

# Natural Runtime

Natural Runtime is used to execute applications that have been written using the development version of Natural for Windows.

This chapter tells you how to port an application from a Natural development workstation to a Natural Runtime workstation. This porting process can be used for a first-time installation of Natural Runtime and for Natural Runtime workstation updates.

You will also learn how to use a service for starting Natural Runtime processes.

The following topics are covered:

- What is not Supported by Natural Runtime?
- Porting Procedure Overview
- Step 1: Packaging the Application on the Development Workstation
- Step 2: Installing Natural Runtime
- Step 3: Installing the Application on the Runtime Workstation
- Step 4: Starting the Application on the Runtime Workstation
- Using the Natural Runtime Startup Service

## **Important:**

Before porting an application to a Natural Runtime workstation, ensure that all objects have been compiled using identical Natural and Natural Runtime versions.

---

## What is not Supported by Natural Runtime?

### System Commands

The following Natural system commands are not supported by Natural Runtime:

CATALL  
CATALOG  
CHECK  
CLEAR  
COMPOPT  
DEBUG  
DELETE  
EDIT  
GLOBALS  
PURGE  
READ  
RENUMBER  
RUN  
SAVE

SCAN  
SCRATCH  
STOW  
STRUCT  
SYSDDM  
SYSMAIN  
UNCATALOG  
UNLOCK

## Editors

Natural editors are not supported by Natural Runtime.

## Utilities

Natural utilities providing developer functionality are not supported by Natural Runtime.

# Porting Procedure Overview

To port an application to a runtime workstation, the following steps (which are described in detail later in this section) are required:

1. Package the application on the development workstation:
  1. Create a collecting directory in your file system.
  2. Customize the global configuration file and copy it to the collecting directory.
  3. Customize the Natural parameter file and copy it to the collecting directory.
  4. Copy or unload all required objects to the collecting directory.
  5. Copy the contents of the collecting directory to a transfer medium (for example, to a CD).
2. Install Natural Runtime on the runtime workstation.
3. Install the application on the runtime workstation:
  1. Copy the global configuration file from the transfer medium to the runtime workstation.
  2. Copy the Natural parameter file from the transfer medium to the runtime workstation.
  3. Copy or load the Natural objects from the transfer medium to the runtime workstation.
4. Start the application on the runtime workstation.

See also *Transferring Natural Generated Programs* in the *Programming Guide*.

# Step 1: Packaging the Application on the Development Workstation

The following topics are covered below:

- Creating a Collecting Directory
- Customizing and Copying the Global Configuration File
- Customizing and Copying the Natural Parameter File
- Copying or Unloading the Objects
- Copying the Collecting Directory to a Transfer Medium

## Creating a Collecting Directory

Use the Windows Explorer to create a new directory in the file system of the development workstation. Use this temporary directory to collect all files which belong to the application.

## Customizing and Copying the Global Configuration File

You have to create a global configuration file which contains all settings required to run your application on the runtime workstation. To do so, you create a backup version of your current global configuration file, make all required changes for the runtime version, copy the customized global configuration file to the collecting directory and then restore your old global configuration file. This is described in detail below.

### To customize and copy the global configuration file

1. Use the Windows Explorer to back up the existing global configuration file.

#### **Note:**

If you do not know where to find the global configuration file, invoke the Configuration Utility, expand the **Local Configuration File** node and select the **Installation Assignments** node. The full path and name of the global configuration file is then shown.

2. Invoke the Configuration Utility and expand the **Global Configuration File** node.
3. Adjust the settings of the global configuration file as required for your application and save your changes. See also the notes below.

#### **Important:**

As long as the global configuration file with the settings for the runtime environment is active, you cannot work with the development version of Natural.

4. Use the Windows Explorer to copy the customized global configuration file to the collecting directory.
5. Use the Windows Explorer to restore the backup version of the global configuration file.

**Notes:**

- The default paths to the FNAT and FUSER system files differ in the development and runtime environments. To simplify the installation of the application in the runtime environment, adjust the paths in this step: select the **System Files** node of the global configuration file and change the FNAT and FUSER paths to the settings of the runtime version.

Example: Change the following default paths of a Natural development version installation:

DBID	FNR	Path
99	100	<i>C:\Program Files\Software AG\Natural\<n.n&gt;\fnat< i=""></n.n&gt;\fnat<></i>
99	101	<i>C:\Program Files\Software AG\Natural\Natapps\Fuser</i>

to the following default paths of a Natural Runtime installation:

DBID	FNR	Path
99	100	<i>C:\Program Files\Software AG\Natural Runtime\<n.n&gt;\fnat< i=""></n.n&gt;\fnat<></i>
99	101	<i>C:\Program Files\Software AG\Natural Runtime\Natapps\Fuser</i>

where *n.n* stands for the version number.

- Make sure that with every new application you are porting, the new settings are compatible with the old settings.

Example: Your first application accesses an SQL database and the DBID entry applies to this SQL database. Your second application, which you are porting at a later date, accesses an Adabas C database. In this case, you must add a second DBID entry for Adabas C. If you do not add a second entry, the new global configuration file will overwrite the SQL database's DBID and your first application will no longer be able to access its database.

## Customizing and Copying the Natural Parameter File

You have to create a Natural parameter file which contains all settings required to run your application on the runtime workstation.

### To customize the Natural parameter file

1. Invoke the Configuration Utility and expand the node for the required parameter file.
2. Adjust the settings of the parameter file as required for your application to run in the runtime environment.

Ensure that the parameter file contains the name of the program to be started. The examples below show different possibilities for this purpose.

#### **Example 1:**

Parameter	Required Setting
AUTO	Must be set to "ON" so that an automatic logon is executed at the start of the Natural session.
INIT-LIB	The name of the library into which the application is to be moved.
STARTUP	The name of the program that is to be started.
USER	The default user ID that is to be set when Natural is started.

**Example 2:**

Parameter	Required Setting
STACK	Must contain the library and the program to be started. For example:  LOGON MYLIB;EXECUTE MYAPP

3. Save the modified parameter file with the name that you want to use in the runtime environment (for example, with the name RUNPARAM).
4. Use the Windows Explorer to copy the customized parameter file (which has the extension "SAG") to the collecting directory.

**Note:**

If you do not know where to find the parameter file, expand the **Local Configuration File** node of the Configuration Utility and select the **Installation Assignments** node. The location of the Natural parameter files is then listed as **Path to parameter**.

## Copying or Unloading the Objects

To make compiled code available with Natural Runtime, you have to copy the cataloged objects from the Natural development environment to the runtime environment.

If the Natural application consists of complete Natural libraries, you can copy the libraries with the copy-and-paste functionality of the Windows Explorer.

Another way for porting the objects is use the Object Handler for unloading the objects in the Natural development environment and for loading them in the runtime environment.

### To copy the objects

1. Use Natural Studio to create a new library which is to contain all objects for the runtime version.

**Important:**

If the application consists of more than one library, create a new library for each library that is used by the application and proceed as described below.

2. Use Natural Studio to copy all cataloged objects, resources and error messages from the development library to the new library. Do not copy the sources.
3. Use the Windows Explorer to copy the entire directory which corresponds to the new library (including the file *FILEDIR.SAG* and the subdirectories *GP*, *RES* and *ERR*) to the collecting directory.

**Notes:**

1. If you do not know where to find this directory, execute the system command `SYSPROF` in Natural Studio. The **Files in File System** tab of the resulting dialog box shows the path to the directory that has been created for the system file `FUSER`. Your new library is a subdirectory of the `FUSER` directory; it has the same name as defined in Natural Studio.
2. You can also find out the path to the `FUSER` directory by using the Configuration Utility: select the parameter file that you have created in a previous step (that is: the parameter file that will be used to start the application in the runtime environment), expand the **Natural Execution Configuration** node and select the **System Files** node. The path to the `FUSER` directory is shown on the **FUSER** tab.
4. If required, rename the copied directory in in the collecting directory: enter the name of the library that is to be used in the runtime environment.

 **To unload the objects**

1. From the **Tools** menu, choose **Development Tools > Object Handler** to start the Object Handler.
2. Start the Unload Wizard.
3. In the first dialog of the unload wizard, select the option button **Unload objects into Natural work file(s)**.
4. In the next dialog in which you have to specify the options settings, define a Natural work file in the **Unload file** text box. This work file must be located in the collecting directory which you have created previously.
5. If the application uses the same library names in both environments, do not specify any information in the next dialog (which can be used to specify parameters).

However, if the application uses library names in the runtime environment which are different from those used in the development environment, select the option button **Use global parameters**, choose the **Set** button and set the name in the resulting dialog box.

6. In the next dialog in which you have to specify the object type, select the option button **Natural library objects**.
7. In the next dialog in which you have to select the Natural library objects to unload, choose the **Details** button.
8. In the resulting dialog box, specify all cataloged objects, resources and error messages contained in the application. Do not unload the sources: from the **S/C-Kind** drop-down list box, choose **Gp**.

9. Proceed to the next dialog and unload the objects.
10. After a successful unload, check the work file that has been created in the collecting directory. Use the Load Wizard to scan the work file for all objects.

## Copying the Collecting Directory to a Transfer Medium

When all files in the collecting directory are ready for porting, use the Windows Explorer to copy the contents of the collecting directory (including all subdirectories) to the transfer medium (for example, to a CD).

## Step 2: Installing Natural Runtime

Install Natural Runtime on the runtime workstation. See the *Installation* documentation for further information.

### Note:

This step is not required when updating applications on the Natural Runtime workstation.

## Step 3: Installing the Application on the Runtime Workstation

The following topics are covered below:

- Copying the Global Configuration File
- Copying the Natural Parameter File
- Copying or Loading the Objects

### Copying the Global Configuration File

Use the Windows Explorer to copy the global configuration file (in which the required DBID has been defined) from the transfer medium to the directory on the runtime workstation which contains the global configuration file.



#### Warning:

**An existing global configuration file will be overwritten.**

### Note:

If you do not know where to find the global configuration file, invoke the Configuration Utility on the runtime workstation, expand the **Local Configuration File** node and select the **Installation Assignments** node. The full path and name of the global configuration file is then shown.

### Copying the Natural Parameter File

Use the Windows Explorer to copy the Natural parameter file from the transfer medium to the directory on the runtime workstation which contains the Natural parameter files.



**Warning:**  
An existing parameter file will be overwritten.

**Note:**

If you do not know where to find the parameter files, select the **Installation Assignments** node of the Configuration Utility as described above. The location of the Natural parameter files is then listed as **Path to parameter**.

## Copying or Loading the Objects

Depending on how the objects have been packaged (see *Copying or Unloading the Objects*), the transfer medium contains either complete libraries or a Natural work file.

If complete libraries have been copied using the Windows Explorer, the transfer medium contains directories with Natural library names. Each directory reflects the Natural library structure: it contains the file *FILEDIR.SAG* and the subdirectories *GP*, *RES* and *ERR*. In this case, you have to copy the libraries as described below.

If the objects have been unloaded into a Natural work file using the Object Handler, the transfer medium contains this work file. In this case, you have to load the objects using the Object Handler as described below.

### ▶ To copy the libraries

- Use the Windows Explorer to copy the libraries (including all subdirectories) to the directory for the FUSER system file.

**Note:**

If you do not know where to find this directory, invoke the Configuration Utility, select the parameter file that you have copied in a previous step, expand the **Natural Execution Configuration** node and select the **System Files** node. The path to the FUSER directory is shown on the **FUSER** tab.

### ▶ To load the objects

1. Invoke the Configuration Utility and make sure that the FUSER settings of the parameter file NATPARAM have the same DBID and FNR as the parameter file that you have copied in a previous step.

**Note:**

The DBID and FNR of the FUSER are shown on the **FUSER** tab which is invoked as described above.

2. From the Windows **Start** menu choose **Programs > Software AG Natural Runtime n.n > Natural Runtime**.

Or:

Use the following shortcut on your Windows desktop.





This invokes Natural Runtime with the standard parameter file NATPARM.

3. Logon to the library SYSOBJH.
4. Execute the program MENU in the library SYSOBJH.

The Object Handler window appears.

5. In the first dialog of the load wizard, select the option button **Load objects from Natural work file(s)**.
6. In the next dialog in which you have to specify the options settings, define your Natural work file in the **Load file** text box. This must be the work file which is located on the transfer medium.

**Note:**

If DBID and FNR of the new parameter file differ from the standard NATPARM settings, enter the values used by the new parameter file in the next dialog (which can be used to specify parameters): Select the option button **Use global parameters** and choose the **Set** button. In the resulting dialog box, select the **Load Target** tab and enter the corresponding values for DBID and FNR in the **Load FUSER** group box.

7. In the next dialog in which you have to specify the object type, select the option button **Load all objects from work file**.
8. Proceed to the next dialog and load the objects.
9. Exit the Object Handler and then exit Natural Runtime.

## Step 4: Starting the Application on the Runtime Workstation

When all required files have been copied to the runtime workstation, you can start your application. It is recommended that you create a shortcut for each application. You can then define the name of the parameter file which is required to run the application in the shortcut.

On the runtime workstation, you can start the application in different ways:

- **With *naturalr.exe***

The user interface, which appears when you do not specify the name of a parameter file with the *naturalr.exe* command, runs invisibly in the background. For example:

```
"C:\Program Files\Software AG\Natural Runtime\n.n.\bin\naturalr.exe" PARM=file-name
```

where *n.n* is the current version number and *file-name* is the name you have assigned to your customized Natural parameter file (without any file extension).

The user interface becomes visible only if the application for which it was started does not terminate this runtime process properly (for example, if the application does not issue a TERMINATE statement). This user interface increases the consumption of system resources, even if it does not appear.

- **With *natrt.exe***

This so-called "mini runtime" does not have a user interface that would allow the user to select an application for execution. It requires that the name of the program that is to be started is defined in the parameter file. For example:

```
"C:\Program Files\Software AG\Natural Runtime\n.n.\bin\natrt.exe" PARM=file-name
```

where *n.n* is the current version number and *file-name* is the name you have assigned to your customized Natural parameter file (without any file extension).

The mini runtime terminates as soon as all commands in the parameter file have been processed.

If a program name is not specified in the parameter file, the mini runtime terminates immediately.

When you use the mini runtime, the consumption of system resources is decreased and it is ensured that the runtime process terminates at the end of the application processing.

## Using the Natural Runtime Startup Service

When the Natural Runtime startup service has been installed and is active, it is possible to start one or more Natural Runtime processes automatically when the PC is booted.

You can define parameter templates which are used to hold Natural parameters. It is thus possible to start a Natural Runtime process with all parameters that are defined in the template.

A Natural Runtime process is normally used to run a Natural application. For this purpose, the Natural Runtime process has to be started with a template in which the `STACK` parameter has been defined as follows:

```
STACK=(LOGON library-name; program-name)
```

When the `STACK` parameter has not been defined, Natural is started without running any application.

By default, the Natural Runtime startup service is not installed. You have to install it as described below.

The following topics are covered below:

- Natural Runtime Startup Service Commands
- Example: Starting a Natural Process Automatically

### Natural Runtime Startup Service Commands

The file *natrtsvc.exe*, which is stored in the *bin* directory of Natural Runtime, is used to execute the service commands.

The following service commands can be specified in the **Command Prompt** window of Windows:

Command	Description
NATRTSVC INSTALL <i>mode</i>	<p>Installs the Natural Runtime startup service. <i>mode</i> can be one of the following:</p> <p>manual      Default. The service is installed and must be started manually (either with the START command or by starting the <b>Software AG Natural Runtime n.n Startup Service</b> in Windows).</p> <p>automatic    The service is installed and is automatically started when the PC is booted.</p>
NATRTSVC REMOVE	Removes the Natural Runtime startup service from the system.
NATRTSVC START	Starts the Natural Runtime startup service if it had not been started yet. The service searches for previously created parameter templates for which the <i>start</i> parameter has been set to "yes". In addition, it starts a Natural Runtime process with the Natural parameters which are also stored in the template.
NATRTSVC START <i>template-name</i>	Starts a Natural Runtime process with the Natural parameters stored in the specified template. If the Natural Runtime startup service has not been started (automatically at boot time or manually by the user) an error message is displayed.
NATRTSVC STOP	Stops the Natural Runtime startup service and all Natural Runtime processes that have been started by the Natural Runtime startup service.
NATRTSVC STOP <i>template-name</i>	Stops the Natural Runtime processes that have been started by the Natural Runtime startup service with the Natural parameters stored in the specified template.
NATRTSVC CREATE <i>template-name</i>	Creates a new parameter template to be started by the Natural Runtime startup service.
NATRTSVC DELETE <i>template-name</i>	Deletes the specified template from the Natural Runtime startup service.

Command	Description
NATRTSVC SET <i>template-name</i> start= <i>mode</i>	Defines whether a Natural Runtime process with the Natural parameters stored in the specified template is to be started when the Natural Runtime startup service is started. <i>mode</i> can be one of the following:  yes      A Natural Runtime process is started.  no        Default. A Natural Runtime process is not started.
NATRTSVC SET <i>template-name</i> <i>Natural-parameters</i>	Stores the Natural parameters in the specified template. For valid Natural parameters, refer to the <i>Parameter Reference</i> . When you specify more than one parameter, you have to separate the parameters with blanks. Instead of parameters, it is also possible to specify the name of a Natural parameter file.
NATRTSVC SHOW	Displays the startup settings and the stored Natural parameters for all templates.
NATRTSVC SHOW <i>template-name</i>	Displays the startup settings and the stored Natural parameters for the specified template.
NATRTSVC STATUS	Displays the status of all templates, that is: whether these templates are active or not active.
NATRTSVC STATUS <i>template-name</i>	Displays the status of the specified template, that is: whether this template is active or not active.

## Example: Starting a Natural Process Automatically

This example explains how to install the Natural Runtime startup service, create a new template and start the corresponding Natural process each time the PC is booted.

### To start a Natural process when the PC is booted

1. Invoke the **Command Prompt** window of Windows.
2. Go to the Natural *bin* directory which contains the file *natrtsvc.exe*.
3. Enter the following command to install the Natural Runtime startup service:

```
NATRTSVC INSTALL automatic
```

The following information is shown:

```
%NATRTSVC-I: Natural Runtime n.n Startup Service
%NATRTSVC-I: Natural Runtime n.n Startup Service successfully installed
%NATRTSVC-I: Path of binary is C:\PROGRAM FILES\SOFTWARE AG\NATURAL RUNTIME\n.n\BIN\NATRTSVC.EXE
%NATRTSVC-I: Startup mode of Natural Runtime n.n Startup Service is 'Automatic'
%NATRTSVC-I: Natural Runtime n.n Startup Service
```

From now on, the Natural Runtime startup service will be started automatically each time the PC is booted.

4. Enter the following command to create an empty parameter template with the name "exa\_temp":

```
NATRTSVC CREATE exa_temp
```

The following information is shown:

```
%NATRTSVC-I: Natural Runtime n.n Startup Service
%NATRTSVC-I: New Natural instance 'exa_temp' created
%NATRTSVC-I: Natural Runtime n.n Startup Service
```

5. Enter the following command to define that a Natural Runtime process with the Natural parameters stored in the parameter template "exa\_temp" is to be started when the Natural Runtime startup service is started:

```
NATRTSVC SET exa_temp start=yes
```

The following information is shown:

```
%NATRTSVC-I: Natural Runtime n.n Startup Service
%NATRTSVC-I: Configuration successfully set
%NATRTSVC-I: Natural Runtime n.n Startup Service
```

6. Enter the following command to store the contents of the Natural parameter file "myparm" in the parameter template "exa\_temp":

```
NATRTSVC SET exa_temp parm=myparm
```

The following information is shown:

```
%NATRTSVC-I: Natural Runtime n.n Startup Service
%NATRTSVC-I: Configuration successfully set
%NATRTSVC-I: Natural Runtime n.n Startup Service
```

7. Reboot your PC.

Since you have defined the automatic startup mode for the Natural Runtime startup service, the defined Natural Runtime processes are started automatically after Windows has been started.

8. Enter the following command to display the status of all parameter templates that are currently defined:

```
NATRTSVC STATUS
```

The following information is shown:

```
%NATRTSVC-I: Natural Runtime n.n Startup Service  
%NATRTSVC-I: Send request to Natural Runtime n.n Startup Service  
%NATRTSVC-I:   exa_temp is active  
%NATRTSVC-I: Natural Runtime n.n Startup Service
```