

# JOB & OUTPUT MANAGEMENT SUITE.

## APPLICATION PROVISIONING



Application management today has to be as efficient as possible. Application Provisioning (AppChange) automates the deployment of new application releases and to avoid human failures in the setup of the software deployment process. The components of new releases are defined in packages. Any number of processing steps can be specified for each component and will run automatically when the package is installed. Checkpoints define segments which need to be checked manually before the next segment is released. Mails can be sent automatically when the deployment has ended successfully or if problems occur during the process.

AppChange can be integrated into the ITIL change management process.

### Benefits

- Reduction of the input required for the deployment of a new release or version
- Reusable definitions for different environments (test, integration, production)
- Reduction of the time required for the deployment process
- Potential errors during implementation reduced to a minimum thanks to pre-checking and proven, standardized procedures
- Simple definition of the scope of the deployment
- Overview and details of the deployment progress and status
- Flexibility thanks to facilities that permit manual intervention
- Supports high security standards

When deploying new versions and releases of application systems, substantial effort is often required on the part of those who maintain the application due to the various processing steps involved for the respective components.

AppChange helps you to automate this deployment process to a very large degree. AppChange contains functions for defining and checking the steps in the process before the deployment begins.

During the deployment, AppChange can optimize the JCL so that the time for the deployment is reduced.

### Defining an AppChange package

All the activities that have to be performed when deploying a new release are described in a package definition, which comprises the following:

- The names of the files in which the individual components are provided
- Information about what should be done with these components (type of processing, sequence of implementation, prerequisite for manual checking etc.)
- A reference to the variable data that apply to a specific environment ("target system") (e.g. file names or name of the database application)
- Further information, e.g. mail addresses of the persons who are to be informed when the deployment of an application reaches a certain status

These package definitions can either be generated by the application managers or by the software developer. They can be transferred interactively either via the administration user interface or via the batch interface in an XML structure.

Prior to the deployment of a new application, it is possible to check whether problems could arise during its use. For example, AppChange detects whether a component is missing in the specified source file. This helps to detect many problems before the

application is actually deployed.

However, if problems occur during the deployment, it is possible to manually intervene in order to restart the process at a given point or from the beginning. AppChange also contains functions that offer support for the "rollback", i.e. with the restore of the previous version.

AppChange lets the user set checkpoints at which a manual release is requested before processing resumes.

Information about the technical procedures to be followed during deployment is stored in the form of skeletons. A skeleton contains the JCL, which for example calls a utility. It also contains logic, e.g. for the optimization of the generated JCL.

Skeletons for standard database systems and other components are included in the AppChange product. If necessary, individual customer- or site-specific skeletons can be created.

### Product functions

- Definition of a package (either by the application managers or by the software developers) with the definition of all processing steps to be executed during the deployment process
  - Routines to check the package definition (e.g. whether all the components are available)
  - Definition of a time frame during which the automatic deployment may start
  - Skeletons for standard software utilities
  - Expandable for other software utilities
  - Functions for backing up the previous version
  - Variables to define a target system (e.g. test, integration, production)
  - Automatic processing of the deployment process with the option to manually intervene
  - Dynamic creation of the JCL during deployment
- Optimization of the deployment jobs / steps to reduce the time for the deployment
  - Display of the current progress of the deployment process
  - Support functions for reverting to the previous version
  - ISPF user dialog
  - Interface to call local functions (user-specific REXX)
  - Batch interface to define packages in an XML structure
  - Interface for sending mails (notification of the progress of the deployment process)
  - Open interface to products integrated into the change management process to ITIL
  - Log function to display all the user and system actions