

IBBSOFT®

Reporting – Scheduling mit

Data2XML

Handbuch

Dieses Handbuch dient als Arbeitsunterlage für Benutzer des Programms Data2XML aus der IBBSOFT® Programmfamilie.

Die in diesem Handbuch enthaltenen Informationen können ohne Vorankündigung geändert werden und stellen keine Verpflichtung seitens IBBSOFT® dar. Die Software, die in diesem Handbuch beschrieben ist, wird im Rahmen einer Lizenzvereinbarung zur Verfügung gestellt.

Das Handbuch ist ausschließlich für unsere Kunden bestimmt. Alle Angaben erfolgen ohne Gewähr. Ohne ausdrückliche schriftliche Erlaubnis des Unternehmens IBBSOFT® darf dieses Handbuch oder ein Auszug hieraus für irgendwelche Zwecke oder in irgendeiner Form mit irgendwelchen Mitteln, elektronisch oder mechanisch, mittels Fotokopie, durch Auszeichnungen oder mit Informationsspeicherungs-Informationswiedergewinnungssystemen nicht reproduziert oder übertragen werden.

© Copyright 2006, 2007 IBBSOFT®

Herausgegeben von:

IBBSOFT®

IBB Industrieautomatisierung / Ingenieurbüro Bräuer

Liebenastr. 27
71111 Waldenbuch

Gewerbestraße 19
Technologiepark
08115 Lichtentanne / OT Stenn

Telefon: +49 (0)7157 22285
Fax: +49 (0)7157 22286
Email: info@ibbsoft.de

Stand der Dokumentation: Juli 2009

Inhalt

1	Einführung / Motivation	5
2	Installation	7
2.1	Setup.exe	7
2.2	Vor dem Starten von Data2XML	10
3	Konfiguration von Data2XML	13
3.1	Übersicht der Schlüsselwörter	13
3.2	Gestaltung des Layouts	16
3.2.1	Schriftformate, Farben, Hintergrund	16
3.2.2	Festlegung des vertikalen Menüs	16
3.2.3	Ansichten des vertikalen Menüs	18
3.3	Detaillierte Beschreibung der Schlüsselwörter einer Abfrage	19
3.3.1	Abschnitt der Abfragen	19
	Anhang A.....	34

Abbildungsverzeichnis

Abbildung 1: Dialog sourcepath.settings	11
Abbildung 2: Auszug Diagramm mit Ziellinie	30
Abbildung 3: Parameterdialog mit voreingestellter Zeitspanne	31

Tabellenverzeichnis

Tabelle 1: Parameter für die Pfadangabe	11
Tabelle 2: Schlüsselwörter für Sourcepath.settings	12
Tabelle 3: Konfiguration - Layout Data2XML	13
Tabelle 4: Schlüsselwörter zur Datenherkunft	14
Tabelle 5: Abfrage automatisch / manuell	14
Tabelle 6: Überschriften im Report	14
Tabelle 7: Hinterlegung der SQL-Abfrage	14
Tabelle 8: Schlüsselwörter für Parameterübergabe	14
Tabelle 9: Schlüsselwörter für ftp-Versand	14
Tabelle 10: Schlüsselwörter für Mail-Versand	15
Tabelle 11: Schlüsselwort für das zu verwendende Arbeitsverzeichnis	15
Tabelle 12: Schlüsselwort für Java-Aufruf	15
Tabelle 13: Schlüsselwort für eine Ziellinie	15
Tabelle 14: Schlüsselwort für einbezogene Subreports und Bilder	15
Tabelle 15: Schlüsselwort für Zeitspanne	15
Tabelle 16: Übersicht - Icons	19
Tabelle 17: Konstanten für DBTyp	20
Tabelle 18: Konstanten für Used	21
Tabelle 19: Konstanten zu cron	21
Tabelle 20: Konstanten zu cron (Fortsetzung)	22
Tabelle 21: Parameter zu cron	22
Tabelle 22: Konstanten für pdfuse	22
Tabelle 23: Parameter zu Überschrift3	23
Tabelle 24: Konstanten für ftpcopy	27
Tabelle 25: Konstanten für email1use	27
Tabelle 26: Konstanten für Periode	32
Tabelle 27: Konfiguration - Layout Data2XML	34
Tabelle 28: Schlüsselwörter Reporteinbindung	35
Tabelle 29: Konstanten zu cron	36
Tabelle 30: Konstanten zu cron (Fortsetzung)	36
Tabelle 31: Parameter zu cron	36



1 Einführung / Motivation

Riesige Datenmengen entstehen heute in Unternehmen aller Ausrichtungen, in Forschungsprojekten an Universitäten, in Verwaltungen oder sogar im Privathaushalt. Hierbei werden insbesondere in Unternehmen mit einer automatisierten Produktion Daten erzeugt, deren Auswertung und Darstellung für das Management die Grundlage für Investitionsentscheidungen, für Maßnahmen zur präventiven Instandhaltung oder zur Steuerung des Produktionsprozesses darstellen könnten. In der Regel werden diese Daten in unterschiedlich gearteten Datenbanken mehr oder weniger strukturiert abgelegt.

SQL ist eine deklarative Abfragesprache für Relationale Datenbanksysteme. Als Quasi-Standard ist SQL in nahezu alle Relationalen Datenbanksysteme integriert. Daher baut das vorliegende Programm auf die Datenerhebung mittels SQL auf.

Infolge der Ausrichtung auf industrielle Umgebungen und die dort geforderte hohe Verfügbarkeit ist das vorliegende System auf die verbreitete Datenbank Oracle® insofern spezialisiert, dass der Zugriff mittels nativen Treibern (Oracle-Client) erfolgt. Darüber hinaus kann auf alle Datenbanksysteme zugegriffen werden, für die ODBC-Treiber verfügbar sind. Das dürfte auf alle verbreiteten und in größerer Menge bzw. Umfang verwendeten Datenbanksysteme zutreffen. Im ODBC-Modus stehen ebenfalls alle Möglichkeiten unserer Software ohne Einschränkungen zur Verfügung. Die Stabilität des Programms ist in diesem Fall jedoch entscheidend von der Implementierung von ODBC abhängig.

Das Programm IBBSOFT® Data2XML führt vordefinierte SQL-Abfragen zyklisch automatisch oder nach Aufforderung durch den Anwender aus. Das Abfrageergebnis wird im Format XML (XML+XSL) erstellt, per FTP oder Email an ein gewünschtes Ziel (z.B. Internet/Intranet-Server) übertragen und es werden bei Überschreitung vordefinierter Grenzwerte Emails an festgelegte Empfänger automatisch versendet.

Weiter erstellt das Programm frei definierte Reporte auch mit graphischen Bestandteilen (Diagramme) aus den Abfrageergebnissen. Für den Entwurf der Reporte steht eine Schnittstelle zu einem "Designer" zur Verfügung.

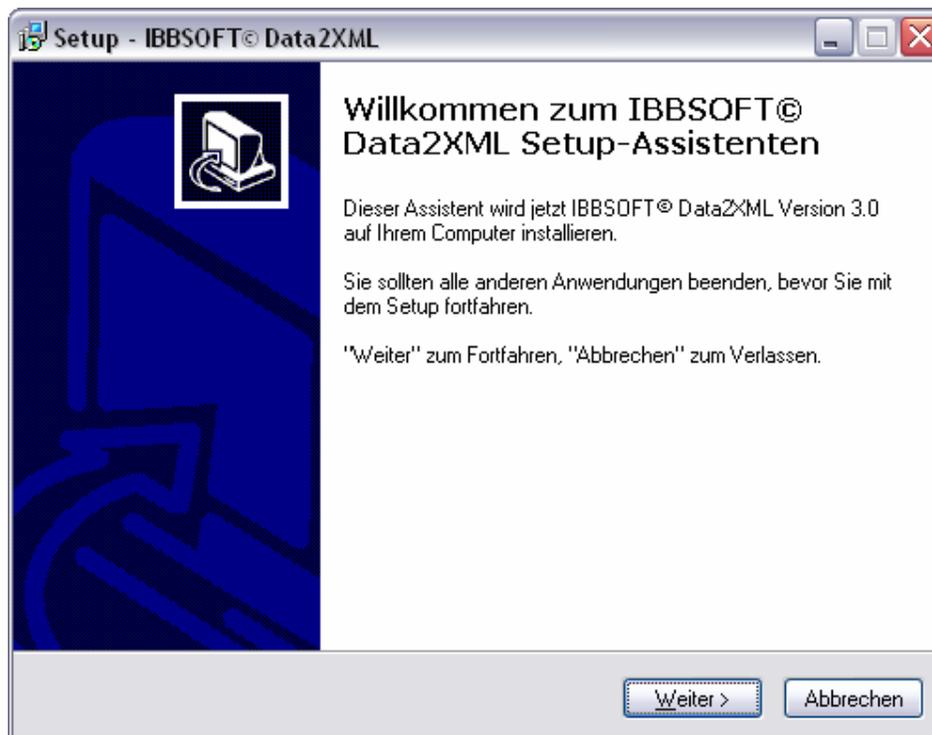
Das Programm ist unter den Windows®-Betriebssystemen lauffähig. Hierbei werden keine besonderen Anforderungen an die Computerarchitektur gestellt. Es kann jedoch nicht garantiert werden, dass die Ablauffähigkeit auf allen Systemen uneingeschränkt gewährleistet ist, auch wenn hier ausdrücklich keine Mindestkonfiguration und Ausstattung genannt wird.

2

2 Installation

2.1 Setup.exe

Legen Sie die Data2XML-CD in Ihr CD- oder DVD-ROM-Laufwerk. Doppelklicken Sie im Inhalt der CD-ROM auf die Datei Data2XML_3_0_0.exe. Diese Datei installiert das Programm komplett auf Ihren Computer. Folgen Sie den Anweisungen des Installationsassistenten.



Betätigen Sie hier bitte den Button „Weiter“.



Akzeptieren Sie bitte an dieser Stelle die Lizenzvereinbarung und betätigen Sie anschließend den Button „Weiter“.

Im nachfolgenden Fenster wählen Sie bitte den Installationsordner aus oder akzeptieren den Vorschlag unseres Installationsprogramms.



Sie können das Programm in einen beliebigen Ordner installieren. Es bestehen keinerlei Notwendigkeiten oder versteckte Abhängigkeiten von bzw. zu bestimmten Installationsorten oder Laufwerken.

Wählen Sie die Bezeichnung der Programm-Verknüpfung im Startmenü-Ordner oder nehmen Sie den Vorschlag an.



Das Programm wird dann durch ein Icon auf dem Desktop bzw. durch einen Eintrag im Startmenü gestartet. Zuvor sollte das Programm jedoch an Ihre Erfordernisse angepasst und konfiguriert werden.

2.2 Vor dem Starten von Data2XML

Bevor das Programm gestartet wird, ist die Datei **Sourcepath.settings** einzurichten. Diese Datei beinhaltet den Pfad einer weiteren Settings-Datei Data2XML.setting, die letztendlich die Abfragen enthält. Auf diese Weise können von verschiedenen Standorten aus auf das gleiche Data2XML zugegriffen und gleichzeitig eigene erstellte Reporte übergeben werden. Beim Starten von Data2XML erfolgt eine Abfrage, auf welche **Data2XML.settings** zugegriffen wird.



Abbildung 1: Dialog sourcepath.settings

Die **Sourcepath.settings** besteht aus zwei Sektionen [path] und [modus]. Unter [path] werden das Verzeichnis und die Bezeichnung der Data2XML.settings angegeben. Insgesamt können neun verschiedene Pfade festgelegt werden.

Pfad	Verzeichnis und Datei-Name	Name bel.	Verzeichnis
path1=	d:\master_Data2XML.settings;	1Quelle 1;	d:\;

Tabelle 1: Parameter für die Pfadangabe

Das Beispiel zeigt einen Ausschnitt einer Sourcepath.settings-Datei. Die Data2XML-Settingsdatei wird hier als Data2XML.settings angegeben. Denkbar wäre an dieser Stelle auch eine andere Benennung. Das Programm holt sich die angegebene Datei aus dem angegebenen Verzeichnis und benennt diese im Arbeitsverzeichnis mit Data2XML.settings.

Beispiel 2.2-1

```
[path]
path1=c:\Programme\Data2XML\sample\Data2XML.settings;1Quelle 1;c:\Programme\Data2XML\sample\;
path2=d:\master_Data2XML.settings2; 2Quelle 2;d:\;
path3=d:\master_Data2XML.settings3; 3Quelle 3;d:\;
path4=d:\master_Data2XML.settings4; 4Quelle 4;d:\;
path5=d:\master_Data2XML.settings5; 5Quelle 5;d:\;
path6=d:\master_Data2XML.settings6; 6Quelle 6;d:\;
path7=d:\master_Data2XML.settings7; 7Quelle 7;d:\;
path8=d:\master_Data2XML.settings8; 8Quelle 8;d:\;
path9=d:\master_Data2XML.settings9; 9Quelle 9;d:\;
[...]
```

Schlüsselwort	Beschreibung	Wert	Link
designer=	Dialog zur Auswahl der Settings-Datei	<i>const</i>	designer
viewonly=	Anzeige der XML-Dateien	<i>const</i>	viewonly
Designerexecutable=	Verwendung des Designers für Jasper®	<i>iReport.exe</i>	designerexecutable

Tabelle 2: Schlüsselwörter für Sourcepath.settings

In der Sektion [modus] wird der Report-Designer festgelegt und eingebunden sowie die generelle Steuerung des Data2XML.

designer=

Der Parameter designer kann den Wert 0 und 1 annehmen. Bei Verwendung eines Report-Designers, wie hier der iReport ist dem Parameter eine 1 zuzuweisen. Zusätzlich wird beim Start der Dialog zur Auswahl möglicher Settingsquellen angezeigt (vgl. Abbildung 1 auf Seite 11). Wird der Wert 0 übergeben, so erfolgt keine Berücksichtigung eines Designers und der Auswahl-Dialog erscheint ebenfalls nicht.

viewonly=

Der Parameter viewonly=1 bedeutet, dass dem Endanwender lediglich die vorliegende XML-Dateien angezeigt werden, die im Arbeitsverzeichnis (siehe hierzu Data2XML.settings) hinterlegt wurden. Wird dem Parameter eine 0 übergeben, können weitere Dateien (z.B. Pdf) angezeigt werden.

designerexecutable=

Parameter designerexecutable benennt den verwendeten Designer. Derzeit existiert eine Schnittstelle zum Report-Designer iReport von Jaspersoft®. Die ausführbare Datei muss im Programmverzeichnis enthalten sein. Sollte dieses Programm in Java erstellt sein, so existieren diverse Abhängigkeiten vom CLASSPATH.

Beispiel 2.2-2

```
[ö]
[modus]
designer=1
viewonly=0
designerexecutable=iReport.exe
```

Das Beispiel zeigt die Anbindung von iReport.

3

3 Konfiguration von Data2XML

Die Konfiguration von Data2XML wird innerhalb einer Settings-Datei vorgenommen. Die Settings-Datei **Data2XML.settings** enthält die gewünschten Parameter. Der Aufbau der Settings-Datei ist dem einer ini-Datei nachempfunden. Das bedeutet, die Datei wird in Sektionen unterteilt und einer Reihe von Schlüsselwörtern können Eintragungen übergeben werden. Die Eintragungen innerhalb einer Sektion gelten für genau einen Report.

Dieses Kapitel befasst sich mit den Konfigurations-Möglichkeiten, die in Data2XML.settings realisierbar sind. Abschnitt 3.1 gibt einen Überblick über die Schlüsselwörter sowie eine kurze Beschreibung zu deren Funktionalität. In den Abschnitten 3.2 und 3.3 erfolgen die ausführlichen Schilderungen und beispielhaft die Anwendung.

3.1 Übersicht der Schlüsselwörter

Im Folgenden sind Tabellen aufgeführt, die die Schlüsselwörter der **Data2XML.settings** enthalten. Spalte „Wert“ birgt die Information, ob dem Schlüsselwort ein beliebiger Wert (*user*) übergeben werden kann oder ob eine Konstante (*const*) als Übergabewert vorgesehen ist. Spalte „Link“ verweist auf die Position der ausführlichen Erläuterung im Dokument.

Schlüsselwort	Beschreibung	Wert	Link
CSSFile=	CSS-Datei	<i>user</i>	CSSFile
Folder=	Zuordnung der Reporte zu einem vertikalen Menüabschnitt	<i>user</i>	Folder
Icon=	Icon-Typ eines Reports im vertikalen Menü	<i>const</i>	Icon

Tabelle 3: Konfiguration - Layout Data2XML

Schlüsselwort	Beschreibung	Wert	Link
DBTyp=	Typ der Datenbank	<i>const</i>	DBTyp
User=	Benutzername für den Zugriff auf die Datenbank	<i>user</i>	User
PassWord=	Passwort des obigen Benutzers	<i>user</i>	PassWord
DataBase=	Name des ODBC-Treibers bzw. Bezeichnung der Datenbank	<i>user</i>	DataBase

Tabelle 4: Schlüsselwörter zur Datenherkunft

Schlüsselwort	Beschreibung	Wert	Link
used=	Report automatisch oder manuell ausführen	<i>const</i>	Used
Cron=	Zeitliche Steuerung der Abfragen	<i>const</i>	Cron
pdfuse=	Anzeige des Reports als pdf-Datei oder als xml-Datei	<i>const</i>	Pdfuse

Tabelle 5: Abfrage automatisch / manuell

Schlüsselwort	Beschreibung	Wert	Link
Ueberschrift1=	Überschrift im Report	<i>user</i>	Ueberschrift
Ueberschrift2=	Unterüberschrift im Report	<i>user</i>	
Ueberschrift3=	Überschrift mit Parametrisierung eines Zeitrahmens	<i>user</i>	

Tabelle 6: Überschriften im Report

Schlüsselwort	Beschreibung	Wert	Link
SelectString=	SQL-Abfrage in einer Zeile	<i>user</i>	SelectString
FileName=	Dateiname zu einer *.sql mit SQL-Abfrage	<i>user</i>	FileName
Fields=	Bezeichnung der Felder zur SQL-Abfrage	<i>user</i>	Fields

Tabelle 7: Hinterlegung der SQL-Abfrage

Schlüsselwort	Beschreibung	Wert	Link
Parameter=	Liste der Parameter (maximal 5)	<i>user</i>	Parameter
Para1ChoiceList=	Dropdownliste für Parametereingabe	<i>user</i>	ChoiceList

Tabelle 8: Schlüsselwörter für Parameterübergabe

Schlüsselwort	Beschreibung	Wert	Link
ftpcopy=	Versand der Reporte an ftp-Server	<i>const</i>	ftp
ftpip=	IP des ftp-Servers	<i>user</i>	
ftpusr=	Benutzername	<i>user</i>	
ftppwd=	Passwort	<i>user</i>	

Tabelle 9: Schlüsselwörter für ftp-Versand

Schlüsselwort	Beschreibung	Wert	Link
email1use=	Aufruf des Email-Versandes	const	
email1smtp=	IP des SMTP-Servers	user	
email1sender=	Absenderadresse	user	
email1usr=	Benutzername für SMPT-Anmeldung	user	
email1pwd=	Passwort für SMTP-Anmeldung	user	email1
email1cond=	Bedingung für Email-Versand	user	
email1subject=	Betreffzeile der Email	user	
email1text=	Inhalt der Email	user	
email1empfaenger=	Email-Adressen der Empfänger	user	

Tabelle 10: Schlüsselwörter für Mail-Versand

Schlüsselwort	Beschreibung	Wert	Link
WorkDir=	Zu verwendende Arbeitsverzeichnis	const	WorkDir

Tabelle 11: Schlüsselwort für das zu verwendende Arbeitsverzeichnis

Schlüsselwort	Beschreibung	Wert	Link
jcall=	Aufruf des Java	const	jcall

Tabelle 12: Schlüsselwort für Java-Aufruf

Schlüsselwort	Beschreibung	Wert	Link
Bourne=	Ziellinie im Balkendiagramm	user	Bourne

Tabelle 13: Schlüsselwort für eine Ziellinie

Schlüsselwort	Beschreibung	Wert	Link
Picture=	Liste von im graphischen Report verwendeten Bilder und Subreports	user	Picture

Tabelle 14: Schlüsselwort für einbezogene Subreports und Bilder

Schlüsselwort	Beschreibung	Wert	Link
Periode=	Angabe der Zeitspanne	user	Periode

Tabelle 15: Schlüsselwort für Zeitspanne



Mit Datei **Data2XML.settings** sind nahezu alle Konfigurationsmöglichkeiten für das Programm verbunden. Eine Zerstörung bzw. ein Verlust dieser Datei versetzt das Programm in den Auslieferungszustand. Daher wird eine Sicherung der Datei auf einen externen Datenträger bzw. auf einen anderen Computer im Netzwerk dringend empfohlen.

Diese Datei **Data2XML.settings** liegt im Installationsverzeichnis des Programms Data2XML.

3.2 Gestaltung des Layouts

3.2.1 Schriftformate, Farben, Hintergrund

CSSFile=

Das Aussehen der erzeugten XML-Dateien kann mit einer CSS-Datei aufgewertet werden. Einzelheiten zu Notationen in CSS-Dateien sind der entsprechenden Fachliteratur zu entnehmen. In Sektion **[CSS]** wird die CSS-Datei angegeben, die sich im Arbeitsverzeichnis von Data2XML befinden muss (vgl. Beispiel 3.2-1). Die Sektion **[CSS]** muss vor der Sektion **[Folder]** stehen.

Beispiel 3.2-1

```
[CSS]
CSSFile=FOREST.CSS
```

3.2.2 Festlegung des vertikalen Menüs

Zu Beginn der Datei **Data2XML.settings** werden die Abschnitte des vertikalen Menüs, in denen später die Reports abgelegt sind, festgelegt. Auf diese Bezeichnungen wird innerhalb der Report-Abschnitte der Settings-Datei Bezug genommen. Dem benannten Abschnitt folgen weitere Sektionen, die sich auf die konkreten Reports beziehen. Betrachtet wird nun die Sektion **[Folder]**.

[Folder]

In Sektion **[Folder]** erfolgt die Auflistung der Abschnitte mit den Schlüsselwort **Folder** und einer dreistelligen fortlaufenden Nummer mit führenden Nullen. Dem Schlüsselwort **Folder** kann ein beliebiger Text zugeordnet werden, der im Programm den Menüabschnitt bezeichnen soll. Die Anzahl der **Folder** ist beliebig erweiterbar (vgl. Beispiel 3.2-2).

Beispiel 3.2-2

[Folder]
Folder001=Report (periodisch)
Folder002=Report (von Hand)
Folder003=Vorstandsversionen



Zur Demonstration wird nun ein weiterer Folder eingefügt. Hierzu tragen wir eine zusätzliche Zeile in die Sektion [Folder] ein.

[Folder]
Folder001=Report (periodisch)
Folder002=Report (von Hand)
Folder003=Vorstandsversionen
Folder004=meine Datenbanken



Als Resultat erhalten wir einen weiteren Menüabschnitt mit der Bezeichnung „meine Datenbanken“.

Dem Bereich [Folder] schließen sich nun die Abschnitte der Abfragen bzw. Reporte an. Eingeleitet werden diese durch eckige Klammern und dem Namen der Abfrage, der vom Benutzer frei wählbar ist. Im Beispiel 3.2-3 sind die Reporte mit **meinReportName1** und **meinReportName2** bezeichnet worden.

Beispiel 3.2-3

[Folder]
Folder001=Report (periodisch)
Folder002=Report (von Hand)
Folder003=Vorstandsversionen
Folder004=meine Datenbanken

[meinReportName1]
 {es folgen Schlüsselwörter und Parameter, die sich auf die Abfrage **meinReportName1** beziehen}

[meinReportName2]
 {es folgen Schlüsselwörter und Parameter, die sich auf die Abfrage **meinReportName2** beziehen}
 usw.

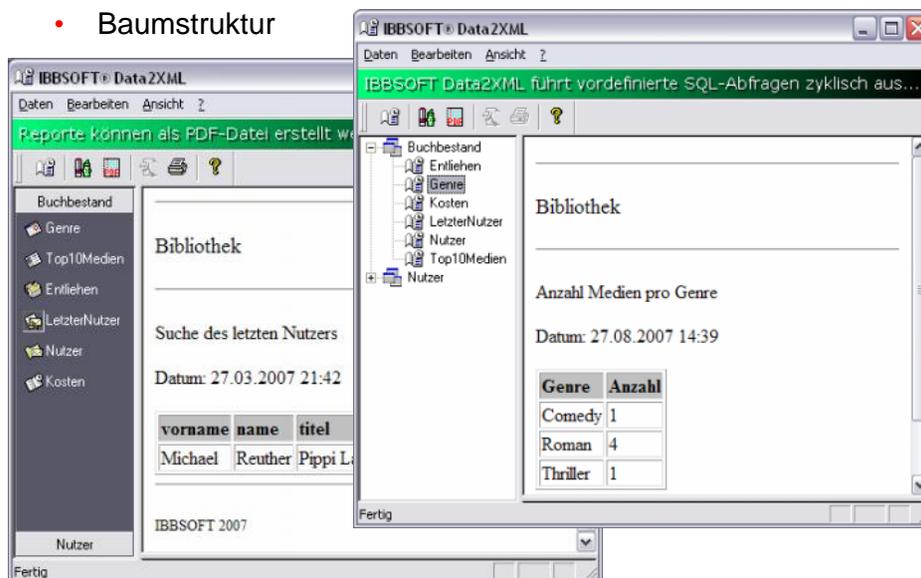


Die Namensgebung der Sektionen der Reporte, d.h. die Bezeichnung innerhalb der eckigen Klammern, ist vollständig frei wählbar. Die Benennungen müssen eindeutig sein und dürfen sich nicht wiederholen.

3.2.3 Ansichten des vertikalen Menüs

Das vertikale Menü kann auf zwei Weisen angezeigt werden.

- Liste großer oder kleiner Icons
- Baumstruktur



Die Baumstruktur kann aktiviert werden, indem eine Datei **tree.cfg** ins Programmverzeichnis gestellt wird. Der Inhalt der Datei ist unwesentlich, sie kann leer sein.

3.3 Detaillierte Beschreibung der Schlüsselwörter einer Abfrage

3.3.1 Abschnitt der Abfragen

Auf die Funktionalität der Schlüsselwörter und Parameter zu den Abfragen wird im Folgenden eingegangen.

Folder=

Die einzelnen Reporte können nun einer der Menüabschnitte untergeordnet werden. Dazu wird dem Schlüsselwort eines aus Sektion [Folder] (vgl. [Folder]) definierten Menüabschnittes zugeordnet.

Beispiel 3.3-1

```
[meinReportName1]
Folder=Folder001
```

Der Report wurde dem ersten Menüabschnitt mit Namen **Report (periodisch)** untergeordnet. (vgl. Beispiel 3.2-3)

Icon=

Im Programm sind einige Icons definiert und integriert, die ausgewählt und im Menü als Bild dargestellt werden. Aus Tabelle 16 kann eine Nummer für einzelne Icons abgeleitet werden.

Icon	groß	Bild	klein	Bild
001	Kalender		Kalender	
002	Rolodex		Visitenkarte	
003	Klemmbrett		Karton	
004	Kalender mit Uhr		Brief und Stift	
005	Gelbe Haftnotiz		Brief	
006	Ordner		Kalender mit Uhr	

Tabelle 16: Übersicht - Icons

Beispiel 3.3-2

```
[meinReportName1]
Icon=004
```

Dem Report wurde als Icon ein Kalender mit Uhr (großes Icon) bzw. Brief mit Stift (kleines Icon) zugeordnet.

DBTyp=
User=
PassWord=
DataBase=

Diese Einträge repräsentieren die Datenbankverbindung. Die Einträge gehören logisch zusammen, daher wird empfohlen, diese Zeilen ebenfalls in der Settings-Datei örtlich zusammen zu halten.

DBTyp erwartet als Übergabewert einen der Konstanten aus Tabelle 17.

Konstante	Beschreibung
ODBC	Das Programm wird angewiesen mittels ODBC-Treiber auf die Datenbank zuzugreifen.
ORANATIVE	Der Zugriff auf Oracle-Datenbanken erfolgt über native Oracle-Client-Treiber.

Tabelle 17: Konstanten für DBTyp

Der Benutzer trägt selbst seine Zugangsdaten für die entsprechende Datenbank hinter die Schlüsselwörter **User=** und **PassWord=** ein. Als Benennung der Datenbank unter Parameter **DataBase** ist der entsprechende Name des angelegten ODBC-Treiber oder nativen Treibers anzugeben. Im Fall von ORANATIVE ist der entsprechende Eintrag aus der Datei tnsnames.ora zu nehmen.

Beispiel 3.3-3

```
[meinReportName1]

DBTyp=ODBC
User=SCOTT
PassWord=TIGER
DataBase=TESTDB
```

Das Beispiel zeigt die Anwendung der Zugangsdaten zu einer Datenbank TESTDB, die über einen ODBC-Treiber angesprochen wird. Zugangsdaten für Benutzer und Passwort lauten SCOTT und TIGER.

Used=

Hier wird gesteuert, ob die Abfrage manuell gestartet wird oder ob sich die Abfrage automatisch starten soll. Die entsprechenden Konstanten sind aus der Tabelle zu entnehmen.

Konstante	Beschreibung
0	Die Abfrage wird manuell (durch den Benutzer) gestartet. Standardwert.
1	Die Abfrage wird automatisch gestartet. Es ergeben sich weitere Optionen. (vgl. Cron)

Tabelle 18: Konstanten für Used

Cron=

Die zeitliche Steuerung der automatischen Aufrufe der Abfragen erfolgt über den Eintrag zu cron. Dieser lautet vollständig:

```
cron=* * * * *;[Zeitspanne] | [Überschrift1] | [Überschrift2] | [Überschrift3] ;
```

Zu beachten sind die genaue Verwendung der Semikolons und Pipes (senkrechter Strich). [Zeitspanne] übergibt eine Konstante, die einen definierten Zeitraum entsprechen. Die Konstanten sind der Tabelle 26 zu entnehmen. Die Überschriften ersetzen die Parameter **Überschrift1=**, **Überschrift2=**, **Überschrift3=** und sind genauso zu handhaben. Überschrift3 kann folglich ebenfalls Zeit-Parameter übergeben werden.

Für den cron-Eintrag sind verschiedene Parameter möglich.

Konstante	Beschreibung
* * * * *	Auszuführende Befehl
	Wochentag (0 - 6) (Sonntag=0)
	Monat (1 - 12)
	Tag (1 - 31)
	Stunde (0 - 23)
	Minute (0 - 59)

Tabelle 19: Konstanten zu cron

Hier die möglichen Einträge für die Zeitangaben:

Zeitangabe	Beschreibung
Minuten	0-59 und * für alle Minuten
Stunden	0-23 und * für alle Stunden
Tage	1-31 und * für jeden Tag
Monate	1-12 und * für jeden Monat
Wochentage	0-6 und * für jeden Wochentag (0 für Sonntag)

Tabelle 20: Konstanten zu cron (Fortsetzung)

Zeitangabe	Beschreibung
0 12,15 ***	Ein Komma „,“ lässt mehrere Zeitangaben zu. Jeden Tag um 12 und 15 Uhr erfolgt eine Abfrage.
0 12-24 ***	Ein Bindestrich „-“ gibt einen Zeitraum an. Jeden Tag von 12-24 Uhr (jede Stunde) wird eine Abfrage durchgeführt.

Tabelle 21: Parameter zu cron

Beispiel 3.3-4

```
[meinReportName1]
used=1
cron=0 12,15 * * *;t|Überschrift1|Überschrift2|Überschrift3;
```

Die automatische Abfrage ist eingestellt. Jeden Tag um 12 und 15 Uhr erfolgt eine Abfrage. Die Zeitspanne beläuft sich auf einen Tag. Der erstellte Report hat die Überschriften *Überschrift1*, *Überschrift2*, *Überschrift3*.

Pdfuse=

Die Anzeige der Daten der Abfragen kann wahlweise als xml- oder pdf-Datei angezeigt werden. Eine pdf-Datei erfordert die Schnittstelle zu iReport. Mittels iReport sind graphische Aufbereitungen der Daten als Kreis-, Linien- oder Balkendiagramme möglich. Tabelle 22 enthält die entsprechenden Einstellungen.

Konstante	Beschreibung
0	Die Abfrage wird als xml-Datei angezeigt. Standardwert.
1	Die Abfrage wird als pdf-Datei angezeigt.

Tabelle 22: Konstanten für pdfuse

Überschrift1=

Überschrift2=

Überschrift3=

Drei Überschriften können vorgegeben werden und erscheinen dann auf den Ausgabeseiten des erzeugten Berichtes. Bei **Überschrift3** besteht zusätzlich

die Möglichkeit, eine in der SQL-Abfrage integrierte Zeitspanne zum Abarbeitungszeitpunkt automatisch ausfüllen zu lassen. Mittels vorgegebener Parameter wird die Überschrift mit aktuellen Werten ergänzt. Es werden die nachfolgenden Parameter verwendet.

Konstante	Beschreibung
#t1	Tag zweistellig
#m1	Monat zweistellig
#y1	Jahr vierstellig
#j1	Jahr zweistellig
#h1	Stunde zweistellig
#x1	Minute zweistellig
#s1	Sekunde zweistellig

Tabelle 23: Parameter zu Überschrift3

Beispiel 3.3-5

Überschrift1=Bibliothek
Überschrift2=Suche des letzten Nutzers
Überschrift3=Datum: #t2.#m2.#y2 #h2:#x2

The screenshot shows a report window titled 'IBBSOFT® Data2XML'. The report content includes the following headers and data:

Bibliothek

Suche des letzten Nutzers

Datum: 27.03.2007 21:42

vorname	name	titel	datumvon
Michael	Reuther	Pippi Langstrumpf	2006-03-01

IBBSOFT 2007

The interface also shows a sidebar with a tree view containing 'Buchbestand', 'Genre', 'Top10Medien', 'Entliehen', 'LetzterNutzer', 'Nutzer', and 'Kosten'. The 'Nutzer' item is selected. At the bottom, there is a 'Fertig' button.

Die Überschriften werden im Bericht angezeigt.

SelectString=

Hinter dem „=“ kann eine SQL-Abfrage stehen. Die Abfrage darf nur eine Zeile umfassen. (siehe auch **FileName=**)

FileName=

Alternativ zu **SelectString** existiert **FileName**. **FileName** wird eine Datei übergeben. In der angegebenen Datei ist der SQL-Befehl, der auszuführen ist, enthalten.

Beispiel 3.3-6

```
SelectString=  
FileName=demo
```

Eine Textdatei demo.sql enthält einen SQL-Befehl. Das Beispiel zeigt die entsprechende Zuordnung. Die Endung .sql wird weggelassen.



Für die Hinterlegung des SQL-Befehls existieren zwei Möglichkeiten. Es muss entweder hinter **SelectString=** ein SQL-Befehl stehen, der nur eine Zeile ausfüllen darf, oder es muss eine Datei vorhanden sein, deren Name aus dem Wort hinter **FileName=** und der Endung .sql besteht. Im Falle der Verwendung einer Datei *.sql kann der SQL-Befehl beliebig formatiert sein.

Fields=

Hinter **Fields** steht eine mit Komma getrennte Auflistung der Feldnamen der Ergebnisse der Datenbankabfrage. Die Namen werden in den Reporten und in den erzeugten XML-Dateien verwendet.

Beispiel 3.3-7

```
Fields=Name,Vorname,Gehalt,Geburtsdatum
```



Es dürfen keine Blanks bzw. Leerzeichen verwendet werden. Anstelle von z.B. „Feld Nummer 1“ muss „Feld_Nummer_1“ benutzt werden.

Parameter=

Die Anfrage kann mittels Parametern dynamisch gestaltet werden. Im Ausführungsfenster (siehe Bild) sind bis zu 5 Textboxen bereitgestellt, die über das Schlüsselwort **Parameter** aktiviert werden. Dafür ist dem Parameter eine Bezeichnung oder eine Liste von Bezeichnern zu übergeben, deren

Anzahl die entsprechende Menge an Textfeldern frei schaltet. Der Inhalt dieser Textfelder ersetzt die Platzhalter in der SQL-Abfrage und bildet ein Suchkriterium. Die Platzhalter benennen sich entsprechend der Anzahl: #param1, #param2, #param3, #param4 und #param5.

Platzhalter im SQL:

```
Select
[ ]
WHERE
[Suchfeld] = '#param1' or
[Suchfeld] = '#param2'
```

Beispiel 3.3-8

Parameter=Titel

The screenshot shows a window titled "Zeitparameter" with a blue header bar containing the text "In der Woche ... vw bestimmt eine Woche von Montag 06.00 Uhr bis Montag 0". Below the header, there are two main sections: "Zeitspanne" and "Parameter".

The "Zeitspanne" section contains two rows of date and time pickers. The first row is labeled "von:" and the second "bis:". Each row has a date picker (showing "30.08.2007") and a time picker (showing "14:01:25"). To the right of these are two dropdown menus labeled "Zeitbereich" and "ISO-KW". There are "Weiter" and "Abbrechen" buttons to the right of the "Zeitspanne" section.

The "Parameter" section contains a table with one column labeled "Titel" and three empty rows. To the right of the table is a button labeled "Zeit/Parameter eintragen".

At the bottom of the window is an "SQL" section with a text area containing the following query:

```
SELECT
benutzer.vorname, benutzer.name, medium.titel, leihstatus.datumvon
FROM
bibo_benutzer, bibo_leihstatus, bibo_medium
WHERE
benutzer.userid=leihstatus.userid and
medium.mediumid=leihstatus.mediumid and
(medium.titel like '%#param1%' or ISNULL(medium.titel))
```

Im Ausführungsfenster wird eine Textbox Titel aktiviert. Der Inhalt dieses Feldes wird an die SQL-Abfrage (Textfenster) übergeben und ersetzt dort einen Platzhalter.

Para1ChoiceList=

Des Weiteren kann anstelle der Textbox eine Auswahlbox stehen, deren Inhalt vom Benutzer bestimmt werden kann. Dazu ist zusätzlich zum Schlüsselwort Parameter eine Auswahlliste Para1ChoiceList mit dem

gewünschten SQL-Befehl anzulegen. Es können maximal 5 Auswahllisten angelegt werden: Para1ChoiceList, Para2ChoiceList, Para3ChoiceList, Para4ChoiceList, Para5ChoiceList.

Beispiel 3.3-9

Parameter=Name
Para1ChoiceList=Select Name, Vorname from bibo.benutzer
order by name

Zeitparameter

Zeitspanne

von: 30.08.2007 13:58:53 Zeitbereich ISO-KW

bis: 30.08.2007 13:58:53

Weiter

Abbrechen

Parameter

Name

Fischer Holger
Meier Regina
Müller Sabine
Reuther Michael
Schneider Ralf
Schniegel Herta
Trommer Rebecca
Werner Matthias
Wetzig Heike

Zeit/Parameter eintragen

SQL

```
SELECT
benutzer.vorname, benutzer.name, benutzer.typ, benutzer.gewogen, benutzer.gewicht
FROM
bib0.benutzer
WHERE
benutzer.name like "%#param1%"
```

Im Ausführungsfenster erscheint eine DropDownBox mit allen Namen der Benutzer in geordneter Reihenfolge. Die DropDown-Liste hat die Bezeichnung Name.

ftpcopy=

ftpip=

ftpusr=

ftppwd=

Je nach vorliegendem Anwendungsfall kann es wünschenswert sein, bei der automatischen Reporterstellung entstehende XML-Dateien (XML/XSL) an einen anderen Computer (z.B. einen Webserver) zu übertragen. Das gilt auch für PDF®-Dateien, die möglicherweise generiert wurden. Übertragen werden die Dateien, wenn **ftpcopy** auf den Wert 1 gesetzt wird, nicht übertragen werden die Dateien bei einem Wert 0.

Konstante	Beschreibung
0	Es findet keine Übertragung statt.
1	Übertragung der Reporte (XML,PDF)

Tabelle 24: Konstanten für ftpcopy

Unter **ftppip** wird die Adresse eines FTP-Servers angegeben, **ftppusr** und **ftppwd** nehmen die Entsprechungen für die Anmeldung am Server auf. Die Dateien werden binär übertragen. Da XML/XSL-Dateien direkt in den meisten Internet-Browsern dargestellt werden können (z.B. Internetexplorer ab Version 5.5), besteht die Möglichkeit, die Abfrageergebnisse direkt in umfangreichere Webprojekte einzubeziehen. Hierbei wird vom Programm die automatische Aktualisierung der Webseite(n) bzw. deren Datenrepräsentation sichergestellt.

Beispiel 3.3-10

```
ftpcopy=1
ftppip=127.0.0.1
ftppusr=scott
ftppwd=meinpasswort
```

- email1use=**
- email1smtp=**
- email1sender=**
- email1usr=**
- email1pwd=**
- email1cond=**
- email1subject=**
- email1text=**
- email1empfaenger=**

Neben der Datenübertragung auf einen FTP-Server können durch das Programm Emails versendet werden. **email1use** kann mit 1 oder 0 belegt werden. Hierdurch wird der Versand der Emails gesteuert.

Konstante	Beschreibung
0	Es findet keine Übertragung statt.
1	Übertragung per Mail kann erfolgen.

Tabelle 25: Konstanten für email1use

Wenn **email1smtp** belegt ist, dann erfolgt eine Anmeldung an einem SMTP-Server. Wird hier keine IP-Adresse angegeben, so wird die zu versendende Email an das aktuelle Emailprogramm auf dem System übergeben. Teilweise öffnet sich hierbei ein Fenster, was dieses Verfahren dann für die

automatische, interaktionsfreie Übertragung ungeeignet macht. Wenn eine Anmeldung am SMTP-Server erforderlich ist, sind den Parameter **email1usr** und **email1pwd** der Benutzername und das Passwort zu übergeben. Sollte keine Anmeldung am eigenen SMTP-Server nötig sein, können die Parameter weggelassen werden.

Die Absenderadresse wird an den Parameter **email1sender** übergeben.

email1cond gibt an, welche Spalte mit einem anderen Wert verglichen werden soll. Ist der Vergleichsausdruck wahr, so wird die Email versendet. Möglich sind >, >=, <, <=, =, !=. Ferner ist der Vergleich == für Textvergleiche möglich. Steht am Ende der Zeile „;all“, so werden alle Mails zu einer Mail zusammengefasst, wenn in mehreren Zeilen das Auslösekriterium für Mails erreicht wurde. Existiert zur Abfrage ein PDF (**pdfuse=1**), so wird bei „;all“ diese Datei als Anhang in der Email versendet.

Wird der Eintrag komplett weggelassen, erfolgt nach Ausführung generell ein Emailversand. Zusätzlich muss der Parameter **email1text** freigelassen werden.

Die Betreffzeile der Mail wird mit **email1subject** angegeben. Des Weiteren kann mittels **email1text** ein Text, etwa eine Meldung zum eingetretenen Anwendungsfall, verfasst werden. Hierbei können die Variablen \$01 - \$64 und die Variablen #01 - #64 verwendet werden. Die Dollar-Variablen verkörpern die Inhalte der Zellen, die Raute-Variablen die Namen der Spalten.

email1empfaenger nimmt eine Liste mit Email-Empfängern an, die durch ein Semikolon getrennt aufgeführt werden. Die maximale Anzahl liegt bei 64 Empfängern.

Beispiel 3.3-11

```
email1use=1
email1smtp=10.11.12.254
email1sender=vonmir@abc.de
email1cond=col3;>;2000;all
email1subject=wichtige Nachricht
email1text=Der Wert in Spalte #03 wurde überschritten und beträgt $03!
email1empfaenger=hugo.test@abcd.de;elfriede.system@abcd.de
```

Wenn die Spalte 3 (beginnend bei 0?) größer als 2000 ist, wird eine Email an die angegebenen Mail-Adressen versandt mit dem Betreff „wichtige Nachricht“.

An dieser Stelle ist angedacht, die Notationsmöglichkeiten um *and*, *or* sowie Kombinationen mit *not* zu erweitern.

```
email1cond=col3;>;2000 and col3;>=;col4 and col5;==;Secretary;all
email1cond=col3;>;2000 or col3;>=;col4 or col5;==;Secretary;all
```

Eventuell ist diese Möglichkeit in der vorliegenden Programmversion bereits integriert.

WorkDir=

Mit diesem Befehl wird das Arbeitsverzeichnis hinterlegt, in welches die .time, .xml, .xsl, .csv abgelegt werden. Wird an dieser Stelle kein Verzeichnis angegeben, dann gilt standardmäßig das Verzeichnis .\Output\.

Beispiel 3.3-12

```
WorkDir=C:\Programme\Data2XML\Report\
```

jcall=

jcall steuert die Art und Weise des Aufrufes von Java und kann die Werte 0 und 1 annehmen.

Beispiel 3.3-13

```
jcall=1
```

Für den Java-Aufruf wird eine DOS-Box gestartet. Im Fall **jcall=0** startet Java ohne DOS-Box.

Bourne=

In die Graphik eines Balkendiagrammes kann zusätzlich eine Ziellinie integriert werden. Der Parameter **Bourne=** wird mit zwei Werten aufgerufen. Der erste Wert bestimmt den Zielwert im Diagramm und muss eine Dezimalzahl sein. Der zweite Wert gibt die Anzahl der Punkte (entspricht der Anzahl der Balken) an. Data2XML erzeugt aus diesen Werten eine Datei **zielvorgabe_absolut.ibb** im Installationsverzeichnis, nach der die Linie in das Balkendiagramm integriert wird. Die Ziellinie gilt für alle Balkendiagramme, die in einem Report enthalten sind.

Beispiel 3.3-14

Bourne=25.0,7

Das Balkendiagramm enthält nun eine Ziellinie bei 25% und besteht aus 7 Punkten (für 7 Balken) (vgl. Abbildung 2).

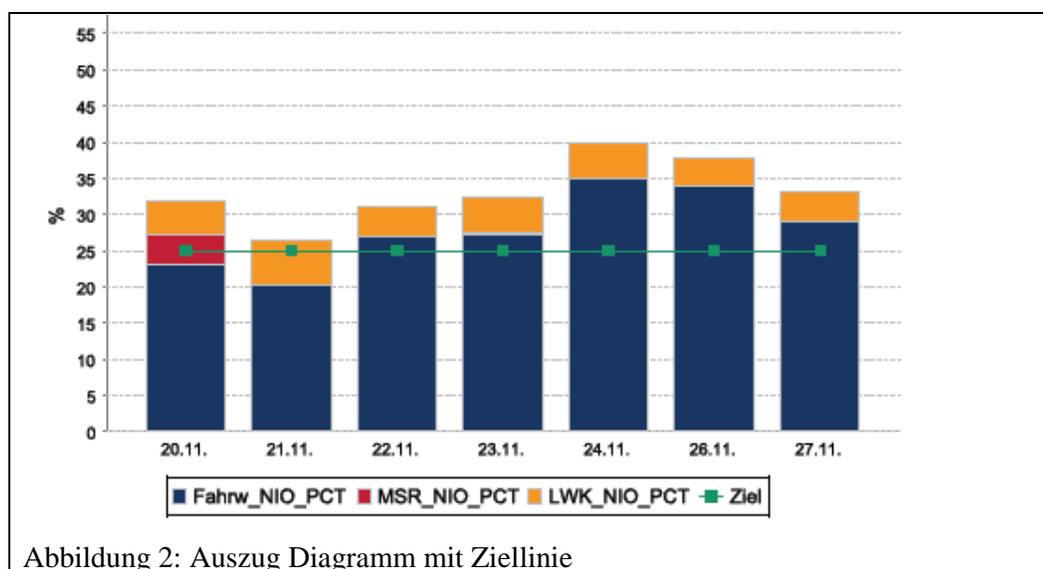


Abbildung 2: Auszug Diagramm mit Ziellinie

Picture=

In graphische Reporte sind Bilder und Subreporte integrierbar, die in gesonderten Dateien beiliegen müssen. Um diese Dateien für den Report bereitzustellen, sind diese dem Programm mitzuteilen. Mittels **Picture=** wird dem Programm eine (ungeordnete) Liste der verwendeten Bilder und Subreports, mit Komma getrennt, übergeben. Diese Dateien sind gleichzeitig im Arbeitsverzeichnis vorzuhalten.

Beispiel 3.3-15

Picture=Logo.gif,MySubreport.jasper

Ein Report enthält ein Bild Logo.gif sowie einen Subreport MySubreport.jasper.

Periode=

Für jeden Report kann ein eigener Abfragezeitraum (Zeitspanne) festgelegt werden. Es sind eine Reihe von Zeitspannen vorgegeben. Die Zeiten werden im Parameterdialog, der sich nach manueller Ausführung eines Reports öffnet, voreingestellt. **Der Parameter wird bei automatisierte Abarbeitung mittels cron nicht benötigt.**

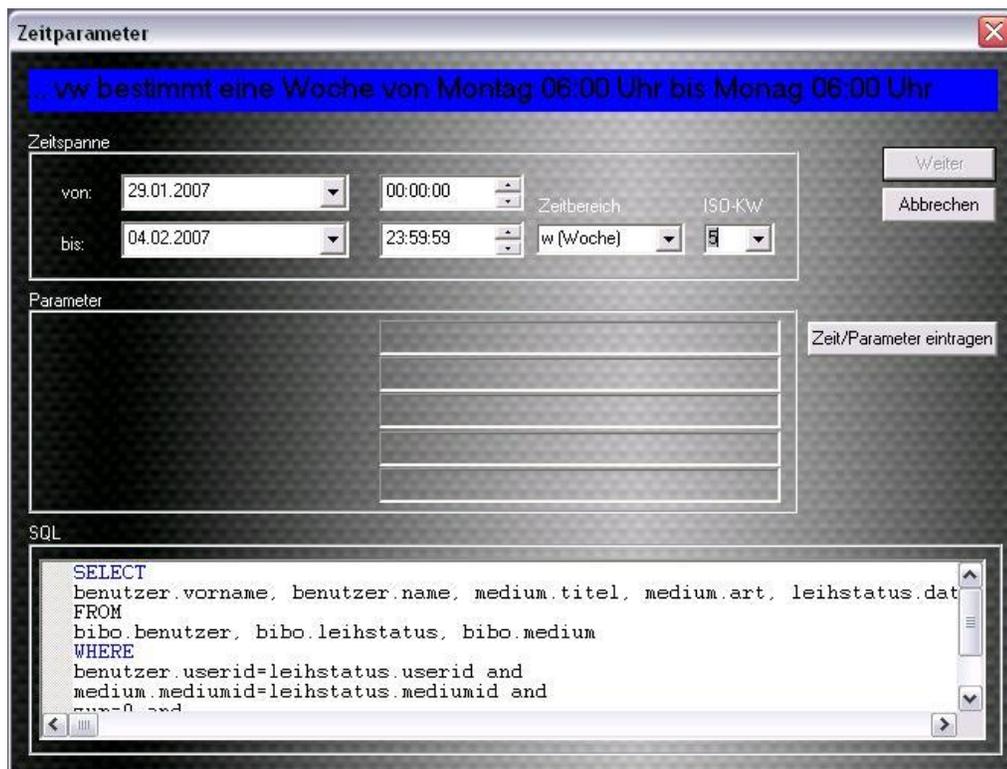


Abbildung 3: Parameterdialog mit voreingestellter Zeitspanne

Konstante	Beschreibung
h	1 Stunde
s1	Nachtschicht: 22:00 – 6:00
s2	Frühschicht: 6:00 – 13:59
s3	Spätschicht: 14:00 – 21:59
t oder d	1 Tag
m	1 Monat
j oder y	1 Jahr
vw	1 Industrie-Woche: Mo 6:00 bis Mo 5:59
it oder id	Letzter vollständiger Tag
iw	Letzte vollständige Woche
im	Letzter vollständiger Monat
ij oder iy	Letztes vollständiges Jahr

Tabelle 26: Konstanten für Periode

Beispiel 3.3-16

```
Periode=t
```

Die Zeitspanne beträgt vom aktuellen Zeitpunkt ein Tag in die Vergangenheit. Alle darin liegenden Daten werden in die Abfrage einbezogen.

Weitere Funktionalität

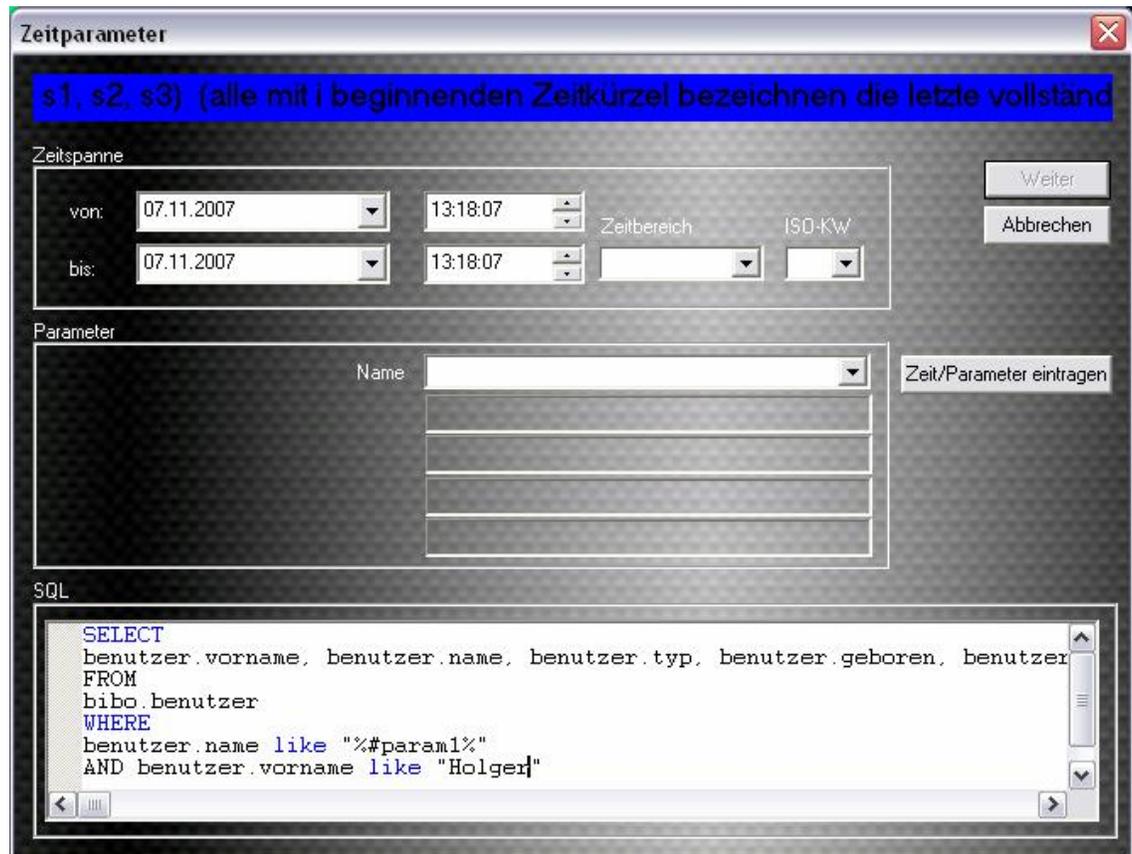
Das Zeitparameter-Fenster verfügt über ein Feld, indem die auszuführende SQL-Anweisung enthalten ist. Der Anwender kann überprüfen, wie die Parameter übernommen werden.

Es besteht die Möglichkeit selbst SQL-Anweisungen einzutragen, die bestehende zu ändern oder zu ergänzen. Die abgebildete SQL-Anweisung wird vom Programm übernommen und ausgeführt.

Beispiel 3.3-17

Im Zeitparameter-Fenster wurde die ursprüngliche SQL-Abfrage durch eine weitere WHERE-Bedingung ergänzt. Diese SQL-Anweisung wird nun vom Programm ausgeführt.

Diese Funktionalität ist für Testzwecke entwickelt worden.



Bei einem erneuten Fenster-Aufruf wird wiederum nur die ursprüngliche SQL-Abfrage angezeigt. Die Änderungen bzw. Ergänzungen gehen verloren. Für eine Dauerhaftigkeit ist die SQL-Anweisung im SelectString bzw. der SQL-Datei zu ändern.



Anhang A

Schlüsselwort	Beschreibung	Wert	Const
CSSFile=	CSS-Datei	<i>user</i>	
Folder=	Zuordnung der Reporte zu einem vertikalen Menüabschnitt	<i>user</i>	
Icon=	Icon-Typ eines Reports im vertikalen Menü	<i>const</i>	001 002 003 ... 006

Tabelle 27: Konfiguration - Layout Data2XML

Schlüsselwort	Beschreibung	Wert	Const
DBTyp=	Typ der Datenbank	<i>const</i>	Oranativ ODBC
User=	Benutzername für den Zugriff auf die Datenbank	<i>user</i>	
PassWord=	Passwort des obigen Benutzers	<i>user</i>	
DataBase=	Name des ODBC-Treibers bzw. Bezeichnung der Datenbank	<i>user</i>	
used=	Report automatisch oder manuell ausführen	<i>const</i>	0,1
Cron01=	Zeitliche Steuerung der Abfragen, mehr siehe Tabelle 29, Tabelle 30, Tabelle 31	<i>const</i>	* * * * * ;t ;
pdfuse=	Anzeige des Reports als pdf-Datei oder als xml-Datei	<i>const</i>	0,1
Ueberschrift1=	Überschrift im Report	<i>user</i>	
Ueberschrift2=	Unterüberschrift im Report	<i>user</i>	
Ueberschrift3=	Überschrift mit Parametrisierung eines Zeitrahmens	<i>user</i>	
SelectString=	SQL-Abfrage in einer Zeile	<i>user</i>	
FileName=	Dateiname zu einer *.sql mit SQL-Abfrage	<i>user</i>	
Fields=	Bezeichnung der Felder zur SQL-Abfrage	<i>user</i>	

Parameter=	Liste der Parameter, auch als sql-Abfrage	<i>user</i>	
Para1ChoiceList=	Parameterliste als Auswahl, max. Para5ChoiceList	<i>user</i>	
ftpcopy=	Versand der Reporte an ftp-Server	<i>const</i>	0,1
ftpip=	IP des ftp-Servers	<i>user</i>	
ftpusr=	Benutzername	<i>user</i>	
ftppwd=	Passwort	<i>user</i>	
email1use=	Aufruf des Email-Versandes	<i>const</i>	0,1
email1smtp=	IP des SMTP-Servers	<i>user</i>	
email1sender=	Absenderadresse	<i>user</i>	
email1usr=	Anmeldung Benutzer für SMTP-Server	<i>user</i>	
email1pwd=	Anmeldung Passwort für SMTP-Server	<i>user</i>	
email1cond=	Bedingung für Email-Versand	<i>user</i>	
email1subject=	Betreffzeile der Email	<i>user</i>	
email1text=	Inhalt der Email	<i>user</i>	
email1empfaenger=	Email-Adressen der Empfänger	<i>user</i>	
WorkDir=	Arbeitsverzeichnis (Standard: .\Output\)	<i>user</i>	
jcall=	Aufruf des Java	<i>const</i>	0,1
Bourne=	Ziellinie in Balken-Diagrammen	<i>user</i>	
Picture=	Aufzählung Bilder oder Subreports	<i>user</i>	
Periode=	Zeitspanne vordefinieren	<i>const</i>	...

Tabelle 28: Schlüsselwörter Reporteinbindung

Beispiel 1

```
[meinReportName1]
used=1
cron01=0 12,15 * * * ;t|Bibliothek|Abfrage jeden Tag um 12 und 15 Uhr|
Datum: #y1-#m1-#d1;
```

Die automatische Abfrage ist eingestellt. Jeden Tag um 12 und 15 Uhr erfolgt eine Abfrage.

Es sind derzeit 9 Einstellungen zu automatischen Abfragen möglich: cron01 bis cron09. Es folgen die Konstanten aus Tabelle 29, Tabelle 30 und Tabelle 31. Semikolon beachten. Nach dem Semikolon steht die Periode (hier t für einen Tag) und | Überschrift 1 | Überschrift 2 | Überschrift 3; (Semikolon nicht vergessen). Zu jeder cron-Abfrage können praktisch andere Überschriften stehen, zum Beispiel als Hinweis, welche Abfrage ausgeführt wurde. In Überschrift 3 können wiederum Zeitparameter übergeben werden (#t1.#m1.#y1)

Konstante	Beschreibung
*****	Auszuführende Befehl
	Wochentag (0 - 6) (Sonntag = 0)
	Monat (1 - 12)
	Tag (1 - 31)
	Stunde (0 - 23)
	Minute (0 - 59)

Tabelle 29: Konstanten zu cron

Hier die möglichen Einträge für die Zeitangaben:

Zeitangabe	Beschreibung
Minuten	0-59 und * für alle Minuten
Stunden	0-23 und * für alle Stunden
Tage	1-31 und * für jeden Tag
Monate	1-12 und * für jeden Monat
Wochentage	0-6 und * für jeden Wochentag (0 für Sonntag)

Tabelle 30: Konstanten zu cron (Fortsetzung)

Parameter	Beschreibung
0 12,15 ***	Ein Komma „,“ lässt mehrere Zeitangaben zu. Jeden Tag um 12 und 15 Uhr erfolgt eine Abfrage.
0 12-24 ***	Ein Bindestrich „-“ gibt einen Zeitraum an. Jeden Tag von 12-24 Uhr (jede Stunde) wird eine Abfrage durchgeführt.

Tabelle 31: Parameter zu cron

Unser Anspruch

- **Kundenorientierung**

In enger Zusammenarbeit mit unseren Kunden entstehen innovative und problemoptimierte Lösungen – maßgeschneidert für jede Anwendung.

- **Zuverlässigkeit**

Von der Anfrage bis zur Inbetriebnahme werden erfahrene Ingenieure Ihre Partner sein.

- **Flexibilität**

Die Freiheit, eigene Produkte zu Ihrem Vorteil mit Komponenten führender Hersteller zu kombinieren, gibt uns die Chance, Ihre Aufgaben gezielt und effektiv zu lösen.

- **Qualität**

Unser Know-how, erfahrene Mitarbeiter und aufwendige Prüfverfahren gewähren ein Höchstmaß an Funktionalität und Qualität.

- **Leistungsfähigkeit**

Rationelle Arbeit, leistungsfähige Produkte und ein stets präsenster Service zu günstigen Konditionen.