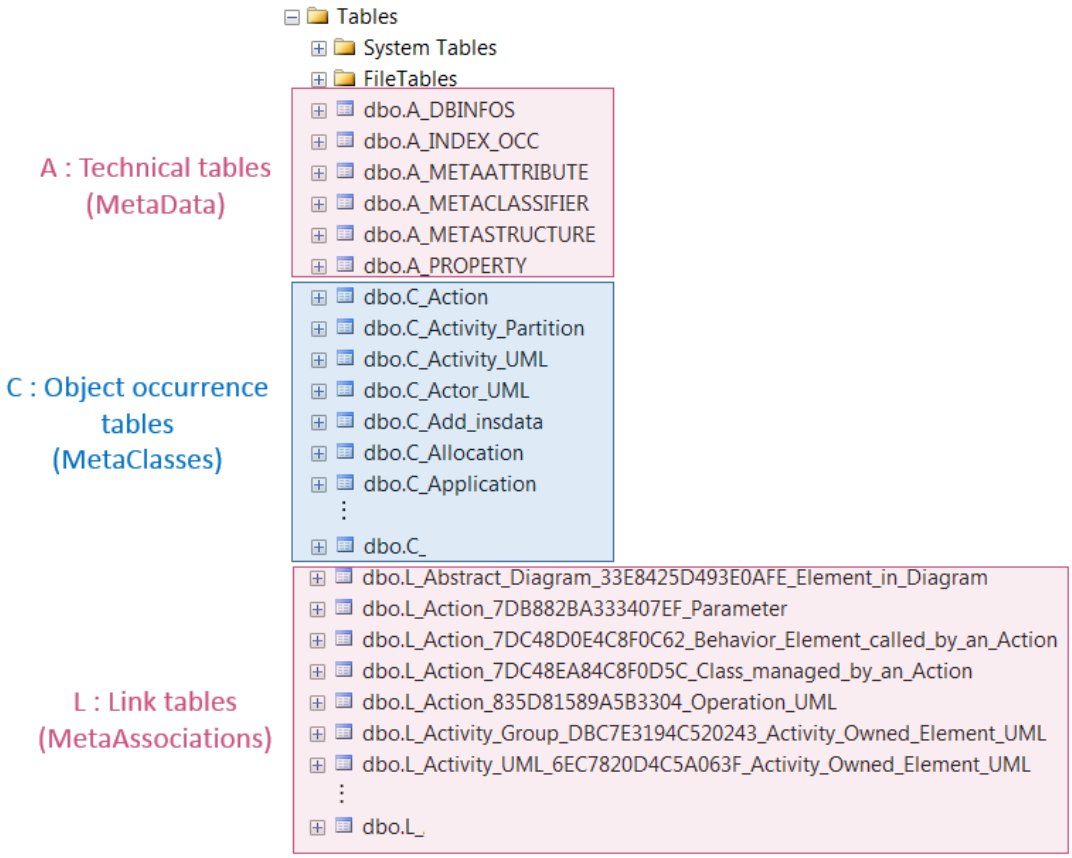
  However, see Launching a diagram synchronization and Launching a calculated MetaAttribute synchronization sections for HOPEX repository performance issues.

7. REPORTING DATAMART DETAILED DESCRIPTION

7.1. Reporting Datamart table classification

In the **Tables** folder, tables are classified according to their content:

- Technical data (MetaData)
- Object occurrences (MetaClass occurrences)
- Links (MetaAssociations)



7.2. Reporting Datamart table name format

Each table name format is as follows:

dbo.<letter>_<table name>

dbo: data base owner

<letter>: **A** for Technical tables

C for object occurrence tables

L for link tables

<table name>: describes the table content

7.2.1. Technical tables

Technical table name is prefixed with: **dbo.A_**.

- dbo.A_DBINFOS
- dbo.A_INDEX_OCC
- dbo.A_METAATTRIBUTE
- dbo.A_METACLASSIFIER
- dbo.A_METASTRUCTURE

- dbo.A_**DBINFOS** table is for internal use only (source repository information).
- dbo.A_**INDEX_OCC** table details all the object IdAbs with their corresponding MetaStructure Idabs. It enables to find out to which table belongs an object.

  - dbo.A_INDEX_OCC
    - Columns
      - IDMETASTRUCTURE (bigint, null)
      - IDABS (nvarchar(17), null)

  For an example see Finding information on an attribute section.

- dbo.A_**METACLASSIFIER** table describes the tables

  - dbo.A_METACLASSIFIER
    - Columns
      - META_IDABS (bigint, null)
      - META_HEXAIDABS (nvarchar(17), null)
      - UNIQUE_CODE (nvarchar(128), null)
      - DESCRIPTION (nvarchar(128), null)
      - META_NATURE (smallint, null)
      - TABLE_TYPE (smallint, null)
      - F_LEGIDABS (bigint, null)
      - S_LEGIDABS (bigint, null)

  - META_IDABS and META_HEXAIDABS are the Metamodel object (MetaClass or MetaAssociation) IdAbs in HOPEX, in numerical (integer) and hexadecimal formats.
  - UNIQUE_CODE is the table name
  - DESCRIPTION is the Metamodel object (MetaClass or MetaAssociation) name in HOPEX
  - F_LEGIDABS and S_LEGIDABS give the MetaAssociationEnd Idabs

  For examples see Finding the MetaClass, and Finding information on an attribute sections.

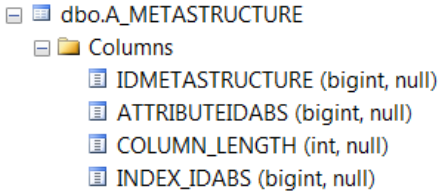
- dbo.A_**METATTRIBUTE** describes the columns (useful for Oracle databases)

  - dbo.A_METAATTRIBUTE
    - Columns
      - ATTRIBUTEIDABS (bigint, null)
      - ATTRIBUTEHEXAIDABS (nvarchar(17), null)
      - UNIQUE_CODE (nvarchar(128), null)
      - DESCRIPTION (nvarchar(128), null)
      - ATTRIBUTEFORMAT (nvarchar(2), null)
      - ATTRIBUTELENGTH (int, null)
      - ATTRIBUTEINDEX (smallint, null)
      - LCID (int, null)

  - ATTRIBUTEIDABS and ATTRIBUTEHEXAIDABS are the MetaAttribute IdAbs in HOPEX

- UNIQUE_CODE is the column name
- DESCRIPTION is the MetaAttribute name in HOPEX

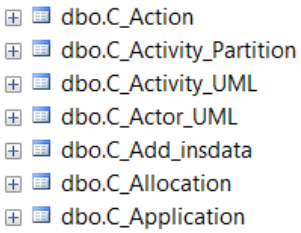
- dbo.A_**METASTRUCTURE** describes the Table x Column matrix



- IDMETASTRUCTURE gives the META_IdAbs of the dbo.A_METACLASSIFIER table
- ATTRIBUTEIDABS gives the MetaAttribute IdAbs in HOPEX.

7.2.2. MetaClass occurrence tables

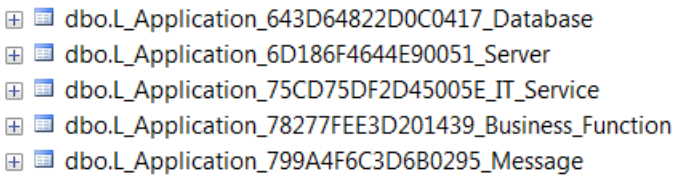
MetaClass occurrence table name is prefixed with: **dbo.C_**.



7.2.3. MetaAssociation tables

MetaAssociaton table name:

- is prefixed with: **dbo.L_**
- is made up of the MetaAssociation IdAbs between the major MetaAssociationEnd name (the first one) and the minor MetaAssociationEnd name (the second one)

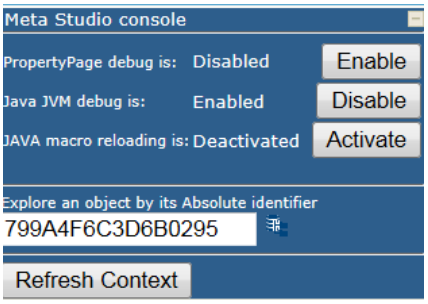


Example:

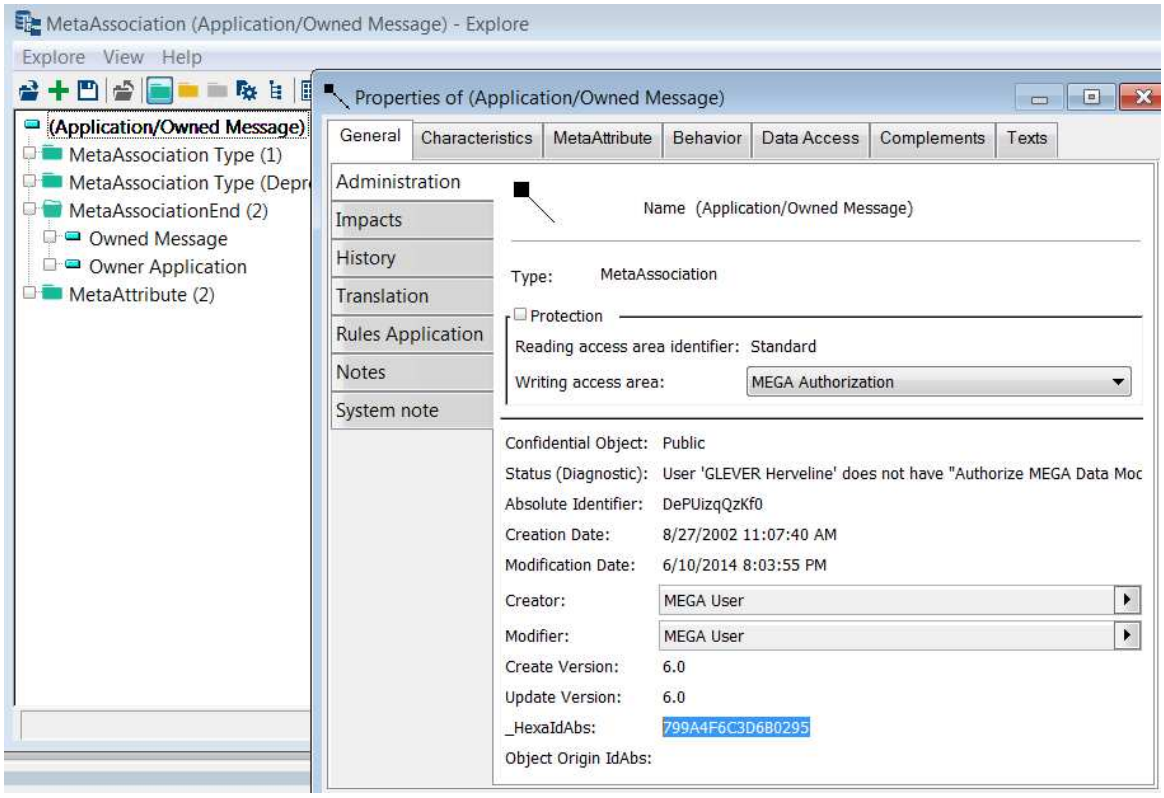
dbo.L_**Application**_799A4F6C3D6B0295_**Message**:

- dbo.L: indicates it is a link (MetaAssociation) table
- **Application**: the first MetaAssociationEnd (major)
- 799A4F6C3D6B0295: the “Application-Message” MetaAssociation IdAbs
- **Message**: the second MetaAssociationEnd (minor)

In HOPEX, with the MetaStudio console, in **Explore an object by its absolute identifier**, enter the 799A4F6C3D6B0295 IdAbs.



You get the MetaAssociation:



7.3. Reporting Datamart columns

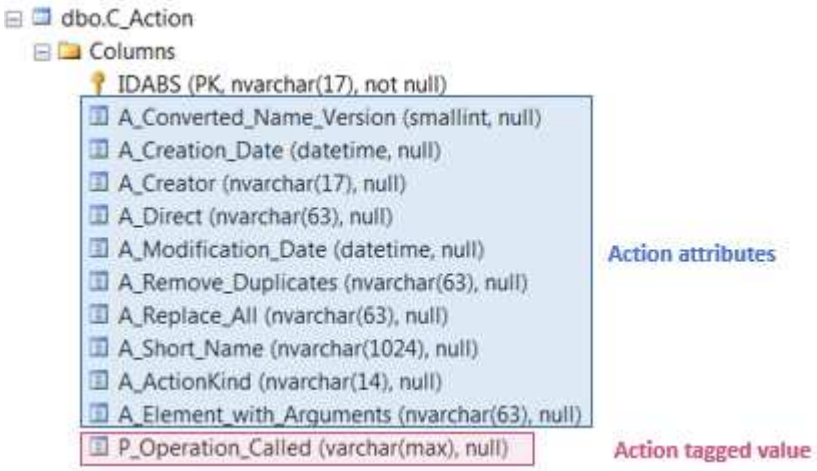
7.3.1. MetaClass occurrence table columns

In **dbo.C_**<MetaClass name>, the **Columns** folder details the attributes or tagged values belonging to the MetaClass occurrence concerned. Each column name is prefixed as follows:

- **A_** for attributes
- **P_** for tagged values

Example:

dbo.C_Action table includes in **Columns** folder all the attributes and tagged values belonging to the Actions.

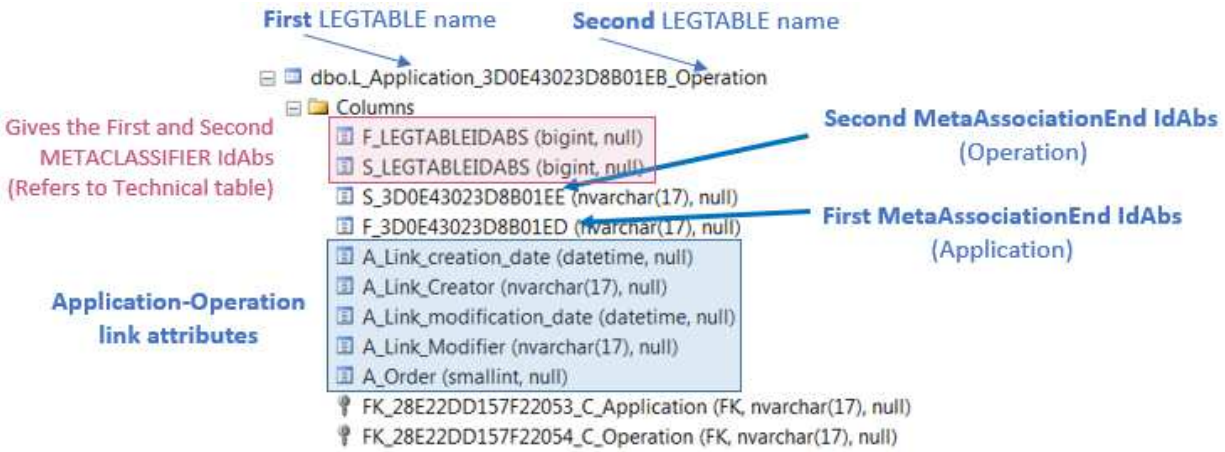


7.3.2. Link table columns

In **dbo.L_<…>**, the **Columns** folder details the attributes, and the first (major) and second (minor) MetaAssociationEnd IdAbs belonging to the MetaAssociation concerned. Each column name is prefixed as follows:

- F_LEGTABLEIDABS: indicates the first (major) table IdAbs
- S_LEGTABLEIDABS: indicates the second (minor) table IdAbs
- F_< IdAbs>: indicates the first (major) MetaAssociationEnd IdAbs
- S_< IdAbs>: indicates the second (minor) MetaAssociationEnd IdAbs
- A_<…> for attributes

Example:



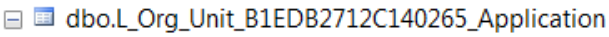
In dbo.L_**Application**_3D0E43023D8B01EB_**Operation** link table, the **Columns** folder is made up of:

- **Application** is the first MetaAssociationEnd (major)

  F_3D0E43023D8B01ED is the Application occurrence IdAbs
- **Operation** is the second MetaAssociationEnd (minor)

  S_3D0E43023D8B01EE is the Operation occurrence IdAbs

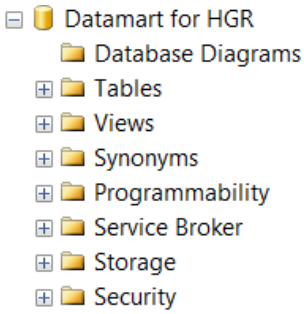
7.4. Use case: reading the Reporting Datamart through a link



7.4.1. Accessing your Reporting Datamart tables

To access your Reporting Datamart:

1) Launch Microsoft SQL Server Management Studio.
2) Access the server where you saved your reporting Datamart.
3) In the server tree view, expand **Databases** folder.
4) Expand your Reporting Datamart folder.



5) Expand the **Tables** folder.

7.4.2. Understanding a link table

Link example: Org_Unit - Application.

dbo.L_Org_Unit_B1EDB2712C140265_Application

Dbo.L: “Org_Unit - Application” link

Org_Unit: major MetaAssociationEnd

B1EDB2712C140265: MetaAssociation IdAbs

Application: minor MetaAssociationEnd

To understand the dbo.L_Org_Unit_B1EDB2712C140265_Application table information:

1) In your Reporting Datamart tree view, **Tables** folder, right-click **dbo_Org_Unit_B1EDB2712C140265_Application** table and select **Select Top 1000 rows**.

   The result is displayed in the right pane.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [F_LEGTABLEIDABS]
      ,[S_LEGTABLEIDABS]
      ,[S_B1EDB2722C14026C]
      ,[F_B1EDB2712C140268]
      ,[A_Link_creation_date]
      ,[A_Link_Creator]
      ,[A_Link_modification_date]
      ,[A_Link_Modifier]
      ,[A_Order]
      ,[FK_28E22CF157F20E97_C_Application]
      ,[FK_28E22CF157F20E98_C_Org_Unit]
  FROM [Datamart for HGR].[dbo].[L_Org_Unit_B1EDB2712C140265_Application]
```

100 %

Results | Messages

	F_LEGTABLEIDABS	S_LEGTABLEIDABS	S_B1EDB2722C14026C	F_B1EDB2712C140268	A_Link_creation_date	A_Link_Creator
1	104475683762647533	103349783855804909	B992C01A3B6E06B7	29E53A213B9607D6	2006-09-19 17:05:37.000	F73EF8C53BCE002E

- 104475683762647533 is the **Org Unit** MetaClass IdAbs (for the first MetaAssociationEnd).

  See Finding the MetaClass corresponding to an IdAbs section.
- 103349783855804909 is the **Application** MetaClass IdAbs (for the second MetaAssociationEnd).
- 29E53A213B9607D6 is the **Org Unit** (first MetaAssociationEnd occurrence) IdAbs.

  See Finding attributes or tagged values belonging to a MetaClass occurrence section.
- B992C01A3B6E06B7 is the **Application** (second MetaAssociationEnd occurrence) IdAbs.

7.4.3. Finding the MetaClass corresponding to an IdAbs

To find out the MetaClass corresponding to the 104475683762647533 IdAbs use the **dbo.A_METACLASSIFIER** table.

To find out the MetaClass corresponding to 104475683762647533 IdAbs:

1) In your Reporting Datamart tree view, right-click **dbo.A_METACLASSIFIER** and select **Select Top 1000 rows**.

2) In the right pane complete the query with : where META_IDABS=104475683762647533

```
FROM [Datamart for HGR].[dbo].[A_METACLASSIFIER] where
META_IDABS=104475683762647533
```

3) **Execute** the query.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [META_IDABS]
      ,[META_HEXAIDABS]
      ,[UNIQUE_CODE]
      ,[DESCRIPTION]
      ,[META_NATURE]
      ,[TABLE_TYPE]
      ,[F_LEGIDABS]
      ,[S_LEGIDABS]
  FROM [Datamart for HGR].[dbo].[A_METACLASSIFIER] where META_IDABS=104475683762647533
```

100 %

Results | Messages

	META_IDABS	META_HEXAIDABS	UNIQUE_CODE	DESCRIPTION	META_NATURE	TABLE_TYPE	F_LEGIDABS	S_LEGIDABS
1	104475683762647533	B1EDB2562C140173	C_Org_Unit	Org-Unit	1	2	NULL	NULL

The query result indicates that 104475683762647533 is the **Org-Unit** MetaClass IdAbs.

7.4.4. Finding attributes or tagged values belonging to a MetaClass occurrence

Use the **dbo.C_<MetaClass name>** table to find the attributes and tagged values belonging to a MetaClass occurrence.

For example 29E53A213B9607D6 is the IdAbs of an Org_Unit occurrence.

To find the Org_Unit table which IdAbs is 29E53A213B9607D6:

1) In your Reporting Datamart tree view, right-click **dbo.C_Org_Unit** and select **Select Top 1000 rows**.

2) In the right pane complete the query with: where IDABS='29E53A213B9607D6'

```
FROM [Datamart for HGR].[dbo].[C_Org_Unit] where IDABS='29E53A213B9607D6'
```

3) **Execute** the query.

```
      ,[A_Org_Unit_Type]
      ,[A_Short_Name]
      ,[A_Use_Default_Schedules]
      ,[A_Validation_State]
      ,[P_Described_by_a_diagram]
      ,[A_Internal_External]
      ,[A_E_mail]
  FROM [Datamart for HGR].[dbo].[C_Org_Unit] where IDABS='29E53A213B9607D6'
```

100 %

Results | Messages

	IDABS	A_Company_Standard	A_Converted_Name_Version	A_Creation_Date	A_Short_Name	A_Creator	A_Hourly_Rate	A_Immutability
1	29E53A213B9607D6	Unknown	29248	2001-09-05 14:43:45.000	Controler	801180463B58001C	0	Modifiable

The query result details all the Org-Unit attributes (IdAbs: 29E53A213B9607D6, Short Name: Controler). It indicates, in particular that its creator IdAbs is: 801180463B58001C.

7.4.5. Finding information on an attribute

You want information on the Creator attribute whose idabs is 801180463B58001C.

You need to:

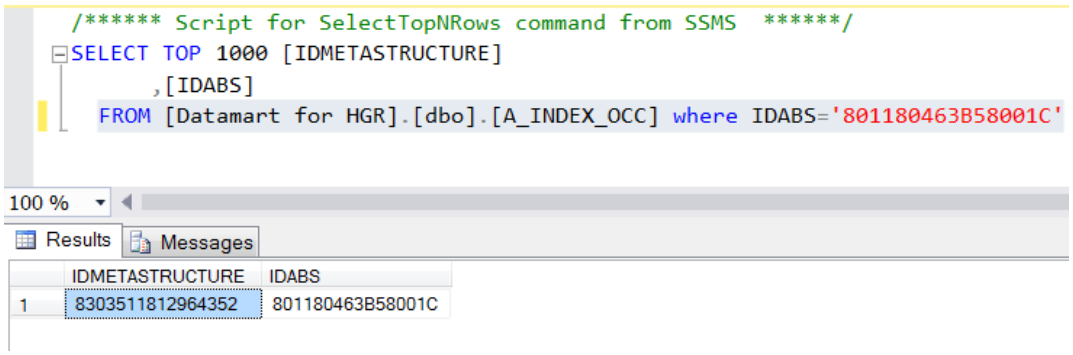
- use the **dbo.A_INDEX_OCC** table to get the MetaClass IdAbs to which the Creator belongs to.
- use the **dbo.A_METACLASSIFIER** table to get the MetaClass name from its Idabs
- use the **dbo.C_<MetaClass name>** table to get the Creator (801180463B58001C IdAbs) information

To find the MetaClass IdAbs of the MetaClass occurrence whose IdAbs is 801180463B58001C:

1) To find out to which MetaClass belongs the occurrence with 801180463B58001C IdAbs, in your Reporting Datamart tree view, right-click **dbo.A_INDEX_OCC** and select **Select Top 1000 rows**.
2) In the right pane complete the query with: `where IDABS='801180463B58001C'`

```
FROM [Datamart for HGR].[dbo].[A_INDEX_OCC] where IDABS='801180463B58001C'
```

3) **Execute** the query.



The query result indicates that the IDMETASTRUCTURE of the 801180463B58001C IdAbs is: 8303511812964352.

4) To find out which MetaClass has the 8303511812964352 IdAbs, in your Reporting Datamart tree view, right-click **dbo.A_METACLASSIFIER** and select **Select Top 1000 rows**.
5) In the right pane complete the query with: `where META_IDABS=8303511812964352`

```
FROM [Datamart for HGR].[dbo].[A_METACLASSIFIER] where
META_IDABS=8303511812964352
```

6) **Execute** the query.

```
        ,[DESCRIPTION]
        ,[META_NATURE]
        ,[TABLE_TYPE]
        ,[F_LEGIDABS]
        ,[S_LEGIDABS]
    FROM [Datamart for HGR].[dbo].[A_METACLASSIFIER] where META_IDABS=8303511812964352
```

100 %

Results | Messages

	META_IDABS	META_HEXAIDABS	UNIQUE_CODE	DESCRIPTION	META_NATURE	TABLE_TYPE	F_LEGIDABS	S_LEGID
1	8303511812964352	000000008000001D	C_Person_System	Person	1	2	NULL	NULL

The query result indicates that 8303511812964352 IdAbs is the **Person System** MetaClass IdAbs.

7) To find out which Person has the 801180463B58001C IdAbs, in your Reporting Datamart tree view, right-click **dbo.C_Person_System** and select **Select Top 1000 rows**.

8) In the right pane complete the query with: `where IDABS='801180463B58001C'`

```
FROM [Datamart for HGR].[dbo].[C_Person_System] where IDABS='801180463B58001C'
```

```
        ,[A_Initials]
        ,[A_ParameterizationOfAlertsScheduling]
        ,[A_Report_Edition_Parameterization]
        ,[A_Data_Language]
        ,[A_Comment]
    FROM [Datamart for HGR].[dbo].[C_Person_System] where IDABS='801180463B58001C'
```

100 %

Results | Messages

	IDABS	A_Creation_Date	A_Creator	A_Modification_Date	A_Short_Name	A_Number_of_Exe
1	801180463B58001C	2016-10-02 23:25:55.000	0000000022222222	2012-04-19 19:13:51.000	Patrick	0

The query result indicates that the 801180463B58001C IdAbs belongs to Patrick.

7.5. Use case: saving the diagram drawings

In **dbo.C_Diagrams** table, the diagrams drawings are in the **A_Drawing** column.

- dbo.C_Diagram
  - Columns
    - IDABS (PK, nvarchar(17), not null)
    - A_Converted_Name_Version (smallint, null)
    - A_Creation_Date (datetime, null)
    - A_Creator (nvarchar(17), null)
    - A_Modification_Date (datetime, null)
    - A_Short_Name (nvarchar(1024), null)
    - A_idRel (nvarchar(128), null)
    - A_DrawingTextFormat (varchar(max), null)
    - A_Drawing (varbinary(max), null)

To save the diagram drawings:

1) Launch Microsoft SQL Server Management Studio.
2) Access the server where you saved your reporting Datamart.
3) Check in your configuration that the following features are unlock (= 1):

```
/*
  CONFIGURATION
  =============
*/
 ---- check the configuration for 'Ole Automation Procedures'. It must be set
to 1
EXEC sp_configure 'Ole Automation Procedures';
GO
 ---- setting 'Ole Automation Procedures' to 1
EXEC sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO
EXEC sp_configure 'Ole Automation Procedures', 1;
GO
RECONFIGURE;
 /*
  EXECUTION
  =========
*/
```

4) Create a folder where you want to retrieve the diagram drawings.

   For example: C:\Pictures.

5) Enter the following code:

```
DECLARE @SQLIMG VARCHAR(MAX),
                @IDABS NVARCHAR(17),
                @NAME NVARCHAR(1024),
                @IMG_PATH VARBINARY(MAX),
                @FILENAME VARCHAR(MAX),
                @ObjectToken INT

DECLARE IMGPATH CURSOR FAST_FORWARD FOR
    SELECT [IDABS], [A_Short_Name], [A_Drawing] from [dbo].[C_Diagram]
OPEN IMGPATH
```

```
FETCH NEXT FROM IMGPATH INTO @IDABS, @NAME, @IMG_PATH

WHILE @@FETCH_STATUS = 0
  BEGIN
    SET @FILENAME = 'C:\Pictures\'+ @IDABS + '_' + @NAME + '.png'

    EXEC sp_OACreate 'ADODB.Stream', @ObjectToken OUTPUT
    EXEC sp_OASetProperty @ObjectToken, 'Type', 1
    EXEC sp_OAMethod @ObjectToken, 'Open'
    EXEC sp_OAMethod @ObjectToken, 'Write', NULL, @IMG_PATH
    EXEC sp_OAMethod @ObjectToken, 'SaveToFile', NULL, @FILENAME, 2
    EXEC sp_OAMethod @ObjectToken, 'Close'
    EXEC sp_OADestroy @ObjectToken

    FETCH NEXT FROM IMGPATH INTO @IDABS, @NAME, @IMG_PATH
 END

CLOSE IMGPATH
DEALLOCATE IMGPATH
```

RDBMS Environment Duplication (EN)

RDBMS Environment Duplication - 1 - SQL Server

The goal of this document is to give detailed instructions as to how duplicating a MEGA environment when it is hosted on an **SQL Server RDBMS**.

We will take the hypothesis that the new databases of the duplicated environment are hosted on a separate SQL Server instance. This way, it gives the appropriate details in case you move data from one server to another.

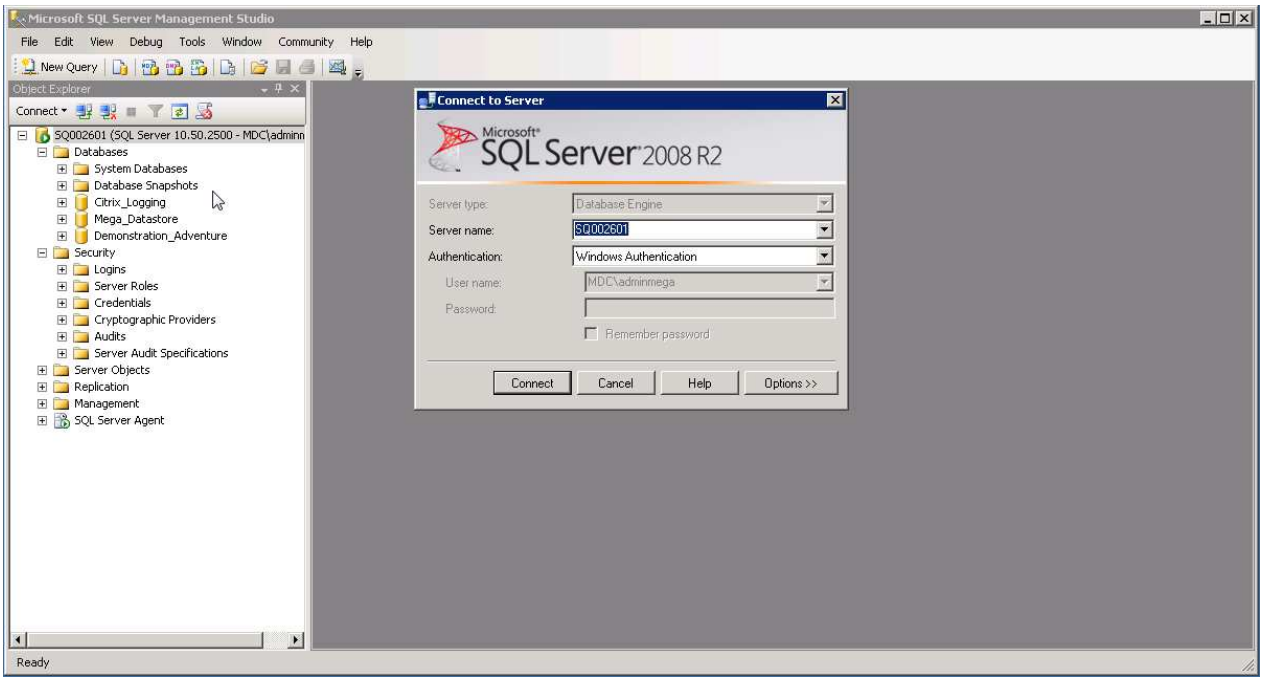
PREREQUISITES

The SQL Server account

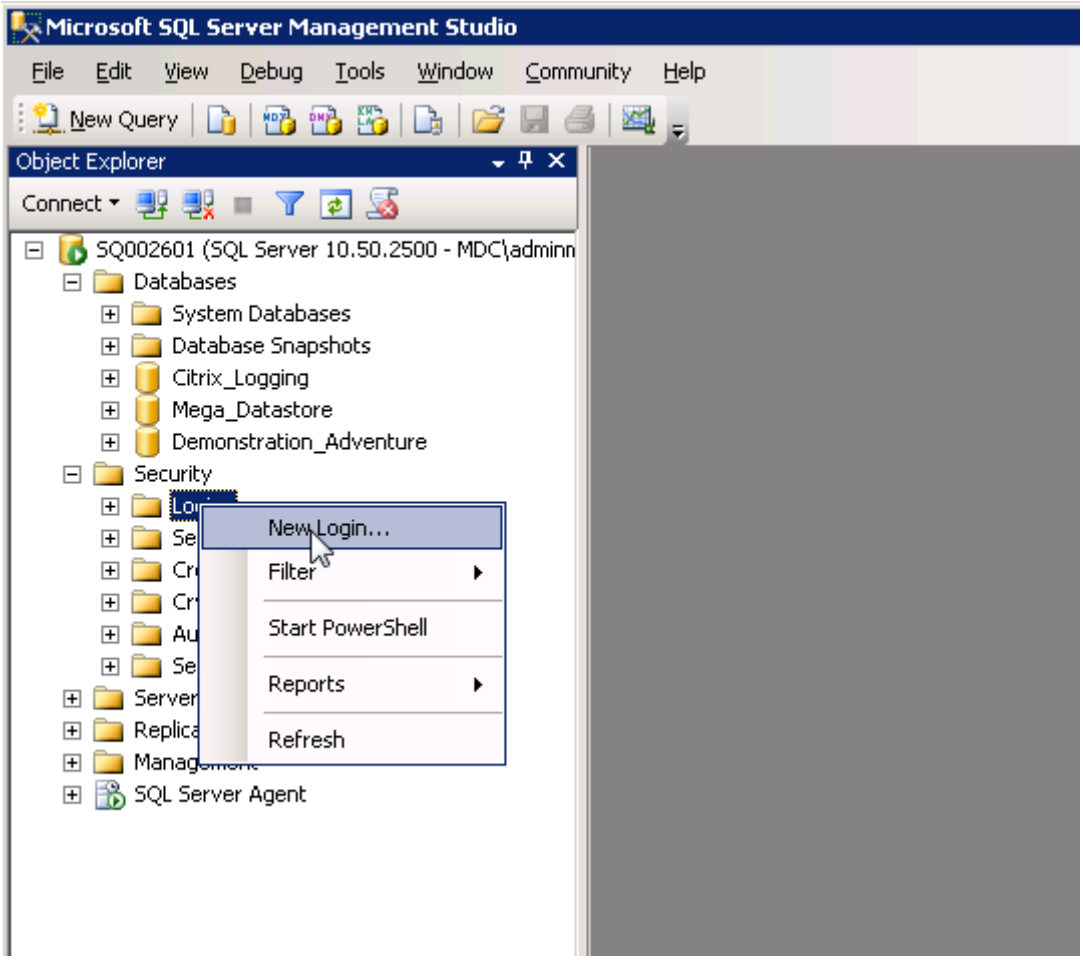
Before you start, you have to make sure that, where the duplicated databases are going to be hosted, you have a user with the proper rights, either:

- you are given the sysadmin right (the super admin) on the instance, and you perform those actions yourself, or
- you have to ask the DBA to create it and grant it.

1. Launch the "Microsoft SQL Server Management Studio" tool.
2. Connect to the SQL Server instance with the user having the sysadmin right.
3. If your Windows account is the one having the sysadmin right, stay in « Windows Authentication » mode. Otherwise, switch to "SQL Server Authentication", and provide the username and password. The field "Server Name" is the name of the instance.



4. In the left pane, expand **Security** file.
5. Right-click **Logins** and select **New Login**.

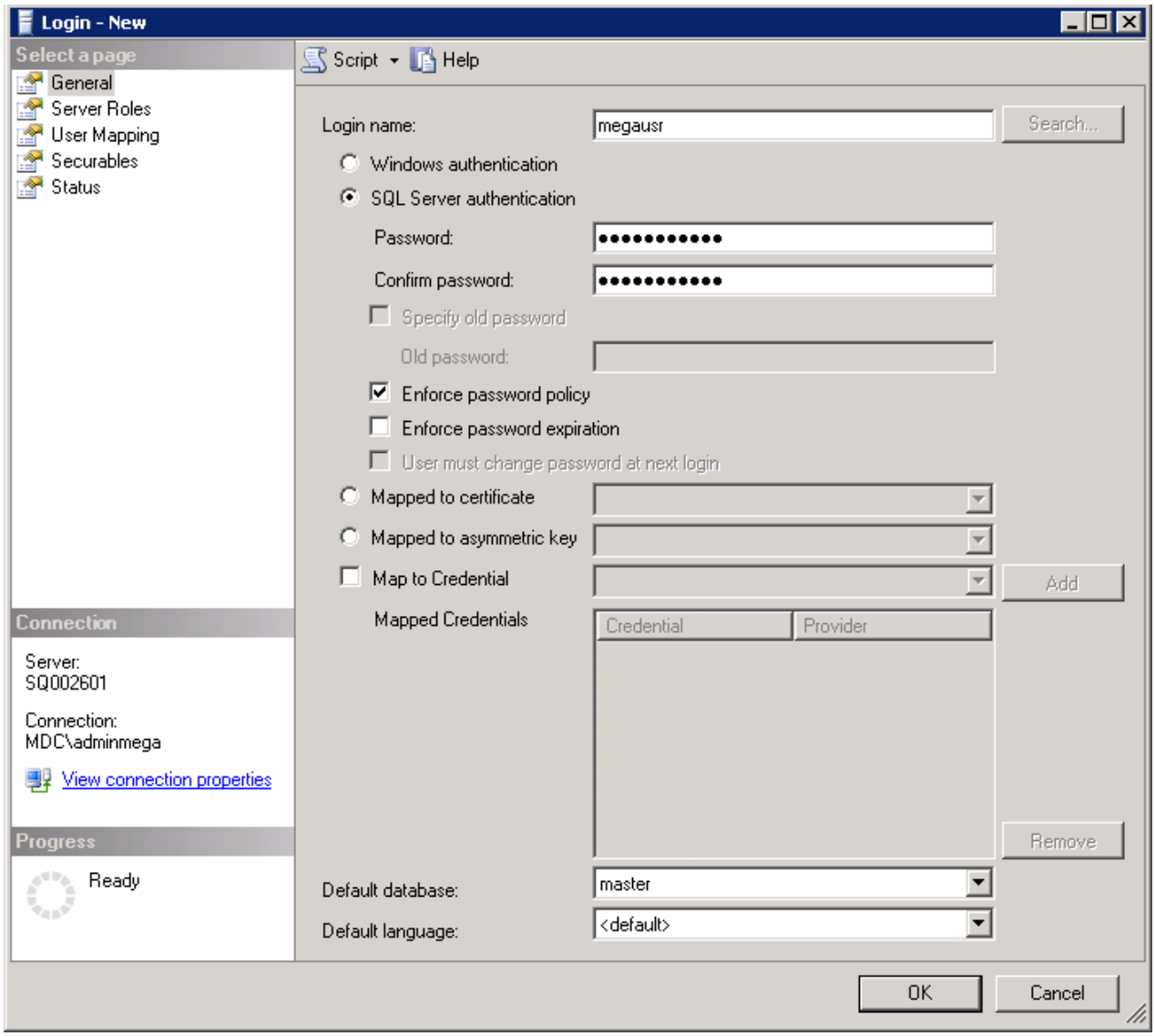


The **Login - New** window appears and displays and the **General** page.

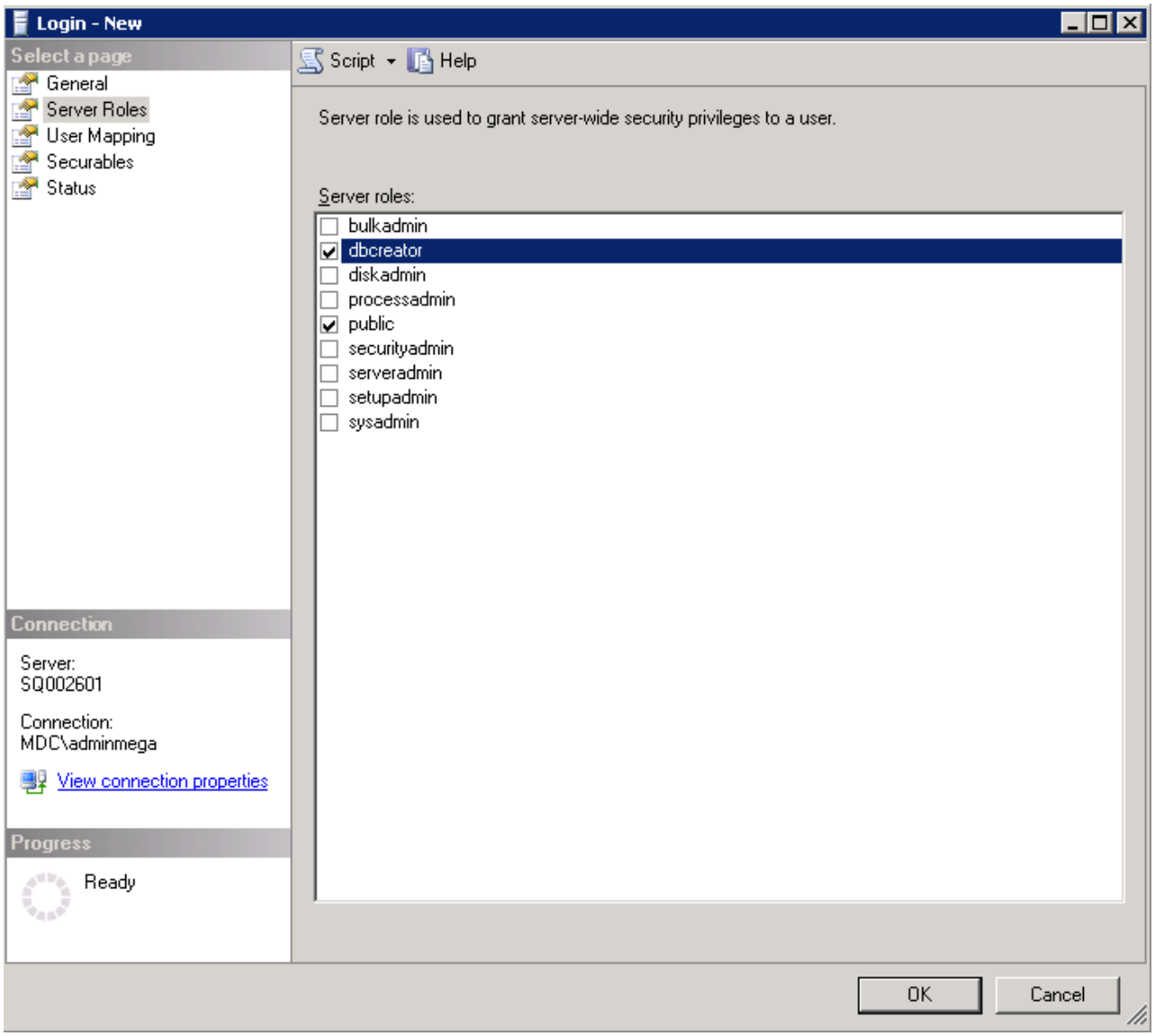
6. In the right pane:
   a. In the field **Login name** enter the login, for exemaple « megausr ».
   b. Select **SQL Server authentication** mode.
   c. In the field **Password**, enter a password that complies with security requirements, for example :

      `Mega2k8!usr`

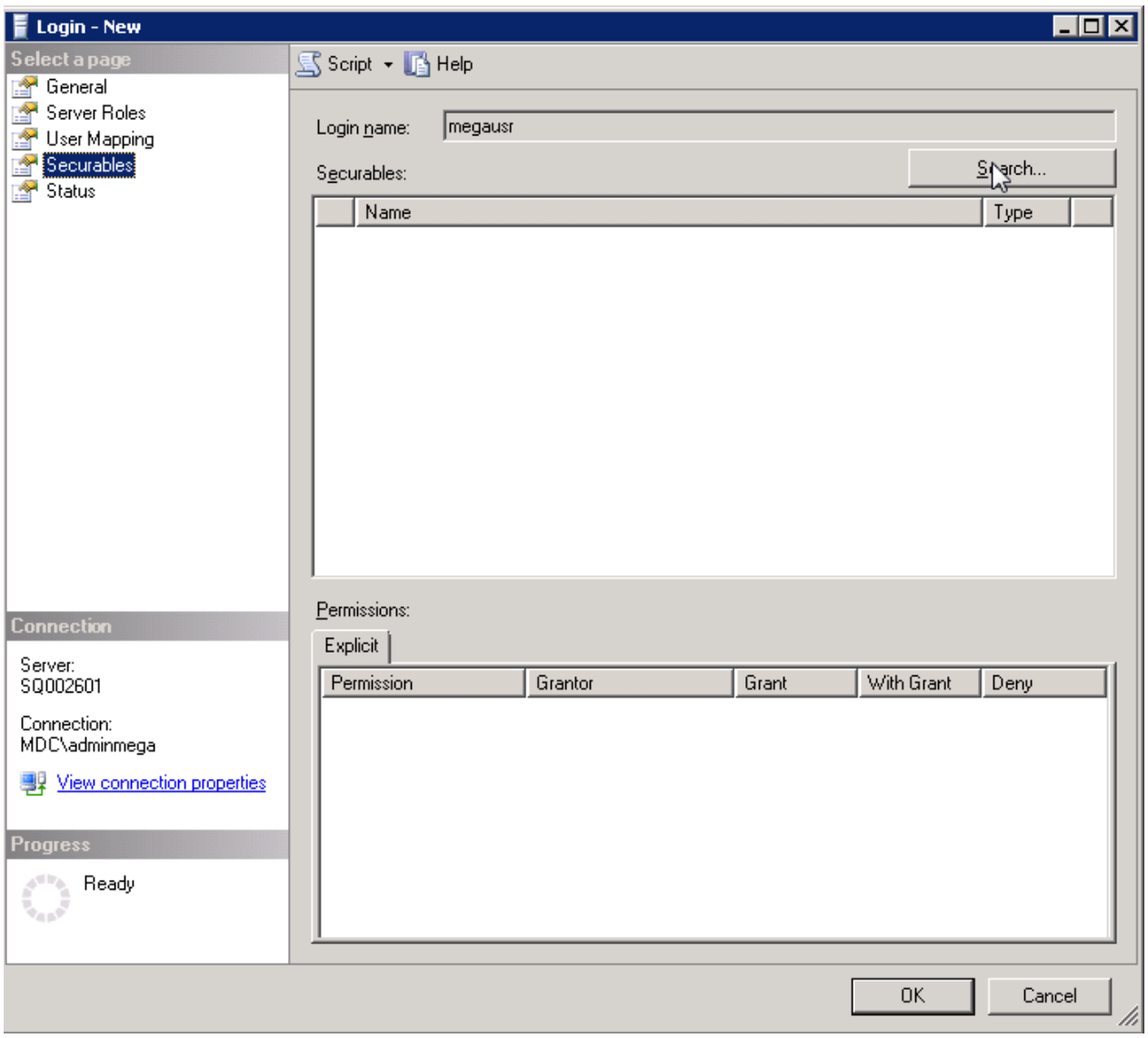
7. Unselect **Enforce password expiration** and User must change password at next login.



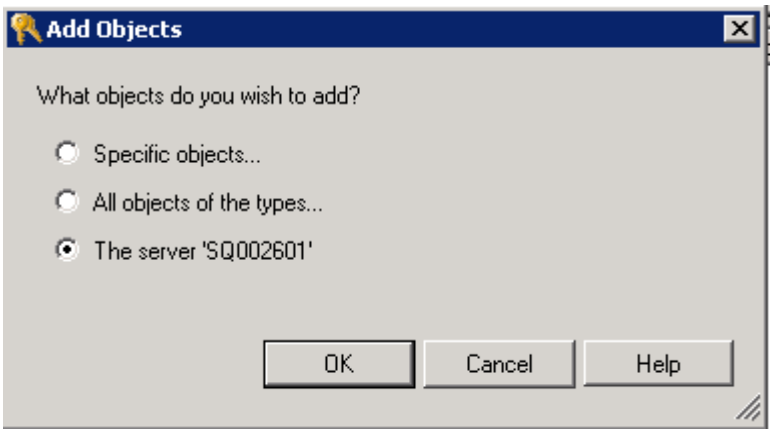
8. In the **Select a page** pane, select **Server Roles**.
9. In the right pane select **dbcreator**.



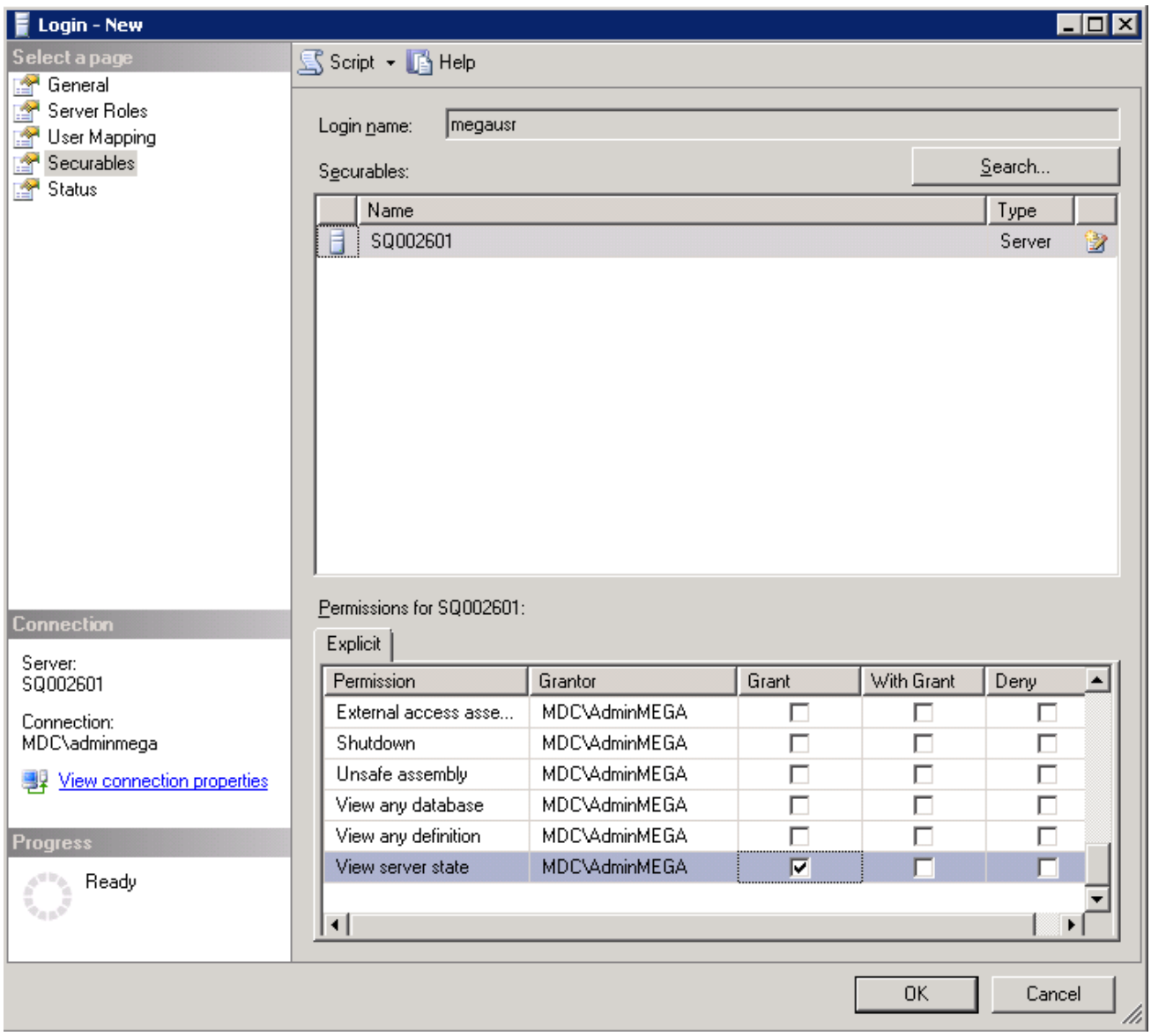
10. In the **Select a page** pane, select **Securables**.
11. In the right pane, click **Search**.



12. Select the instance to browser for its objects. Here we were on instance SQL002601 :



13. Click **OK**.
14. In the **Explicit** tab, for the **View Server state** permission select « Grant ».



15. Click **OK** to create the user.

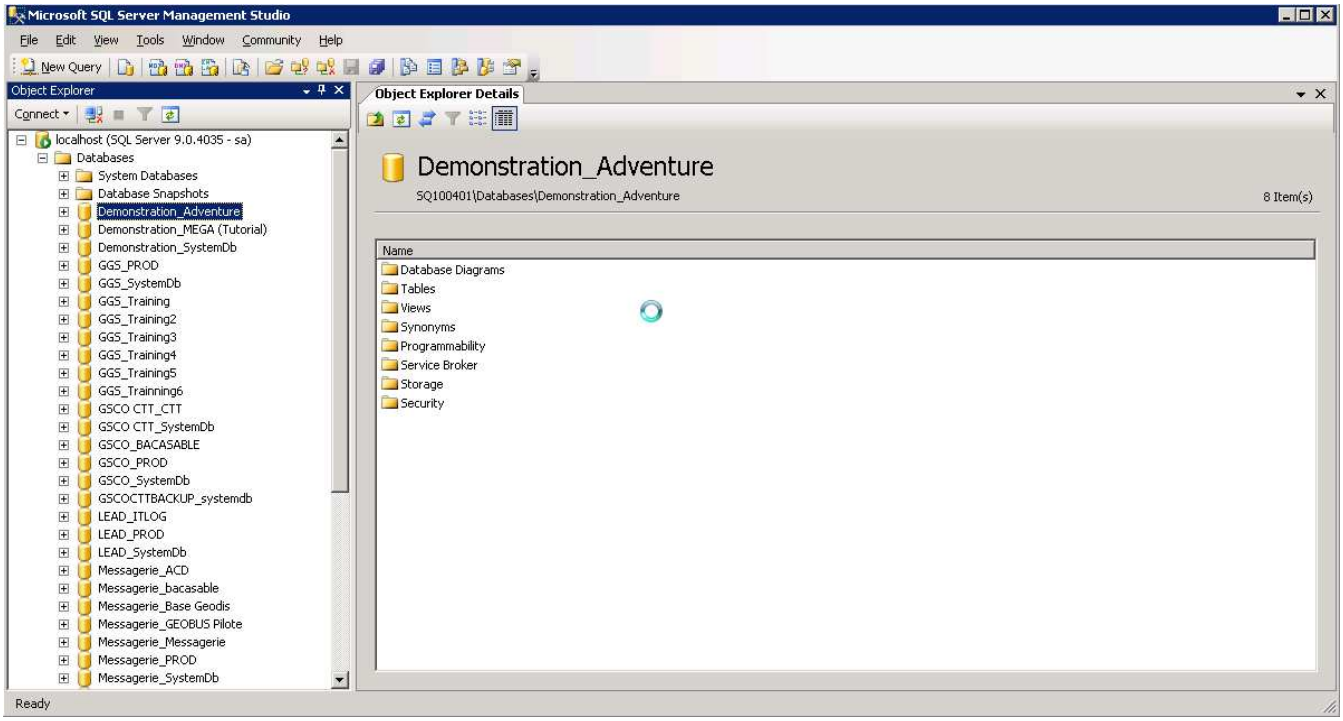
Backup and file transfer

1. Connect to the server hosting the source database.

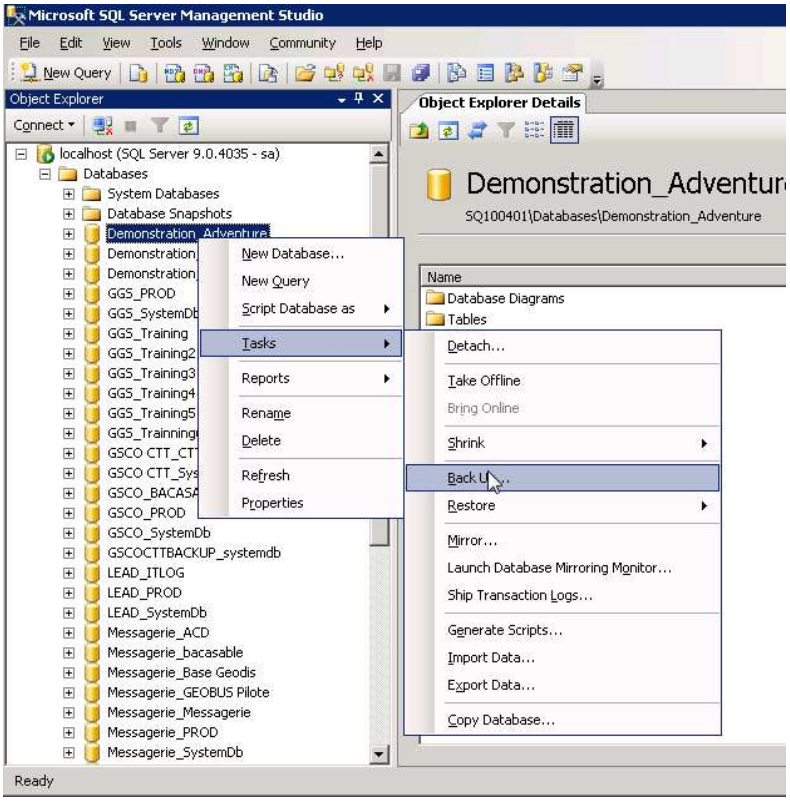
   Example: SQ100401.

2. Launch **Microsoft SQL Server Management Studio**.

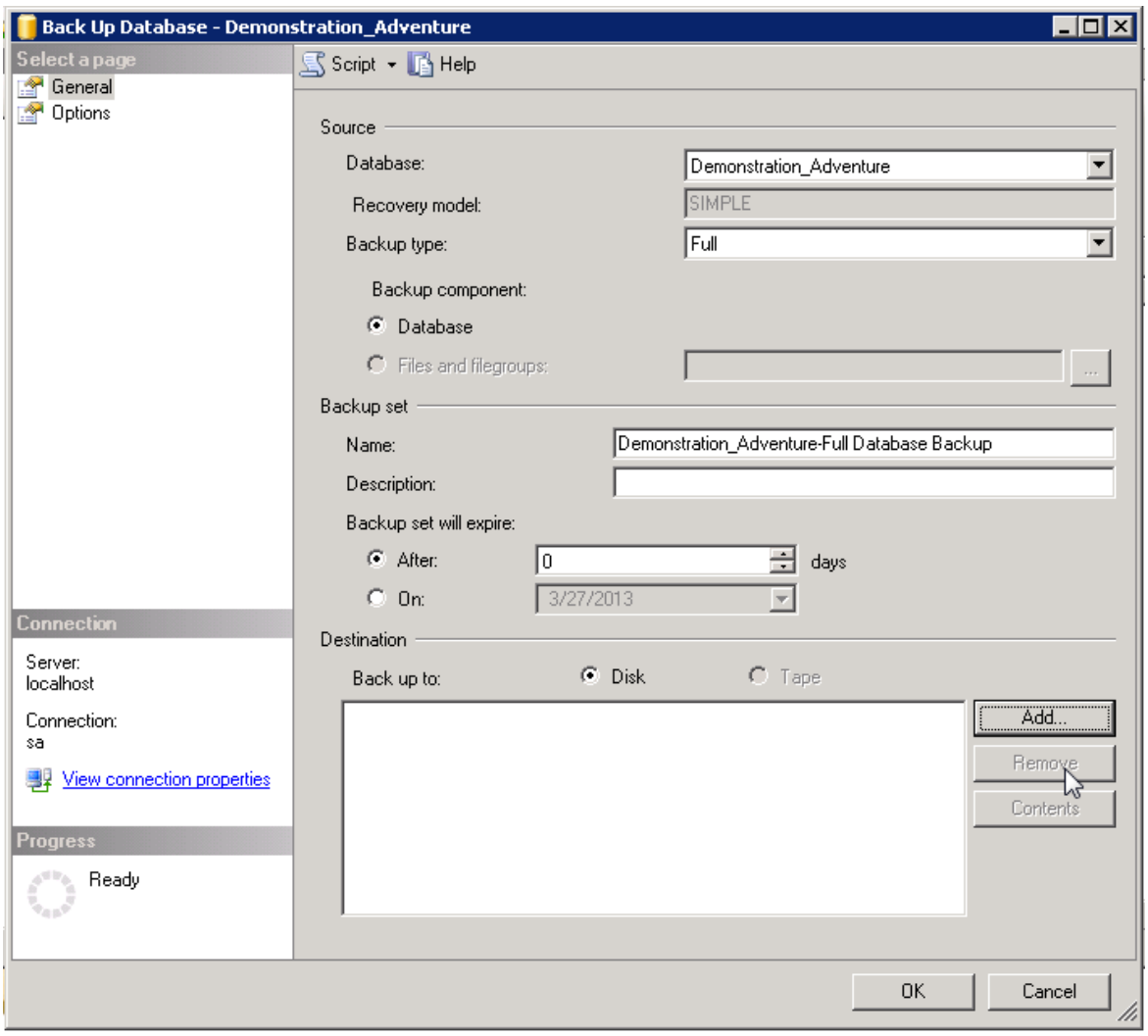
   If possible, connect with a Windows user that will have been granted sysadmin rights on the instance. Otherwise, you might encounter issues when creating the backup file to a specific location, or later, when you need to restore the database on the target instance.



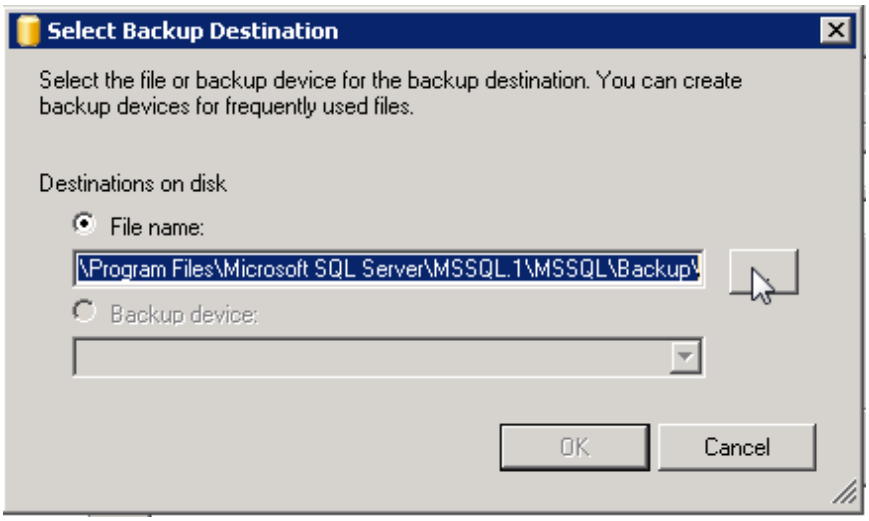
3. To make a backup of the source database (for example, Demonstration_Adventure), right-click the database and select **Tasks > Back Up**.



The **Back Up Database** window appears.

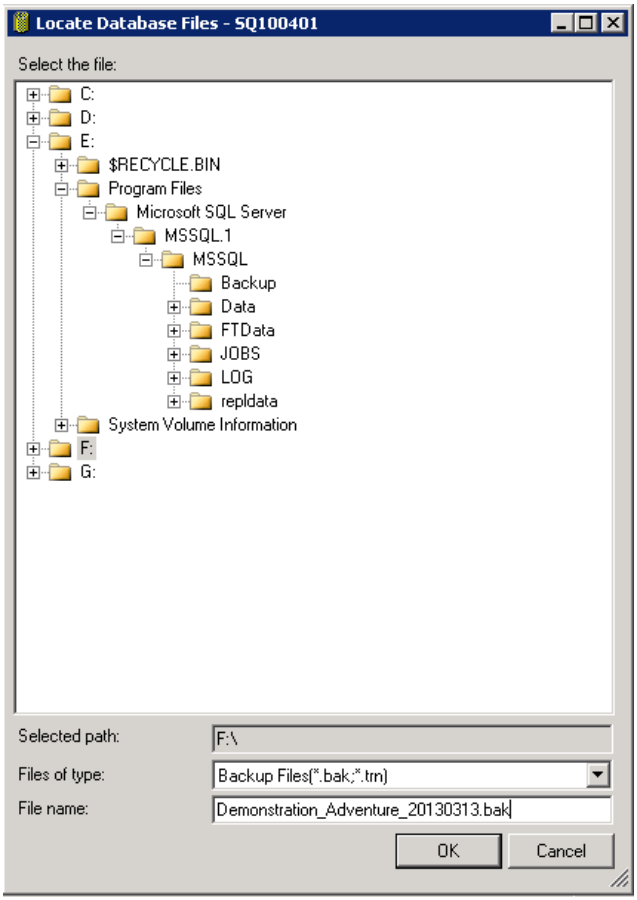


4. In the right pane, in the **Destination** pane, make sure that the destination list is empty. If it isn't, select each line and click **Remove**. Once it is empty, click **Add**.
5. In the **Select BackUp Destination**, click ...

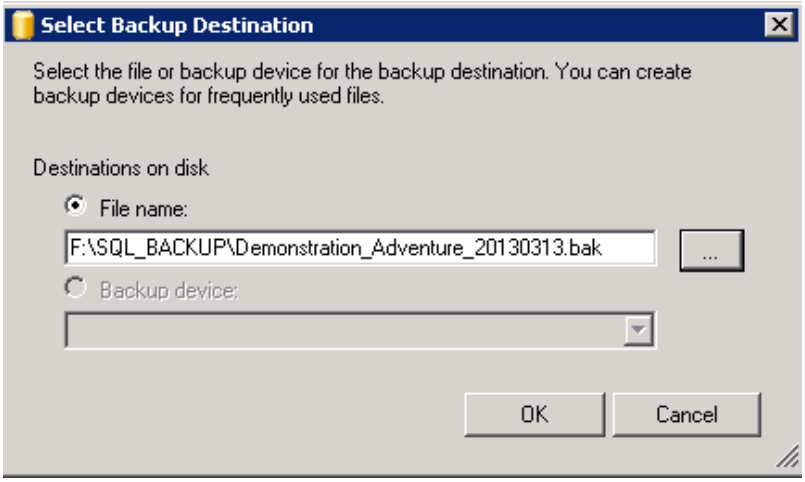


6. Choose a location where you know that the user you authenticated with, has rights to write on. In this example in « F:\SQL_BACKUP » of the F drive, and give the name of the backup file to create (here Demonstration_Adventure_20130313.bak).

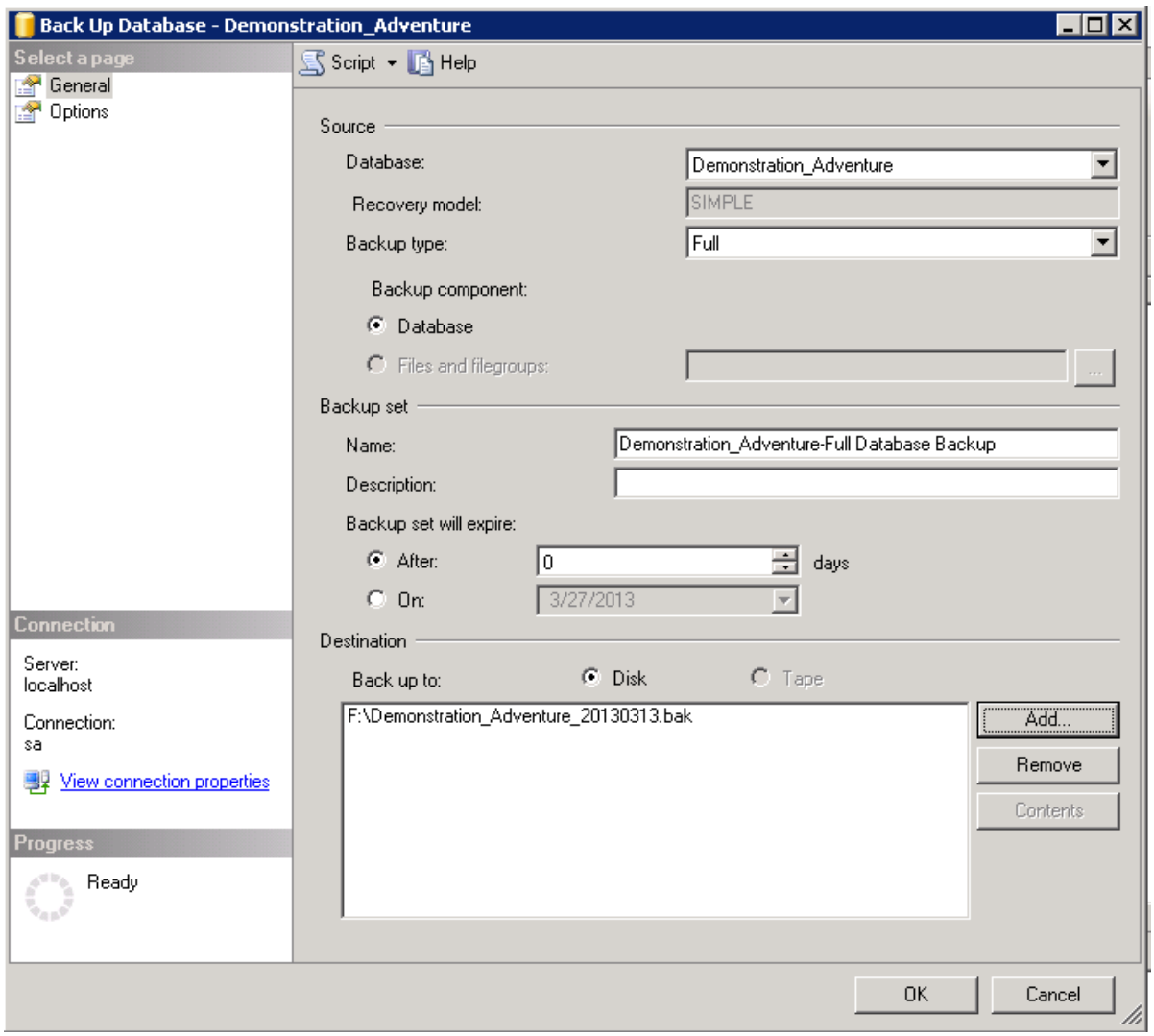
Please note that the known format of a full backup in SQL Server is **.BAK** files. You have to explicitely put it in the file name, otherwise it will not have any format.



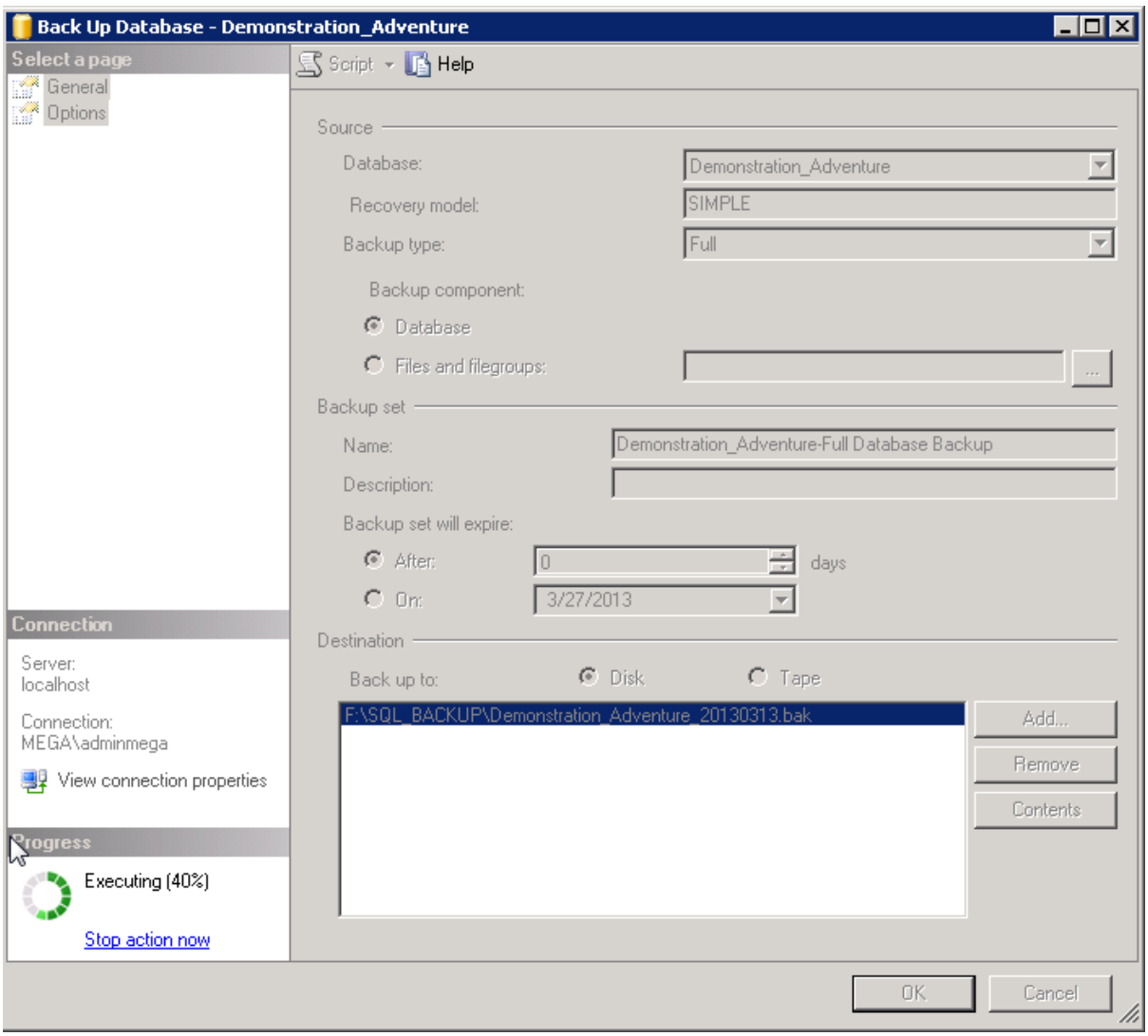
7. Click **OK**.



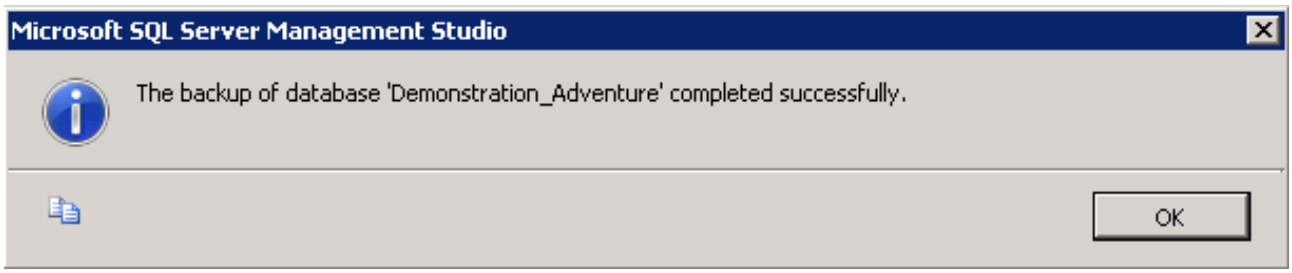
8. Click **OK**.



9. Click **OK** and check the progress of the restore by looking at the left-bottom section of this window (here, we are 40% done with the restore).



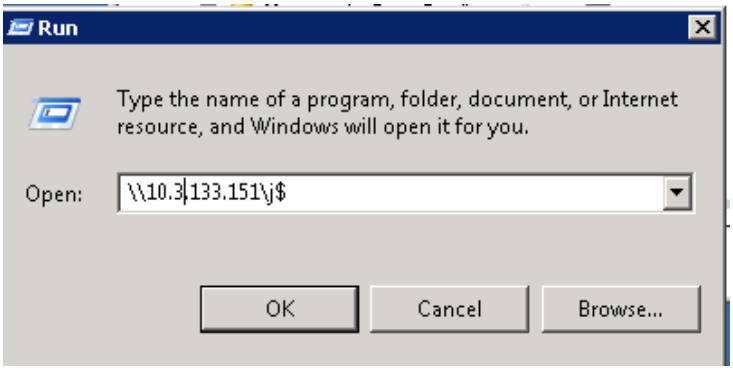
10. Check that the database was fully restored, and click **OK**.



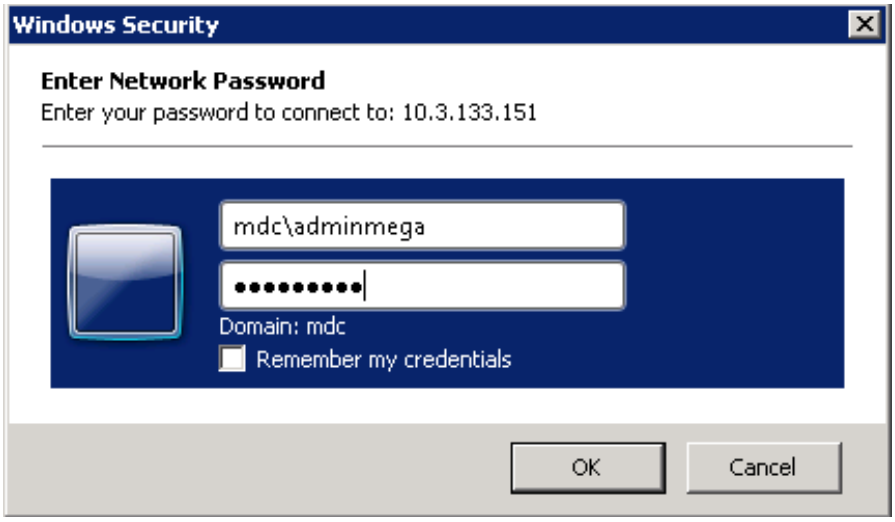
11. Transfer the backup file created on the SQL Server server, to the target server (for example, here it is SQL002601).

In this example, the drive hosting the databases on the target instance, as well as the daily backups, is the J drive.

We used its IP address instead of its name, as we were working over two different domains, that did not see each other.

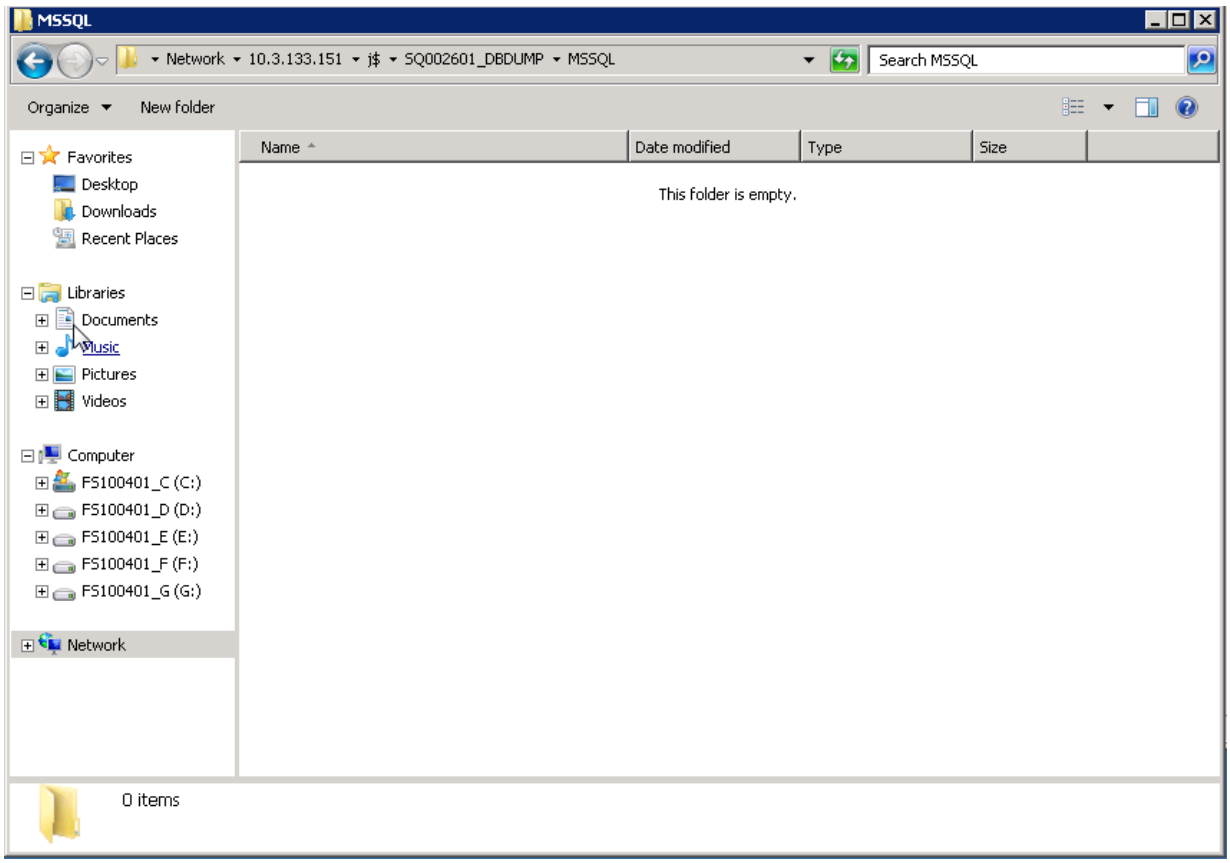


This is also the reason why we had to authenticate with another user.

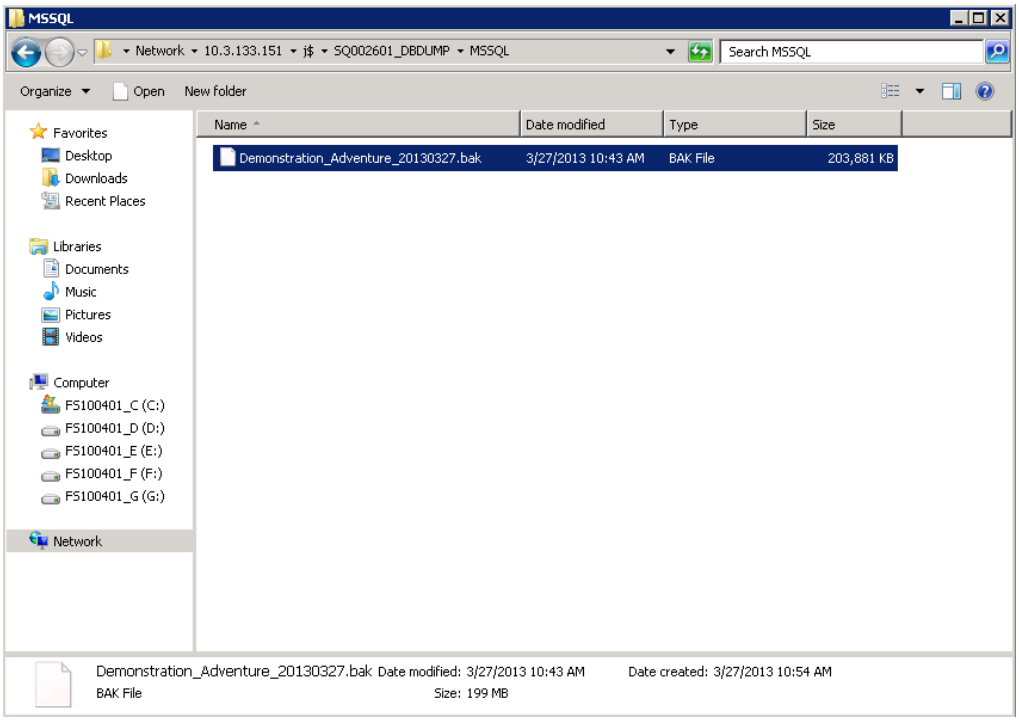


12. Go to the subfolder hosting the daily backups.

    (for example: « SQ002601_DBDUMP\MSSQL » on the J drive)

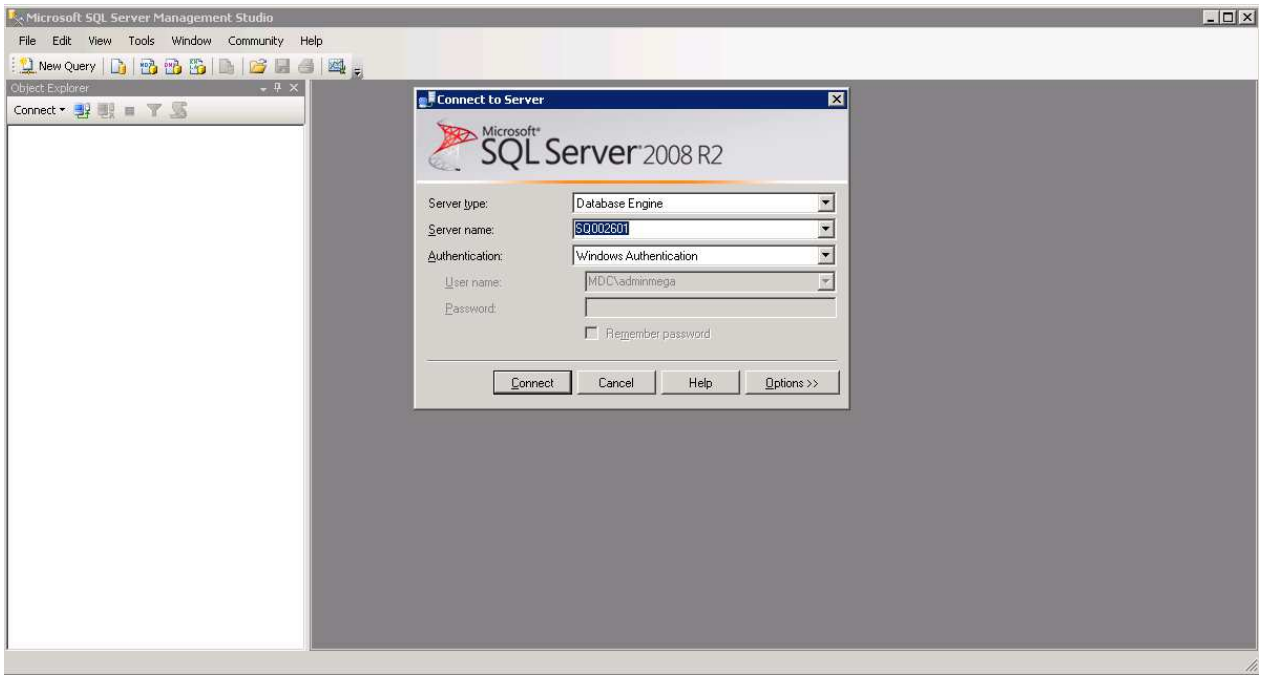


13. Copy the file from the source folder to this one.

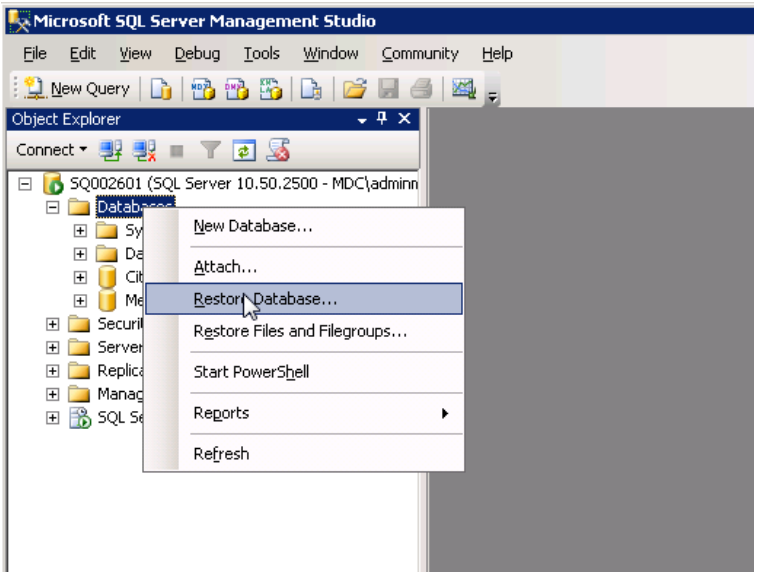


Restore

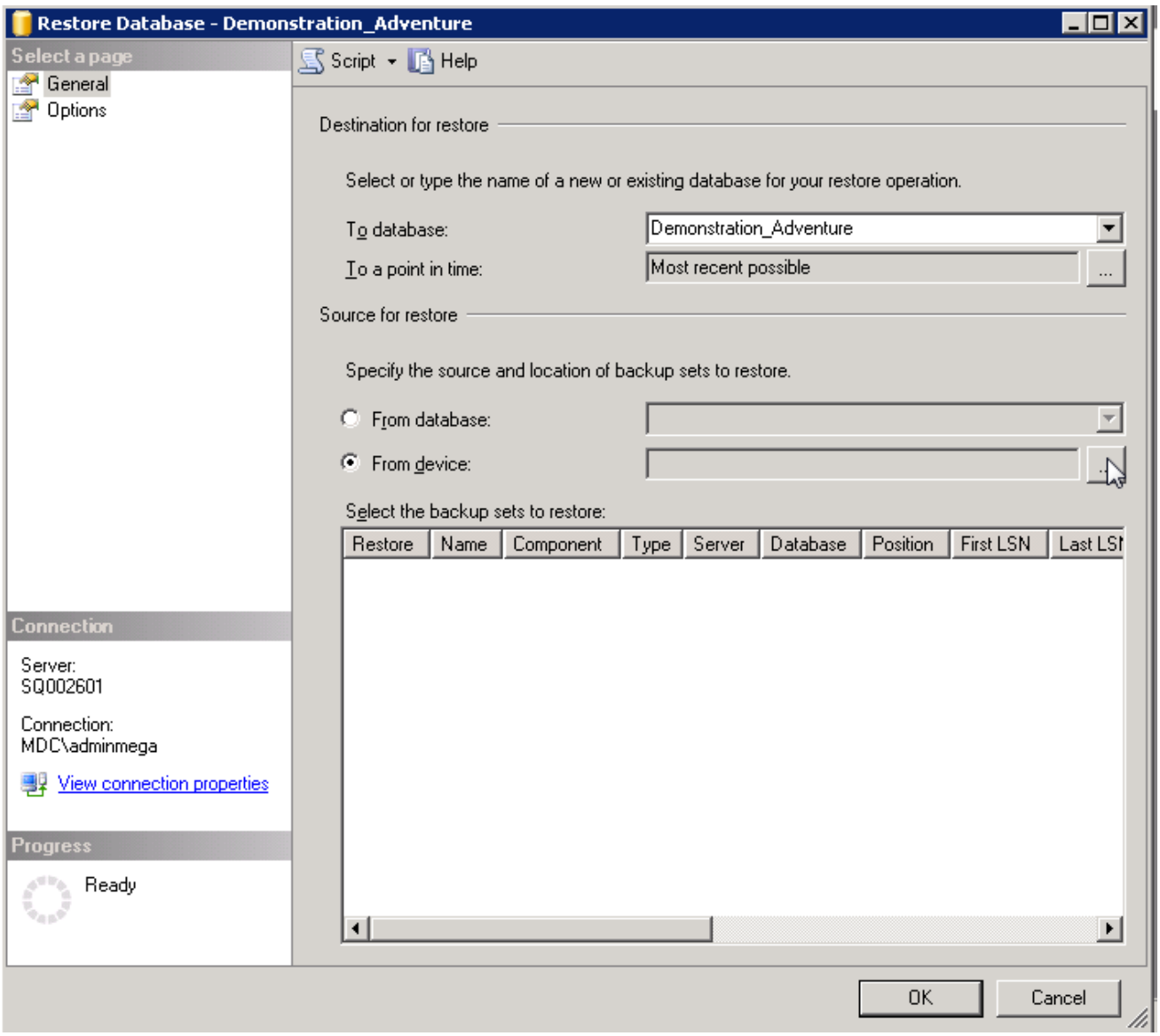
1. Connect to the target SQL Server server (example: SQL002601).
2. Launch the Management Studio, and connect to the instance using, if possible, a Windows account that is both sysadmin, and has access rights to the folder where the backup file was copied (example: mdc\adminmega).



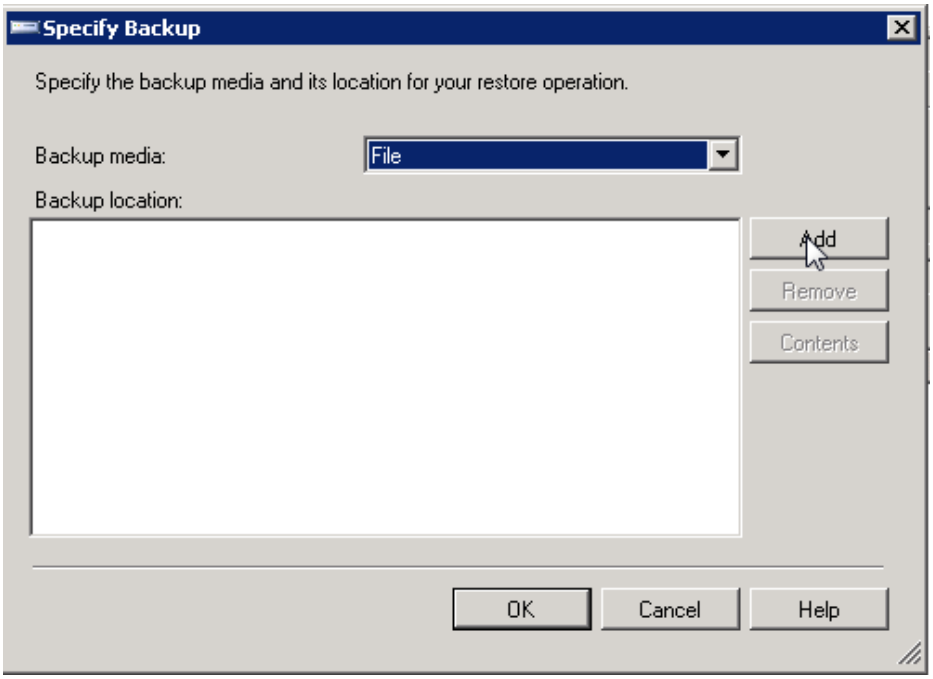
3. Right-click **Databases**, and select **Restore Database**.



4. From the **General** page, in the **Destination for restore** pane, in the **To database** field, provide the name of the database that will be created (example: Demonstration_Adventure).
5. In the **Source for restore** pane, select **From Device** option and click ...


Restore Database - Demonstration_Adventure
Select a page
General
Options
Script
Help
Destination for restore
Select or type the name of a new or existing database for your restore operation.
To database:
Demonstration_Adventure
To a point in time:
Most recent possible
Source for restore
Specify the source and location of backup sets to restore.
From database:
From device:
Select the backup sets to restore:
Restore
Name
Component
Type
Server
Database
Position
First LSN
Last LSN
Connection
Server:
SQ002601
Connection:
MDC\adminmega
View connection properties
Progress
Ready
OK
Cancel

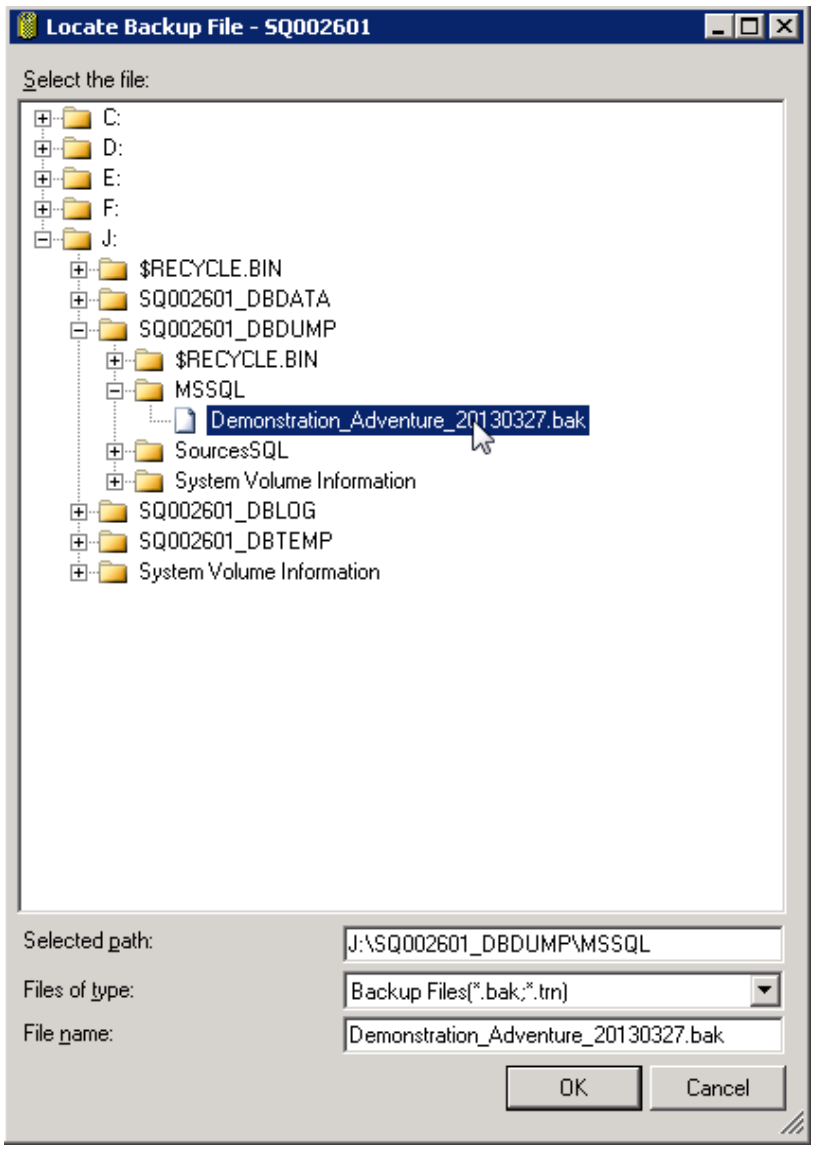
6. Click **Add**.



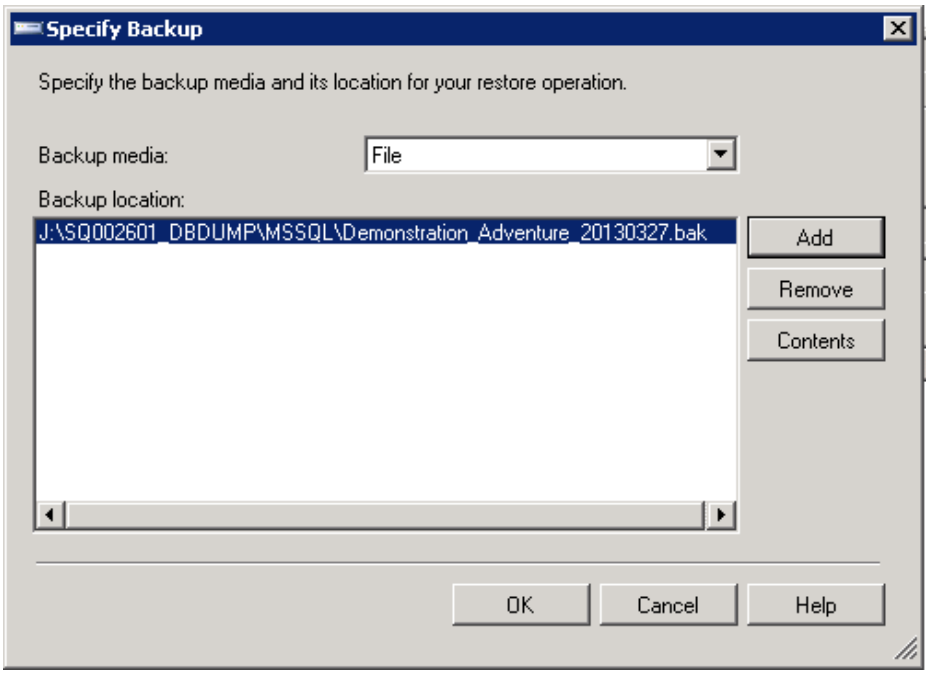
7. Check that we are correctly put in the folder « J:\SQ002601_DBDUMP\MSSQL».

   Otherwise, browse the folders to get to it. Then, click the .bak file that you just copied on the server (for example: « Demonstration_Adventure_20130327.bak »).

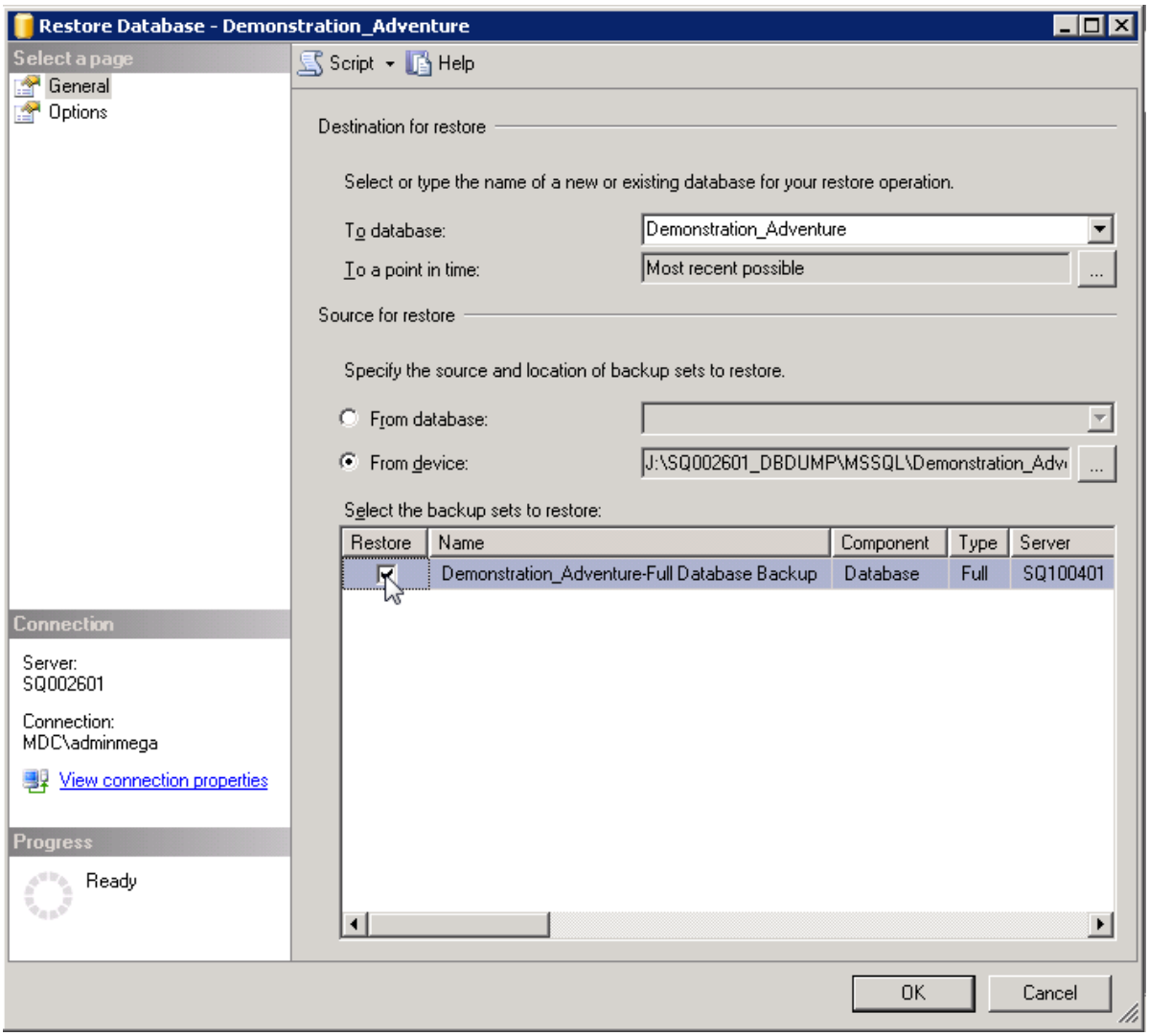
   Check that the « File Name » field is correctly filled, and click **OK**.


Locate Backup File - SQ002601
Select the file:
C:
D:
E:
F:
J:
$RECYCLE.BIN
SQ002601_DBDATA
SQ002601_DBDUMP
$RECYCLE.BIN
MSSQL
Demonstration_Adventure_20130327.bak
SourcesSQL
System Volume Information
SQ002601_DBLOG
SQ002601_DBTEMP
System Volume Information
Selected path:
J:\SQ002601_DBDUMP\MSSQL
Files of type:
Backup Files(*.bak;*.trn)
File name:
Demonstration_Adventure_20130327.bak
OK
Cancel

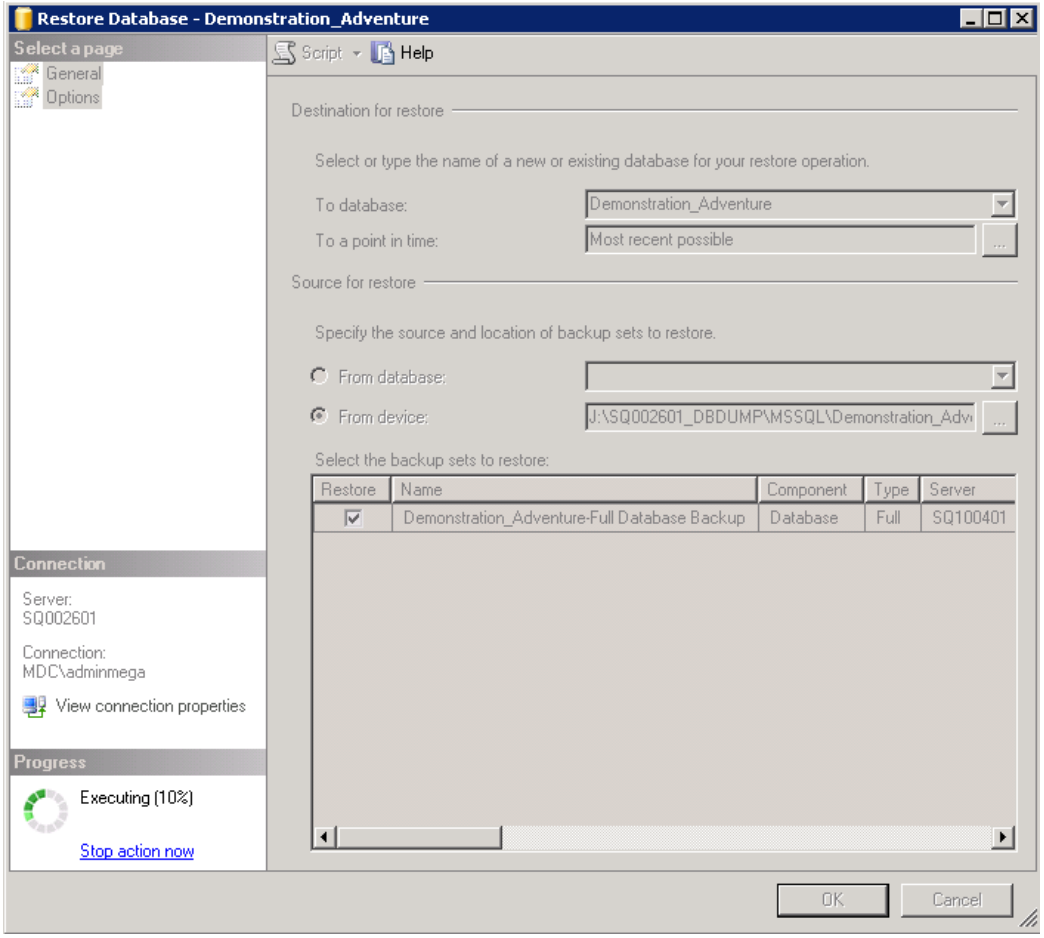
8. Click **OK**.


Specify Backup
Specify the backup media and its location for your restore operation.
Backup media:
File
Backup location:
J:\SQ002601_DBDUMP\MSSQL\Demonstration_Adventure_20130327.bak
Add
Remove
Contents
OK
Cancel
Help

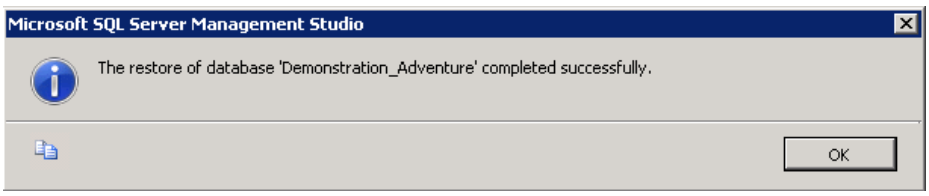
9. In the **Source for restore** pane, in the **Select the backup sets to restore** table select **Restore** at the beginning of the line.



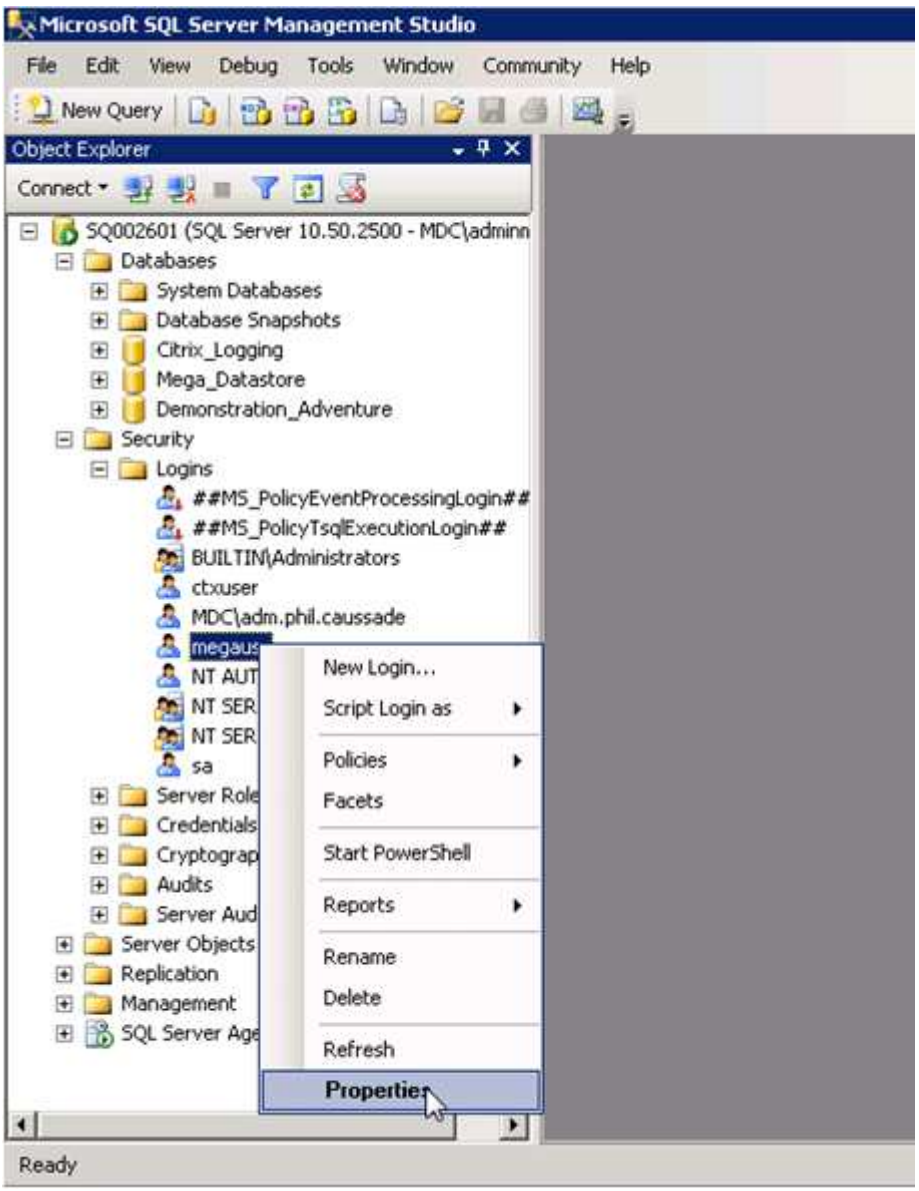
10. click **OK**.



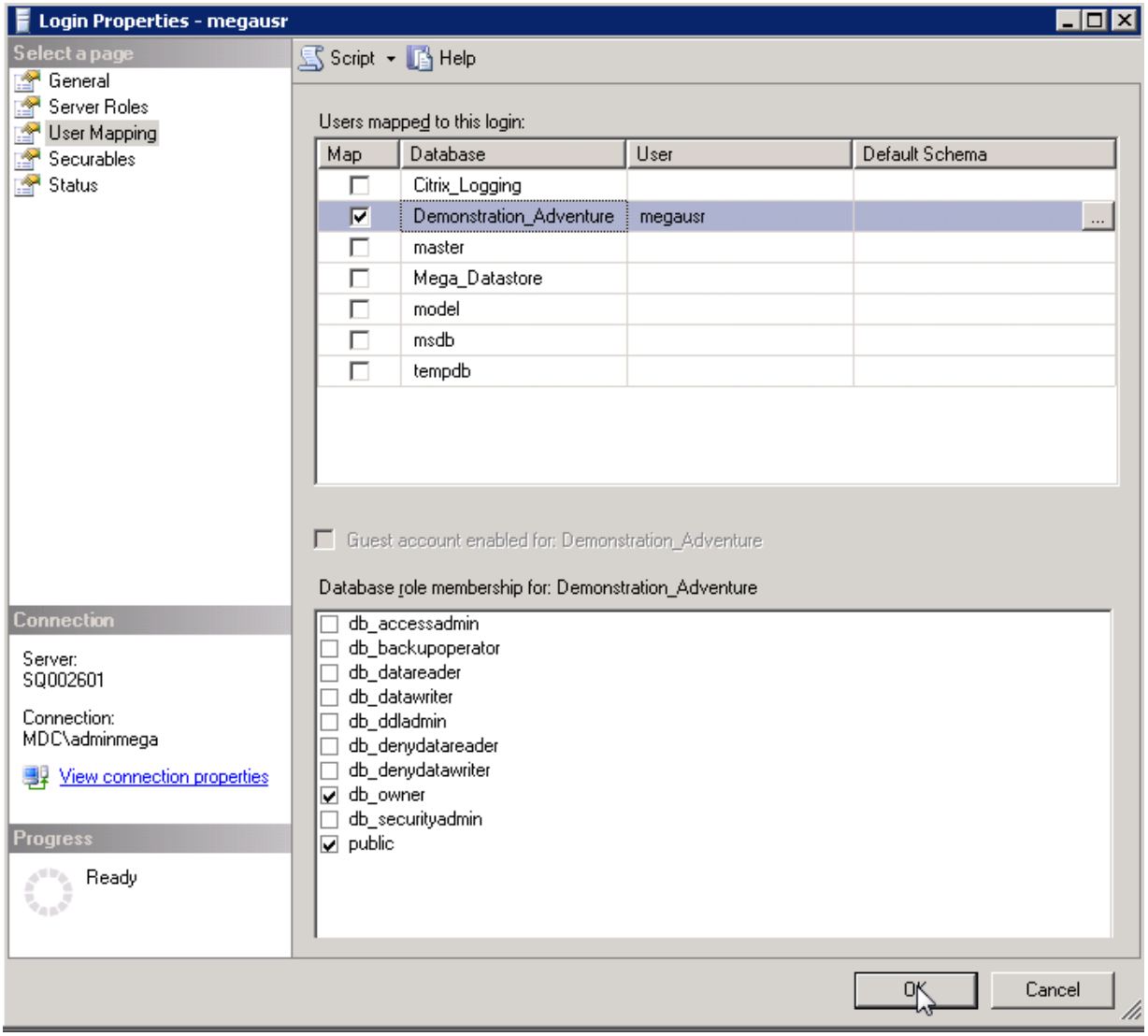
11. Once the restore completed successfully, click **OK**.



12. In ManagementStudio, expand **Security** folder, and **Logins** folder.
13. Right-click the account that you will use to connect the application to SQL Server (for example: login « megausr ») and select **Properties**.



14. In the **Login Properties - <login>** window, select **User Mapping** page, and select the **Map** corresponding to the database lin you just restored.
15. In the **Database role membering for: <database name>** pane, select « db_owner ».
16. Click **OK**.



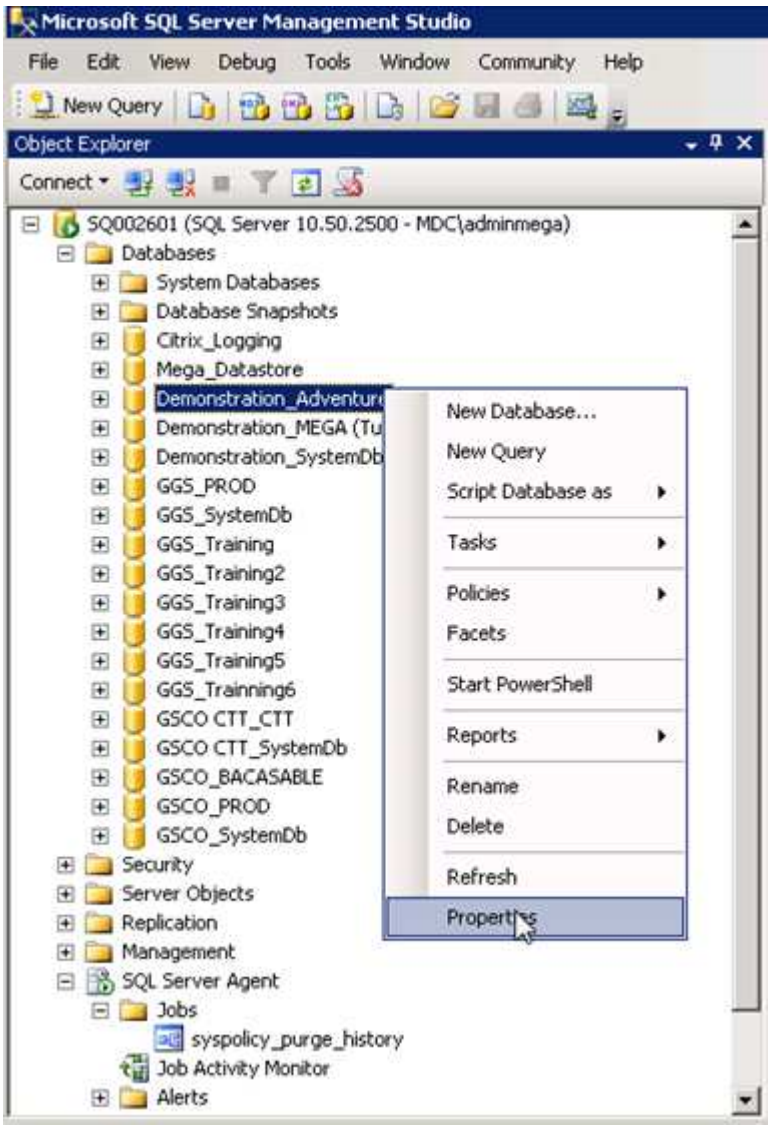
IF you moved from a version of SQL Server to a more recent one, you have to upgrade the compatibility level of your database.

In this example, the source database came from an SQL Server 2005 instance, and was restore on an SQL Server 2008 instance.

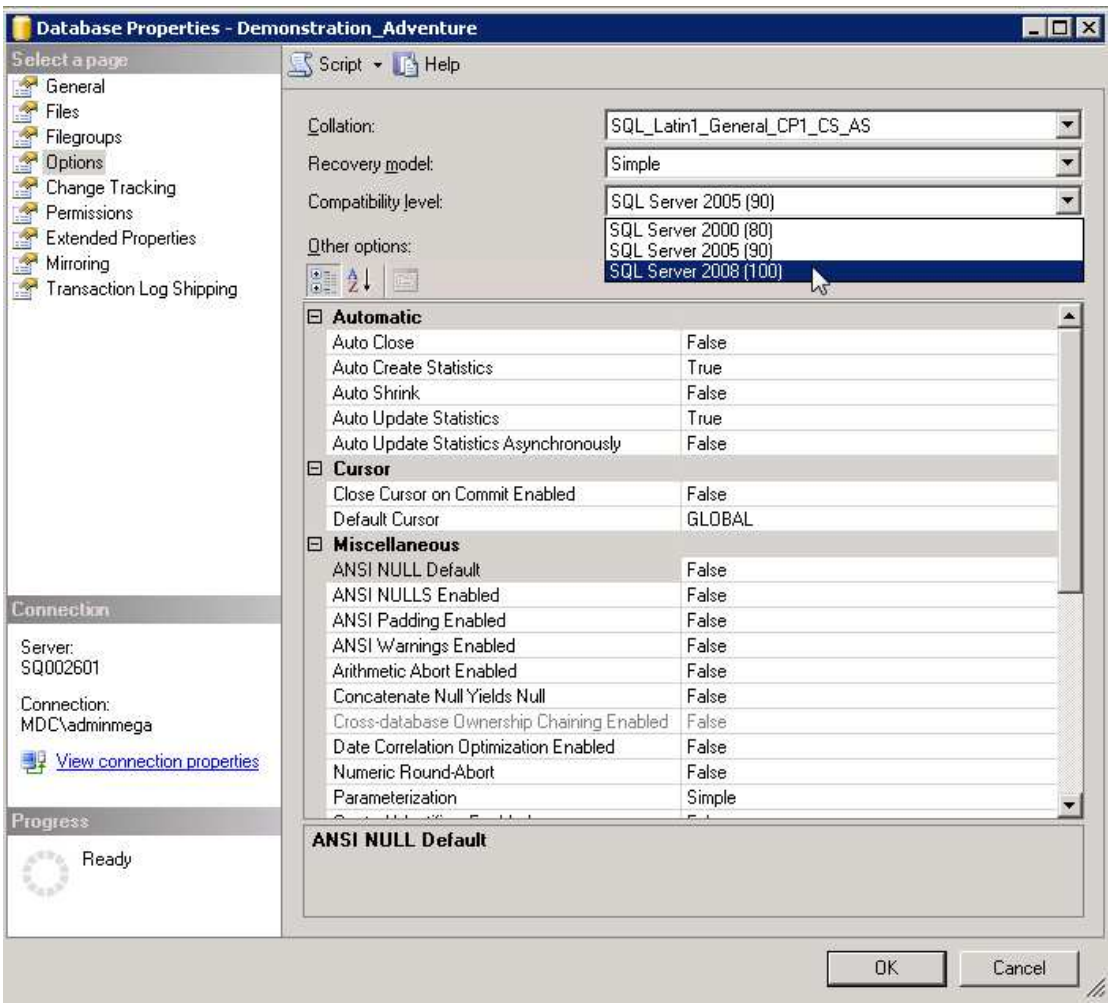
If you stay on the same version, go directly to the next section.

To upgrade:

1. Right-click the database and select **Properties**.



2. Select the **Options** page, in the **Compatibility level** drop down list select the appropriate version (for example « SQL Server 2008 (100) »).


Database Properties - Demonstration_Adventure
Select a page
General
Files
Filegroups
Options
Change Tracking
Permissions
Extended Properties
Mirroring
Transaction Log Shipping
Script
Help
Collation:
SQL_Latin1_General_CP1_CS_AS
Recovery model:
Simple
Compatibility level:
SQL Server 2005 (90)
SQL Server 2000 (80)
SQL Server 2005 (90)
SQL Server 2008 (100)
Other options:
Automatic
Auto Close
False
Auto Create Statistics
True
Auto Shrink
False
Auto Update Statistics
True
Auto Update Statistics Asynchronously
False
Cursor
Close Cursor on Commit Enabled
False
Default Cursor
GLOBAL
Miscellaneous
ANSI NULL Default
False
ANSI NULLS Enabled
False
ANSI Padding Enabled
False
ANSI Warnings Enabled
False
Arithmetic Abort Enabled
False
Concatenate Null Yields Null
False
Cross-database Ownership Chaining Enabled
False
Date Correlation Optimization Enabled
False
Numeric Round-Abort
False
Parameterization
Simple
ANSI NULL Default
Connection
Server:
SQ002601
Connection:
MDC\adminmega
View connection properties
Progress
Ready
OK
Cancel

3. Click **OK**.

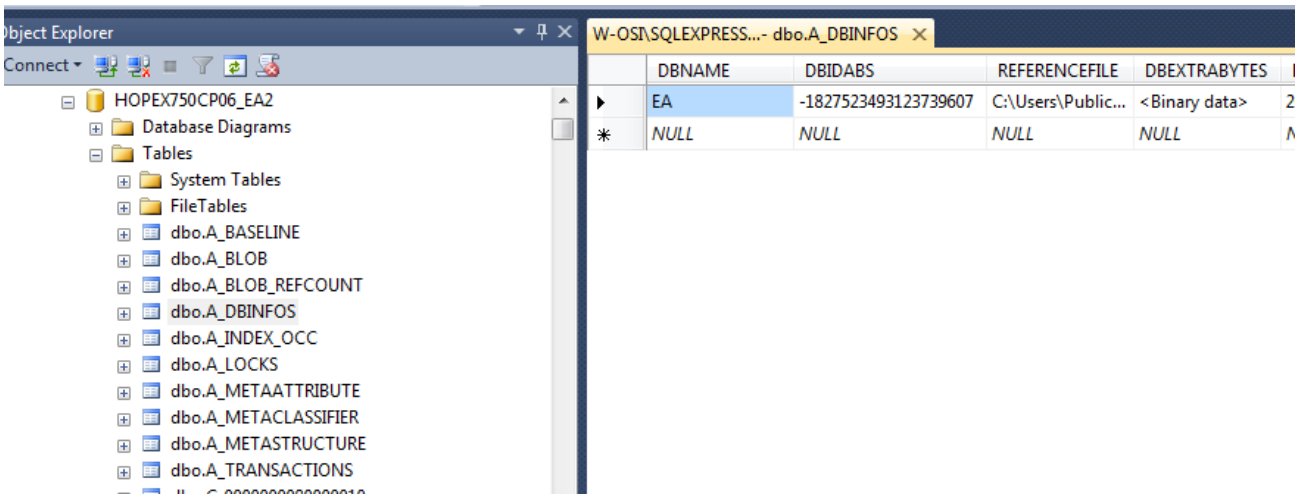
When duplicating a repository in an environment – Expert mode

This section is only if you are very confident about your Mega and RDBMS skills, since you have to modify data directly in some tables of your database.

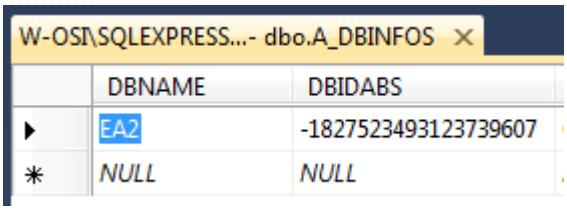
In case you want, within an environment, duplicate a specific repository, you need to tweak the data manually in two tables of your restored database.

For example: you want to duplicate repository "EA" to have a repository "EA2" in the environment "HOPEX750CP06".

1. You have restored database EA twice, in databases "HOPEX750CP06_EA" and "HOPEX750CP06_EA2".
2. Using SQL Server Management Studio, open database "HOPEX750CP06_EA2", and edit the table "dbo.A_DBINFOS":

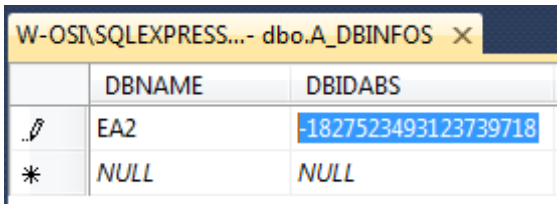


3. Modify the **DBNAME** field of that line, to enter the name of your duplicate, i.e. how you want it to appear in your environment (for example: here, "EA2").



4. Modify the **DBIDABS** field that is the unique identifier of the repository within your environment.

   You need to change its value, and make sure that it is unique for all other repositories, so you need to check in all schemas. Make sure that when you change one or two characters, it will create a string that is not used by any other (for example: here, we modified the last 3 digits from " 607" to "718").



5. Save your updates and close the table.

RDBMS environment duplication - 1 - Oracle

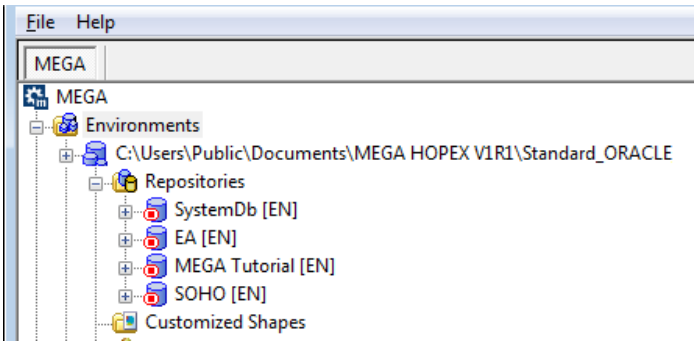
GENERALITIES

This document purpose is to explain the way of duplicating MEGA data that are stored in an Oracle RDBMS and how to use this data to create a duplicate MEGA environment that will point on this duplicated data.

It is important to have **no end user activity** on the MEGA environment while the export procedure is carried out **to avoid inconsistency** in the exported data.

You can go through the whole document to carry out the entire scenario. You can also jump directly to the Oracle data import section if you already have the Oracle dump files containing the SCHEMAs hosting the MEGA environment data.

MEGA recommends using a different Oracle SCHEMA for each MEGA repository. For example, if you have to deal with a MEGA environment like this one



It would mean that there are 4 SCHEMAs involved (also 1 for the SystemDb). It is technically possible to have every repository hosted in the same SCHEMA but **not recommended.**

For the following scenario, a local installation of Oracle will be used for exporting and importing data. The service name of this instance is EDE2.

Oracle supported versions

Refer to "*Web Front-End Architecture Overview*" or "*Windows Front-End Architecture Overview*" guides to find out the Oracle supported versions of your version.

Oracle tools used

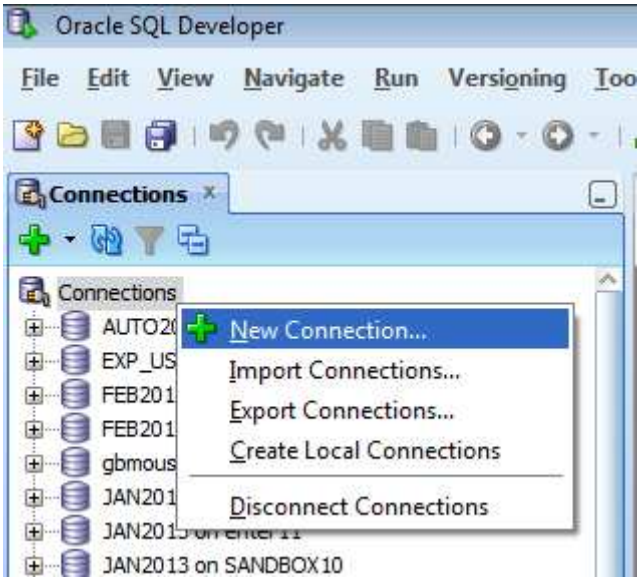
The Oracle tools that will be used are:

- Oracle SQL Developer, a free graphical tool that can be used for Oracle administration tasks. See Oracle documentation for details.
- Oracle Data Pump to export and import the physical Oracle data. There are different Data Pump components but you will be using only the command-line clients **expdp** and **impdp**. See Oracle documentation for details.

USING ORACLE SQL DEVELOPER

Oracle SQL Developer can be downloaded for free from the Oracle web site, on the Oracle software download page.

Using SQL Developer, the first thing to do is to set up a connection to be able to browse the content of the database for the SCHEMA that hosts MEGA data.



Reminder:

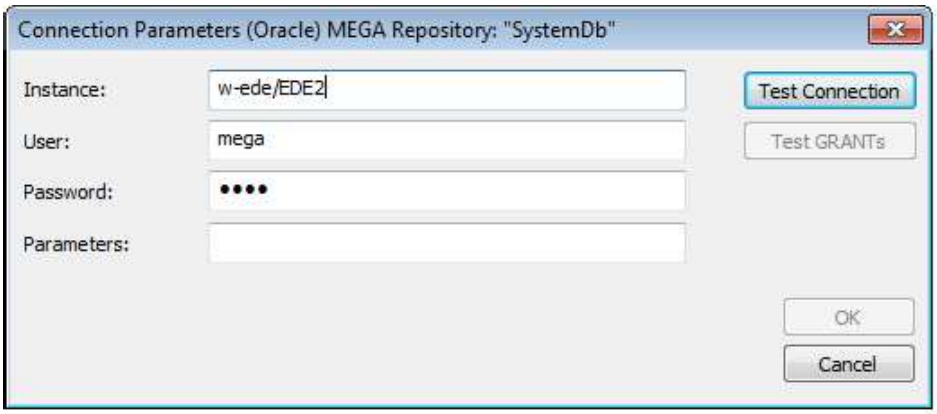
Note that the concepts of Oracle USER and Oracle SCHEMA are very similar but not exactly the same. Refer to Oracle documentation to clarify this point but here is what really needs to be understood:

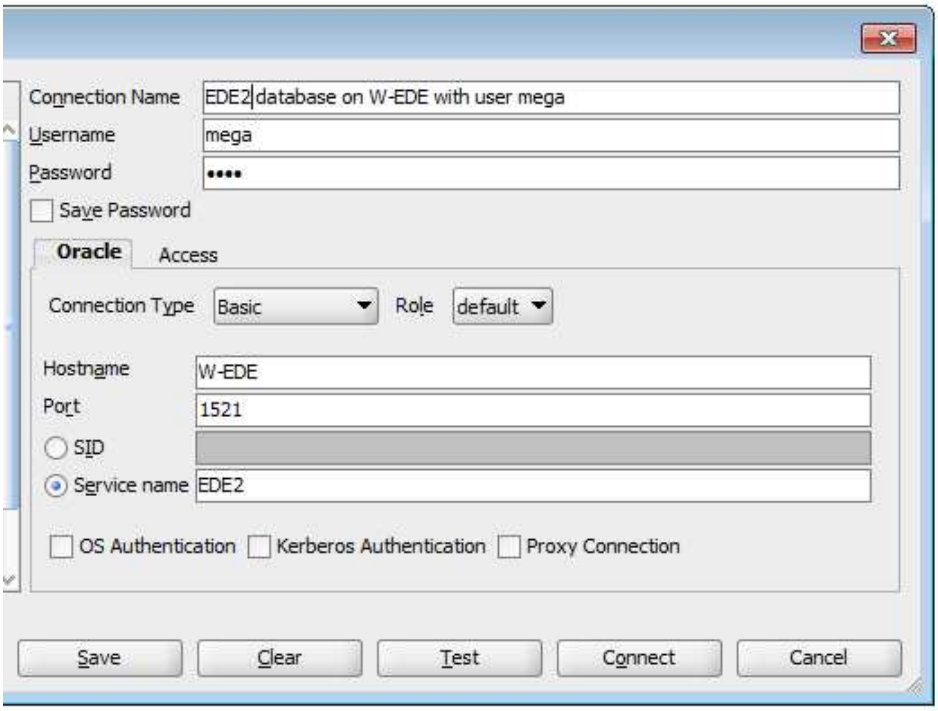
> A **schema** is a collection of logical structures of data, or schema objects. A schema is owned by a database user and has the same name as that user. Each user owns a single schema.

Fill in the connection parameters according to MEGA Administration settings

Note that you have to enter a port number. Port number 1521 is the default one. If another port was used, you would see it in the connection parameters that were entered in MEGA.

For example: w-ede:1234/EDE2



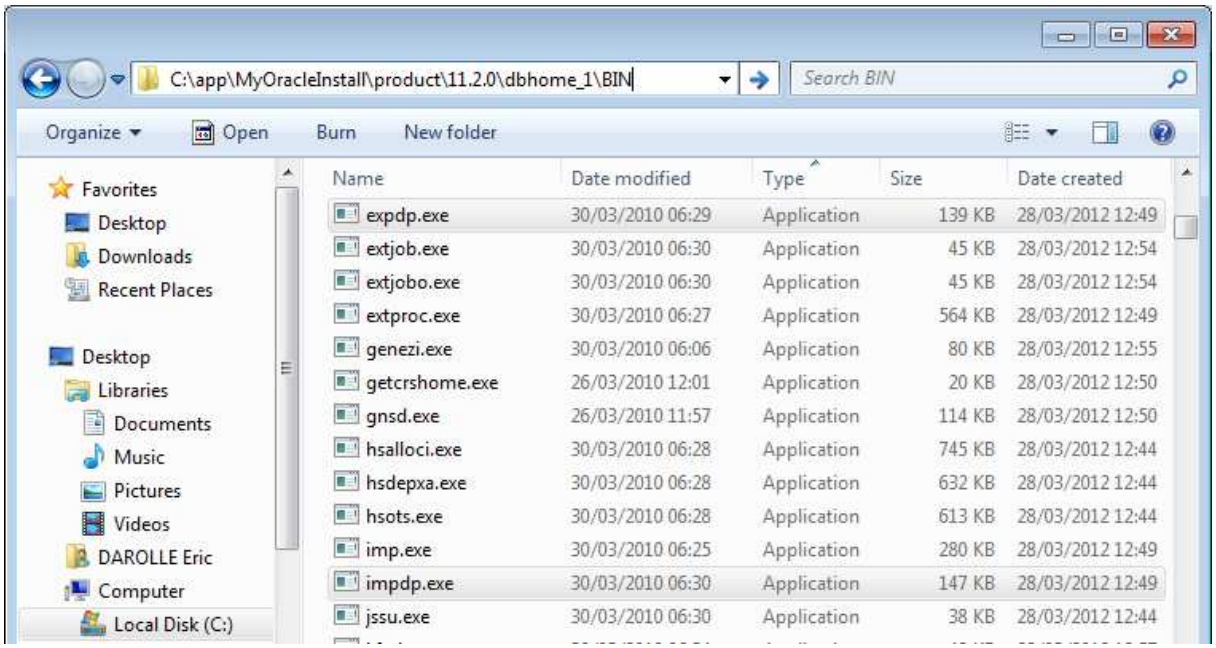


In this example, the connection parameters in MEGA were entered using a method called **EASY CONNECT NAMING**. They could also be entered in another form, using the **CONNECT DESCRIPTOR** method. This method is discussed further in this document.

THE ORACLE DATA EXPORT WITH EXPDP

The **expdp** client is a program that uses the Data Pump API. It is invoked by using the **expdp** command. See Oracle documentation for the parameters that are available in the export's command-line mode.

The **expdp** client can be found there for a typical Oracle installation on a Windows operation system.



Location of the Data Pump utilities

Prerequisites for using expdp utility

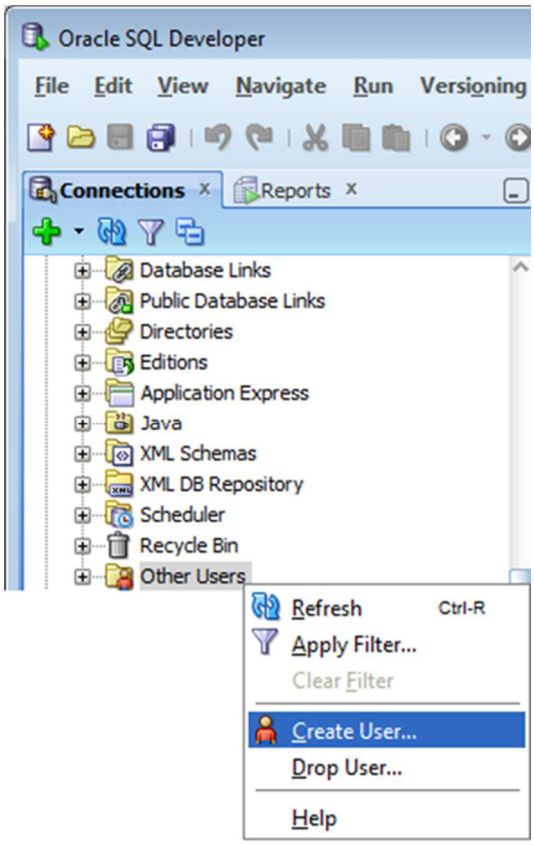
No one must be connected to MEGA when the export starts as it could lead in a set of **inconsistent data**.

Prerequisites to use the Data Pump export, you need to correctly configure the following Oracle Objects:

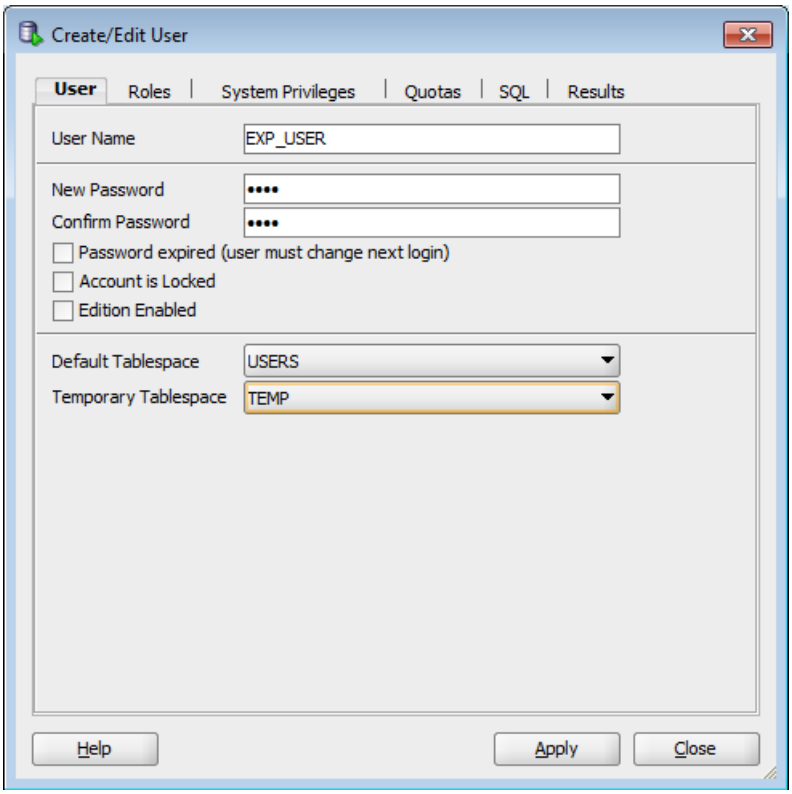
- an Oracle USER that has the rights to use Data Pump
- an Oracle DIRECTORY object that maps to a folder that will be used for putting the dump files.

Oracle USER configuration

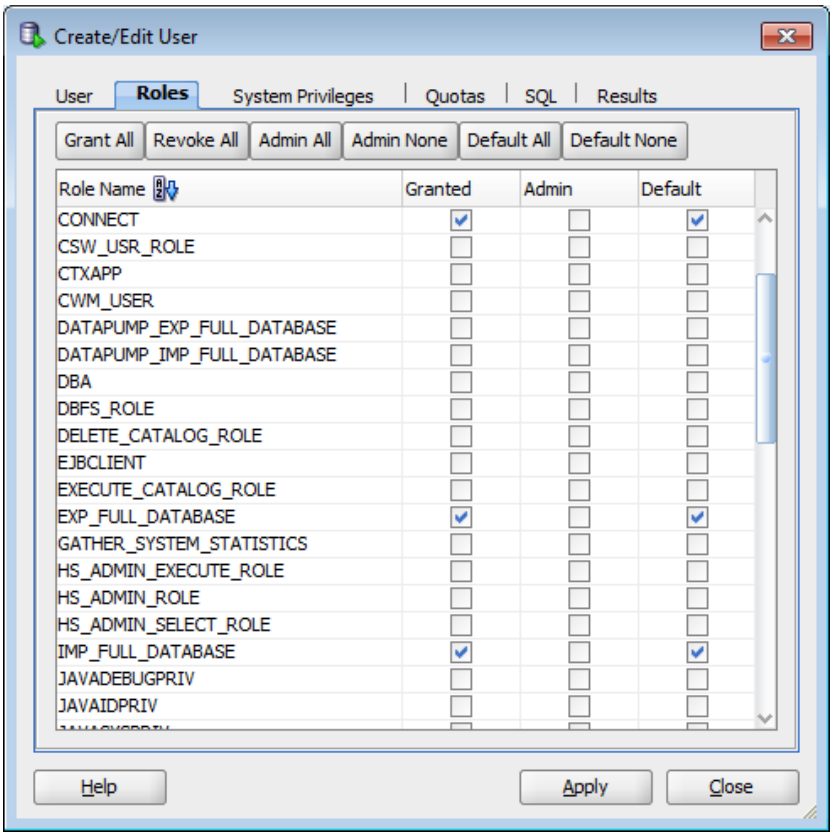
1. Launch Oracle SQL developer (other tools might be used) and connect to the Oracle database that hosts the original data to be duplicated with the USER SYSTEM (or any other USER that has the right to create USERs and to GRANT rights to these USERs).
2. From Oracle SQL developer, create an Oracle USER for exporting the data.



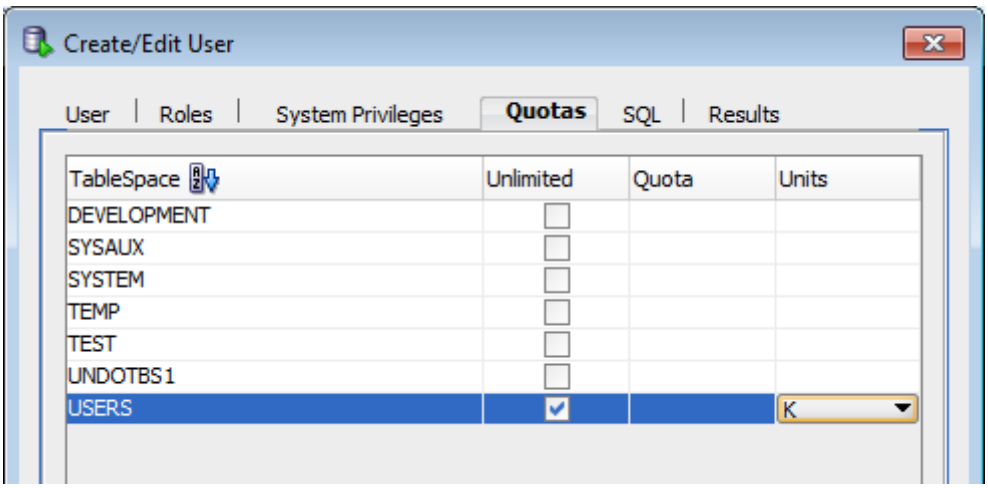
3. In the **User** tab, name the user: EXP_USER.



4. In the **Roles** tab, select **Granted** for the following roles:
   - CONNECT
   - EXP_FULL_DATABASE
   - IMP_FULL_DATABASE



5. In the **Quotas** tab, Quotas must be assigned on the tablespace hosting the data.



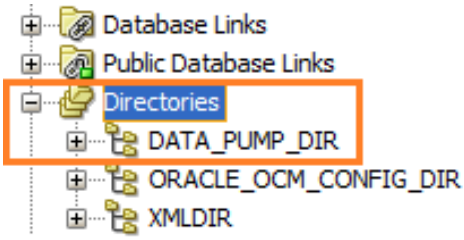
The resulting SQL script:

```
-- USER SQL
CREATE USER EXP_USER IDENTIFIED BY EXP_PWD
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP";
ACCOUNT UNLOCK;

-- ROLES
GRANT "CONNECT" TO EXP_USER ;
GRANT "DATAPUMP_IMP_FULL_DATABASE" TO EXP_USER ;
GRANT "DATAPUMP_EXP_FULL_DATABASE" TO EXP_USER ;

-- SYSTEM PRIVILEGES
GRANT CREATE ANY DIRECTORY TO EXP_USER ;

-- QUOTAS
ALTER USER EXP_USER QUOTA UNLIMITED ON USERS;
```

Note: there is an extra SYSTEM PRIVILEGE **CREATE ANY DIRECTORY** that was not mentioned before. It is not compulsory but nice to have if you want this user-to-be-created to be able to create DIRECTORIES for receiving the generated files.

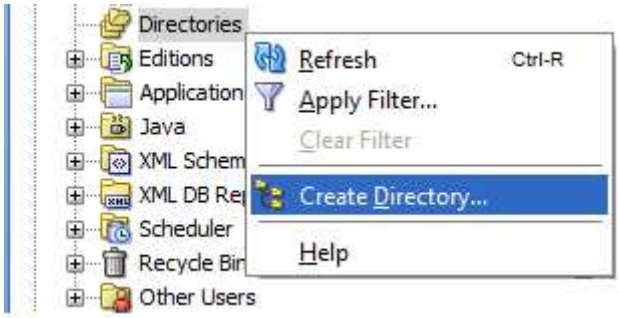
Oracle DIRECTORY object configuration

There is already a default DIRECTORY object in Oracle for using Data Pump with. It is called DATA_PUMP_DIR.

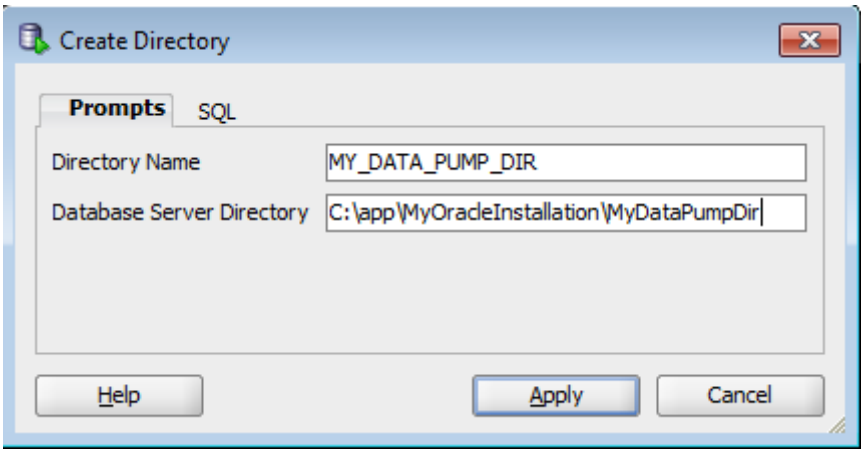


Since it could be mapped to a folder where writing is not allowed (for instance in a secured production environment), let us see how to create a new one where we are sure to be able to write and for which we know for sure that the remaining space available will be enough.

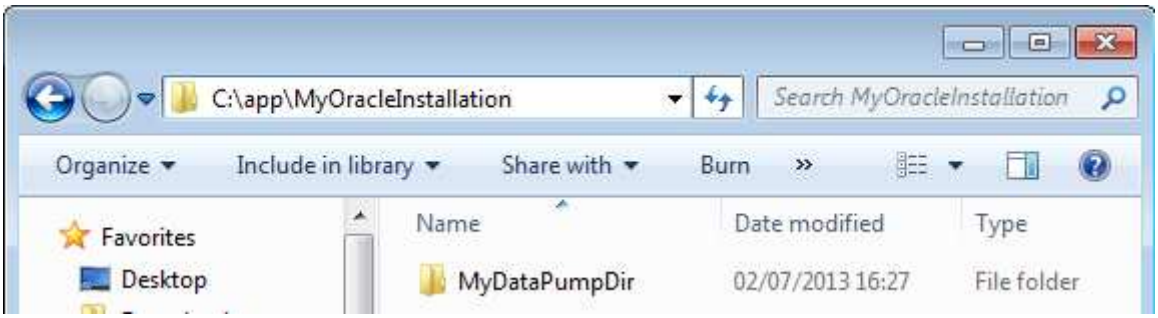
1. In Oracle SQL developer, with the same connection used to create the Oracle USER, right-click **Directories** node and select **Create Directory**.



2. Enter a name for the DIRECTORY object to be created (for example: MY_DATA_PUMP_DIR).
3. Enter the path to the actual folder for mapping the DIRECTORY created (for example: C:\app\MyOracleInstallation\MyDataPumpDir).



Make sure that this folder is writable in your context.


C:\app\MyOracleInstallation
Search MyOracleInstallation
Organize
Include in library
Share with
Burn
Favorites
Desktop
Name
Date modified
Type
MyDataPumpDir
02/07/2013 16:27
File folder

Initiating the export

This section describes how to launch the expdp export utility to create the dump files for each SCHEMA to duplicate each MEGA repository's data.

> **Note from Oracle documentation:**
> Do not invoke Export as SYSDBA, except at the request of Oracle technical support. SYSDBA is used internally and has specialized functions; its behavior is not the same as for general users.

expdp parameters explained

Here are the command and the parameters to use:

expdp

invokes the Data Pump export utility

user_name/user_password@address_of_Oracle_database_instance

specifies the Oracle USER that launches the export

must be followed by the USER's password

USER name and password are followed by the @ symbol and the Oracle database instance to export from. It could be seen as the "address" of the database or the way to identify it. There are more than one type of variables that can follow the @ symbol but we will show the example with one type only: a **CONNECT DESCRIPTOR** (since it is the one that will always be usable). See Oracle documentation for more details on CONNECT DESCRIPTORS.

The CONNECT DESCRIPTOR is of the following format:

(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = <DB server Id>) (PORT = <listener port number>)) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = <Oracle instance name>)))

DIRECTORY=

Specified the Oracle DIRECTORY object's name that will be the container of the dump file and the log file.

DUMPFILE=

The name of the resulting dump file. The dump file contains the data exported.

LOGFILE=

The name of the file that will log the actions that occur during the export.

SCHEMAS=

The name of the Oracle SCHEMAS to be exported. In our example the dump file will contain the data of only one SCHEMA.

Since a SCHEMA is supposed to match a MEGA repository (if MEGA recommendations have been followed), we will have as many dump files as the number of MEGA repositories in our environment (the SystemDb also counts for one repository).

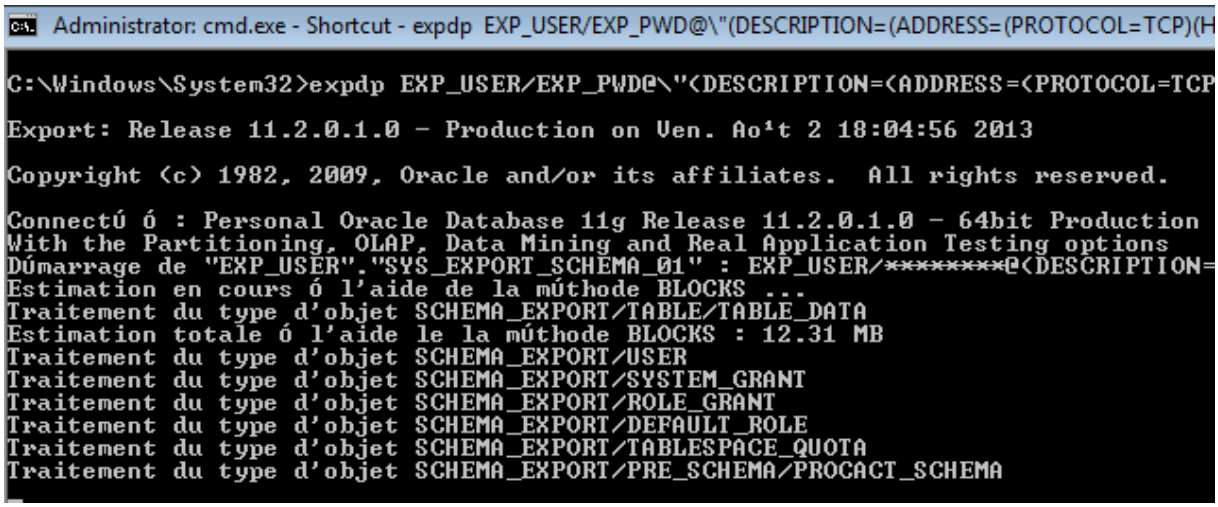
The actual export

MEGA recommends using one SCHEMA for hosting only one MEGA repository. This is for making Oracle administration tasks much easier. If, for any reason, this cannot be achieved, and many MEGA repositories are hosted in the same Oracle SCHEMA, a slightly different approach is to be used (see **Error! Reference source not found.** section).

SystemDb repository export

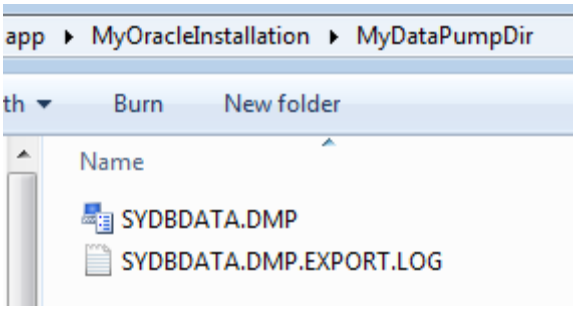
Here is the command used in our example, starting with the export of the SCHEMA used for the **SystemDb** repository. In this example the SCHEMA is named MEGASYSDB:

```
expdp EXP_USER/EXP_PWD@\"(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = W-EDE) (PORT=1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = EDE2)))\" DIRECTORY = MY_DATA_PUMP_DIR DUMPFILE =
SYDBDATA.DMP LOGFILE = SYDBDATA.DMP.EXPORT.LOG SCHEMAS = MEGASYSDB
```



The export starts

Once it is done the data and log files are created in the directory.



User data repositories export

Now we will take care of the export for the SCHEMA(s) hosting the data corresponding to the User data repositories. To do this, refer to the export technique that has just been described. Only change the names for the files (dump and log) and use the correct corresponding SCHEMA.

Let us see how the command would look like the MEGA repositories hosting the MEGA User data (i.e. repository **Adventure** of the environment **Demonstration**). In this example the SCHEMA is named MEGADATA1:

Here is the command used (same as before with changes highlighted):

```
expdp EXP_USER/EXP_PWD@\"(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = W-EDE) (PORT=1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = EDE2)))\" DIRECTORY = MY_DATA_PUMP_DIR DUMPFILE =
MEGADATA1.DMP LOGFILE = MEGADATA1.DMP.EXPORT.LOG SCHEMAS = MEGADATA1
```

Repeat the operation **for each MEGA repository**.

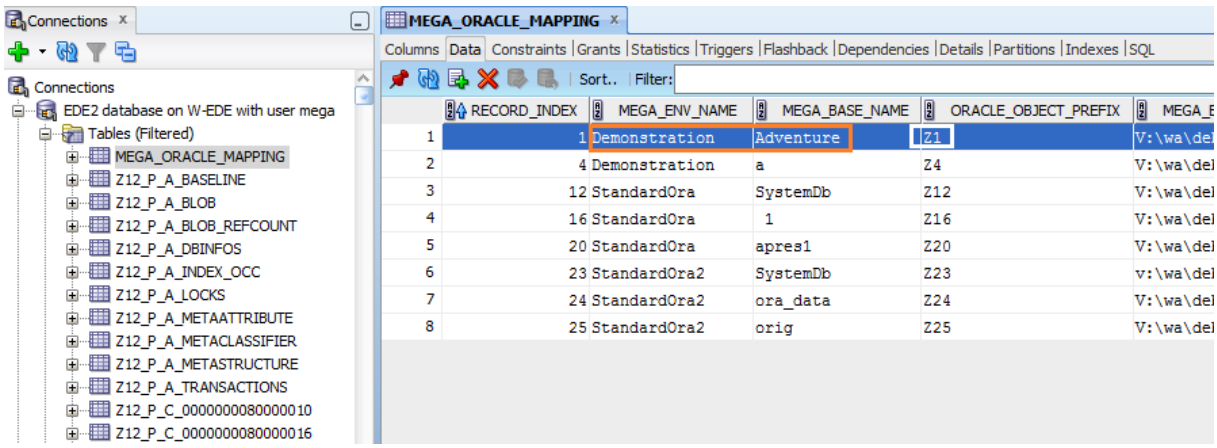
Special case (not recommended): many repositories for one SCHEMA

1. In that case, identify the repository to be exported.
2. Change the expdp command so that it filters only the tables corresponding to the repository to be exported.

Identifying the set of data corresponding to the MEGA repository

1. In SQL Developer, browse the SCHEMA of the original data.
2. Look for the table called MEGA_ORACLE_MAPPING.
3. Select the table name.
4. In the right pane select the **Data** tab.

   Here, we are interested in the value of ORACLE_OBJECT_PREFIX for the row where MEGA_ENV_NAME is **Demonstration** and MEGA_BASE_NAME is **Adventure**.



RECORD_INDEX	MEGA_ENV_NAME	MEGA_BASE_NAME	ORACLE_OBJECT_PREFIX
1	Demonstration	Adventure	Z1

The value is 'Z1'.

Filtering the export to include only that specific repository data

To filter the export so that it contains only the data of the **Adventure** repository, you need to provide the Data Pump export utility with something that says:

"In my export file, I want only the tables starting with 'Z1' ". To do that, you need to add the following instruction at the end of the command:

```
[INCLUDE=TABLE:]
```

Because the prefix used for the tables for the Adventure repository starts with 'Z1', it means that the tables we are interested in really start with 'Z1_'.

The underscore character ('_') is a special character in SQL queries for Oracle meaning "any character". So when you ask the export utility to export tables starting with 'Z1_', it will understand tables starting with 'Z1'. So table which names start with 'Z12' will also be exported…

➔ There is a work around: the ESCAPE clause. With this clause, you can tell Oracle that a special character (here you will use '!') will be used to make Oracle interpret the '_' literally, rather than as a special pattern matching character.

Here is the command that could be used:

```
expdp EXP_USER/EXP_PWD@\"(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = W-EDE) (PORT=1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = EDE2)))\" DIRECTORY = MY_DATA_PUMP_DIR DUMPFILE =
MEGADATA1.DMP LOGFILE = MEGADATA1.DMP.EXPORT.LOG SCHEMAS = MEGADATA1
INCLUDE = TABLE:\"LIKE 'Z1!_%' ESCAPE '!'\"
```

The set of data **wouldn't be complete** if you forget to include the table **MEGA_ORACLE_MAPPING** during export. This needs not to be taken care of when exporting a full SCHEMA in the situation of a one to one SCHEMA – MEGA repository.

To include the MEGA_ORACLE_MAPPING table in the exported file, you need to change the way of filtering as the [INCLUDE=TABLE:] clause cannot handle the OR operator.

You could use a filtering query:

```
expdp EXP_USER/EXP_PWD@\"(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = W-EDE) (PORT=1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = EDE2)))\" DIRECTORY = MY_DATA_PUMP_DIR DUMPFILE =
MEGADATA1.DMP LOGFILE = MEGADATA1.DMP.EXPORT.LOG SCHEMAS = MEGADATA1
INCLUDE = TABLE:"IN (SELECT TABLE_NAME FROM ALL_TABLES WHERE
TABLE_NAME='MEGA_ORACLE_MAPPING' OR TABLE_NAME LIKE 'Z1!_%' ESCAPE
'!')"
```

There is still a problem with this last expdp command. It will not work as it is because of all the special characters that need to be escaped in that string:

```
"IN (SELECT TABLE_NAME FROM ALL_TABLES WHERE TABLE_NAME =
'MEGA_ORACLE_MAPPING' OR TABLE_NAME LIKE 'Z1!_%' ESCAPE '!')"
```

to make it possible for the operating system's command prompt to pass a meaningful string to the expdp utility.

It is getting too complicated.

At that point you can use an expdp special feature that allows you to pass parameters in an easier way: the [PARFILE=] parameter. It is used to point to a file that contains all the parameters to pass to the expdp command.

It is used like this:

```
expdp EXP_USER/EXP_PWD@\"(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = W-EDE) (PORT=1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = EDE2)))\" PARFILE = "C:\parm.txt"
```

having the parm.txt file containing:

```
DIRECTORY = MY_DATA_PUMP_DIR

DUMPFILE = MEGADATA1.DMP

LOGFILE = MEGADATA1.DMP.EXPORT.LOG

SCHEMAS = MEGADATA1

INCLUDE = TABLE:"IN (SELECT TABLE_NAME FROM ALL_TABLES WHERE
TABLE_NAME='MEGA_ORACLE_MAPPING' OR TABLE_NAME LIKE 'Z1!_%' ESCAPE
'!')"
```

Initiating the import

Now that the data is in a "transportable format" (i.e. dump files), we will take care of recreating a set of duplicated data based on it.

Since we saw in the export part of this document how to restrict the data contained in the dump file to contain only what is needed, we will assume that the Oracle dump files to be imported need no filtering.

Location of the target Oracle database

In many cases, the import phase will not take place in the same Oracle environment as the one used for the export.

Let us consider the scenario in which nothing is the same in the source and the target Oracle environments:

- The machine hosting the Oracle database is different.
- The Oracle service_name is not the same.
- The data pump DIRECTORY will not be the same.
- The SCHEMA in which the exported data are to be put in has a different name.
- The TABLESPACE of the source SCHEMA and the target SCHEMA have different names. We have not gone through the TABLESPACE concept yet, but we will explain it further.
- The Oracle USER that will be used for importing is not the same (here the User will be called IMP_USER). The GRANTS needed are very similar to the ones needed for exporting with expdp. See Oracle USER configuration section.

impdp parameters explained

The impdp utility is symmetric to the expdp utility. So for example, let us see a quick example of how an import command would look like (changes are highlighted):

```
impdp IMP_USER/IMP_PWD@\"(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = W-EDE) (PORT=1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = EDE2)))\" DIRECTORY = DATA_PUMP_DIR DUMPFILE =
MEGADATA1.DMP LOGFILE = MEGADATA1.DMP.IMPORT.LOG SCHEMAS = MEGADATA1
```

Let us see the impdp specific parameters that can be used when the SCHEMA and the TABLESPACE are not the same.

Let us say that the origin SCHEMA is called **MEGADATA1** and the one available in the target Oracle environment is called **MEGAUSER**.

Also, the origin TABLESPACE is called **MEGA_TABLESPACE** and the TABLESPACE in the target Oracle environment is called **MEGAUSER_TBLSPC**.

REMAP_SCHEMA=

Used to tell Oracle that during the import, it should put the data extracted from **MEGADATA1** and put it into **MEGAUSER**

Example: REMAP_SCHEMA=source_schema:target_schema

REMAP_SCHEMA=MEGADATA1:MEGAUSER

See Oracle documentation for more details

REMAP_TABLESPACE=

Used to tell Oracle that during the import, it should put the data extracted from **MEGADATA1** that was in the TABLESPACE **MEGA_TABLESPACE** and put it into **MEGAUSER**'s TABLESPACE **MEGAUSER_TBLSPC.**

Example: REMAP_TABLESPACE=source_tablespace:target_tablespace

REMAP_TABLESPACE=MEGA_TABLESPACE:MEGAUSER_TBLSPC

See Oracle documentation for more details

Here is the command used (for a host called OHX03 and a service_name called PROD12)

```
impdp IMP_USER/IMP_PWD@\"(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = OHX03) (PORT=1521)) (CONNECT_DATA = (SERVER = DEDICATED)
(SERVICE_NAME = PROD12)))\" DIRECTORY = DATA_PUMP_DIR DUMPFILE =
MEGADATA1.DMP LOGFILE = MEGADATA1.DMP.IMPORT.LOG SCHEMAS = MEGADATA1
REMAP_SCHEMA= MEGADATA1:MEGAUSER REMAP_TABLESPACE= MEGA_TABLESPACE:
MEGAUSER_TBLSPC
```

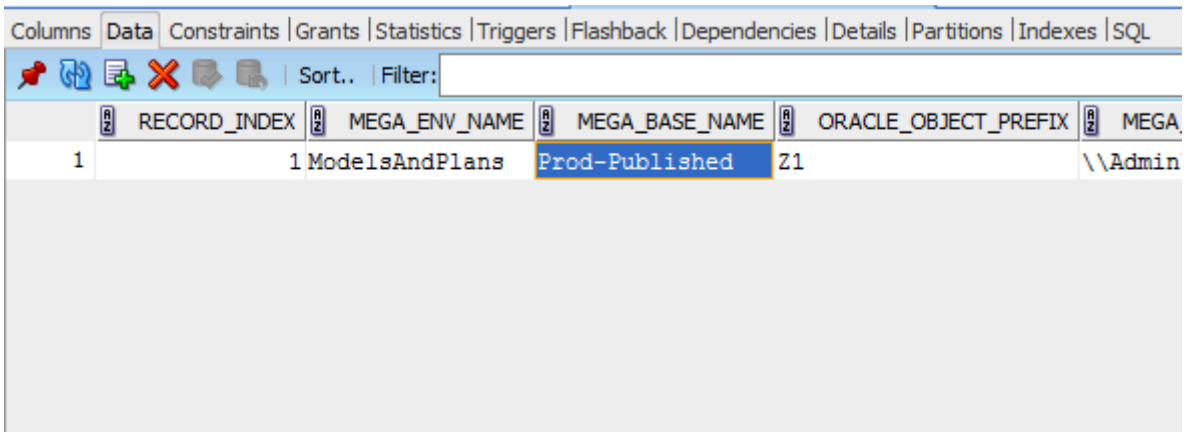
When duplicating a repository in an environment – Expert mode

This section is only if you are very confident about your Mega and RDBMS skills, since you have to modify data directly in some tables of your schema.

In case you want, within an environment, to duplicate a specific repository, you need to tweak the data manually in two tables of your restored schema.

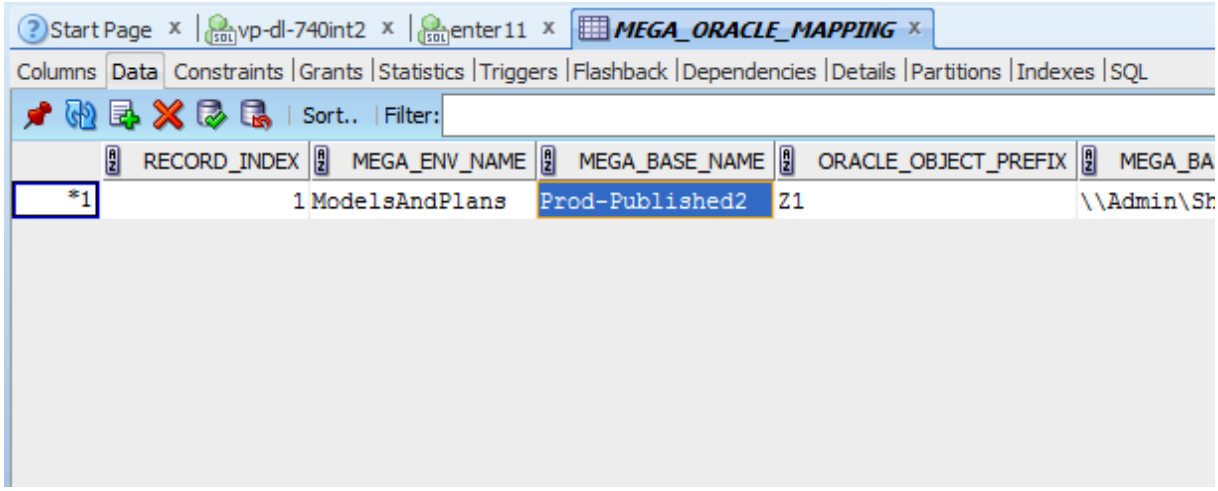
Proceed as follows:

1. Using for example SQL Developer, locate the "MEGA_ORACLE_MAPPING" table of your schema, and the line of your repository:

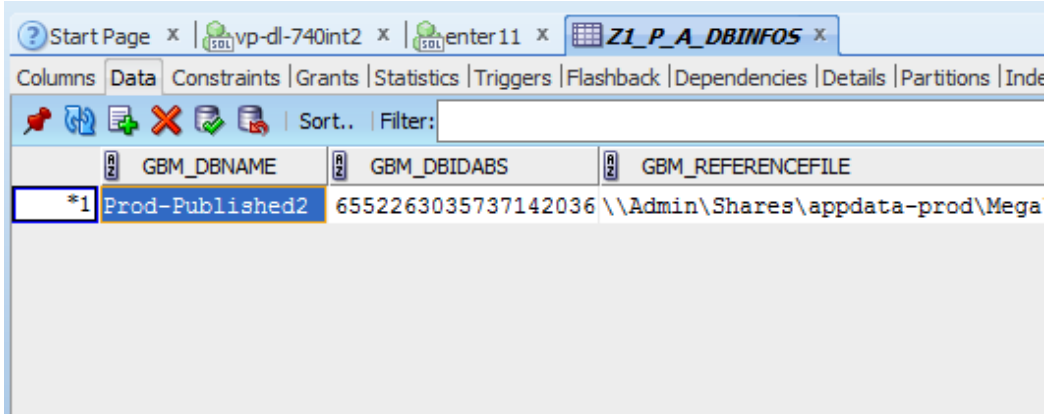


2. In the **MEGA_BASE_NAME** field of that line, enter the name of your duplicate, i.e. how you want it to appear in your environment.

   (for example here: "Prod-Published2")

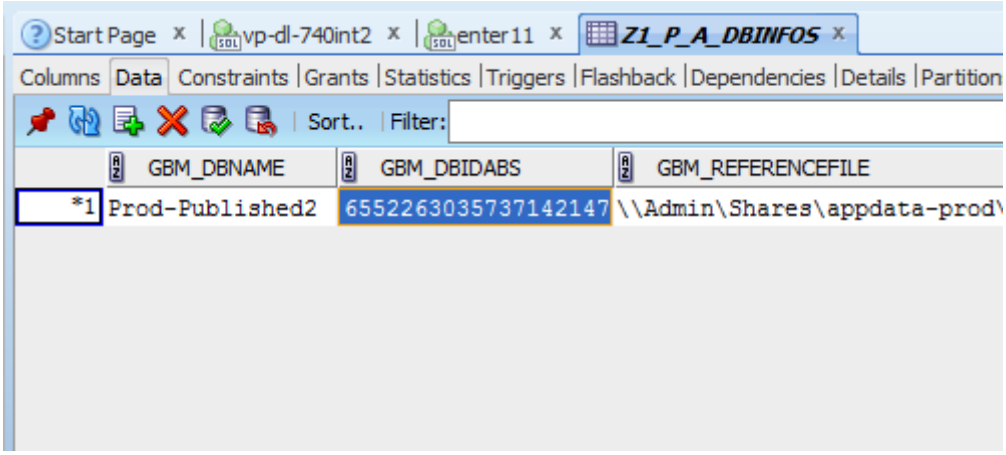


3. Commit the change and close the table.
4. Go to table that ends with "*_P_A_DBINFOS", '*' being the prefix of your repository (in this example, the prefix is "Z1", as shown in the previous screenshots), and locate the line of your repository.
5. In **GBM_DBNAME** field enter the same name as before (for example here: "Prod-Published2").



6. In **GBM_DBIDABS** field, which is the unique identifier of the repository within your environment, modify its value, and make sure that it is unique for all other repositories, so you need to check in all schemas.

   Make sure that when you change one or two characters, it will create a string that is not used by any other (for example here, we modified the last 3 digits from "036" to "147").



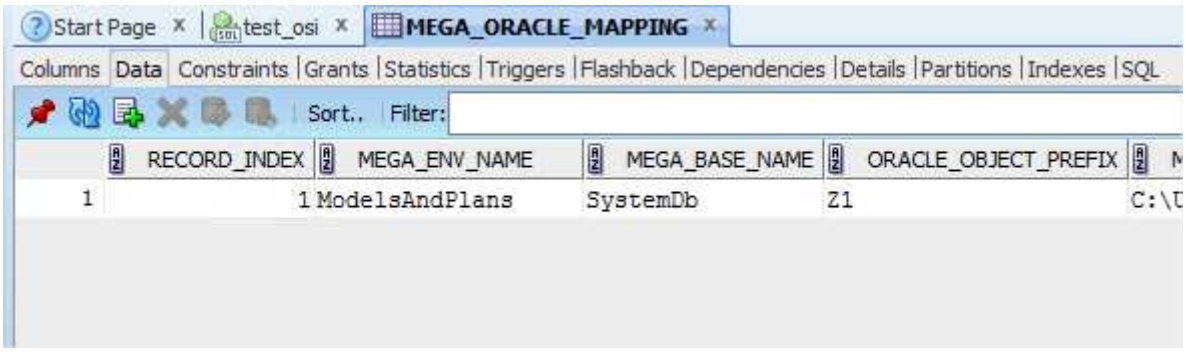
   Commit your updates and close the table.

When renaming an environment – Expert mode

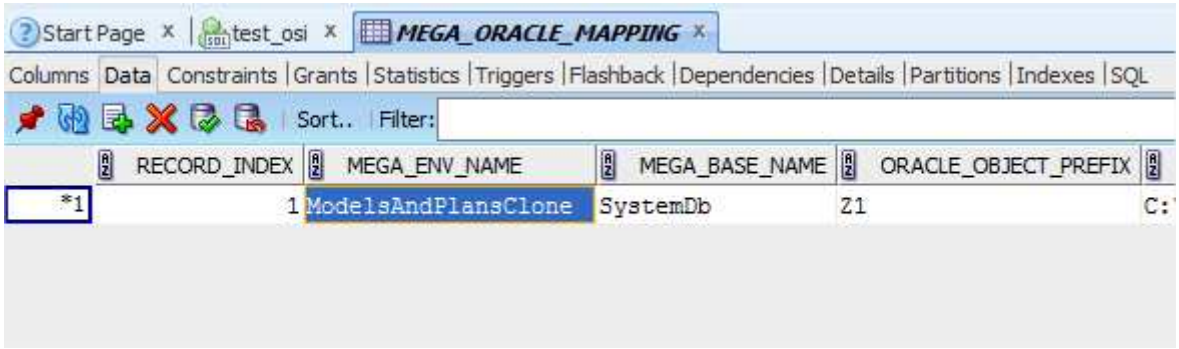
This section is only if you are very confident about your Mega and RDBMS skills, since you have to modify data directly in some tables of your schema.

In case you want to change the name of the environment after restore, to have something different from the source, you need to tweak the data manually in a table of your schema.

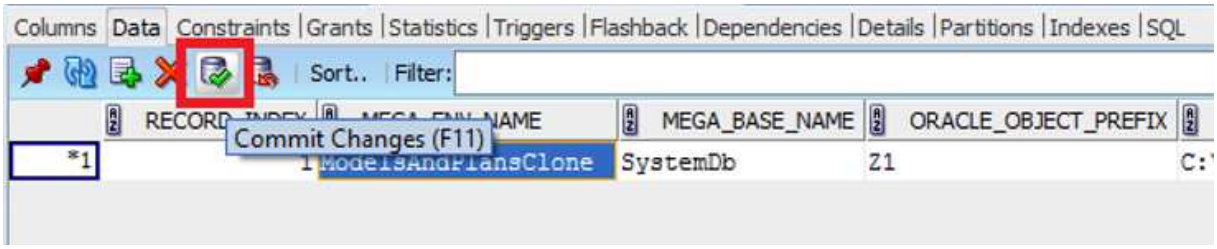
1. Using for example SQL Developer, locate the "MEGA_ORACLE_MAPPING" table of your schema, and the line of your repository:



2. In the **MEGA_ENV_NAME** field of that line, enter the name of the environment as you want it to appear in the application.
   (for example here: "ModelsAndPlansClone")



3. Commit your change:



4. Repeat those steps for all the repositories that could be stored in that schema. And for all schemas that contain repositories of this environment.

RDBMS Environment Duplication - 2 - Post Processing Tasks

INTRODUCTION

This document describes the tasks you need to cary out after duplicating the MEGA data that were stored in an SQL Server or an Oracle database.

You need to have gone through one of the following document first, depending on the RDBMS target:

- *RDBMS Environment Duplication - 1 - Oracle*

or

- *RDBMS Environment Duplication - 1 - SQL Server*

The first part details what to do to reattach to the set of duplicated data from MEGA point of view.

As all is not stored inside the RDBMS, this document details what you need to copy from the original MEGA environment to the new MEGA environment pointing at the duplicate RDBMS data.

Since the tasks that need to be carried out are almost exactly the same, this document does not differentiate between Oracle and SQL Server except for a few examples: the difference is highlighted in that case.

POST INSTALLATION TASKS

Create/Attach an environment in Mega

Creating an environment

To create an environment:

1. Connect to **MEGA Administration**.
2. In the navigation tree, right-click **Environments** and select **New**.
3. Enter the **Name** of the new environment.

   The environment name must respect Windows folder naming constraints.

   For SQL Server: It is the beginning of the name of the system database. It should be called <environment>_SystemDb, where <environment is the string that you put in the Mega admin tool. In this example, the environment is called "GGS".

   For Oracle: the environment name can be found in the MEGA_ORACLE_MAPPING table that can be found in the Oracle SCHEMA corresponding to the SystemDb of the environment. It will show in the MEGA_ENV_NAME column. See p.15 of "RDBMS environment duplication - 1 – Oracle" for more details.
4. (Optional) Location of the environment is specified by default; you can modify it if necessary using the **Browse** button.

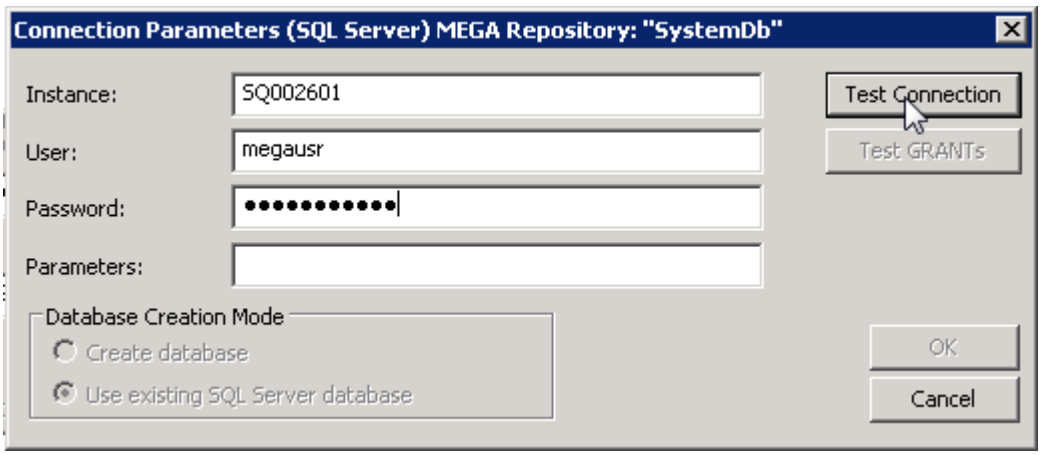
   In this example, we created a share on the server hosting the SQL Server instance, that will host the environment files, and that is called \\sq002601\EnvironnementsMega.
5. Select an RDBMS repository server type for the new environment: SQL Server or Oracle.
6. Select **Restore** which allows to connect to an existing MEGA formatted set of data on an instance,
7. Click **OK**.
8. Enter the connection parameters for the **Instance**.

   Do not enter anything in the **Parameters** field.

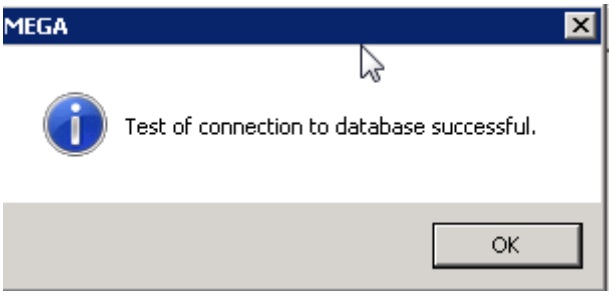
   Note:

   The syntax for a named instance on **SQL Server** is : server_name\instance_name. In this example, the instance is the default one, without a specific name. That is why we only provide the name of the server where the instance is running.

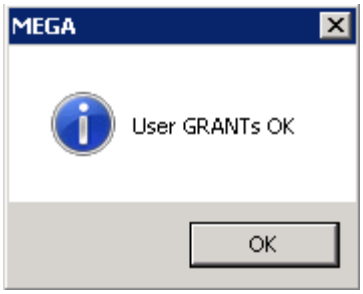
   For **Oracle**, the backslash needs to be replaced with a forwardslash server_name/instance_name.



9. Click **Test Connection**.

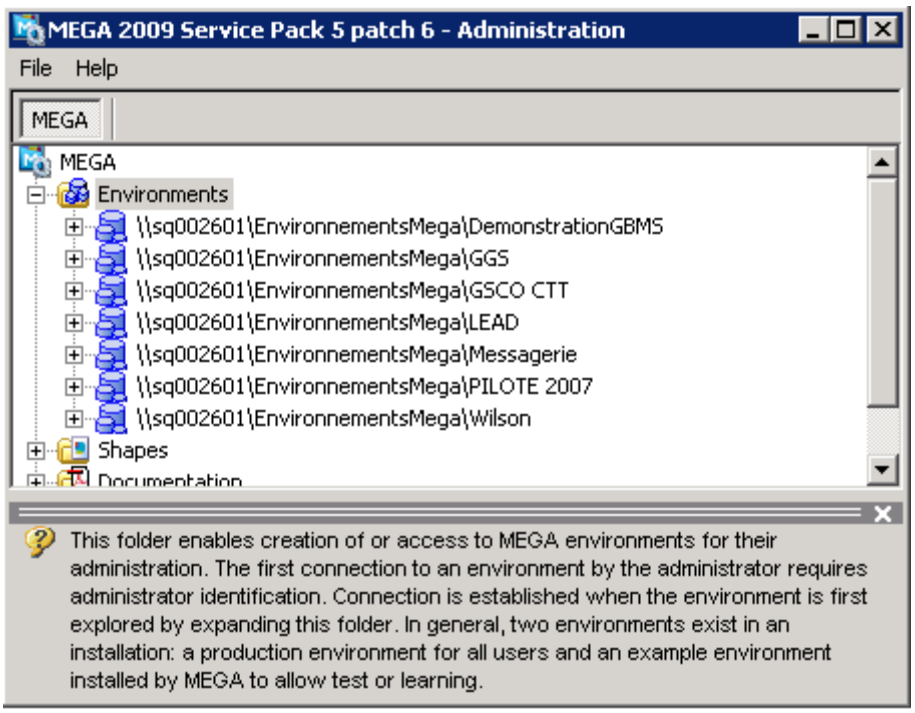


10. Click **Test GRANTS**.



11. Click **OK**.

A popup indicating that the environment was successfully created appears, and the new environment is displayed in the list (for example here « \\sql002601\EnvironmentsMega\GGS).



Attaching the working database(s) to the environment

Once your environment is created, you still have to attach the working database(s) to this environment.

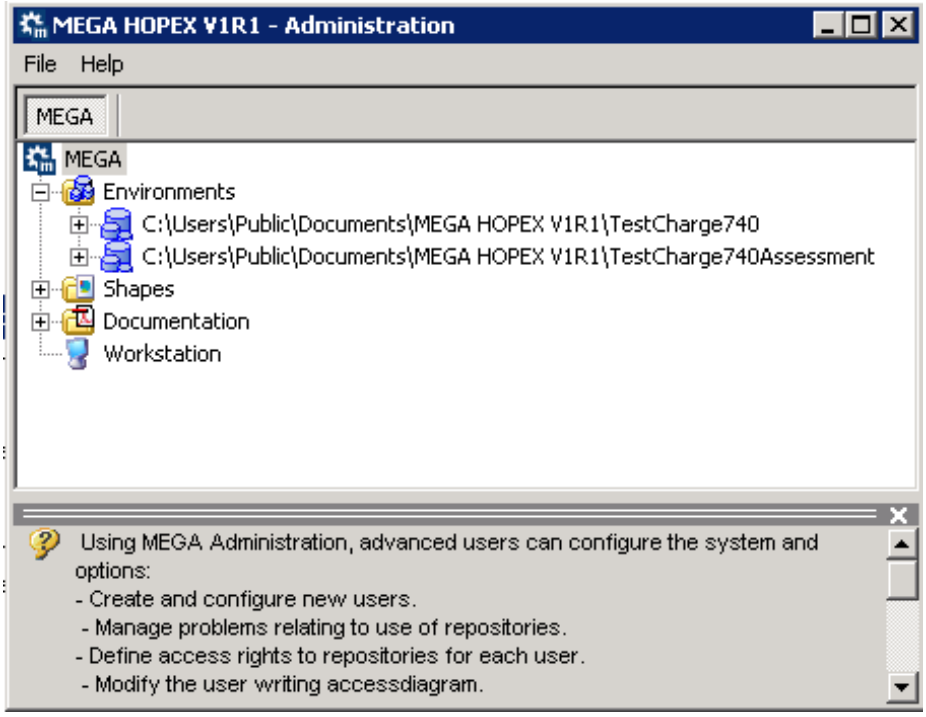
In **Repositories** you only have the SystemDb.

Now you have to connect to the environment, and add the Mega repositories. The first time you connect, you receive warning message(s) telling you that database "X" ("X" being the name of the repository that existed in the source environment, and most likely one of the repositories you are trying to attach) is not referenced.

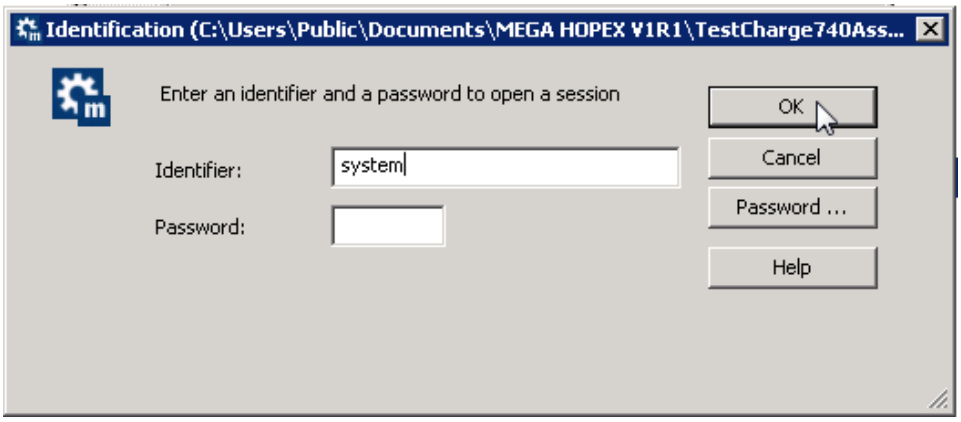
This is because the SystemDb database contains some information about the working repositories. You can discard this warning, and continue.

To attach a database, previously restored in SQL Server Management Studio:

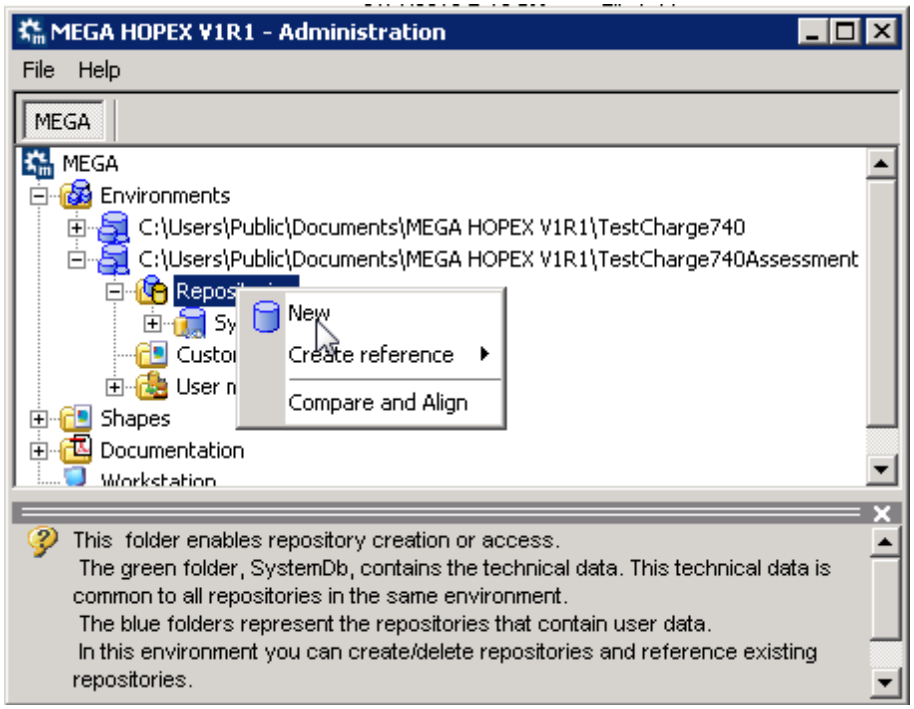
1. Connect to **MEGA Administration**.



2. Connect to the environment concerned (for example with **System** user).



3. Right-click **Repositories** and select **New**.

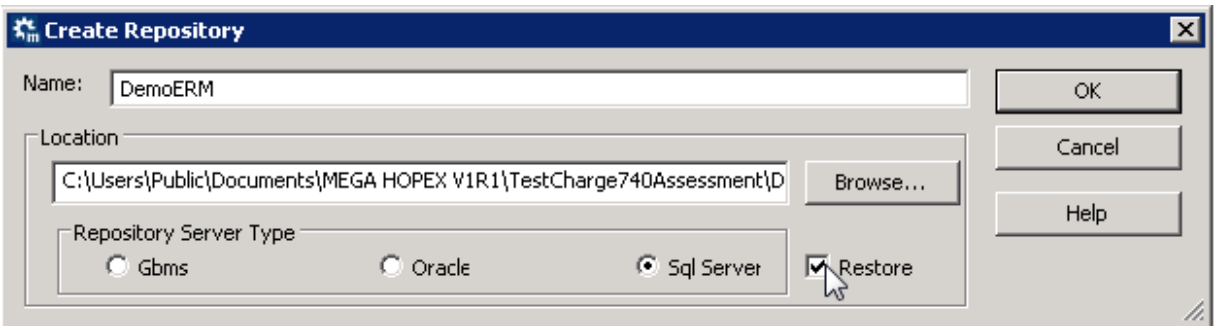


4. Enter the Name of the Repository (in this example erver rmati

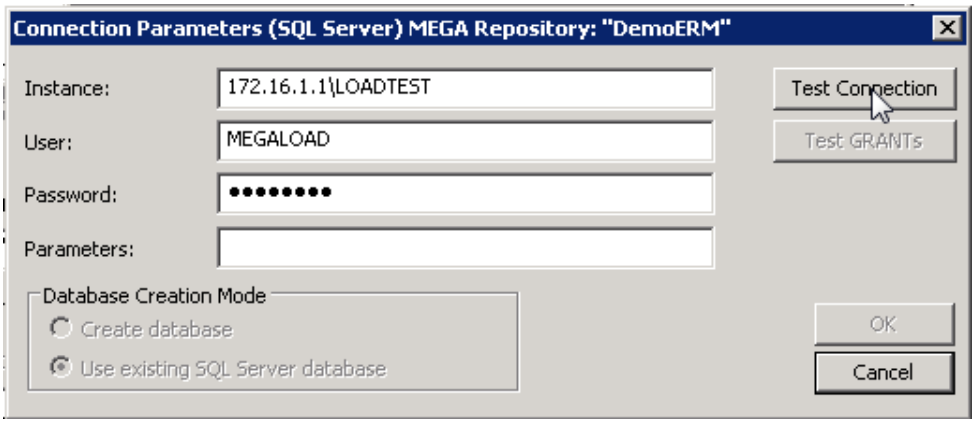
   In **SQL Server**, you should have a database called this example erver rmat_DemoERM" existing, with the native SQL Server user db_owner of the database.

   In **Oracle**, you need to know which SCHEMA is "CHEMA is to know ser dbase called this example erver rmation about the working repositories. You can discard this warning, and continue.

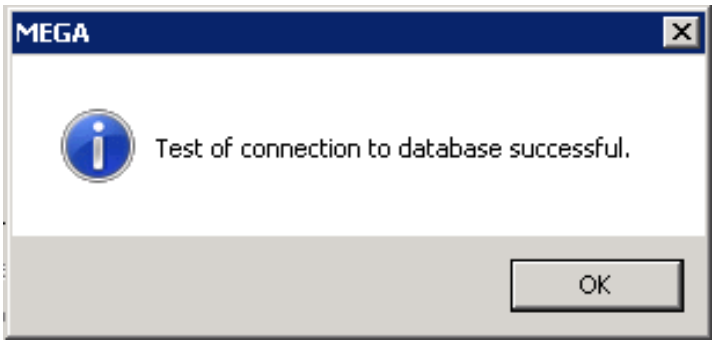
5. In the **Repository Server Type** pane, select **SQL Server** or **Oracle**.
6. Select **Restore**.



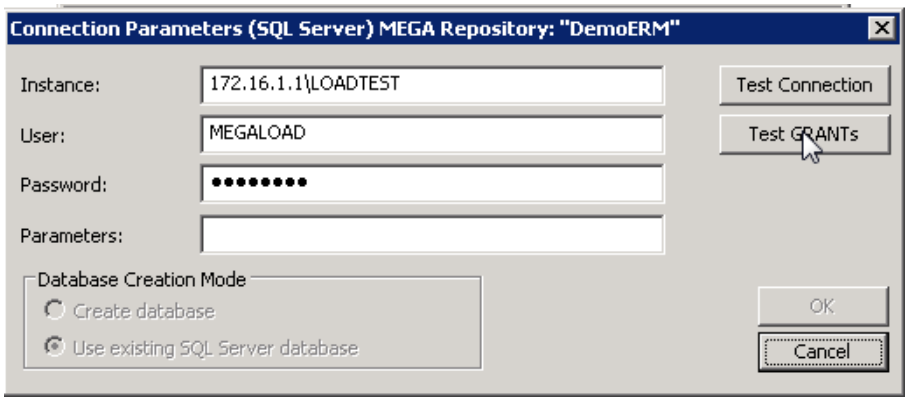
7. Click **OK**.
8. The connection parameters are already set. Check that they are correct.

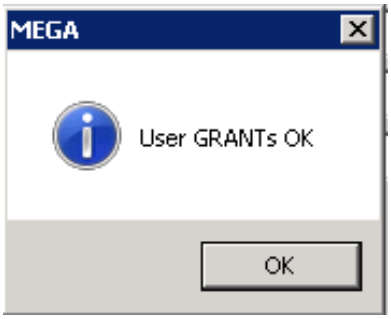


9. Click **Test Connection**.

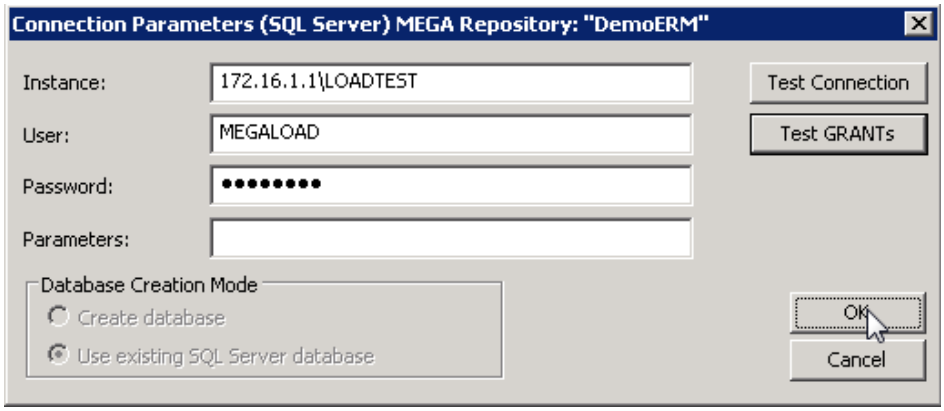


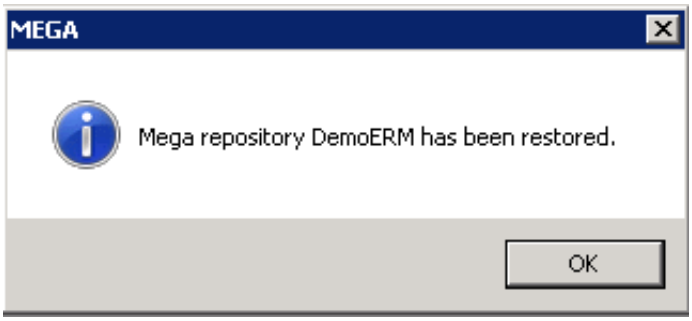
10. Click **Test GRANTs**.



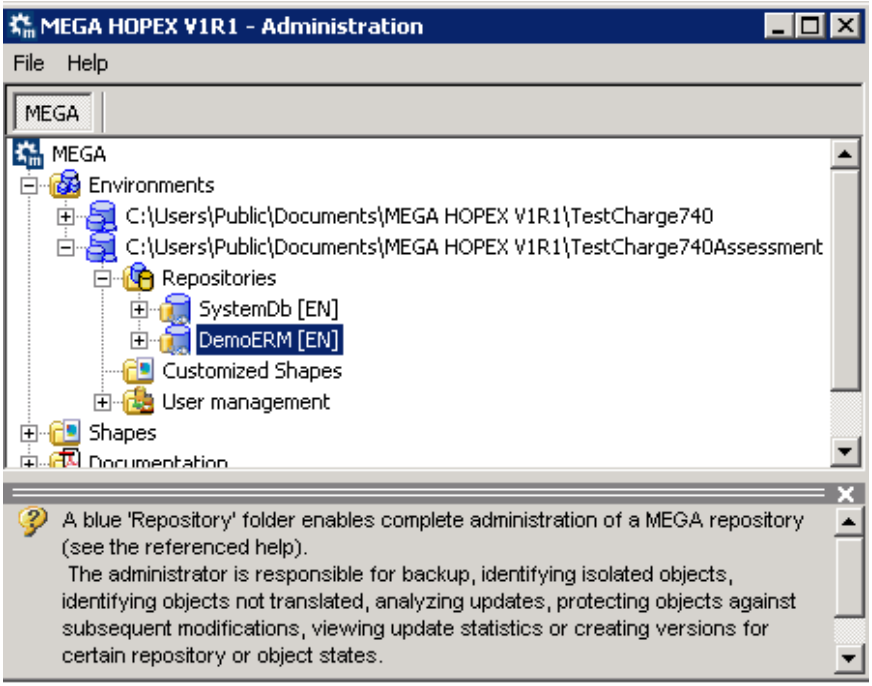


11. Click **OK** to create the repository.





The "DemoERM" repository appears in the **Repositories** list



When duplicating a repository

In case you are also creating a clone of an existing repository within your environment, you made preliminary steps while restoring either your databases (in SQL Server), or schemas (in Oracle).

Since this is done, you just have to perform again the Attaching the working database(s) to the environment procedure, and adapt it to give the name of your duplicated repository.

In the above example, you have "DemoERM". So we assume that you have also a database for the clone of the repository that would be called "DemoERM2" for instance.

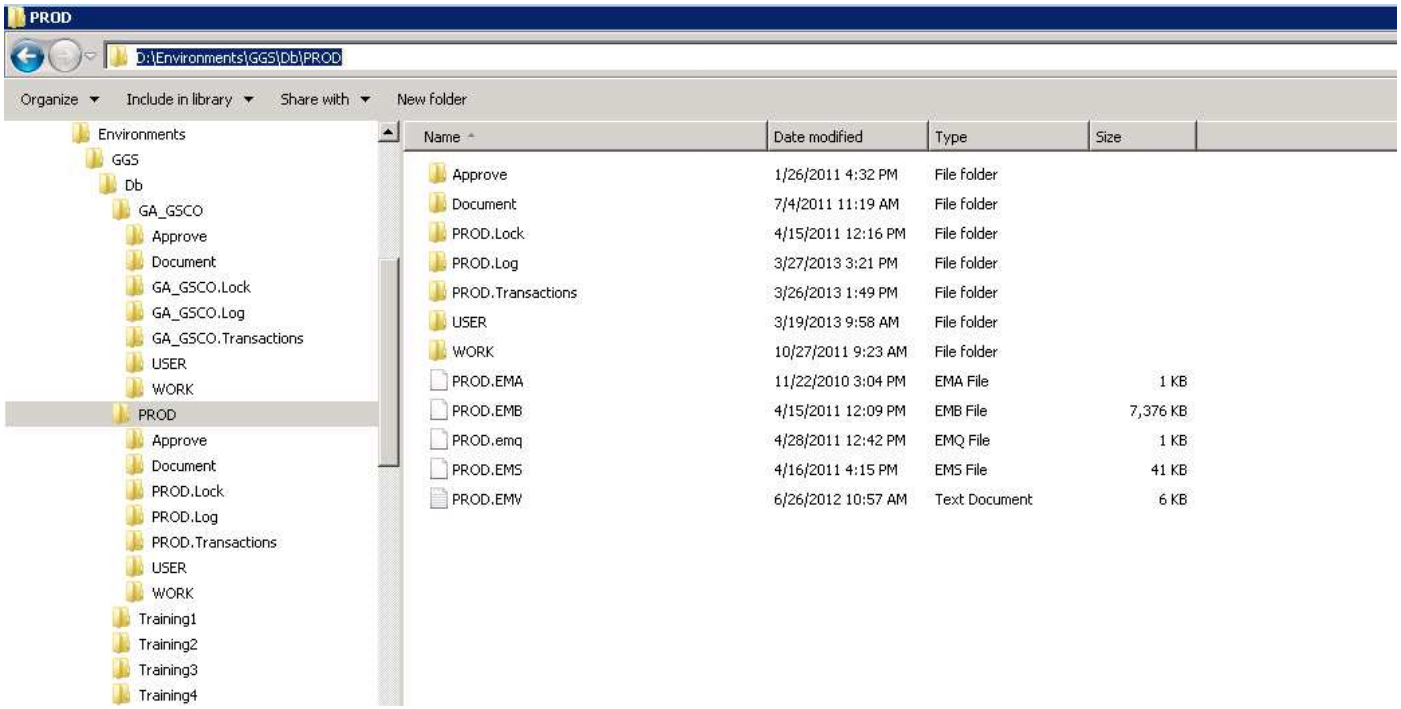
The same actions should allow you to add this copy in your environment, without any impact on the "DemoERM" repository.

Copy the documents from source to target

You have to copy the documents from source to target for each repository.

The Word/RTF documents

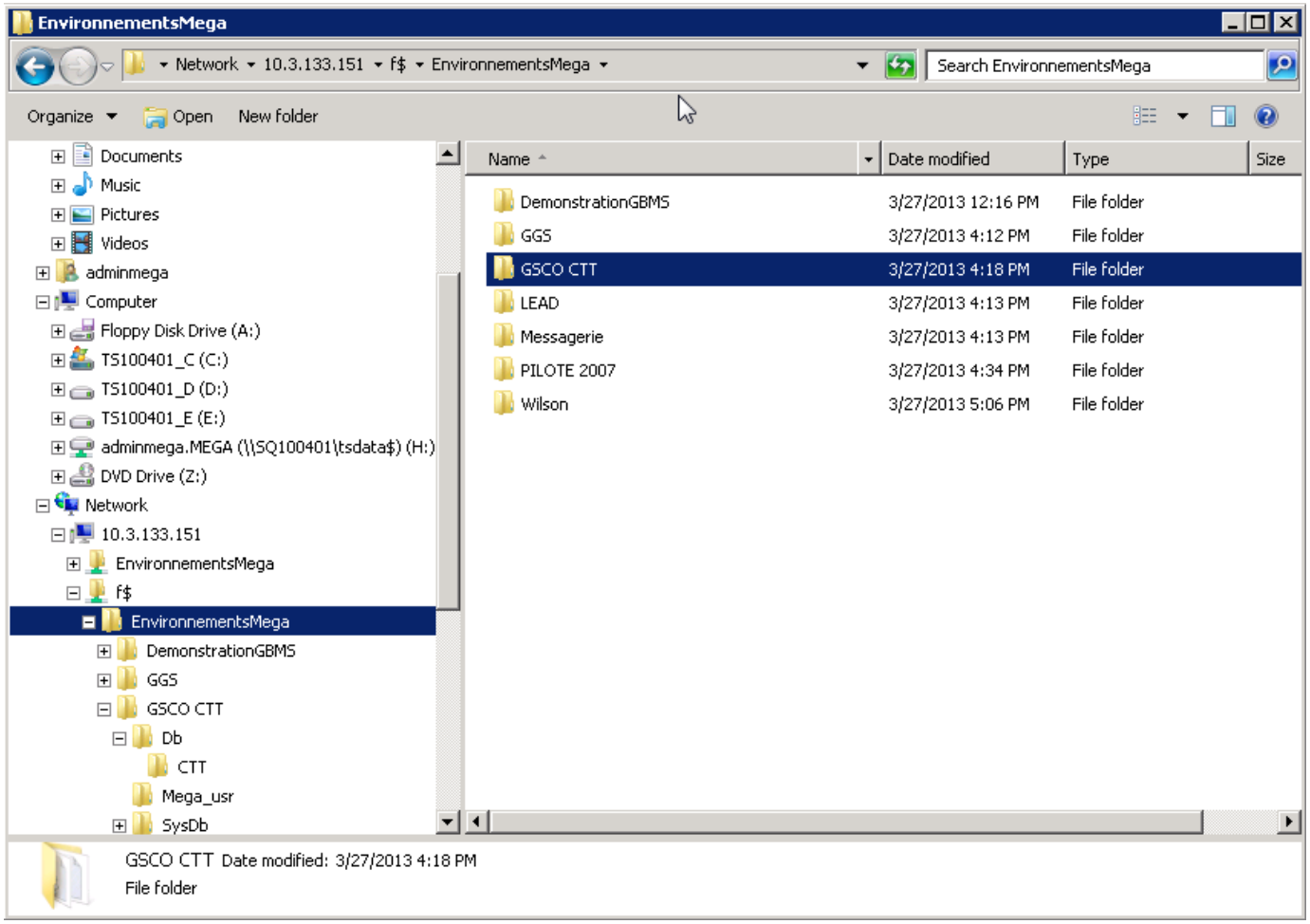
1. Connect to the source server.
2. Go to the folder hosting the environment (for example environment "GSCO CTT").
3. Go to the **Db** folder, and in the sub-folder of each repository you migrated (in this example « D:\Environments\GGS\Db\PROD » contains the data of the repository called "CTFTM_OLD").



4. From the source server, open an explorer on the target server, in the same folder.

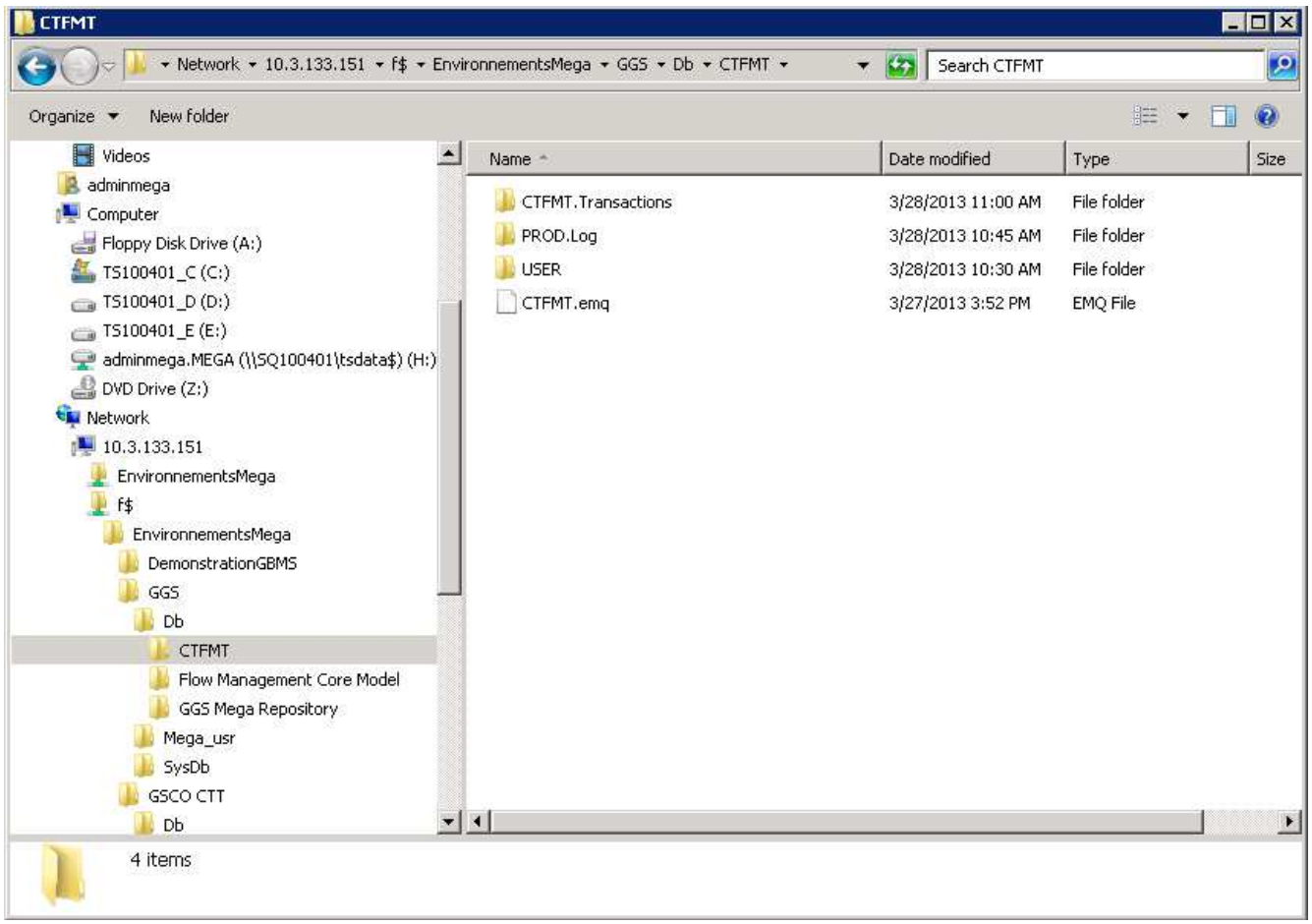
   In this example, the environments on the target server are hosted on the server with 10.3.133.151 IP address, and on that server the environments are located on the F drive in the "EnvironnementsMega" folder, so that the syntax is:

   \\10.3.133.151\f$\EnvironnementsMega\

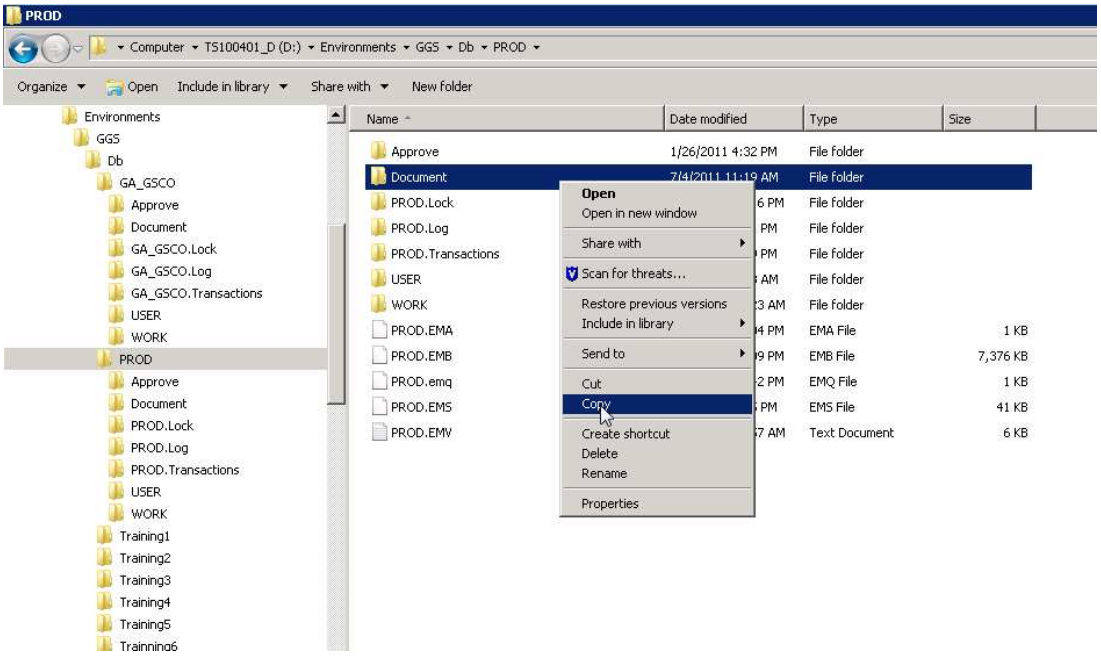


5. Go to the same sub-folder « …\GGS\Db\CTFMT ».

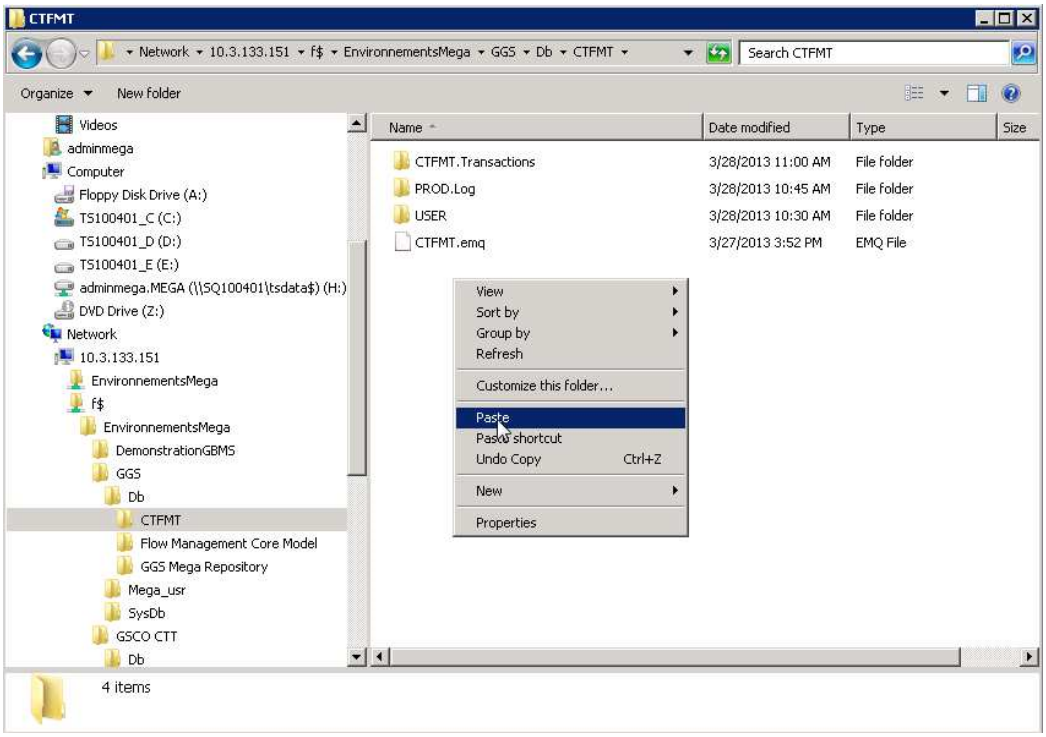
   Note: If you are wondering why it is not the same environment name and the same repository name, that is because in this example, this repository was taken from one environment to another, with a rename.



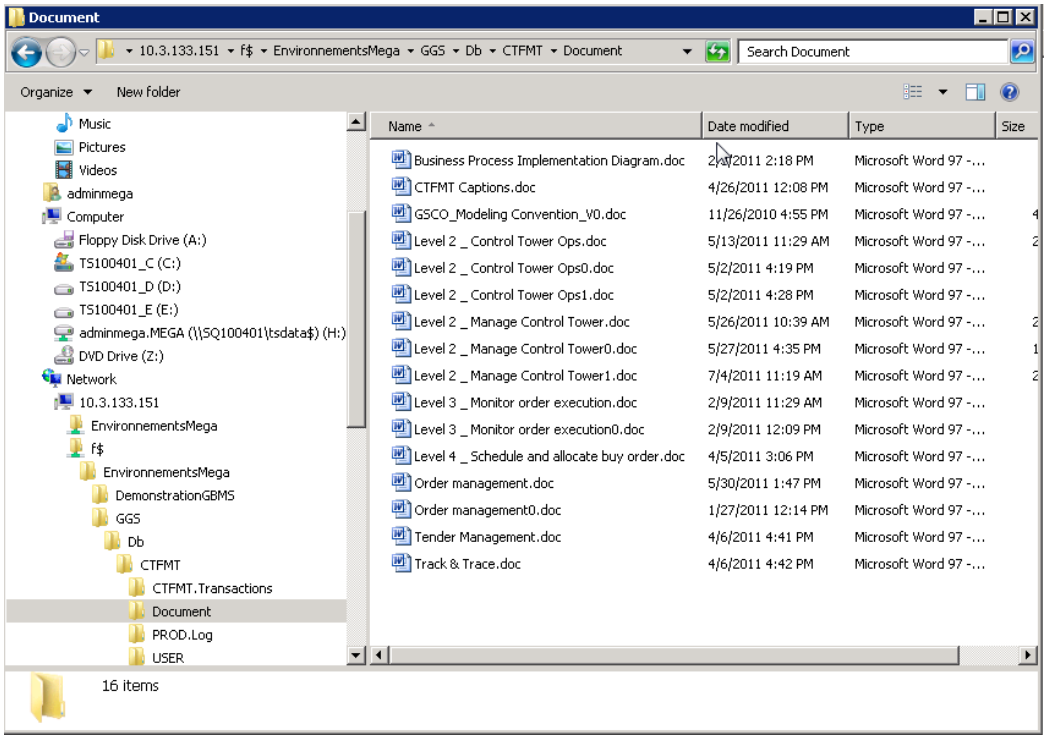
6. Right-click the **Document** folder from the source and select **Copy**.



7. Paste it on the target (future Production).



8. Check that the documents are all in the target folder.



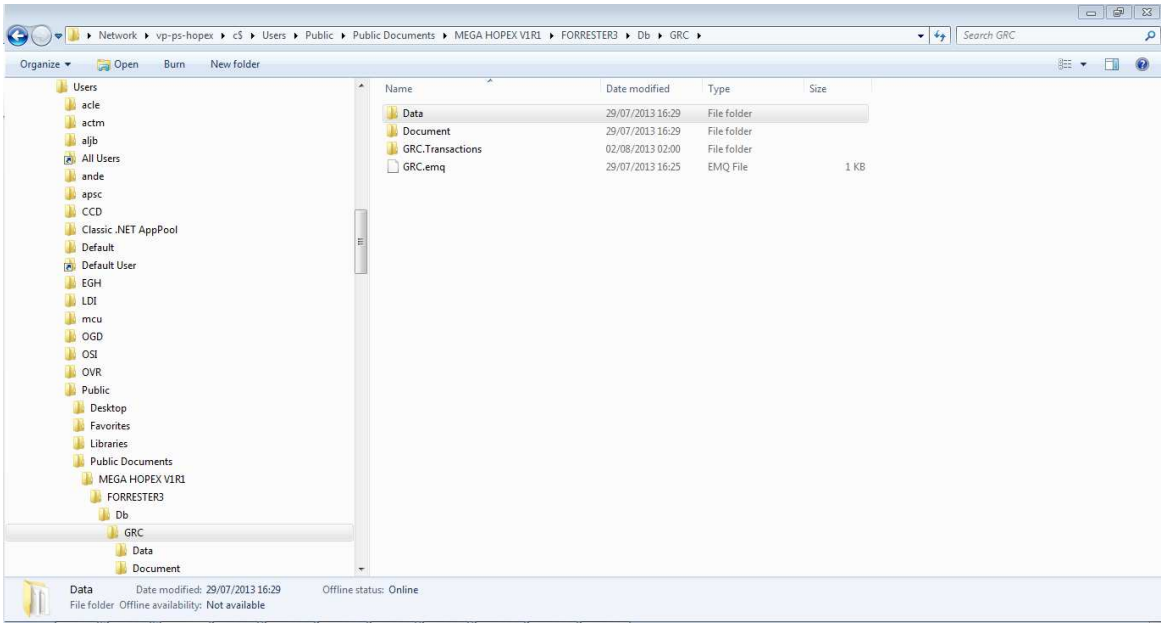
The internal documents (.DAT files)

Some documents can also be stored partially inside the SQL Server/Oracle databases (as references) and partially in .DAT files (generated in the folder of each repository of the migrated environment).

By restoring the SQL Server/Oracle databases, you already retrieved the references. Now you have to copy a folder from source to target.

1. From the source environment, expand the **Db** folder and the sub-folder of the database.

   In this example, on server vp-ps-hopex, we are looking at environment py aRESTER3 from source to target.the migrated environment).ry name, that is because in tX V1R1\FORRESTER3\Db\GRCxa



2. Right-click the **Data** folder and select **Copy**.
3. Paste it at the same level on the target folder.

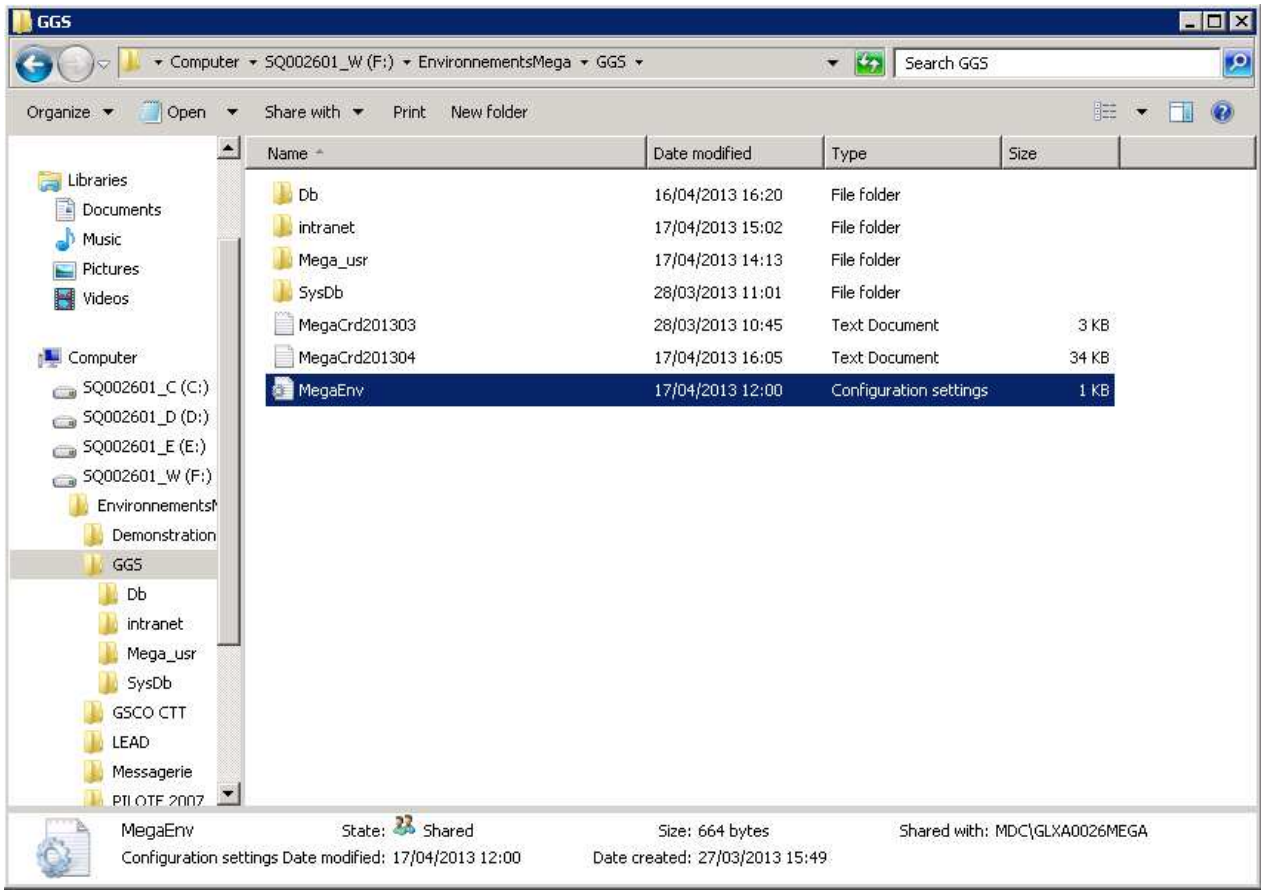
Get the parameters of the environment

For each environment, you need to retrieve certain types of configuration included in:

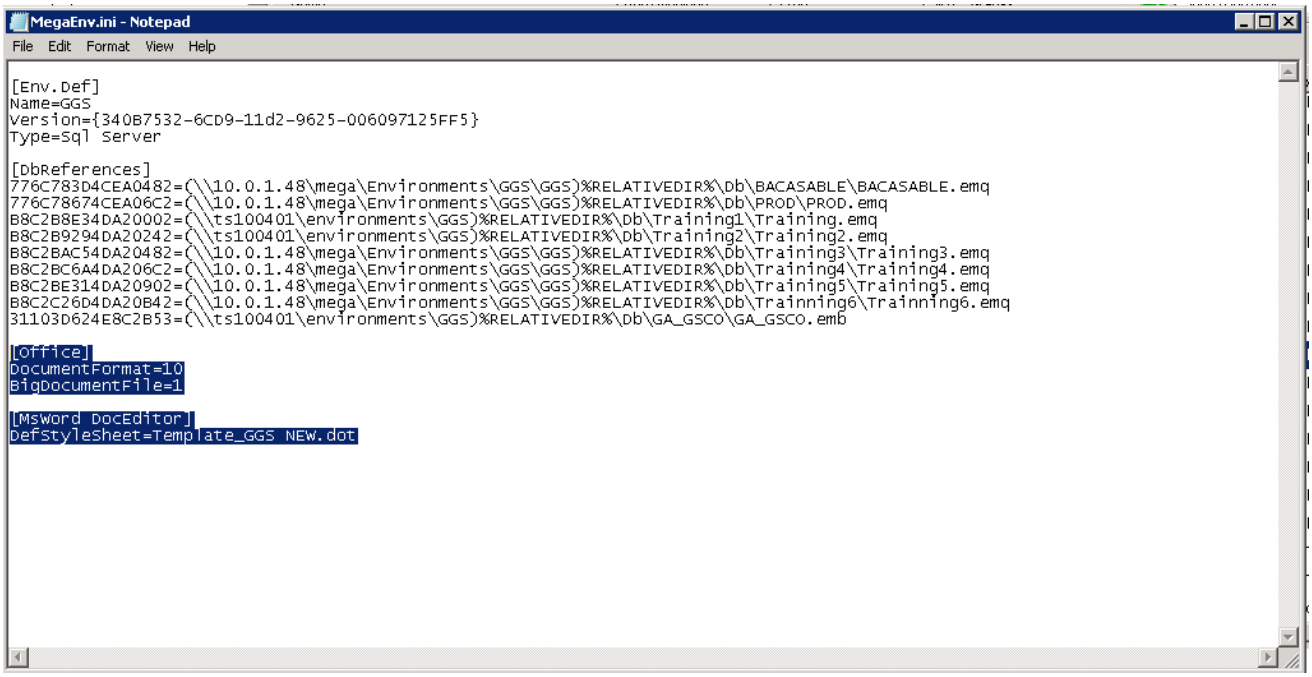
- the **MegaEnv.ini** file
- the **Mega_Usr** folder.

MegaEnv.ini file

The MegaEnv.ini file is located at the root level of each environment (in this example, in the folder tione database. In thga\GGSy afor the GGS environment) on the source server.



1. Open the **MegaEnv.ini** file on the source server.

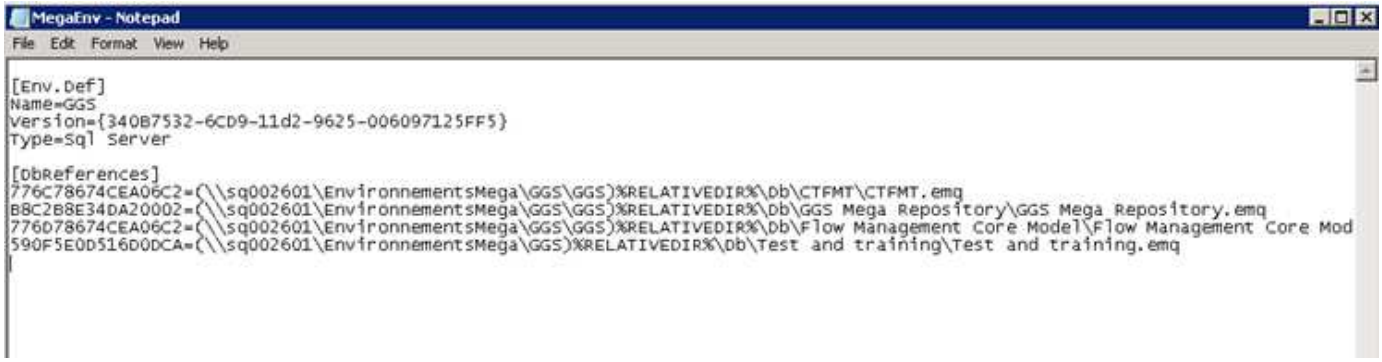


```
MegaEnv.ini - Notepad
File  Edit  Format  View  Help

[Env.Def]
Name=GGS
Version={340B7532-6CD9-11d2-9625-006097125FF5}
Type=Sql Server

[DbReferences]
776C783D4CEA0482=(\\10.0.1.48\mega\Environments\GGS\GGS)%RELATIVEDIR%\Db\BACASABLE\BACASABLE.emq
776C78674CEA06C2=(\\10.0.1.48\mega\Environments\GGS\GGS)%RELATIVEDIR%\Db\PROD\PROD.emq
B8C2B8E34DA20002=(\\ts100401\environments\GGS)%RELATIVEDIR%\Db\Training1\Training.emq
B8C2B9294DA20242=(\\ts100401\environments\GGS)%RELATIVEDIR%\Db\Training2\Training2.emq
B8C2BAC54DA20482=(\\10.0.1.48\mega\Environments\GGS\GGS)%RELATIVEDIR%\Db\Training3\Training3.emq
B8C2BC6A4DA206C2=(\\10.0.1.48\mega\Environments\GGS\GGS)%RELATIVEDIR%\Db\Training4\Training4.emq
B8C2BE314DA20902=(\\10.0.1.48\mega\Environments\GGS\GGS)%RELATIVEDIR%\Db\Training5\Training5.emq
B8C2C26D4DA20B42=(\\10.0.1.48\mega\Environments\GGS\GGS)%RELATIVEDIR%\Db\Trainning6\Trainning6.emq
31103D624E8C2B53=(\\ts100401\environments\GGS)%RELATIVEDIR%\Db\GA_GSCO\GA_GSCO.emb

[Office]
DocumentFormat=10
BigDocumentFile=1

[MsWord DocEditor]
DefStyleSheet=Template_GGS NEW.dot
```

2. Under the [DbReferences] section, you find the environment-specific parameters, like the type of document (DocumentFormat=10 meaning that they were converted from .DOC to .RTF) or the templates used by default.

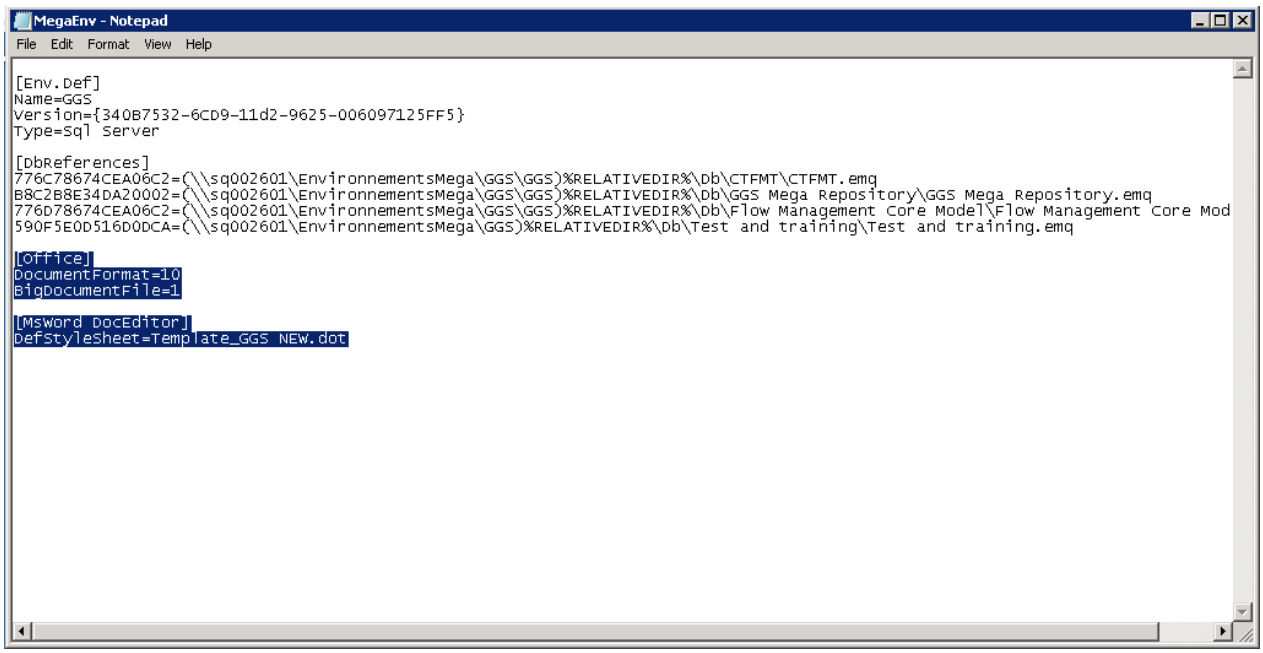
   Copy this specific section.

3. Open the **MegaEnv.ini** of the target server.



```
MegaEnv - Notepad
File  Edit  Format  View  Help

[Env.Def]
Name=GGS
Version={340B7532-6CD9-11d2-9625-006097125FF5}
Type=Sql Server

[DbReferences]
776C78674CEA06C2=(\\sq002601\EnvironnementsMega\GGS\GGS)%RELATIVEDIR%\Db\CTFMT\CTFMT.emq
B8C2B8E34DA20002=(\\sq002601\EnvironnementsMega\GGS\GGS)%RELATIVEDIR%\Db\GGS Mega Repository\GGS Mega Repository.emq
776D78674CEA06C2=(\\sq002601\EnvironnementsMega\GGS\GGS)%RELATIVEDIR%\Db\Flow Management Core Model\Flow Management Core Mod
590F5E0D516D0DCA=(\\sq002601\EnvironnementsMega\GGS)%RELATIVEDIR%\Db\Test and training\Test and training.emq
```

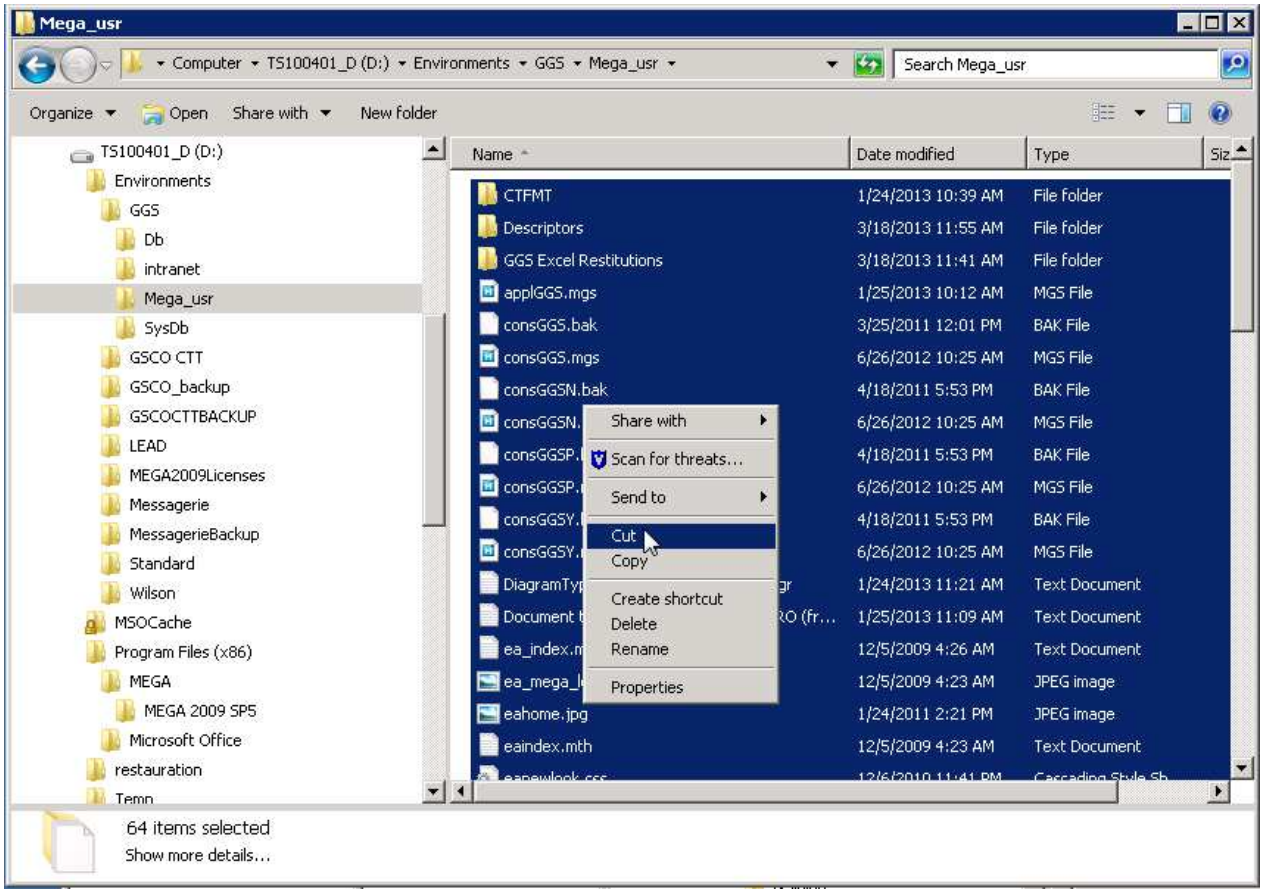
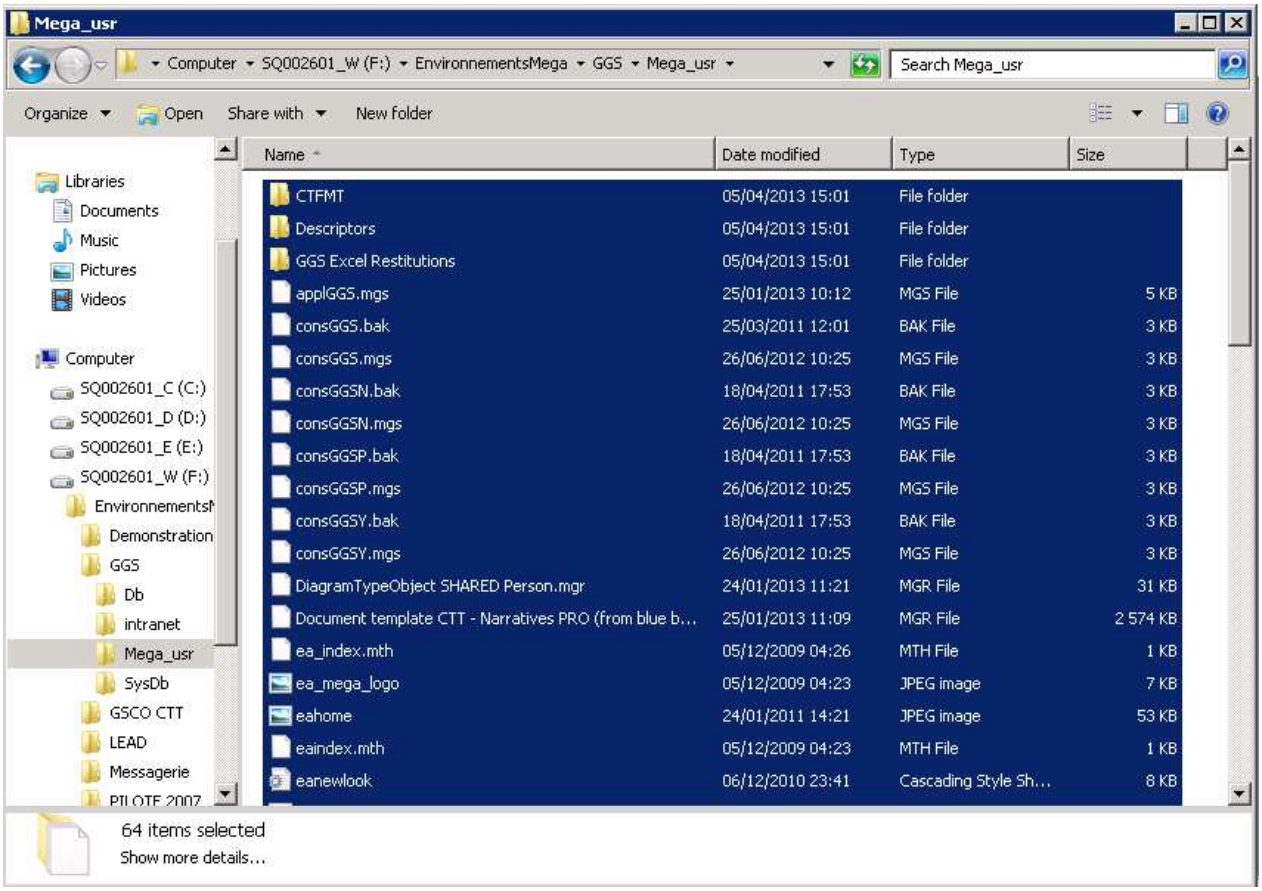
4. Paste this section at the same level in the file and save the document.



Mega_Usr folder

In each environment, there is a **Mega_Usr** ega_Usr, there is a the same level in the file and save tese are environment-specific and correspond to objects in the Mega respositories. It is mandatory to transfer these if you want to keep your customizations.

1. In the source server, go to the root level of your environment.
2. Copy everything included in the **Mega_Usr** folder.



3. Go to the target server in the folder of the duplicated environment.
4. Paste everything in the **Mega_usr** folder.